Development of Advanced Predictive Functional Control Strategies for SISO Dynamic Processes

Muhammad Saleheen Aftab

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

The University of Sheffield

Faculty of Engineering

Department of Automatic Control and Systems Engineering

August 2022

To my beloved family . . .

Abstract

Predictive Functional Control (PFC) is a heuristic Model Predictive Control (MPC) algorithm that offers intuitive, transparent and simple designs, along with the basic predictive control characteristics, in the cost and complexity roughly similar to that of a standard PID (Proportional-Integral-Derivative) design. But despite these advantages, its practical utilisation has largely been confined to a relatively small selection of simple industrial applications which exhibit benign first-order dynamics. This is mainly due to the use of over-simplified assumptions within the algorithm that generally work well for simpler systems but often cause undesirable tuning difficulties in slightly more complicated higher order applications. Another critical issue that limits its practicality is the lack of consistency and reliability in closed-loop performances while handling severely underdamped and/or open-loop unstable processes.

Therefore, the primary objective of this research is to overcome these prominent deficiencies and hence extend the scope of PFC to a broader range of SISO applications by proposing: (i) a performance oriented controller tuning method which uniquely bases parameter selection on the expected control activity for a well-informed and more meaningful tuning decision, (ii) a new PFC algorithm based on relative measures with far simpler controller tuning as compared to the standard practices of parameter selection, and (iii) a systematic design framework integrating the concept of pre-stabilised or closed-loop predictions within the overall formulation for efficient control of challenging processes. Furthermore, the thesis also investigates a relatively unexplored application of PFC in the area of nonlinear predictive control and therefore presents an efficient and cost-effective PFC design for a class of nonlinear systems as the final contribution.

The efficacy of these proposals has been investigated through numerous simulation studies which suggest marked performance improvements over the conventional PFC, and indeed the PID, in real-world scenarios. It has been observed that: (i) the new tuning proposal for conventional PFC and the proposed Relative PFC algorithm both provide approximately upto 30% faster closed-loop settling times as compared to the existing tuning methods which are either too aggressive for practical implementation or somewhat conservative to have a meaningful impact on the closed-loop behaviour, (ii) the use of the proposed pre-stabilised, or closed-loop, predictions ensure output stability and recursive feasibility under constraints where the direct utilisation of challenging open-loop predictions within PFC fail to perform reliably, and (iii) the proposed Nonlinear PFC algorithm, being inherently better at handling process nonlinearities, provides approximately $2 - 4$ times faster closed-loop responses than the linear PFC (and the PID), and is therefore a natural choice in processes involving wider operating ranges.

Acknowledgements

First and foremost, I would like to thank Almighty Allah for giving me the strength and potential to carry out this research project.

Secondly, I would like to express my earnest gratitude to my Supervisor Dr. Anthony Rossiter for his continuous guidance, encouragement and support throughout this journey. I am also thankful to Dr. George Panoutsos for his valuable advice and discussions.

I am sincerely grateful to the University of Sheffield for funding my PhD studies at the Department of Automatic Control and Systems Engineering.

Finally, I am deeply indebted to my parents, siblings and wife without their unconditional love, support, prayers and encouragement, I would not have been able to accomplish what I have managed to achieve in my life today.

Statement of Originality

Unless otherwise stated in the text, the work described in this thesis was carried out solely by the candidate. None of this work has already been accepted for any degree, nor is it concurrently submitted in candidature for any degree.

Candidate: __________________________
Muhammad Saleheen Aftab

Supervisor: __________________________
John Anthony Rossiter

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces the readers to the research project, with a concise topic background presented in Section 1.1, the investigated research problems in Section 1.2, the primary thesis contributions in Section 1.3, and the thesis organisation in Section 1.4.

1.1 Background

Model Predictive Control (MPC) is an advanced optimal control strategy with powerful and well-defined procedures for complex multivariate processes [2]. However, its computation-heavy nature has traditionally favoured applications with slower dynamics, although the availability of advanced computing resources has significantly widened its scope and utility in the recent years [1, 3]. Nevertheless, there are areas and applications, for example industrial servo loops, where implementing such expensive strategies would be logically and financially infeasible and where cost-effective approaches such as PID naturally make more sense [4]. However, it is evident that PID often falls short when it comes to processes with significant dead-times and physical constraints [5], though popular workarounds, such as Smith predictors [6] and anti-windup techniques [7], have overcome these problems to a certain extent. Nonetheless, these solutions are generally ad hoc which often degrade other performance attributes; for example, poor robustness to parametric uncertainties and/or modelling errors is one prominent side-effect of such post-design alterations.

This highlights the need for more systematic yet simpler and cost-effective model-based approaches, and over the years Predictive Functional Control (PFC) has proved its efficacy as a viable alternative [8–12]. Although PFC in essence is a model-based predictive controller, it operates with fairly simplified design assumptions akin to the standard PID algorithm [13]. Yet in comparison, it features many desirable attributes by default; process dead-times and constraints are managed more systematically with a certain degree of robustness owing to the use of receding horizon [11]. However, it differs from more advanced predictive controllers in the parametrisation of the manipulated variable which, in the case of PFC, is pre-defined as a linear combination of some simple polynomial basis functions [14]. The design is further simplified by noting that constant set-point tracking is achievable with constant predicted control moves, which eliminates the necessity of the complex optimisation routines for control development [15]. Although, unlike the advanced MPC approaches, PFC's heuristic nature merely provides sub-optimal solutions without concrete a priori stability guarantees [16], its simplistic design traits have attracted numerous industrial applications as reported in the literature [10–12, 17, 18].

1.2 Research Problems

Although PFC offers numerous benefits including computational simplicity, design transparency and intuitive parameter tuning, these attributes are somewhat restricted to relatively benign dynamics. Particularly, previous studies have highlighted tuning difficulties with more involved overdamped and non-minimum phase processes along with far more pronounced performance degradation when handling severely underdamped and/or open-loop unstable systems [10, 11, 19]. Despite recent advancements, such as the development of Laguerre PFC [20], model decomposition [21], and input shaping [22] strategies, the literature still lacks cost-effective solutions to overcome these difficulties. Specifically, it is noted that:

(i) Controller tuning in PFC is inherently heuristic and, apart from simple first-order dynamics, lacks clear cut procedures for parameter selection. Furthermore, the existing tuning guidelines, such as [10, 16, 19], often fail to produce the expected results when dealing with slightly more involved but stable higher order dynamics.

(ii) The oversimplified constant future input assumption in PFC may not be flexible enough to efficiently handle dynamics with poor damping and/or open-loop instability. Notably, the current literature lacks well-structured design guidelines to utilise pre-stabilised or closed-loop prediction dynamics with such systems [11].

Another area of research which has remained relatively unexplored is the development of computationally efficient and inexpensive nonlinear predictive control alternatives utilising PFC. More specifically, the existing nonlinear PFC solutions, [23–26], are fairly complicated as opposed to the linear counterparts, with far more ambiguous controller tuning. Hence, this thesis attempts to fill this and the aforementioned voids in the available literature by providing a number of novel contributions for improving the conventional PFC strategy in a straightforward manner and hence enable efficient control of a variety of Single Input Single Output (SISO) dynamic applications.

1.3 Thesis Contributions

The main contributions of this study are summarised below:

(i) Development of a performance oriented tuning algorithm, primarily for stable and well-damped higher order dynamics, which uniquely bases parameter selection on the expected control activity for a well-informed and more meaningful tuning decision.

(ii) Development of a novel PFC algorithm with significantly simpler performance tuning as compared to the conventional methods of parameter selection.

(iii) Development of a systematic design framework to handle difficult open-loop dynamics efficiently, integrating the concept of pre-stabilised or closed-loop predictions within the PFC formulation.

(iv) Development of various classical feedback compensation schemes for straighforward transformation of underdamped and/or unstable open-loop behaviour into stable, well-damped and monotonically convergent (if possible) prediction dynamics for straightforward PFC implementation.

(v) Development of an efficient constraint handling algorithm to accommodate the impact of pre-stabilisation on the functionality of PFC.

(vi) Development of a novel approach to PFC for a class of nonlinear systems along with straightforward and computationally efficient constraint management procedure using a simple saturation policy.

Remark 1.1. *It is worth mentioning that a majority of these contributions successfully retain the inherent simplicity of conventional PFC while improving its design and functionality. In the same spirit, its fundamental weaknesses, such as the lack of rigorous a priori stability guarantees especially under constraints [16], are also retained. Nevertheless, it is noted that despite being a predictive conroller, PFC falls directly under the category of low-cost control approaches, such as PID, for which the absence of mathematical rigour is not a major concern and a posteriori analysis is generally acceptable [11].*

1.4 Thesis Layout

This thesis is divided into seven chapters along with several appendices which comprise the published research articles accompanying this work.

1.4.1 Organisation of Chapters

Chapter 1 introduces the research project with a concise topic background, to-be-investigated research problems, main contributions and the thesis organisation.

Chapter 2 summarises the findings of the literature review conducted in the field of predictive functional control.

Chapter 3 reviews the technical aspects of conventional PFC, discusses the associated tuning weaknesses for stable higher order dynamics and proposes an improved and performance oriented method of parameter selection. Details of proposal are available in the publication attached in Appendix E.

Chapter 4 presents a novel PFC algorithm that considerably simplifies parameter selection and overcomes the inherent tuning deficiencies of the conventional PFC. This chapter is based on two publications which are attached in Appendices G and H.

Chapter 5 discusses the concept of pre-stabilisation and proposes a systematic design framework for controlling difficult processes with PFC. This chapter summarises the contents of various publications which are available in Appendices A-G.

Chapter 6 presents a novel approach to PFC for nonlinear systems, as a direct extension of the approach presented in Chapter 4, which is based on the publication attached in Appendix I.

Chapter 7 presents the final conclusions along with various recommendations for potential future studies.

1.4.2 Organisation of Appendices

Appendix A ([27]) M. S. Aftab, J. A. Rossiter, and Z. Zhang, "Predictive Functional Control for unstable first-order dynamic systems," in Lecture Notes in Electrical Engineering. Springer International Publishing, Sep 2020, pp. 12-22.

Appendix B ([28]) M. S. Aftab and J. A. Rossiter, "Predictive Functional Control with explicit pre-conditioning for oscillatory dynamic systems," in 2021 European Control Conference (ECC). IEEE, Jun 2021.

Appendix C ([29]) M. S. Aftab and J. A. Rossiter, "Pre-stabilised Predictive Functional Control for open-loop unstable dynamic systems," IFAC-PapersOnLine, vol. 54, no. 6, pp. 147-152, 2021.

Appendix D ([30]) 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.

Appendix E ([31]) M. S. Aftab and J. A. Rossiter, "Predictive functional control for challenging dynamic processes using a simple pre-stabilization strategy," Advanced Control for Applications, Wiley, Mar 2022.

Appendix F ([32]) M. S. Aftab, J. A. Rossiter and G. Panoutsos, "Predictive Functional Control for difficult second-order dynamics with a simple pre-compensation strategy," in 13th UKACC International Conference on Control (CONTROL 2022). IEEE, Apr 2022.

Appendix G ([33]) M. S. Aftab, J. A. Rossiter and G. Panoutsos, "Predictive Functional Control for difficult dynamic processes with a simplified tuning mechanism," in 13th UKACC International Conference on Control (CONTROL 2022). IEEE, Apr 2022.

Appendix H ([34]) J. A. Rossiter, M. S. Aftab and G. Panoutsos, "Exploiting Laguerre polynomials and steady-state estimates to facilitate tuning of PFC," in 2022 European Control Conference (ECC). IEEE, Jul 2022.

Appendix I ([35]) J. A. Rossiter, M. S. Aftab, G. Panoutsos, and O. G. Villarreal , "A novel approach to PFC for nonlinear systems," European Journal of Control, Elsevier, June 2022.

Chapter 2

Literature Review

This chapter summarises the findings of the literature review conducted in the field of Predictive Functional Control (PFC). Section 2.1 first presents a brief introduction to Model Predictive Control (MPC). The focus then shifts to the basic characteristics of predictive functional control algorithm in Section 2.2. Next, tuning challenges associated with PFC for higher order processes are discussed in Section 2.3. Section 2.4 critiques the existing PFC modifications for controlling difficult dynamic processes. A brief discussion on nonlinear predictive control is then presented in Section 2.5. Finally, the chapter concludes in Section 2.6.

2.1 A Brief Introduction to Model Predictive Control

Model Predictive Control is an advanced optimal control strategy with powerful and well-defined procedures for complex multivariate processes [2]. Its unparalleled ability to manage physical constraints and straightforward tuning methods have revolutionised industrial process control [14], so much so that more than 4500 successful implementations had been reported by the end of the past century [1]. Since then, MPC has extended its utility to other markets including, agriculture [36], renewable energy [37], power systems [38], building and environment [39, 40], and so on.

The basic philosophy behind MPC is straightforward [41]: at the current time step, the control trajectory is determined via current and predicted process behaviour over a finite horizon by optimising a performance objective subject to some equality and inequality constraints. Only the first control move is implemented, and the whole process is repeated at the next time step. It is this *receding horizon* approach that distinguishes MPC from classical optimal control theory [42], and implicitly adds robustness in the design [11]. Nevertheless, a more important feature is the emphasis on predicted future behaviour, and hence accurate prediction models. Clearly, achievable performance is limited by model precision, and unreliable system representation most certainly leads to ill-posed decision making [11].

Owing to wider industrial acceptance, MPC practitioners have traditionally favoured discrete-time linear Single Input Single Output (SISO) and Multiple Input Multiple Output (MIMO) prediction models. For instance, Dynamic Matrix Control (DMC), a popular variant of MPC, employs predictions based on finite impulse response (FIR) models of the dynamic process [43]. Another popular algorithm, Generalised Predictive Control (GPC), utilises transfer function models in the prediction structure [44]. However, it is noted that implementations of MPC on MIMO processes, while straightforward in principle, may not always be easy. The advent of modern control theory has triggered more focus on state-space approaches in MPC, e.g. [45]; this facilitates straightforward model extensions, if and when more process variables are accessible. On the flip side, state-estimation is almost always necessary with such representations, especially if the system states cannot be measured directly.

With all its attractive features, MPC has one major drawback, that is the implied control law is not guaranteed stabilising by default unlike traditional optimal control approaches [46]. Although there are popular modifications to warrant stability, such as inclusion of terminal constraints to ensure input and output convergence [47], or pre-stabilisation of dynamics for convergent predictions [45], instability and infeasibility often arise with the introduction of constraints. To maintain recursive feasibility, that is a well-posed optimisation problem at each time step, and hence stability, control theorists often utilise the concept of set invariance [14]. Of course such a set may not exist for all types of problems and conditions and thus this is an active area of research. Moreover, most industrial processes have stable open-loop dynamics, and practitioners usually combine heuristics with prudence and system know-how to maintain overall stable operating conditions [10].

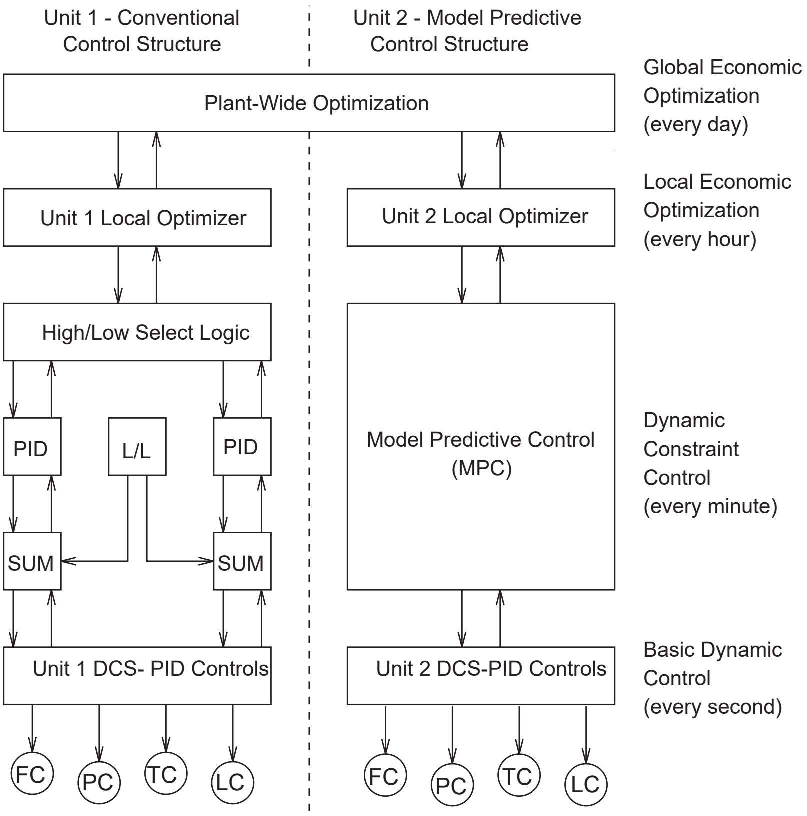

Unit 1 - Conventional Control Structure
Unit 2 - Model Predictive Control Structure
Global Economic Optimization (every day)
Plant-Wide Optimization
Local Economic Optimization (every hour)
Unit 1 Local Optimizer
Unit 2 Local Optimizer
High/Low Select Logic
PID
L/L
PID
Model Predictive Control (MPC)
Dynamic Constraint Control (every minute)
SUM
SUM
Basic Dynamic Control (every second)
Unit 1 DCS- PID Controls
Unit 2 DCS-PID Controls
FC
PC
TC
LC
FC
PC
TC
LC

Figure 2.1: A typical industrial application of MPC [1].

A typical plant-wide hierarchical control framework [1], generally implemented in large process industries, is depicted in Figure 2.1. As evident, MPC is positioned at level-1 or dynamic constraint control level, whose primary objective is to safely drive the system states to updated set-points with minimal constraint violations. However, it is pertinent to note that MPC in such scenarios is only effective as long as level-0 or basic dynamic control is performing satisfactorily. The industrial control practitioners generally prefer PID for low level applications, owing to cheap and easy implementation and maintenance. Nevertheless, the standard PID algorithm has its own shortcomings when it comes to processes with significant dead-times and tighter actuation limits, though popular workarounds, such as Smith predictors [6] and anti-windup techniques [7], have overcome these problems to a certain extent. Nonetheless, these solutions are generally ad hoc and more often degrade other performance aspects; poor robustness to uncertainties is one prominent side-effect of such post-design alterations.

Evidently, there is a need for simpler yet cost-effective MPC approaches, and over the years PFC has proven its efficacy as a viable alternative [10, 17].

2.2 PFC – A Heuristic Approach to Predictive Control

In late 1970s, Richalet et al. [8] introduced a heuristic approach to model predictive control which greatly simplified the coding and algebraic requirements of MPC. Instead of numerically optimising control moves, the strategy was fixed by pre-defining a reference trajectory and then predicting future control moves based on an internal model to ensure reference tracking. The procedure was repeated at each sample to recalculate control moves in order to minimise the difference between actual and predicted behaviour due to modelling uncertainties. The proposal advocated using independent models for predictions that granted straightforward introduction of constraints on control and process variables. The design was inherently robust, but the need for accurate enough models and achievable reference trajectories was emphasised for a stable performance. The efficacy was demonstrated with three industrial case-studies, including a power plant, a chemical plant and an oil refinery, where the proposal had been successfully operating for over an year.

This concept was later formalised as PFC [48–50], and to date numerous successful implementations have been reported in the open literature [9–11, 17, 51–53].

2.2.1 The Basic Algorithm

Although PFC inherits most design attributes from the mainstream MPC family, it differs significantly from other predictive controllers in the parametrisation of the future input [11]. Unlike other predictive controllers, the input trajectory in PFC is pre-defined as a linear combination of some simple polynomial functions whose order depends upon the characteristics of the set-point [14, 15, 17]. Thus, for constant set-points, the future input parametrises to just one degree-of-freedom, eliminating the need for the complex optimisation routines generally associated with high-end MPC algorithms. This on one hand simplifies computations, but on the other hand necessitates heuristics for constrained predictive control problems; simple clipping or saturation has been the commonly deployed constraint management protocol within PFC [8, 14, 17, 54].

There are three basic elements of PFC [17]: a prediction model, a reference trajectory and a disturbance estimator. The prediction model and the reference trajectory work in conjunction to determine the control action over a finite prediction horizon, more specifically the *coincidence horizon*[1]. The disturbance estimator is necessary to ensure offset-free tracking when external disturbances and uncertainties cause mismatches between the predicted and actual behaviour.

Furthermore, the reference trajectory sets a performance objective by defining the desired exponential path between the process output and the set-point. This exponential convergence, which is the main design parameter, is dependent on the characteristics and capabilities of the physical system, and determined by the closed-loop time response (CLTR) i.e. the time *desired* to reach and stay within 95% of the steady-state. It is usually more convenient to represent the target dynamics as a discrete first-order pole placed at $e^{-T_s/\tau}$, with T_s and τ being the sampling period and target time constant respectively where $\tau \approx CLTR/3$ [11].

2.2.2 Constraint Management

One of the key benefits of PFC over similarly placed approaches, such as PID, is its ability to integrate constraints within the design systematically instead of treating them as an afterthought [13]. Although the resulting constrained solution is often sub-optimal, the implicit integral action in PFC simplifies implementing common procedures, such as input clipping or saturation, without causing integral wind-up which is very common with traditional PID techniques [10]. While input constraints in PFC are straightforward to implement, constraint management for internal states or outputs appear to have no clear solutions or are not systematic [14].

In the earlier reported works (such as [9, 17, 54]), the usual practice was to tune two PFCs: the first to ensure reference tracking, and the other to provide conservative control action for constraint adherence. Both controllers worked in parallel and a logic supervisor directed the appropriate command to the process. This clearly was an idealistic approach only effective as long as external perturbations did not interfere. Nevertheless,

[1] Chapter 3 discusses the technical aspects of PFC, refer to Section 3.1 for details.

this traditional method is obsolete now and replaced by more efficient constraints handling proposals developed recently [11, 55].

For example, one proposal recommends utilising model predictions to validate state and/or output constraints using long enough validation horizons [11]. However, it is noted that constraint verification in this way may induce a computational overhead leading to slightly elevated cost and complexity. Another recent study extends this concept further by incorporating the so called *closed-loop* predictions in the control law providing significant improvements in the tuning efficacy and the overall constrained control performance [55]. Nevertheless, it does not explicitly discuss the design of closed-loop parametrisation, which is highly critical for an efficient PFC functionality.

2.2.3 Robustness of PFC

The continuous re-evaluation of control action, on one hand, escalates the computational burden, but on the other hand, also ensures a certain degree of robustness in the design. Yet it is important to assess the controller's ability in managing disturbances, measurement noise and modelling uncertainties for stable closed-loop operation, especially in the presence of constraints.

The effects of various modelling structures on noise and disturbance sensitivities has been investigated [56], which demonstrate marked differences in PFC performance with CARIMA (with and without T-filters) [11] and independent [17] model types. Although such analysis is highly dependent on open-loop characteristics, the CARIMA structure plus T-filter generally provides better noise minimisation for stable and well-behaved dynamics, at the price of relatively slower disturbance rejection.

A similar study on independent models to evaluate Laguerre PFC performance [57] demonstrates slightly aggressive control moves, and hence increased sensitivity to noise and disturbances. Nevertheless, it is noted that better noise filtration alongside fast disturbance rejection is, in fact, a conflicting requirement [58], but being in different frequency regions, a satisfactory design is achievable with prudent selection of controller parameters [59].

Another study [60] implements a robust filter (initially proposed for Smith predictor, see [61]) with prediction model to minimise the effect of dead-time variations for well damped first-order and second-order systems. The proposed configuration achieves stable constrained performance, as opposed to designs without robustifying filter, for which the response often destabilises due to parametric uncertainty.

2.2.4 Comparisons with PID

Evidently it would not be very sensible to compare PFC's performance with that of more mainstream predictive controllers owing to the significant differences in the underlying optimisation. Nevertheless, multiple studies suggest its remarkable benefits over similar approaches like PID [13, 62]. In this context, one comparative analysis [13] highlights the efficacy of PFC in terms of tuning, constraints handling, coping with dead-times, ease of coding, and robustness. Moreover, a recent study conducted to compare and contrast closed-loop performances in automobile cruise control has further strengthened the superiority of PFC over the standard PID algorithms [62]. Nevertheless, competition is not the only possibility; combining these two approaches could further enhance performance beyond that which may be achieved by each compensator in isolation [63–65].

A fault tolerant architecture of PFC-PI is presented in [63] to improve overall reliability. To achieve this, a bumpless switching mechanism transfers the controls from one controller to the other without causing discontinuities in the manipulated variable. A new PID tuning method is proposed in [64] in which the proportional, integral and derivative gains are tuned in an incremental fashion using PFC theory. A case-study on coke-furnace pressure control shows improved disturbance rejection as compared to other PID tuning mechanisms. In another application [65], both PFC and PID are combined to cope with process nonlinearity in an electro-hydraulic servo system, where the merged controller provides offset-free sinusoidal tracking, something both PFC and PID could not achieve individually.

2.3 Tuning Challenges with Higher Order Dynamics

Despite its immense practical appeal, tuning has remained an area of ambiguity within the PFC community [10, 11, 15]. In fact, with the exception of very benign dynamics [16], the literature has lacked rigorous *a priori* stability proofs; empirical suggestions combined with common sense and intuition have mainly guided the tuning process.

For example, a common practice to tune PFCs for slower dynamic systems, such as heat exchangers, is to choose a reference trajectory with CLTR four to five times faster than the open-loop behaviour of the system [66]. It has been learned through experience that such selection generally provides stable performance [67]. In another study [68], the significance of target dynamics and coincidence horizon on performance and closed-loop stability has been analysed experimentally for industrial heat exchange system. It was shown that too quick exponential convergence (smaller CLTR) or too short coincidence horizon may cause severe oscillations in the process output.

However, it is evident that the lack of systematic approaches in literature inhibits rigorous stability analysis on theoretical grounds.

2.3.1 Target Pole and Coincidence Point Selection Criterion

A study conducted to analyse the impact of coincidence horizon on closed-loop performance has found strong influence of coincidence point on the efficacy of reference trajectory [19]. As a general rule of thumb, the target pole gradually becomes ineffective as one shifts the coincidence point further away in the future. It is noted that very large horizons essentially transform PFC into a mean-level controller [11], an approach whose sole purpose is offset-free tracking and constraints handling without worrying about the closed-loop transient performance. An obvious, but implicit, requirement with mean-level controllers is the satisfactory and stable open-loop system behaviour, which is directly reflected in the closed-loop performance.

Nevertheless, for stable first-order dynamics, it is indeed possible to preserve 100% efficacy of the target pole by ensuring one-step ahead coincidence [10, 17, 19]. A similar statement, however, may not hold for stable higher order systems, especially if the dynamics involve significant amount of initial lag or a non-minimum phase characteristic [19]. Hence, for higher order systems, it is recommended to select the appropriate coincidence point first before designing the reference trajectory [10, 11, 19]. One suggestion [9, 10] is to use the point of inflection, i.e. the point of maximum gradient on the step response curve, as the coincidence point. However, [19] argues that parameter tuning on this criterion alone may be flawed, especially if the dynamics exhibit significant non-minimum phase characteristics. As per recommendation though, a more sensible choice is to enforce coincidence within the time window when the open-loop step response approximately rises from 40% to 80% with significant gradient.

These suggestions make intuitive sense as long as the predicted response progresses monotonically towards steady-state after coincidence, but in cases where high frequency oscillations or divergence are prevalent, reliable and consistent performance tuning seems unattainable [11, 19]. Another empirical suggestion for under-damped processes is to choose a prediction horizon that covers at least one cycle of oscillations [66]. However, no mathematical justification or proof of efficacy is provided.

2.3.2 Incorrect Use of Feedforward Component

Irrespective of the tuning policy, it is noted that the link between target pole and actual performance gradually weakens as the coincidence horizon is increased. Recent studies have investigated the underlying issue, and concluded that the prediction mechanism for higher order dynamics in standard PFC formulation may have severe anomalies [69, 70]. The problem lies in the incorrect use of feedforward information, more specifically initialisation of target trajectory on the process output whose slower progression, compared to the first-order reference, embeds delays into the future target values. Nevertheless, the proposed modification in the control law utilises trajectory information from previous samples, instead of simply relying on the current measure of tracking error [70]. With this trivial adjustment, simulation results have demonstrated a relatively stronger link and faster convergence than the original PFC algorithm.

2.3.3 Existing Solutions for Reliable and Effective Tuning

A common solution to address tuning difficulties with higher order systems is by decomposing the prediction model into multiple first-order subsystems using partial fractions [15, 21, 71]. It is noted that such decomposition is only possible due to the *independent* nature of prediction model generally utilised in PFC [72].

Parallel decomposition is explored in [71], which guarantees a prediction structure similar to first-order dynamics. Two equally tuned PFCs are implemented on a third-order process and its reduced first-order version, and simulations demonstrate smoother and less aggressive control action with model decomposition.

Reference [15] presents a cascade decomposition scheme, which complicates the prediction structure to some extent, but preserves first-order tuning and standard constraints handling procedures. Interestingly, it has been pointed out that parallel or cascade decomposition may not suit unstable, complex or non-minimum phase systems, due to the presence of unfactored dynamics in the process model.

Another parallel decomposition strategy for over-damped systems [21] takes a slightly different approach. Each first-order model is treated as a standalone process, with PFC implemented individually. The total manipulated variable is the linear combination of individual control actions, which guarantees target behaviour, provided all input weighting coefficients add up to one.

In view of the apparent deficiency of constant control action, another proposal implements a Laguerre polynomial in the control law to differently parameterise the manipulated variable [20, 73, 74]. It is noted that the additional tuning parameter, i.e. the Laguerre pole, demonstrates direct correlation with the output convergence [73]. Hence, the general recommendation is to select a Laguerre pole that replicates the target performance.

Two PFC algorithms based on Laguerre functions have been proposed [20], with different parametrisation schemes. However, both algorithms demonstrate slightly different performance attributes even with equal Laguerre poles. Furthermore, a new constraint handling mechanism developed for Laguerre PFC [74] ensures less conservative responses under constraints as compared to the original PFC, without increasing the coding complexity.

Nevertheless, it is noted that these proposals fundamentally assume stable and monotonically convergent behaviour, and therefore are not tailored to systems with challenging dynamics.

2.4 PFC and Difficult Dynamics

The preceding discussion assumed well-behaved and monotonically convergent process behaviour, but what if this is not the case? Difficult dynamics, such as open-loop instability, have particularly been challenging for control practitioners. The problem with divergent predictions is that simple and intuitive stability results, which hold for stable dynamics, are no longer relevant.

For guaranteed stability, researchers in predictive control generally impose some form of terminal constraints to reach the stable input and output values within the prediction horizon [11, 47, 75, 76]. Nevertheless, early studies [45, 77] have shown that such restrictive constraints may provide a dead-beat response that may be too aggressive to maintain feasibility in constrained environment in addition to poorer robustness against uncertainties.

A cautious approach to control design [45, 77, 78] has been suggested as an alternative, which bases the decision-making on closed-loop predictions and allows the use of less restrictive terminal constraints to ensure less active input and output. As an added benefit, this type of predictive controller provides improved robustness to model mismatches as opposed to its ancestors. To maintain feasibility under normal constraints, however, it results in increased degrees-of-freedom, which could be utilised to improve performance or reduce control horizon for efficient computation [79].

Unlike MPC, predictive functional control formulates constraints heuristically [14]. As such systematic inclusion of terminal constraints for guaranteed stability may not be achievable. Moreover, the decision making in PFC is generally restricted to just one degree-of-freedom allowing only constant control moves. Thus instability is almost always guaranteed beyond coincidence, even if the predicted behaviour appears satisfactory within this time-frame [80]. Hence, the obvious solution to control difficult processes, with PFC, is to seek different parametrisations of the degree-of-freedom either via explicit pre-stabilisation (for example [10, 81–84]), or implicit modification of the control law (such as [22, 85]).

2.4.1 Pre-stabilisation of Predictions

The early reported work on model pre-stabilisation [80, 81, 86, 87] is based on one of the typical model decomposition structures proposed in [9], wherein the unstable prediction model is decomposed into two stable parts. Such stabilisation of dynamics has been effective in avoiding prediction divergence due to the constant future input assumption.

In [80], the main decision variables, the control changes, are split into two parts as a direct consequence of pre-stabilisation. One part is calculated solely from the past inputs and outputs of the decomposed model, whereas the other part is evaluated with PFC. Such a structure ensures stabilising control laws but may also cause aggressive input activity. To manage hard constraints with such active controllers, a novel constraints handling mechanism is proposed in [87]. The proposal employs two PFCs, one well-tuned and the other slightly detuned with less aggression and hence reduced risk of output overshoot. The predicted response is evaluated at each time sample; if no violation occurs, the well-tuned PFC drives the process, otherwise the detuned controller takes charge. The level of detuning is flexible and depends upon the amount of violation. This algorithm has guaranteed feasibility property in the nominal case.

The study [86] has conducted an unconstrained performance analysis to evaluate the benefits of pre-stabilised model predictions. The results suggest a marked performance difference, as utilising open-loop unstable predictions tends to be unreliable and gives varied PFC success. On the other hand, pre-stabilised control laws always ensure nominal closed-loop stability, but often with a dead-beat type response. To overcome this undesirable behaviour, a modified control law, based on pre-stabilisation, is presented in [81]. The paper argues that allowing too few control moves causes extra aggression in manipulated variable, and proposes a modified prediction class by forcing the *tail* in subsequent optimisations. The numerical simulations demonstrate relatively better performance with a smoother and less aggressive input activity.

The concept of *transparent control* is introduced in [10], whereby one implements an internal proportional control to ensure loop stability in case unexpected disturbances act on the plant. To ensure purely algebraic transference of constraints, however, the proposal discourages employing integral or derivative action within the transparent control architecture. The study [83] takes this concept a step further by controlling integrating processes. The results, nevertheless, suggest a relatively conservative control with constraints as compared to a different parametrisation based on Laguerre functions. Further studies [82, 84] reveal that for both types of parametrisations i.e. transparent and Laguerre, it is indeed possible to achieve equivalent performance, at least in unconstrained setting, with considerable improvements over the conventional PFC.

The manual for predictive functional control practitioners [10] has numerous *make-shift* approaches to overcome difficulties on case-by-case basis. These tweaks may have been successful in practice, but more importantly lack well-structured guidelines for tuning and stability in general.

2.4.2 Input Prediction Shaping Strategy

The core concept behind input shaping [22, 85] is to obtain a set of convergent output predictions by identifying and subsequently cancelling the undesirable features from the prediction class. To achieve this, the predicted input must implement appropriate modification at each time step; however, the choice of input parameterisation has direct impact on the closed-loop performance.

A minimal-order realisation of input shaping is presented in [22], which unsurprisingly results in fairly aggressive control actions. Nevertheless, the proposal improves efficacy of the PFC tuning parameter, the closed-loop target pole, and reduces the adverse effect of large coincidence horizons associated with difficult dynamics. The simulation results demonstrate drastic performance improvement in controlling open-loop unstable and poorly damped systems. Moreover, the algorithm guarantees recursive feasibility in the presence of constraints, which cannot be claimed with non-shaped conventional PFC.

The modified shaping solution [85] provides further improvements by allowing more control moves to drive the predicted output to the steady-state. Three different pole shaping strategies have been presented for integrating, under-damped and unstable systems. These proposals are simple and intuitive, but their execution requires additional calculations which may increase the coding or algebraic complexity. Nevertheless, the algorithm has been tested and validated on numerous real-world case studies and with a real-time servo speed control experiment. The results have shown significant improvements over the minimum-move shaping PFC [22] and the conventional PFC algorithms.

Although input prediction shaping has provided a new dimension to PFC in controlling difficult systems, the impact of such algorithmic modifications on loop sensitivity and robustness is yet to be explored. An initial investigation, nonetheless, recommends utilising a CARIMA with T-filter prediction structure which leads to relatively less loop sensitivity against measurement noise, without overly compromising the disturbance rejection properties of PFC [12].

2.4.3 Model Decomposition for Poorly Damped Processes

Apart from open-loop instability, controlling dynamics with significant under-damping has also been problematic for practitioners [10, 11, 19]. The main problem lies in the PFC premise which assumes smooth prediction convergence after coincidence, which is clearly an unreasonable expectation with oscillatory dynamics. Moreover, constant control does not possess enough flexibility to damp down the predicted outputs. Consequently these issues often leave the designer with very restricted parameter choices for a meaningful performance [19]. Although such systems can be controlled with the shaping strategies discussed above, a relatively simpler approach, similar to the one for overdamped higher order dynamics, is to decompose the process model into multiple first-order subsystems using partial fractions [88, 89].

It is understood that partial fraction decomposition of under-damped models would result in complex residues. Despite that, the work in [88] reports considerable reduction in algebraic requirements, and more importantly guarantees a real implementable control law. Even then it is noted that the existing general purpose industrial PLCs may not be suitable for complex computation. In view of this argument, [89] proposes a modified tuning algorithm that does not rely on complex algebra, instead the real and imaginary parts are handled separately. This, however, comes at the price of slightly burdened computational complexity with increased costs. Nevertheless, numerical simulations and real-time experiment have validated the efficacy of this proposal.

A more recent decomposition strategy suggests separating the minimum and non-minimum phase dynamics for straightforward pre-compensation of oscillatory poles [90]. While the efficacy of the proposal has been successfully validated with three industrial case studies, the inner compensation is primarily based on pole cancellation using crude inversion of prediction dynamics which may have poorer robustness to external disturbances and modelling uncertainties. Furthermore, the proposed constraint handling is based on back calculation (see [10]), which is inherently inefficient and often leads to relatively conservative constrained performance [11].

2.5 PFC and Nonlinear Dynamics

A vast majority of real-world processes operate around fixed and well-defined operating points. For such systems, linear prediction models are sufficiently adequate in order to implement a successful MPC design [91, 92]. Nevetheless, there exists a class of highly nonlinear dynamic applications which either cannot be suitably approximated by linear models, or have very large operating windows that prevent a reasonable linear MPC operation [93]. These limitations necessitate utilising nonlinear model predictive control (NMPC), which is largely both complex and computationally demanding as evidenced in the literature [91–94].

2.5.1 Challenges with Nonlinear Predictive Control

The primary challenge in the development of efficient NMPC solutions is posed by the non-convex nature of the underlying optimisation problem [95]. Specifically, the presence of multiple local minimas in the cost functions demands for significantly slower and intensive computations to obtain real-time optimal solutions. However, this problem is commonly rectified by linearising the prediction model at each sampling instant, essentially transforming the nonlinear programming into the more manageable conventional quadratic programming (QP) [96–99]. Yet, the computational overhead related to subsequent relinearisations and optimisations is still quite significant.

The second hurdle is the development of suitable nonlinear models that adequately capture the true process behaviour. Although white-box models derived from first principles are relatively easier to develop for simpler systems, these become exceedingly difficult to obtain for processes with moderate to high system complexity [92, 100]. In comparison, black-box models, such as NARX [101], Volterra [102], fuzzy logic [103] and neural networks [104], can be identified from input-output data without in-depth technical knowlegde of the underlying process dynamics. Nevertheless, their performance is primarily limited to the training data range and generally degrades outside this design limit.

Another alternative is to utilise block-oriented cascade models [105]. The most commonly employed members of this family in NMPC are Wiener and Hammerstein types in which the linear dynamic part appears in series with the nonlinear steady-state part [97, 98, 106, 107]. These strategies are particularly useful in the development of hybrid or grey-box representations, wherein the steady-state nonlinearities could be captured by the aforementioned black-box techniques, such as neural netwroks [97, 98]. Nevertheless, the computational complexity pertaining to online nonlinear optimisation still restricts the generic application of these NMPC proposals to inherently slower dynamic processes.

2.5.2 Available Nonlinear PFC Solutions

Although it is expected that a simplified predictive controller, such as PFC, could significantly reduce the cost and complexity associated with the mainstream NMPC techniques, surprisingly this area of research has remained relatively unexplored, despite immense practical appeal. In fact only a few nonlinear PFC (NPFC) proposals are available in the current predictive control literature [23–26].

An artificial neural network based NPFC was developed and applied to a Continuous

Stirred Tank Reactor (CSTR) [23]. Although simulation results demonstrated a superior performance as compared to the PID, the underlying neural modelling around a single operating point was quite restrictive. Another proposal utilised pseudo partial derivative (PPD) for dynamical linearisation of the nonlinear prediction model [24]. The developed adaptive PFC algorithm was tested on a linear time-delayed plant and a nonlinear chemical reactor, providing a relatively better closed-loop performance than various PI/PID controllers especially under the influence of external disturbances and modelling uncertainties.

More recent NPFC approaches are based on iterative learning control (ILC) and general regression neural network (GRNN) models [25, 26]. An ILPFC algorithm for nonlinear batch processes is proposed, which combines the benefits of ILC with model predictive control [25]. Furthermore, the proposal is utilised for trajectory tracking in unmanned ground vehicles (UGVs) and batch chemical reactions, which provides similar closed-loop results but with relatively faster processing as compared to the available high-end ILMPC algorithms.

In [26], a GRNN model-free PFC (GRNN-MFPFC) is proposed and applied to a highly nonlinear plant and an injection moulding industrial process. Although the simulation results confirm considerably better closed-loop control in both examples as compared to a model-free adaptive controller (MFAC) and a model-free adaptive predictive controller (MFAPC), the controller tuning in GRNN-MFPFC is achieved via hit and miss – a tedious and time consuming activity which clearly limits the practicality of this approach.

2.6 Chapter Summary

Over the years, PFC has not only cemented its place as a viable alternative to PID for a majority of SISO industrial control problems, but also has gone through various interesting phases of improvement. Nevertheless, a review of the open literature has identified some obvious gaps in the current knowledge base which will be examined thoroughly in this reseach study.

As discussed in Section 2.3, the most critical discrepancy of conventional PFC lies in its ambiguous tuning procedures for higher order dynamics. While numerous solutions have been reported for improvements, these are either too complicated for a cheaper implementation or lead to significantly conservative performances under the influence of process constraints. Chapters 3 and 4 of this thesis will address these tuning difficulties with relatively simpler modifications in the conventional PFC algorithm.

Another prominent deficiency is the unavailability of systematic procedures to control challenging dynamics with PFC (Section 2.4). For instance, references [9, 10] propose numerous modifications in the basic PFC algorithm to deal with difficult scenarios, but do not provide well-structured reasoning for their application. Instead what we have is a plethora of tweaks and adjustments, each only relevant to a particular situation. Chapters 5 of this thesis will present a systematic design framework utilising the concept of pre-stabilisation to cater for such difficult applications.

Finally as discussed in Section 2.5, it is noted that the nonlinear predictive control literature in general lacks simpler and inexpensive alternatives to handle critical process nonlinearities. More specifically, the existing NPFC solutions are fairly complicated, as opposed to the linear counterparts, with far less clear-cut controller tuning. Hence, Chapter 6 of this thesis will attempt to fill this void by proposing a novel approach to PFC for a class of nonlinear systems with reasonably straightforward design procedures.

Chapter 3

Conventional PFC

This chapter serves two purposes. Firstly, it introduces the readers to the technical aspects of conventional PFC throughout Sections 3.1-3.2, emphasising on standard tuning practices and their shortcomings. Secondly, to address these deficiencies it proposes a new and improved parameter selection method in Section 3.3, which partially covers the contents of ACA paper [31] attached in Appendix E. A detailed efficacy analysis of the proposal using two numerical examples is included (Section 3.4), before concluding the chapter in Section 3.5.

3.1 Technical Details of Conventional PFC

This section reviews the technical characteristics of the conventional PFC algorithm.

For stable and well-damped dynamics, PFC works as follows: at every time sample k, the current control u_k is used to enforce a match between the predicted output y_k and a pre-defined reference trajectory r_k at a coincidence point n_y samples ahead. However, unlike the mainstream MPC algorithms (such as DMC [43] and GPC [44]), this is done assuming a constant future input (for step targets), i.e. $u_k = u_{k+1} = \cdots = u_{k+ny}$, whose recomputation at each subsequent sample implicitly establishes a virtual feedback that drives the response closer to the set-point. To ensure a meaningful decision-making, however, an implicit but fundamental requirement is a stable and monotonically convergent prediction behaviour beyond coincidence, without which the achieved closed-loop performance may be unreliable [11, 19].

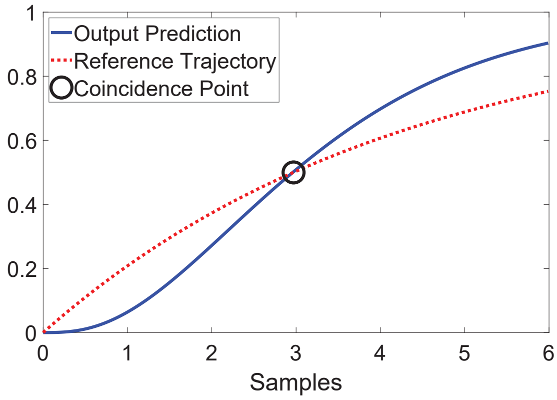


Figure 3.1: Example of coincidence between output prediction and reference trajectory at $n_y = 3$.

Nevertheless, there are three key elements fundamental to PFC: a reference trajectory, a prediction model, and a mechanism to obtain unbiased predictions. These are discussed next.

3.1.1 Reference Trajectory

The reference trajectory provides an *ideal* or desired path to the set-point, which is chosen mainly for computational simplicity as a first-order exponential initiated on the current output measurement, represented by the following relationship [11, 17, 19]:

$$r_{k+i} = R - (R - y_k)\rho^i, \quad i \geq 1 \tag{3.1}$$

where R is the set-point, y_k is the current output and ρ is the target pole, the main tuning parameter, defined as $\rho = e^{-T_{samp}/\tau}$ with T_{samp} and τ being the sampling time and the target time constant respectively. Therefore, at the point of coincidence n_y, which is the second tuning knob, one enforces the prediction to match the reference, as shown in Figure 3.1, such that:

$$y_{k+n_y|k} = R - (R - y_k)\rho^{n_y} = r_{k+n_y} \tag{3.2}$$

where the notation $k + x|k$ means x-step ahead prediction made at the current sample k.

3.1.2 Prediction Model

Owing to the practical appeal, PFC vendors traditionally prefer transfer function models to predict the future behaviour [10], although the *independent* prediction structure employed within the framework may allow utilising other model types as well (for example, see reference [11] which enlists many alternative representations including state-space, CARIMA and FIR).

Let us consider a discrete-time model $G(z)$, relating the model output $\hat{y}_k$ to the input u_k, given as follows:

$$G(z) = \frac{b(z)}{a(z)} = \frac{b_1 z^{-1} + b_2 z^{-2} + \cdots + b_n z^{-n}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}} \tag{3.3}$$

where $G(z)$ is strictly proper with order n. The output predictions are recursively obtained from the model expression $a(z)\hat{y}_k = b(z)u_k$ and can be written as [11]:

$$\begin{aligned}
\hat{y}_{k+1} + a_1\hat{y}_k + a_2\hat{y}_{k-1} + \cdots + a_n\hat{y}_{k-n+1} &= b_1 u_k + b_2 u_{k-1} + \cdots + b_n u_{k-n+1} \\
\hat{y}_{k+2} + a_1\hat{y}_{k+1} + a_2\hat{y}_k + \cdots + a_n\hat{y}_{k-n+2} &= b_1 u_{k+1} + b_2 u_k + \cdots + b_n u_{k-n+2} \\
\hat{y}_{k+3} + a_1\hat{y}_{k+2} + a_2\hat{y}_{k+1} + \cdots + a_n\hat{y}_{k-n+3} &= b_1 u_{k+2} + b_2 u_{k+1} + \cdots + b_n u_{k-n+3} \\
&\vdots
\end{aligned} \tag{3.4}$$

or in the compact matrix form as:

$$\mathbf{C}_a \underset{\rightarrow}{\hat{\mathbf{y}}}_{k+1} + \mathbf{H}_a \underset{\leftarrow}{\hat{\mathbf{y}}}_k = \mathbf{C}_b \underset{\rightarrow}{\hat{\mathbf{u}}}_k + \mathbf{H}_b \underset{\leftarrow}{\hat{\mathbf{u}}}_{k-1} \tag{3.5}$$

where the corresponding matrices and vectors are defined below:

$$\mathbf{C}_a = \begin{bmatrix} 1 & 0 & \dots & 0 \\ a_1 & 1 & \dots & 0 \\ a_2 & a_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix} \qquad \mathbf{H}_a = \begin{bmatrix} a_1 & a_2 & \dots & a_n \\ a_2 & a_3 & \dots & 0 \\ a_3 & a_4 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix} \tag{3.6}$$

$$\mathbf{C}_b = \begin{bmatrix} b_1 & 0 & \dots & 0 \\ b_2 & b_1 & \dots & 0 \\ b_3 & b_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix} \qquad \mathbf{H}_b = \begin{bmatrix} b_2 & b_3 & \dots & b_n \\ b_3 & b_4 & \dots & 0 \\ b_4 & b_5 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix} \tag{3.7}$$

$$\underset{\rightarrow}{\hat{\mathbf{y}}}_{k+1} = \begin{bmatrix} \hat{y}_{k+1} \\ \hat{y}_{k+2} \\ \vdots \\ \hat{y}_{k+i} \end{bmatrix}; \quad \underset{\leftarrow}{\hat{\mathbf{y}}}_k = \begin{bmatrix} \hat{y}_k \\ \hat{y}_{k-1} \\ \vdots \\ \hat{y}_{k-n+1} \end{bmatrix}; \quad \underset{\rightarrow}{\mathbf{u}}_k = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+i} \end{bmatrix}; \quad \underset{\leftarrow}{\mathbf{u}}_{k-1} = \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \\ u_{k-n} \end{bmatrix} \tag{3.8}$$

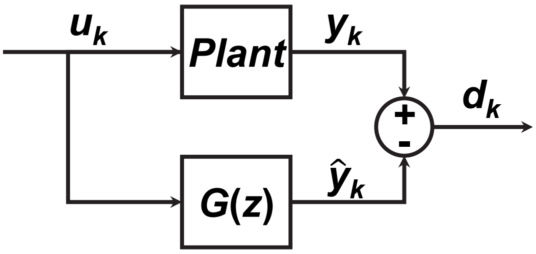


Figure 3.2: Disturbance estimation with independent model structure.

Hence, the i-step ahead output prediction can be obtained from:

$$\hat{y}_{k+i|k} = \mathbf{H}_i \underrightarrow{\mathbf{u}}_k + \mathbf{P}_i \underleftarrow{\mathbf{u}}_{k-1} + \mathbf{Q}_i \underleftarrow{\hat{\mathbf{y}}}_k \tag{3.9}$$

where $\mathbf{H}_i$, $\mathbf{P}_i$ and $\mathbf{Q}_i$ denote the i^{th} row vectors of the corresponding matrices $\mathbf{H} = \mathbf{C}_a^{-1}\mathbf{C}_b$, $\mathbf{P} = \mathbf{C}_a^{-1}\mathbf{H}_b$ and $\mathbf{Q} = -\mathbf{C}_a^{-1}\mathbf{H}_a$ respectively.

3.1.3 Mechanism for Unbiased Predictions

PFC employs an independent model structure wherein the model $G(z)$ is simulated in parallel with the real process, as shown in Figure 3.2, which allows predictions to be determined solely from the past model states and the past/present inputs. However, the presence of external disturbances and modelling mismatches is unavoidable, which typically induces bias causing a non-zero offset in the steady-state. Hence, to ensure bias-free predictions, a disturbance estimator d_k is employed in (3.9) such that one gets:

$$y_{k+i|k} = \mathbf{H}_i \underrightarrow{\mathbf{u}}_k + \mathbf{P}_i \underleftarrow{\mathbf{u}}_{k-1} + \mathbf{Q}_i \underleftarrow{\hat{\mathbf{y}}}_k + d_k \tag{3.10}$$

where d_k, the estimate of disturbance/bias at sample k, is computed from:

$$d_k = y_k - \hat{y}_k \tag{3.11}$$

3.1.4 Conventional PFC Control Law

Hence, the unbiased output prediction (3.10) is matched with the ideal reference trajectory (3.2) at the coincidence point n_y, keeping the current input constant, i.e. $u_{k+i} =$

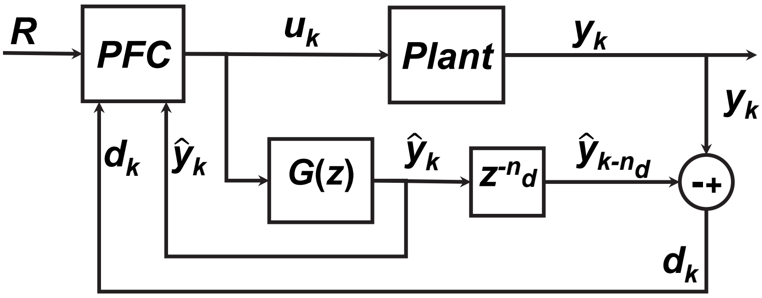


Figure 3.3: The conventional PFC control architecture with delay compensation.

u_k, $\forall i \geq 1$. Consequently, one can deduce the following control law:

$$u_k = \frac{1}{h_{n_y}}\Big[R - (R - y_k)\rho^{n_y} - (\mathbf{P}_{n_y}\underleftarrow{\mathbf{u}}_{k-1} + \mathbf{Q}_{n_y}\underleftarrow{\hat{\mathbf{y}}}_{k} + d_k)\Big] \tag{3.12}$$

where $h_{n_y} = \mathbf{H}_{n_y}\mathbf{L}_{n_y}$ with $\mathbf{L}_{n_y} = [1 \quad 1 \ldots 1]^T_{1\times n_y}$.

3.1.5 Coping with Large Delays

In order to handle process deadtimes (represented here by n_d samples), PFC utilises a structure similar to Smith predictor as shown in Figure 3.3. But in essense, this configuration forces the model predictions (3.9) to lag the current output by n_d samples which must be corrected through suitable changes in the control law. Therefore (3.12) is updated with the delayed output prediction given as $E[y_{k+n_d|k}] = \hat{y}_k + d_k$ [11], implying that:

$$u_k = \frac{1}{h_{n_y}}\Big[R - (R - E[y_{k+n_d|k}])\rho^{n_y} - (\mathbf{P}_{n_y}\underleftarrow{\mathbf{u}}_{k-1} + \mathbf{Q}_{n_y}\underleftarrow{\hat{\mathbf{y}}}_{k} + d_k)\Big] \tag{3.13}$$

where $E[\,.\,]$ represents the n_d-step ahead estimate/prediction and d_k is now computed using $d_k = y_k - \hat{y}_{k-n_d}$. Note that when $n_d = 0$, both (3.12) and (3.13) are no different.

3.1.6 Constraints Handling

Assume that the process being controlled is subject to some inequality constraints, defined on the input and output as follows:

$$\underline{u} \leq u_k \leq \bar{u}, \quad \Delta\underline{u} \leq \Delta u_k \leq \Delta\bar{u}, \quad \underline{y} \leq y_k \leq \bar{y} \tag{3.14}$$

where $\Delta = 1 - z^{-1}$ represents the sample-wise rate of change. One of the core benefits of PFC over similarly placed techniques, such as PID, is its ability to integrate these constraints within the design systematically instead of treating them as an afterthought [13]. Specifically, the constant future input facilitates straightforward implementation of a simple saturation policy to predict and validate input constraint adherence using just the following four inequalities at each k:

$$\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} u_k \leq \begin{bmatrix} \bar{u} \\ -\underline{u} \\ \Delta\bar{u} + u_{k-1} \\ -\Delta\bar{u} - u_{k-1} \end{bmatrix} \tag{3.15}$$

Output/state constraints, if present, can also be implemented efficiently using model predictions [11], such as (3.10), 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 (3.12), such that:

$$\underline{y} \leq h_i u_k + \mathbf{P}_i \underleftarrow{\mathbf{u}}_{k-1} + \mathbf{Q}_i \underleftarrow{\hat{\mathbf{y}}}_k + d_k \leq \bar{y}; \quad i = 1, 2, \ldots, n_c \tag{3.16}$$

Remark 3.1. *For constraint validation, it is often sufficient to select n_c corresponding to 95% settling time of the open-loop step respons of $G(z)$, i.e. the time to reach and stay within about 95% of the implied steady-state output [11].*

Since all constraints must be verified simultaneously, it is often more convenient to stack (3.15) and (3.16) into a single set of $2(2 + n_c)$ linear inequalities as shown below:

$$\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ h_i \\ -h_i \end{bmatrix} u_k \leq \begin{bmatrix} \bar{u} \\ -\underline{u} \\ \Delta\bar{u} + u_{k-1} \\ -\Delta\bar{u} - u_{k-1} \\ \bar{y} - \left(\mathbf{P}_i \underleftarrow{\mathbf{u}}_{k-1} + \mathbf{Q}_i \underleftarrow{\hat{\mathbf{y}}}_k + d_k\right) \\ -\underline{y} + \left(\mathbf{P}_i \underleftarrow{\mathbf{u}}_{k-1} + \mathbf{Q}_i \underleftarrow{\hat{\mathbf{y}}}_k + d_k\right) \end{bmatrix} \tag{3.17}$$

Remark 3.2. *The process of constraint validation based on* (3.17) *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 [11, 85].*

3.2 Parameter Tuning in PFC

In conventional PFC, there are two main tuning parameters: the target pole ρ and the coincidence point n_y. The primary tuning parameter ρ represents the ideal (first-order exponential) speed of convergence of the tracking error, i.e. how fast or slow the predicted response approaches the set-point. It is clear from (3.2) that:

$$e_{k+n_y|k} = \rho^{n_y} e_k; \quad e_k = R - y_k \tag{3.18}$$

While the significance of ρ is obvious, its efficacy is highly dependent on the judicious selection of the coincidence point n_y. To elaborate this, we analyse the impact of tuning parameters on the control effort produced by conventional PFC.

3.2.1 Effect of Tuning on Controller Activity

The dynamics of 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 behaviour of the controlled system [10]. Assuming zero initial conditions and no uncertainty for simplicity, it is straightforward to show using (3.12) that for a change in R:

$$u_{1,n_y} = \frac{R}{h_{n_y}}(1 - \rho^{n_y}) \tag{3.19}$$

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

- The initial input is directly proportional to the magnitude of the desired set-point R which is expected since tracking a large target 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.

Hence, it is obvious that different pairings of (ρ, n_y) will result in different closed-loop performances. Nevertheless, making the most appropriate selection is the core tuning dilemma which often arises specifically with higher order dynamics [11, 19].

3.2.2 Is $n_y = 1$ a Good Choice?

Seemingly $n_y = 1$ stands out as the most plausible solution since it preserves complete efficacy of the target pole. In fact, this has been the default suggestion when dealing with very benign dynamics [10, 11, 21, 71], such as those dominated by first-order behaviour, and for which establishing concrete guarantees of tuning efficacy, performance and closed-loop stability has been extremely straightforward [16, 19]. The only problem is $n_y = 1$ cannot be generalised.

A majority of real-world processes exhibit higher order dynamics which may not be as simple to work with. For example, heavily damped systems often have slower transient behaviour for which one-step ahead coincidence may necessitate a significant amount of control energy well beyond system's capacity [11, 19]. This becomes even more critical with non-minimum phase dynamics, whose inverse output response can easily lead to instability with inappropriate coincidence horizons [11, 19].

Hence, one is often left with no other choice but to enforce coincidence further ahead in the future, which clearly reduces the efficacy of the target pole, but in return produces a more practical and easily implementable control input. Obviously, this then raises a very important question: what is a suitable coincidence horizon and how should it be selected? Surprisingly, there is no clear-cut answer. The following section summarises the standard tuning guidelines currently available in the literature.

3.2.3 Standard Tuning Practices

Although the literature lacks a systematic and well-defined parameter tuning procedure, there have been few empirical suggestions which have traditionally guided parameter selection without overarching justification, with the exception of first-order dynamics for which strong mathematical results have been furnished [11, 16, 19]. These guidelines are stated below:

(i) For higher order dynamics, one suggestion is to use the point of inflection, i.e. the point of maximum gradient on the open-loop step response curve, as the coincidence point [10, 17]. However, it is argued that tuning on this criterion alone may be flawed, especially if the dynamics in question are non-minimum phase [19].

(ii) An alternative approach for higher order dynamics is to enforce coincidence within the time window when the open-loop step response has risen from 40% to 80% of the implied steady-state with significant gradient [11, 19].

(iii) Another recommendation is to choose n_y such that the condition $2h_{n_y} > G(1)$ is met, where $G(1)$ is the steady-state gain [16]. Although it is sufficient, but not necessary, to guarantee nominal closed-loop stability for simple dynamics, it often leads to performances very similar to the second approach discussed above [16].

(iv) In order to ensure the least aggressive initial control, a less common approach is to enforce coincidence at the so-called *Turpin point*, i.e. the horizon n_y which minimises the initial input u_{1,n_y} [10].

Nevertheless, these guidelines have some clear shortcomings. For example, there is no explicit consideration of the expected initial input in parameter selection; the obvious exception is suggestion (iv), but being far too cautious, it is rarely implemented in practice [16]. Another core issue is the partial efficacy of target dynamics, which often arises with $n_y >> 1$ [11, 19]. Hence to compensate for longer horizons, faster ρ's must be used, which in turn may lead to overactive control with higher likelihood of constraint violations and closed-loop instability. Evidently these guidelines merely provide a non-systematic way to reach the best possible selection of $(\rho,\, n_y)$, which more often ends up being a global search without a clear enough link to the desired performance.

3.3 Performance Oriented Parameter Selection

This section presents a new tuning method with the aim of systematically overcoming the mentioned discrepancies of the standard tuning guidelines. In particular, the proposal explicitly utilises the initial input information which makes parameter selection more meaningful and performance oriented.

With regards to the expected initial input (3.19), 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 $\rho^{n_y} \to 0$. Thus knowing the controller activity for both cases can provide a better understanding of the expected closed-loop performance for various choices of n_y in between.

Theorem 3.3. *For a given set-point R and a target pole ρ, the initial control u_{1,n_y}*

(i) for $n_y = 1$ is given by:

$$u_{1,n_y} = \frac{R}{b_1}(1-\rho); \quad n_y = 1 \tag{3.20}$$

where b_1 is the lead coefficient of $b(z)$.

(ii) for $n_y \to \infty$ is given by:

$$u_{1,n_y} = u_{ss}; \quad n_y \to \infty \tag{3.21}$$

where u_{ss} is the implied steady-state input.

Proof. Both (3.20) and (3.21) can be deduced easily from the forgoing discussion as follows:

(i) The one-step ahead prediction ($n_y = 1$) obtained from (3.4) can be rearranged as:

$$\hat{y}_{k+1} = b_1 u_k + [b_2 \quad b_3 \ldots] \underleftarrow{\mathbf{u}}_{k-1} + [-a_1 \quad -a_2 \quad -a_3 \ldots] \underleftarrow{\hat{\mathbf{y}}}_{k}$$

from which it is clear that with $h_1 = b_1$, (3.19) leads to (3.20).

(ii) When $n_y \to \infty$, it is known from a previous study [19] that h_{n_y} approaches the static gain $G(1)$ of the system, where:

$$G(1) = \frac{b(1)}{a(1)} = \frac{b_1 + b_2 + \cdots + b_n}{1 + a_1 + a_2 + \cdots + a_n} = \frac{\sum_{i=1}^{n} b_i}{1 + \sum_{i=1}^{n} a_i} \tag{3.22}$$

Hence, (3.19) reduces to:

$$u_{1,\infty} = \frac{R}{G(1)}; \quad n_y \to \infty \tag{3.23}$$

By definition, the steady-state input is $u_{ss} = \dfrac{R}{G(1)}$ which proves $u_{1,\infty} = u_{ss}$.

□

Remark 3.4. *In practice, $u_k = u_{ss}$, or close to, will be achieved with a very large horizon; indeed the choice $n_y = n_c$ (implying the minimum variance control [108]) usually works well, where n_c corresponds to 95% settling time of the open-loop step response. So, choosing a large horizon in essence means you are choosing mean-level control [11].*

Corollary 3.5. *For well-damped minimum phase and monotonically convergent prediction dynamics, a usable initial input lies within the following n_y range:*

$$1 \leq n_y \leq n_c \tag{3.24}$$

provided both $b_1 \neq 0$ and $G(1) \neq 0$.

Proof. This is obvious from (3.20) and (3.23). □

Remark 3.6. *For non-minimum phase dynamics with otherwise well-damped open-loop characteristics, a usable initial input lies within the following sub-range of n_y:*

$$n_{dip} < n_y \leq n_c \tag{3.25}$$

where n_{dip} corresponds to the delay (in samples) due to response reversal, after which the predicted output surpasses the initial state and proceeds monotonically in the correct direction.

Therefore, notwithstanding the lack of mathematical rigour, a sensible choice of parameters could be the one that amplifies u_{ss} by a reasonable amount, such that the resulting initial control is not too aggressive, i.e. practically achievable. Algorithm 3.1 presents the exact details.

Algorithm 3.1. ***(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 prediction model $G(z)$, plot (3.19) over a long enough range of n_y, preferably up to one time constant n_τ (i.e. the time required to reach approximately 63% of the implied steady-state response, equivalent to $n_c/3$). Select a combination of ρ and n_y which gives $u_{1,n_y} \approx \theta u_{ss}$, where θ is the input aggression factor roughly chosen within $1 < \theta \leq 5$.*

As opposed to standard tuning practices, the core benefit of Algorithm 3.1 is its explicit utilisation of transient input activity for controller tuning, which makes parameter

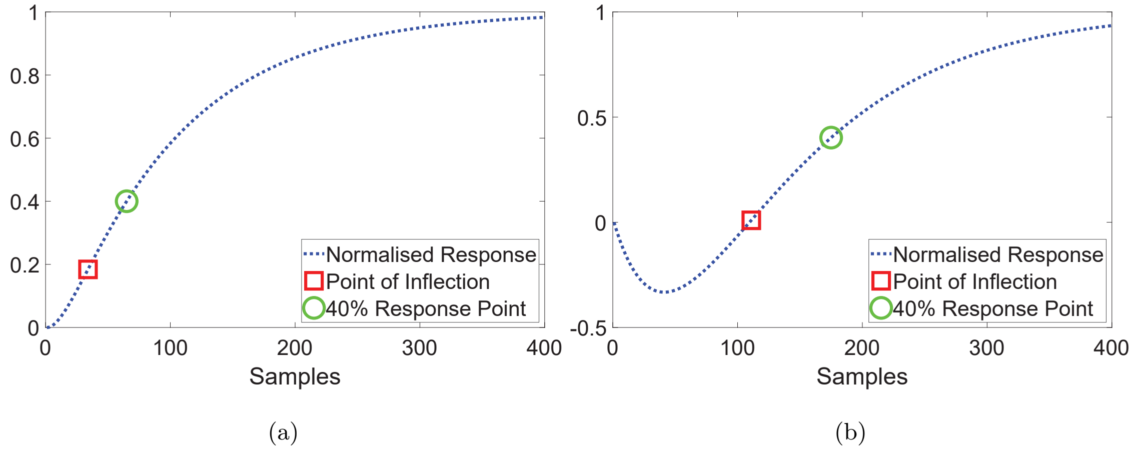


Figure 3.4: Normalised open-loop step response of (a) G_1 and (b) G_2 to obtain coincidence points based on standard tuning guidelines.

selection more meaningful and performance oriented; indeed conversations with the originator J. Richalet suggested that this is a design method he would recommend. The next section validates its usefulness with two numerical examples.

3.4 Simulation Examples

Consider open-loop stable processes G_1 and G_2:

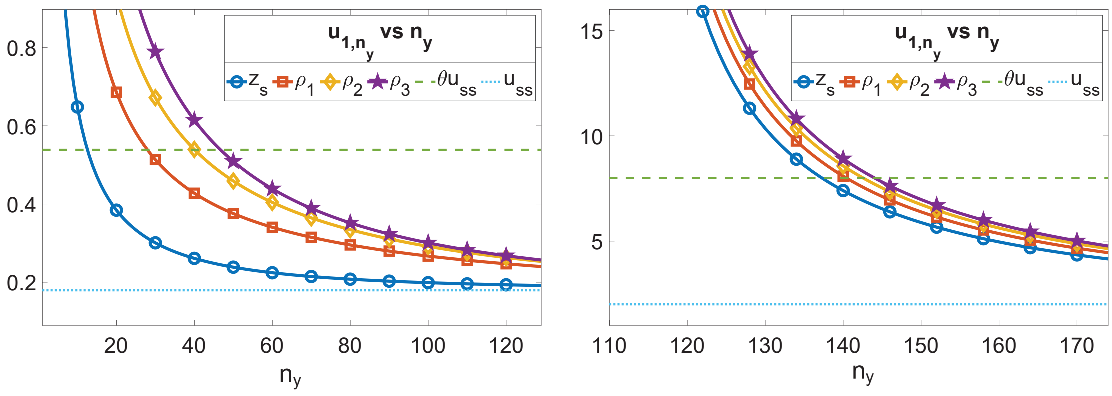
$$G_1 = \frac{0.02z^{-1} + 0.04z^{-2}}{1 - 1.4z^{-1} + 0.45z^{-2}} \qquad G_2 = \frac{-0.0098z^{-1} + 0.0099z^{-2}}{1 - 1.9702z^{-1} + 0.9704z^{-2}} \tag{3.26}$$

where G_1 is overdamped and G_2 exhibits non-minimum phase behaviour. We compare and contrast efficacy of the proposal against standard guidelines under nominal conditions; effects of undesirable perturbations, such as external disturbances and measurement noise as well as modelling uncertainties, will be studied in the following chapters.

3.4.1 Parameter Selection

Figure 3.4 shows the normalised open-loop step responses of both systems, with highlighted inflection points (red squares) and instants at which the response reaches 40% steady state value (green circles). These sample points have been selected as possible coincidence horizons as per conventional tuning guidelines (i) and (ii) respectively (see Section 3.2.3).

- For G_1: $n_y = 34$ (inflection point) and $n_y = 65$ (point of 40% steady-state)



(a) (b)

Figure 3.5: Initial input activity vs n_y for (a) G_1 with $z_s = 0.9895$, $\rho_1 = 0.9795$, $\rho_2 = 0.9695$, $\rho_3 = 0.9595$, $u_{ss} = 0.1795$, $\theta = 3$ and $R = 1$ (b) G_2 with $z_s = 0.9898$, $\rho_1 = 0.9873$, $\rho_2 = 0.9848$, $\rho_3 = 0.9823$, $u_{ss} = 2$, $\theta = 4$ and $R = 1$

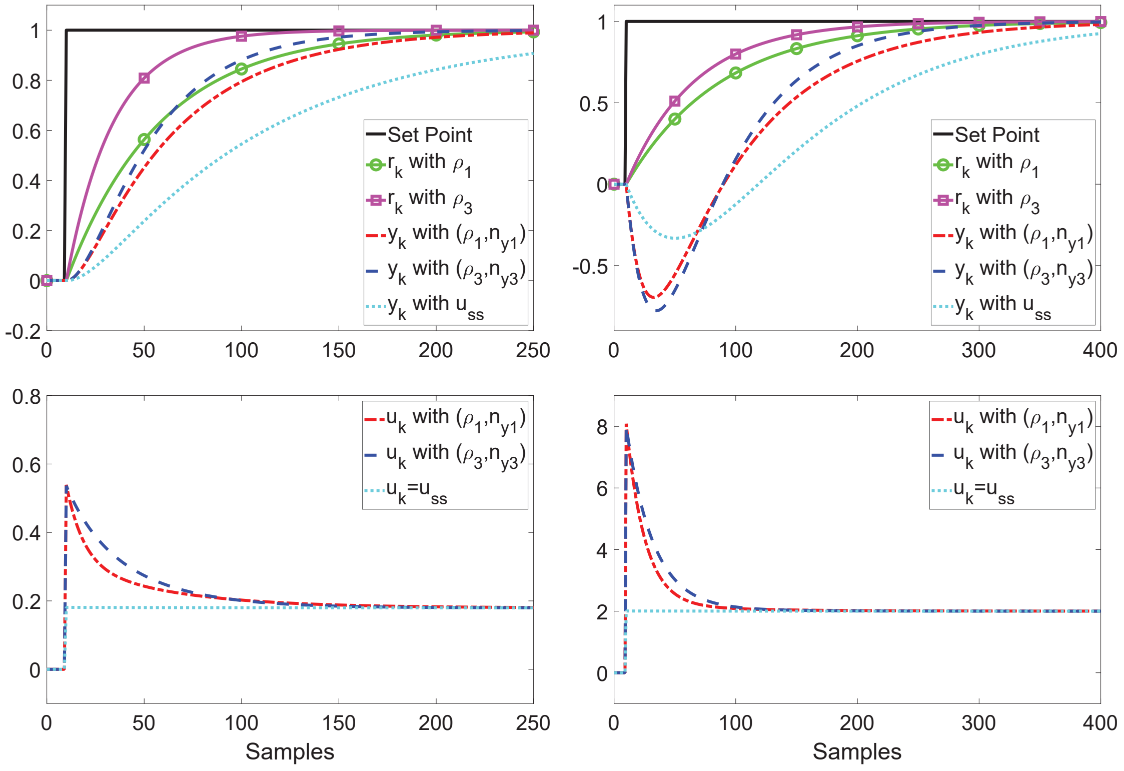
- For G_2: $n_y = 111$ (inflection point, after inversion delay $n_{dip} = 110$ samples) and $n_y = 175$ (point of 40% steady-state)

Algorithm 3.1 is applied next, with Figure 3.5 displaying the initial input activity as a function of coincidence horizon for both G_1 (Figure 3.5a) and G_2 (Figure 3.5b) with various choices of the target pole.

Evidently the choice $n_y = 1$ for overdamped G_1 produces an overactive control well above the chosen aggression factor ($\theta = 3$), necessitating longer horizons in practice. For G_2, on the other hand, coincidence can only be enforced beyond n_{dip} (i.e. $n_y > 110$), limiting the efficacy of ρ to some extent. The relative closeness between various u_{1,n_y} curves (see Figure 3.5b) despite differing target poles confirms this limitation.

An interesting feature is observed near the system's time constant (120 and 170 samples for both G_1 and G_2 respectively), when the initial inputs are almost identical, irrespective of ρ, and about twice as much as u_{ss}. Although controller tuning around this point may seem intuitive especially with tighter actuation limits, the resulting performance may be quite conservative in general, and hence may not always be acceptable.

For relatively smaller horizons, it is observed that different pairings of (ρ, n_y) can produce similar input activity. Predicted inputs with faster target poles, however, tend to



(a) (b)

Figure 3.6: Closed-loop step response of (a) G_1 with $(\rho_1 = 0.9795, n_{y1} = 28)$ and $(\rho_3 = 0.9595, n_{y3} = 47)$, (b) G_2 with $(\rho_1 = 0.9873, n_{y1} = 140)$ and $(\rho_3 = 0.9823, n_{y3} = 144)$

intercept the θu_{ss} limit at slightly longer coincidence points, apparently suggesting weaker tuning efficacy.

To investigate if this is indeed the case, we select two distinct pairs with similar initial inputs: $\rho_1 = 0.9795$,$n_{y_1} = 28$ and $\rho_3 = 0.9595$,$n_{y_3} = 47$ for G_1, and $\rho_1 = 0.9873$,$n_{y_1} = 140$ and $\rho_3 = 0.9823$,$n_{y_3} = 144$ for G_2. The next section examines how these choices affect the closed-loop performance in nominal conditions.

3.4.2 Efficacy of the Proposed Method

Figure 3.6 displays some interesting closed-loop characteristics obtained with Algorithm 3.1. Despite similar initial inputs, the combination $(\rho_3$,$n_{y3})$ provides relatively quicker transition to the set point as compared to the choice $(\rho_1$,$n_{y1})$, which is ascertained by the dynamics of u_k (bottom figures) where the inputs with faster targets proceed rather grad-

Table 3.1: Positioning of closed-loop poles obtained with the proposed tuning algorithm

	θ	(ρ,n_y)	Open-loop Poles	Closed-loop Poles
G_1	3	(0.9795,28) (0.9595,47)	0.9895,0.9331	0.9805,0.8922 0.9715,0.9266
G_2	4	(0.9873,140) (0.9823,144)	0.9898,0.9804	0.9872,0.9368 0.9822,0.9549

ually towards the implied u_{ss}.

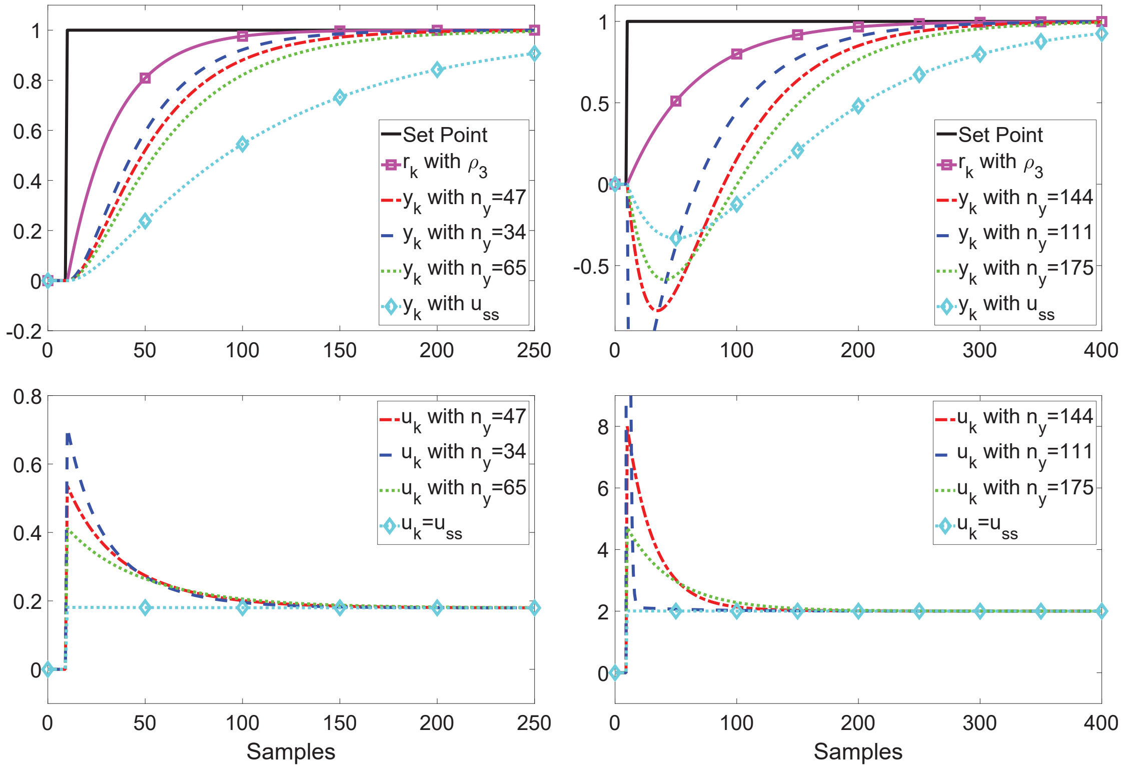
Moreover, Table 3.1 presents the achieved closed-loop pole positioning, wherein a stronger association between the selected ρ's and dominant closed-loop poles can be seen specifically for smaller n_y's, which nonetheless weakens to some extent with slightly longer horizons. However, this is not something unexpected and should not be an issue in practice as long as a sensible n_y is selected, i.e. the one that does not undermine the desirable effect of the chosen faster target pole.

3.4.3 Comparison with Standard Guidelines

Figure 3.7 presents a comparison of nominal closed-loop performance obtained using standard tuning guidelines (i) and (ii) along with the proposed tuning algorithm, and Table 3.2 tabulates the corresponding closed-loop poles and the resulting input aggression factor θ.

Evidently, enforcing coincidence at the point of inflection provides the fastest closed-loop response but with significant input activity ($\theta = 3.9$ for G_1 and 53.4 for G_2), which may not be achievable in practice. On the other hand, tuning with horizons corresponding to 40% steady state output appears somewhat cautious; input aggression remains within the range $2 \leq \theta \leq 2.5$ along with a slightly weaker link between the target and dominant closed-loop poles.

In comparison, Algorithm 3.1 clearly outperforms both conventional alternatives, exhibiting comparatively stronger tuning efficacy along with a reasonable input aggression adhering to its pre-defined values used for parameter selection. This is expected to be beneficial in practical scenarios involving process constraints, where one can simply select



(a)

(b)

Figure 3.7: Comparison of nominal closed-loop performance with various tuning methods for: (a) G_1 with $\rho_3 = 0.9595$ and $n_y = 34$ (point of inflection), $n_y = 65$ (point of 40% steady-state output) and $n_y = 47$ (Algorithm 3.1), and (b) G_2 with $\rho_3 = 0.9823$ and $n_y = 111$ (point of inflection), $n_y = 175$ (point of 40% steady-state output) and $n_y = 144$ (Algorithm 3.1)

a suitable θ according to the underlying actuation capacity for a more realistic parameter selection.

3.5 Chapter Summary

The key highlights of this chapter are as follows:

- The analytics associated with conventional PFC are reviewed in Section 3.1, which shows straightforward control development for processes with dead-times and constraints.

- Two parameters ρ and n_y are identified as the main performance tuning knobs,

Table 3.2: Comparison of closed-loop poles and input activity with different tuning methods

	Method	ρ	n_y	Open-loop Poles	Closed-loop Poles	θ
G_1	Inflection Point	0.9595	34	0.9895,0.9331	$0.9667, 0.9171$	3.9
	40% Steady State		65		$0.9768, 0.9309$	2.3
	Algorithm 3.1		47		$0.9715, 0.9266$	3.0
G_2	Inflection Point	0.9823	111	0.9898,0.9804	$0.9821, 0.4029$	53.4
	40% Steady State		175		$0.9828, 0.9727$	2.4
	Algorithm 3.1		144		$0.9822, 0.9549$	4.0

which work exceptionally well for simple processes but may become significantly ineffective for slightly more involved higher order dynamics. Various conventional tuning methods for such systems along with their shortcomings are discussed in Section 3.2.

- An improved parameter selection method based on a global search (implicitly recommended by J. Richalet) is proposed for significantly overdamped and non-minimum systems in Section 3.3, which explicitly utilises the predicted input aggression for a more meaningful and performance oriented controller tuning.

- Simulation results, presented in Section 3.4, demonstrate comparatively superior performance efficacy of the proposal against the commonly employed standard tuning guidelines presented in Section 3.2.

Chapter 4

Relative PFC

This chapter presents a new PFC algorithm with simplified tuning as the primary contribution. Section 4.1 first introduces the notion of relative tuning before formally presenting the Relative PFC algorithm in Section 4.2. Section 4.3 discusses whether utilising a simple Laguerre function for input parametrisation could be as beneficial as it usually is within the conventional PFC formulation. Two simulation studies are presented in Section 4.4 to analyse and validate the efficacy of the proposal. Finally the chapter concludes in Section 4.5 which summarises the key findings of the discussion.

4.1 Concept of Relative Tuning in PFC

Before introducing the readers to the concept of *relative* tuning, it is important to highlight the core weaknesses and deficiencies of the conventional PFC algorithm that have led to the development of various modifications (see for example [19–22]) in order to ensure a more meaningful and effective performance tuning. These challenges are summarised in the following section.

4.1.1 Weaknesses of Conventional PFC

Firstly, and perhaps ironically, it is noted that the use of constant future input for control synthesis, which was originally proposed to simplify computations [8, 10, 11], in essence embeds open-loop characteristics in the closed-loop decision making, and therefore may not always yield desirable results, or could even backfire in applications that involve poorly damped and/or unstable prediction dynamics [19]. Although a commonly deployed remedy to rectify this deficiency is to implement a differently parametrised input function [20, 22], doing so often increases design complexity negating the core notion of simplicity and cost-effectiveness associated with PFC. To address this issue, the next chapter will present some simple and inexpensive modifications by utilising the concept of pre-stabilised or closed-loop prediction dynamics within the PFC framework.

Secondly, tuning in conventional PFC revolves around two main parameters, namely the target pole ρ and the coincidence point n_y, that directly influence the closed-loop behaviour but their peculiar relationship often makes an intuitive selection somewhat difficult especially with slightly complex higher order dynamic systems. The previous chapter has discussed this issue in detail wherein an improved parameter selection method was developed to facilitate a more meaningful performance tuning. Nevertheless, this and other standard approaches have one major limitation in that the ultimate tuning decision relies on a form of global search conducted in the parameter space, which may not always be as intuitive or straightforward to implement. Besides, recent studies have reported significant anomalies in the standard definition of coincidence point which makes inconsistent use of target/disturbance information and often results in additional lag in the responses [69, 70]. Due to these reasons, the tuning is not as effective as desired.

This chapter specifically focuses on the second challenge discussed above, and therefore presents a *relative* PFC algorithm with simplified tuning as the core contribution. More specifically, the idea is to replace the fairly ambiguous two-parameter tuning in conventional PFC with something simpler and far more *transparent.* The following section presents the concept in detail.

4.1.2 The Relative Tuning Proposal

Recall from Section 3.2 that the nominal relationship bewteen the initial input activity u_{1,n_y} and the controller parameters (ρ, n_y) is given by:

$$u_{1,n_y} = \frac{R}{h_{n_y}}(1 - \rho^{n_y}) \tag{4.1}$$

where, as per Algorithm 3.1, it is recommended to select those values of (ρ, n_y) that provide $u_{1,n_y} \approx \theta u_{ss}$ with $u_{ss} = R/G(1)$. Nevertheless, instead of globally searching for the appropriate (ρ, n_y) pairing, the proposal is to use θ directly as the core tuning parameter which appears to be far simpler than the conventional tuning methods.

Theorem 4.1. *The closed-loop performance can be tuned relative to the open-loop (or steady-state) benchmark input u_{ss} by simply using the input aggression factor θ as the main tuning parameter rather than finding ρ and n_y globally on absolute terms.*

Proof. Using the expression $u_{1,n_y} = \theta R/G(1)$ in (4.1) results in:

$$\theta = \frac{u_{1,n_y}}{u_{ss}} = \frac{G(1)}{h_{n_y}}(1 - \rho^{n_y}) \tag{4.2}$$

from which it is obvious whether one uses the left-hand side with θ or the right-hand side with (ρ, n_y), the relationship in (4.2) will tune the closed-loop performance by suitably adjusting the initial input u_{1,n_y} with respect to the mean-level (or steady-state) control u_{ss}. Although both methods attempt to fulfill the same performance specification, using θ directly as the sole tuning parameter is apparently far more efficient and straightforward than the conventional tuning methods. □

Hence, the proposed relative tuning is expected to be beneficial in the following ways:

(i) The controller tuning would simplify to merely answering just one trivial question, i.e. how much faster (or slower) one wishes the closed-loop system to respond as compared to the open-loop or mean-level behaviour?

(ii) No explicit requirement of ρ and n_y would by default prevent the incorrect use of feedforward information in the control law, and thus rectify the performance issue arising from the inadvertent addition of undesirable lag in the closed-loop response.

The technical details of the proposal are available in the attached UKACC 2022 publication (see Appendix G), which will now be presented in the following section.

4.2 Relative PFC Algorithm

Just like the conventional PFC (see Section 3.1 for details), the relative PFC (RPFC) algorithm too relies on three key elements for implementation: a suitable prediction model, a benchmark response and a mechanism to ensure bias-free predictions. Nevertheless, instead of using an ideal exponential trajectory and the concept of coincidence, an open-loop (or mean-level) system response is utilised as a benchmark for the closed-loop performance tuning.

4.2.1 RPFC Control Law

Let us consider the discrete-time transfer function model (3.3) of a well-damped open-loop process $G(z)$, i.e. $a(z)\hat{y}_k = b(z)u_k$, whose recursive use yields the following prediction expression:

$$y_{k+i|k} = \mathbf{H}_i u_k + \mathbf{P}_i \underset{\leftarrow}{\mathbf{u}}_{k-1} + \mathbf{Q}_i \underset{\leftarrow}{\hat{\mathbf{y}}}_k + d_k; \quad d_k = y_k - \hat{y}_k \tag{4.3}$$

where $\mathbf{H}_i$, $\mathbf{P}_i$ and $\mathbf{Q}_i$ are determined from the model $a(z)$ and $b(z)$. If one selects $u_{k+i} = u_{ss}$, $\forall i \geq 0$ where u_{ss} is the expected steady-state input, the control law then obtained is the so-called mean level PFC [11], which mirrors the open-loop transient performance in the closed-loop response along with offset free tracking. The implied u_{ss} for $G(z)$ (in nominal conditions) is obtained from the following relationship:

$$u_{ss} = \frac{R}{G(1)} \qquad \because y_{ss} = y(1) = R \tag{4.4}$$

where $G(1)$ is the steady-state system gain given in (3.22). 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 open-loop step response, although as mentioned previously in Chapter 3, the choice $i \approx n_c$ is generally sufficient. Hence, with target R and $u_{k+i} = u_{ss}$ $\forall i \geq 0$, the predicted error converges as follows:

$$e_{ss}(k+i) = R - (h_{n_c} u_{ss} + \mathbf{P}_{n_c} \underset{\leftarrow}{\mathbf{u}}_{k-1} + \mathbf{Q}_{n_c} \underset{\leftarrow}{\hat{\mathbf{y}}}_k + d_k); \tag{4.5}$$

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

$$e(k+i) = R - (h_{n_c} u_k + \mathbf{P}_{n_c} \underset{\leftarrow}{\mathbf{u}}_{k-1} + \mathbf{Q}_{n_c} \underset{\leftarrow}{\hat{\mathbf{y}}}_k + d_k) \tag{4.6}$$

where in (4.5) and (4.6) above, $h_{n_c} = \mathbf{H}_{n_c}\mathbf{L}_{n_c}$ with $\mathbf{L}_{n_c} = [1 \quad 1 \ldots 1]^T_{1\times n_c}$. Thus, to ensure a faster convergence than the benchmark (4.5), one must select a u_k correspondingly more active than u_{ss}. Lemma 4.2 below formalises this concept.

Lemma 4.2. *In the nominal state and zero initial conditions, the choice* $u_k = \theta u_{ss}$ *for the target* R *provides an error convergence which is* γ *times* (4.5) *such that:*

$$\gamma = \frac{G(1) - h_{n_c}\theta}{G(1) - h_{n_c}} \tag{4.7}$$

Proof. With d_k, $\underleftarrow{\mathbf{u}}_{k-1}$ and $\underleftarrow{\hat{\mathbf{y}}}_k$ all zero, and $u_k = \theta u_{ss}$ the initial errors are related as follows:

$$R - h_{n_c}\theta u_{ss} = \gamma(R - h_{n_c}u_{ss})$$

using (4.4) then implies:

$$1 - \frac{h_{n_c}\theta}{G(1)} = \gamma\left(1 - \frac{h_{n_c}}{G(1)}\right)$$

which simplifies to (4.7) after simple manipulations. □

Theorem 4.3. *For the chosen input activity* θ *and the error convergence* γ *defined in* (4.7) *above, the Relative PFC (RPFC) control law is given by:*

$$u_k = \gamma u_{ss} + \frac{1-\gamma}{h_{n_c}}\left[R - \left(\mathbf{P}_{n_c}\underleftarrow{\mathbf{u}}_{k-1} + \mathbf{Q}_{n_c}\underleftarrow{\hat{\mathbf{y}}}_k + d_k\right)\right] \tag{4.8}$$

Proof. Using Lemma 4.2 and equations (4.5)-(4.6), it is clear that:

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

implying that:

$$R - (h_{n_c}u_k + \mathbf{P}_{n_c}\underleftarrow{\mathbf{u}}_{k-1} + \mathbf{Q}_{n_c}\underleftarrow{\hat{\mathbf{y}}}_k + d_k) = \gamma\left[R - (h_{n_c}u_{ss} + \mathbf{P}_{n_c}\underleftarrow{\mathbf{u}}_{k-1} + \mathbf{Q}_{n_c}\underleftarrow{\hat{\mathbf{y}}}_k + d_k)\right]$$

which simplifies to the control law (4.8). □

Notably, the controller (4.8) is an alternative representation of the standard PFC control law derived in (3.12), albeit with a single tuning parameter γ, obtained directly from the chosen aggression factor θ, which replaces the conventional parameters ρ and n_y to ensure considerably simpler and efficient controller tuning as compared to the conventional PFC algorithm.

4.2.2 Parameter Tuning in RPFC

The primary benefit of the proposed RPFC algorithm is obvious from the preceding discussion, as it reduces the closed-loop performance tuning to simply answering one statement: how fast or slow one wants the closed-loop system to respond as compared to the benchmark open-loop behaviour? More specifically, the proposal provides three distinct tuning choices, i.e. $0 < \theta < 1$, $\theta = 1$ and $\theta > 1$, each with the following interpretation:

- $0 < \theta < 1$ 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.

Nevertheless, it is advised not to select too large θ or the initial input could be too aggressive to achieve practically. Although a commendable closed-loop performance is usually attainable with θ up to 5 (given a well-damped and stable open-loop dynamic behaviour), clearly a more practical approach is to adhere to the possible set-point changes (recall that the initial input is directly proportional to R) as well as the actuating capacity of the physical system while making the ultimate tuning selection.

4.2.3 Managing Deadtimes and Constraints

In addition to significant tuning simplifications, another benefit of the proposal is that it handles process delays and constraints in the conventional manner, as discussed in Section 3.1, without requiring any further modifications in the control law. Hence to incorporate a deadtime of n_d samples, one simply needs to compute the offset correcting term in (4.8) such that $d_k = y_k - \hat{y}_{k-n_d}$. As for constraint handling, it is sufficient to verify the $2(2 + n_c)$ linear inequalities given in (3.17) at each sample k in the standard manner.

4.2.4 Reative PFC in a Nutshell

The following algorithm sums up the foregoing discussion:

Algorithm 4.1. ***(Relative PFC Design)*** *For a given target R, first compute the implied steady-state input u_{ss} from* (4.4)*. Next select an appropriate aggression factor θ as per the desired control specification and calculate the error convergence rate γ using* (4.7)*. Then at each sample k, compute u_k from the relative PFC control law derived in* (4.8)*, and update the plant and the model accordingly.*

4.3 Including Laguerre Function in Relative PFC

For completeness, this section analyses whether a differently shaped input utilising a simple first-order Laguerre polynomial in the control law could further enhance the tuning efficacy of the relative PFC algorithm. According to some recent studies (see for example [20, 74]), the suggested modification within conventional PFC typically improves the closed-loop tuning efficacy by enhancing the overall prediction consistency. Nevertheless as a downside, the non-constant nature of the input parametrisation also induces slight complications in constraint validation, which may also increase the overall cost and computational burden by significant proportions.

Note that the discussion in the following section is based on the proposal presented in the ECC 2022 paper [34] which is attached in Appendix H of this thesis.

4.3.1 Laguerre RPFC (LRPFC) Control Law

A first-order Laguerre function, in essence, represents an ideal exponential decay which parametrises the future input such that:

$$\underrightarrow{\mathbf{u}}_{k} = \mathbf{H}_{\lambda}\eta + \mathbf{L}_{n_c}u_{ss}; \quad \mathbf{H}_{\lambda} = [1 \quad \lambda \quad \dots \quad \lambda^{n_c-1}]^T_{n_c} \tag{4.9}$$

where λ is the decay factor to be chosen and η is the new degree-of-freedom that will be computed from the RPFC algorithm. Hence with this input parametrisation, the predicted error converges as follows:

$$e_L(k+i) = R - (h_{\lambda}\eta + h_{n_c}u_{ss} + \mathbf{P}_{n_c}\underleftarrow{\mathbf{u}}_{k-1} + \mathbf{Q}_{n_c}\underleftarrow{\hat{\mathbf{y}}}_{k} + d_k) \tag{4.10}$$

where $h_\lambda = \mathbf{H}_{n_c}\mathbf{H}_\lambda$. Therefore for a given aggression factor θ, the parameter γ_L is found to be (using a similar procedure as shown in Lemma 4.2):

$$\gamma_L = 1 - \frac{(\theta - 1)h_\lambda}{G(1) - h_{n_c}} \tag{4.11}$$

which finally leads to the Laguerre RPFC control law:

$$\eta = \frac{1-\gamma_L}{h_\lambda}\left[R - \left(h_{n_c}u_{ss} + \mathbf{P}_{n_c}\underleftarrow{\mathbf{u}}_{k-1} + \mathbf{Q}_{n_c}\underleftarrow{\hat{\mathbf{y}}}_k + d_k\right)\right] \tag{4.12}$$

with $u_k = \eta + u_{ss}$.

Remark 4.4. *The detailed procedure to perform constraint management with Laguerre PFC is well documented in literature (see for example the references [20, 74]), which is applicable in an exact manner to the Laguerre RPFC algorithm derived here.*

4.3.2 Efficacy of Laguerre RPFC

Although including a first-order Laguerre function is proven effective in the conventional PFC setting [20, 74], surprisingly the following analysis shows that both relative PFC variants, i.e. RPFC and LRPFC, are equivalent with no obvious benefit of utilising an exponential decay in the overall closed-loop performance.

Lemma 4.5. *For a given R and θ, the indirect tuning parameters γ and γ_L are related with each other as per the following expression:*

$$\frac{1-\gamma_L}{1-\gamma} = \frac{h_\lambda}{h_{n_c}} \tag{4.13}$$

Proof. From (4.11), it is easy to show that:

$$1 - \gamma_L = \frac{(\theta - 1)h_\lambda}{G(1) - h_{n_c}} \tag{4.14}$$

Similarly, one may deduce the following result from (4.7):

$$1 - \gamma = \frac{(\theta - 1)h_{n_c}}{G(1) - h_{n_c}} \tag{4.15}$$

Hence, dividing (4.14) by (4.15) leads to the required expression given in (4.13). □

Theorem 4.6. *The RPFC and LRPFC control laws derived in* (4.8) *and* (4.12) *respectively are equivalent.*

Proof. Using (4.13) in the LRPFC control law (4.12) leads to:

$$\eta = \frac{1-\gamma}{h_{n_c}}\left[R - \left(h_{n_c}u_{ss} + \mathbf{P}_{n_c}\underset{\leftarrow}{\underline{\mathbf{u}}}_{k-1} + \mathbf{Q}_{n_c}\underset{\leftarrow}{\hat{\mathbf{y}}}_{k} + d_k\right)\right] \tag{4.16}$$

which after simple manipulations can be written as:

$$\eta + u_{ss} = \gamma u_{ss} + \frac{1-\gamma}{h_{n_c}}\left[R - \left(\mathbf{P}_{n_c}\underset{\leftarrow}{\underline{\mathbf{u}}}_{k-1} + \mathbf{Q}_{n_c}\underset{\leftarrow}{\hat{\mathbf{y}}}_{k} + d_k\right)\right] \tag{4.17}$$

which is equivalent to the RPFC control law (4.8) as $u_k = \eta + u_{ss}$. □

Hence, reparametrising the future input with a first-order Laguerre function in relative PFC has no obvious impact on the closed-loop performance tuning, which clearly is a significant result because the current conventional PFC literature in this regard suggests otherwise [20, 74]. Perhaps this could be attributed to the more efficient usage of feedforward information in RPFC, which in the case of conventional PFC is not true as reported in these references [69, 70].

4.4 Simulation Examples

Let us reconsider the numerical examples from Section 3.4:

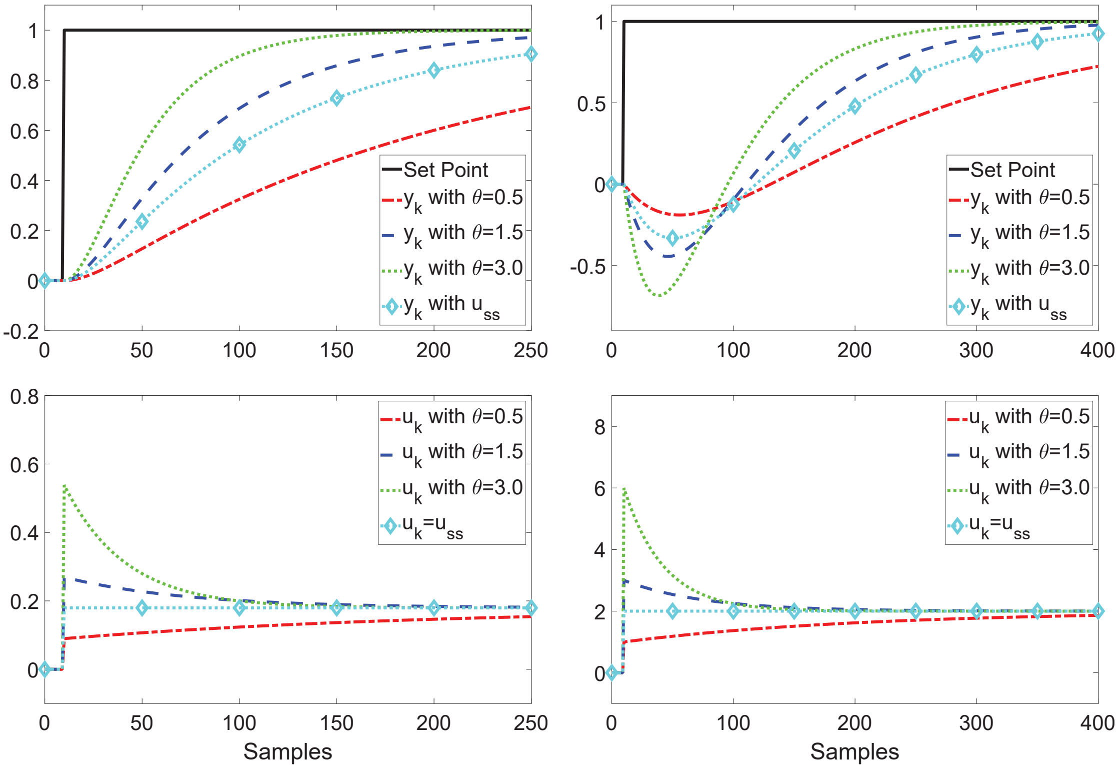
$$G_1 = \frac{0.02z^{-1} + 0.04z^{-2}}{1 - 1.4z^{-1} + 0.45z^{-2}} \qquad G_2 = \frac{-0.0098z^{-1} + 0.0099z^{-2}}{1 - 1.9702z^{-1} + 0.9704z^{-2}} \tag{4.18}$$

where G_1 represents an overdamped system and G_2 exhibits significant non-minimum phase behaviour. Note that the following analysis utilises the initial input plots in Figure 3.5 for tuning purposes.

4.4.1 Analysis of Tuning Efficacy

The tuning efficacy of the proposed RPFC algorithm for the systems G_1 and G_2 has been analysed, with the results shown in Figure 4.1. It is clear that the parameter θ is successful in slowing down (with $\theta = 0.5$) or speeding up (with $\theta = 1.5, 3.0$) the closed-loop response by correspondingly changing the controller input as compared to the benchmark u_{ss}.

Evidently, performance tuning in this way is far more straightforward and meaningful than finding ρ and n_y with the conventional methods (for example see Section 3.2.3),



(a) (b)

Figure 4.1: Tuning efficacy of RPFC with $\theta = [0.5, 1.5, 3.0]$ for (a) G_1 and (b) G_2 under nominal conditions.

which often requires tedious offline analysis of open-loop characteristics and yet merely provides multiple parameter choices with no definitive selection criterion matching the desired performance specification.

4.4.2 Comparisons with Algorithm 3.1

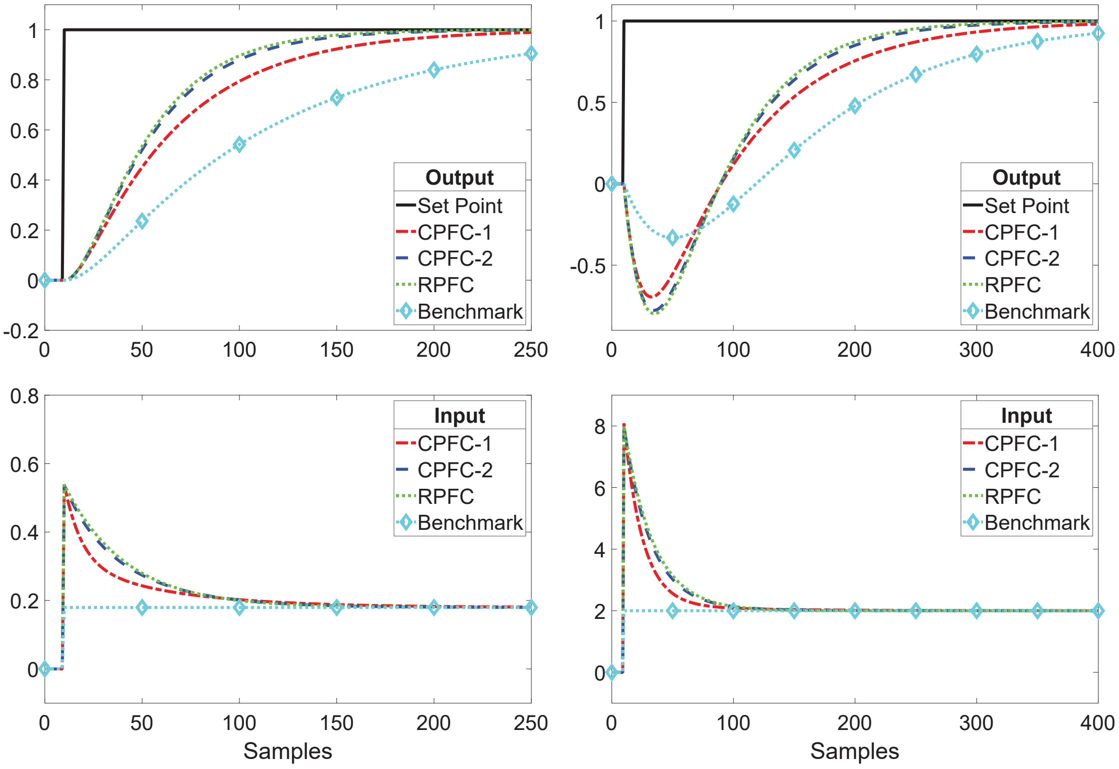
In this section, we compare and analyse the efficacy of RPFC with that of a conventional PFC tuned using Algorithm 3.1. For a given input aggression θ, two distinct pairs of (ρ, n_y) have been selected (from Section 3.4.1), which are given in Table 4.1.

The simulation results are shown in Figure 4.2, from which it is evident that while the implementation of Algorithm 3.1 ensures adherence to the predefined input aggression θ, slower target poles, i.e. ρ_1 in both examples, generally tend to decelerate the closed-loop response and therefore should not be preferred. Instead, the faster pole choice with

Table 4.1: Selected parameters of RPFC and CPFC for analysis in Section 4.4.2

Example	RPFC θ	CPFC-1 (ρ_1, n_{y_1})	CPFC-2 (ρ_2, n_{y_2})
G_1	3.0	(0.9795,28)	(0.9595,47)
G_2	4.0	(0.9873,140)	(0.9823,144)

relatively longer coincidence horizon, i.e. the pairing (ρ_2, n_{y_2}), competes much closely with the relative PFC, and hence a comparatively better option when tuning a conventional PFC with Algorithm 3.1.



(a) (b)

Figure 4.2: Comparison of tuning efficacy between Conventional and Relative PFC control laws under nominal conditions for (a) G_1, and (b) G_2 with the controller parameters given in Table 4.1.

Table 4.2: Selected parameters of RPFC and CPFC for analysis in Section 4.4.3

Example	RPFC-1 θ_1	RPFC-2 θ_2	CPFC-1 (ρ_1,n_{y_1})	CPFC-2 (ρ_2,n_{y_2})
G_1	3.0	2.0	(0.9595,47)	(0.9595,78)
G_2	4.0	2.5	(0.9823,140)	(0.9823,170)

4.4.3 Performance Analysis with Uncertainties

This section analyses the closed-loop performance in the presence of uncertainties, with the results shown in Figures 4.3 and 4.4 for disturbance rejection and noise filteration (with modelling mismatches) respectively. Note that the Conventional PFC in both examples is tuned using the faster ρ choice but at different input aggression levels resulting in two different coincidence horizons (see Table 4.2).

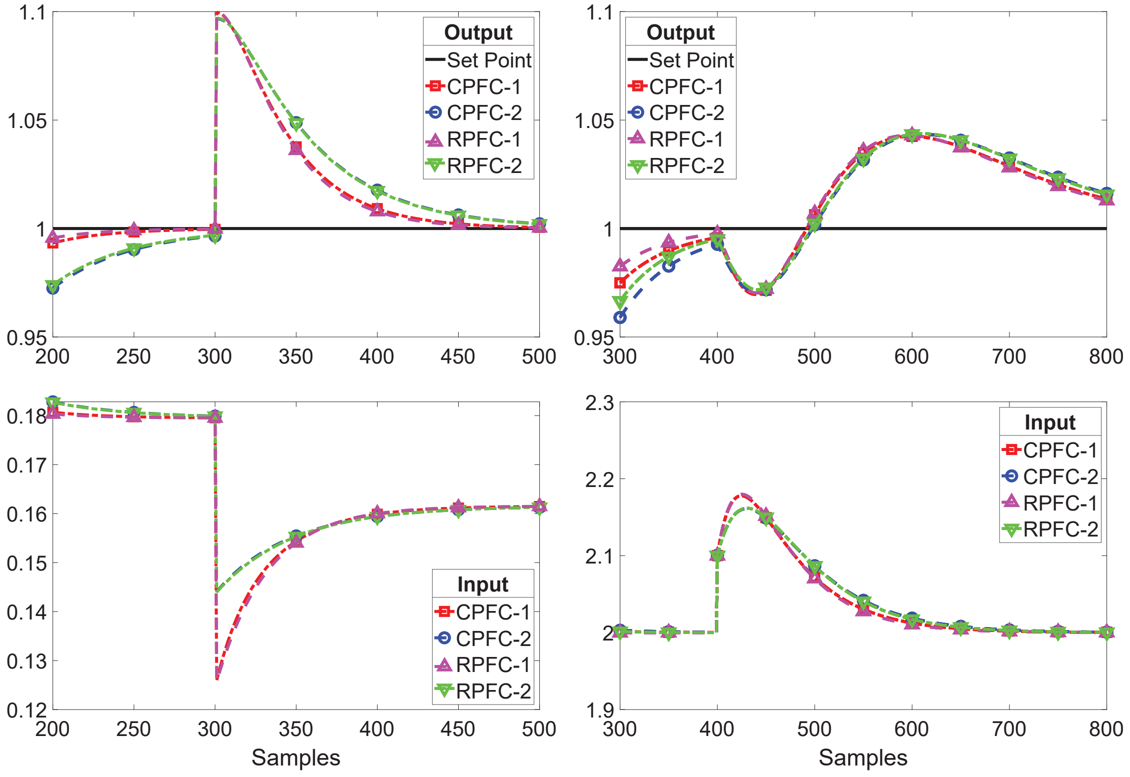
In both scenarios, no remarkable difference in the closed-loop performance is observed between both RPFC and CPFC, which further authenticates the findings from Figure 4.2. Despite these similarities, it is reiterated that controller tuning in RPFC is evidently far more efficient and straightforward as compared to CPFC, which in these examples required relatively tedious offline plotting and analysis of various u_{1,n_y} curves for parameter selection in Section 3.4.1.

4.5 Chapter Summary

The key highlights of this chapter are summarised below:

- The concept of relative tuning is introduced in Section 4.1, which is fundamentally based on the proposed Algorithm 3.1 of parameter selection but in comparison only uses the input aggression factor θ as the main tuning parameter replacing ρ and n_y from the original formulation.

- The Relative PFC control law (4.8), developed in Section 4.2, is in essence an alternative representation of the Conventional PFC (3.12) but with significantly simpler design traits and far more intuitive controller tuning.

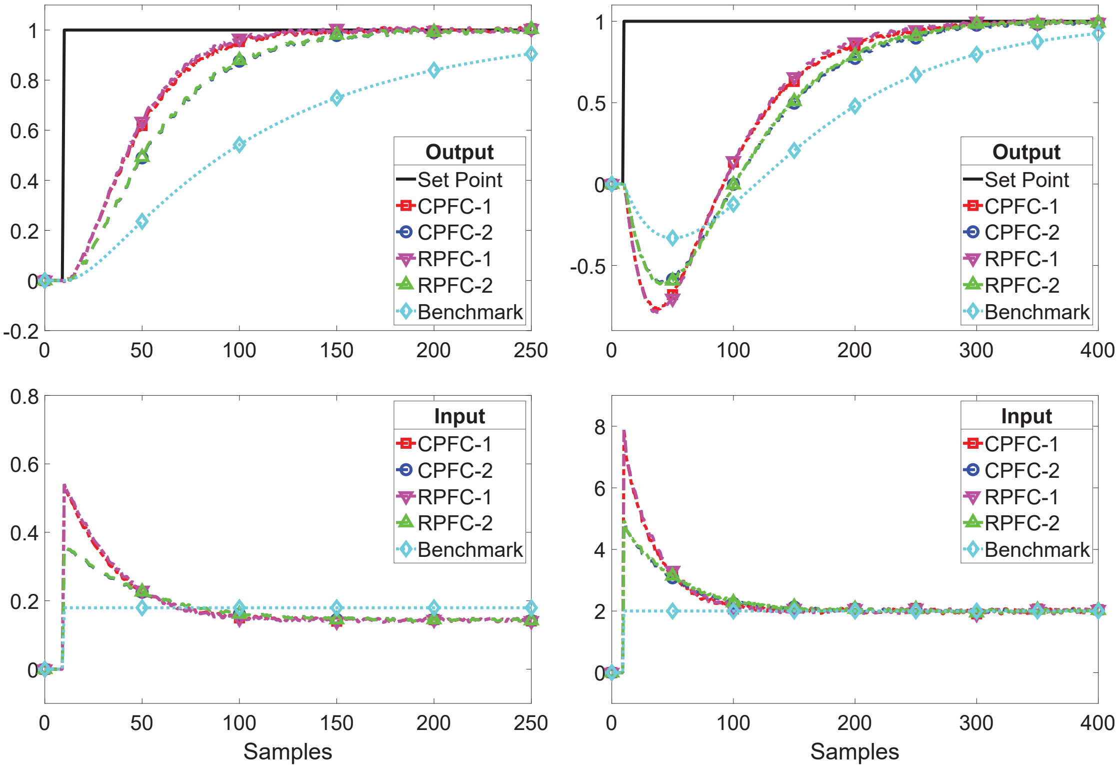


(a) (b)

Figure 4.3: Comparison of disturbance rejection between Conventional and Relative PFC control laws for (a) G_1 with 10% output disturbance, and (b) G_2 with 5% input disturbance using controller parameters listed in Table 4.2.

- It is shown in Section 4.3 that a different form of future input parametrisation using a first-order Laguerre function does not yield any further improvements in the proposed relative PFC algorithm. In fact, both control laws, i.e. simple RPFC (4.8) and Laguerre RPFC (4.12), have been proven equivalent for a given set-point and θ.

- Simulation analysis in Section 4.4 validates the efficacy of the relative PFC algorithm with two numerical examples. Furthermore, comparisons with conventional PFC, tuned using Algorithm 3.1, generally demonstrate similar closed-loop performance if one opts for faster target dynamics along with relatively longer points of coincidence.



(a) (b)

Figure 4.4: Comparison of tuning efficacy between Conventional and Relative PFC control laws along with measurement noise and plant-model mismatches for (a) G_1 with 25% multiplicative uncertainty, and (b) G_2 with an unmodelled pole at $z = 0.5$ using controller parameters listed in Table 4.2.

Chapter 5

Pre-stabilised PFC

This chapter introduces the concept of pre-stabilisation and discusses how it improves the functionality of PFC in relatively challenging applications. Section 5.1 first presents the rationale behind pre-stabilisation in the context of PFC, followed by the development of a generic control design framework in Section 5.2. Various pre-compensation strategies are discussed next in Section 5.3, before presenting the proposal for constraint management with pre-stabilisation in Section 5.4. The simulation case studies follow in Section 5.5 which analyses the efficacy of various pre-compensation schemes utilised within the framework of pre-stabilised PFC. Finally, the chapter concludes in Section 5.6.

5.1 Rationale Behind Pre-stabilisation

Industrial processes exhibiting unstable and/or poorly damped dynamics are generally difficult to control with simple and low-cost approaches, such as PID, and therefore often require relatively expensive and more sophisticated designs for successfull operation [5]. Unsurprisingly, the simplistic design attributes mean that PFC too struggles and performs rather poorly in these applications, primarily due to the use of constant future input within predictions which lacks flexibility to handle such challenging behaviour in an efficient manner [19, 22, 85]. Moreover, parameter tuning for such systems is far less intuitive, and in many cases it may not be possible to guarantee the standard feasibility results either [11, 19]. While the closed-loop system may occasionally work due to the use of the receding horizon, the performance is indeed unreliable and prone to failure especially under the influence of external perturbations and/or tight actuation limits.

To tackle these shortcomings, a commonly deployed remedy is to implement a more flexible input parametrisation, such as [22, 85], which in this study is achieved via pre-stabilisation of difficult open-loop dynamics [45, 109]. Pre-stabilisation is, in essence, a two-stage design methodology wherein the undesirable prediction model is first compensated using a well understood classical feedback control approach before implementing PFC in a cascade structure. Although the use of an additional inner control loop often complicates the standard constraint validation to some extent [10, 11], in most cases pre-stabilisation successfully accomplishes far more reliable and consistent performance tuning (see for example the Processes 2021 paper [30] in Appendix D), which significantly outweighs the slight intricacy in constraint handling that may arise due to the use of inner loops within the design.

The following sections will present a concise summary of the proposal; interested readers are referred to the accompanying publications [27–33] attached in Appendices A-G for an in depth technical analysis and discussion.

5.2 Framework of Pre-stabilised PFC

For challenging dynamics, the concept of pre-stabilised PFC works systematically in two simple steps: reparametrising the degrees-of-freedom by forming stable and well-damped closed-loop predictions using a classical feedback compensator, and implementing PFC on the pre-compensated model to compute the reparametrised decision variable. The framework 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 technical details, available in the ACA 2022 paper [31] attached in Appendix E, will be summarised next in the following sections.

5.2.1 Reparametrising the Degree-of-Freedom

Consider a challenging open-loop process modelled as a n^{th}-order strictly proper transfer function $G(z)$ given as:

$$G(z) = z^{-n_d} G_0(z); \quad G_0(z) = \frac{\hat{y}_k}{\hat{u}_k} = \frac{b(z)}{a(z)} \tag{5.1}$$

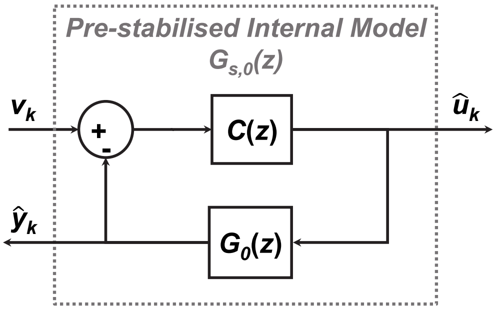


Figure 5.1: Pre-stabilisation loop structure.

where $\hat{y}_k$ and $\hat{u}_k$ are the model output and input respectively, the polynomials $a(z)$ and $b(z)$ represent plant estimates with $a(z) = 1 + a_1 z^{-1} + \cdots + a_n z^{-n}$, $b(z) = b_1 z^{-1} + \cdots + b_n z^{-n}$, n_d is the process delay in samples, and $a(z)$ factors the open-loop unstable and/or complex conjugate poles.

In order to obtain stable and well-damped output predictions, the delay-free model $G_0(z)$ is compensated using a m^{th} order bi-proper feedback controller $C(z)$, as shown in Figure 5.1. Note that:

$$C(z) = \frac{q(z)}{p(z)} \tag{5.2}$$

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 delay-free prediction model is then given by:

$$G_{s,0}(z) = \frac{\hat{y}_k}{v_k} = \frac{q(z)b(z)}{p(z)a(z) + q(z)b(z)} = \frac{\beta(z)}{\alpha(z)} \tag{5.3}$$

where v_k is now the decision variable computed via an outer PFC loop as shown in Figure 5.2. 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, i.e. when $d_k = 0$) as detailed in [31] (see Appendix E for complete derivation). Here, we state the final result:

$$u_k = B_0 v_k + f_k; \quad f_k = -\mathbf{A}\underleftarrow{\mathbf{u}}_{k-1} + \mathbf{B}\underleftarrow{\mathbf{v}}_{k-1} + \mathbf{E}\underleftarrow{\mathbf{d}}_{k} \tag{5.4}$$

where vectors $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{E}$ are obtained from the parameters $a(z)$, $\alpha(z)$, $p(z)$ and $q(z)$ as follows:

$$\begin{aligned} A(z) &= \alpha(z)p(z) = 1 + A_1 z^{-1} + A_2 z^{-2} + \ldots \\ B(z) &= q(z)a(z)p(z) = B_0 + B_1 z^{-1} + B_2 z^{-2} + \ldots \\ E(z) &= -\alpha(z)q(z) = E_0 + E_1 z^{-1} + E_2 z^{-2} + \ldots \end{aligned} \tag{5.5}$$

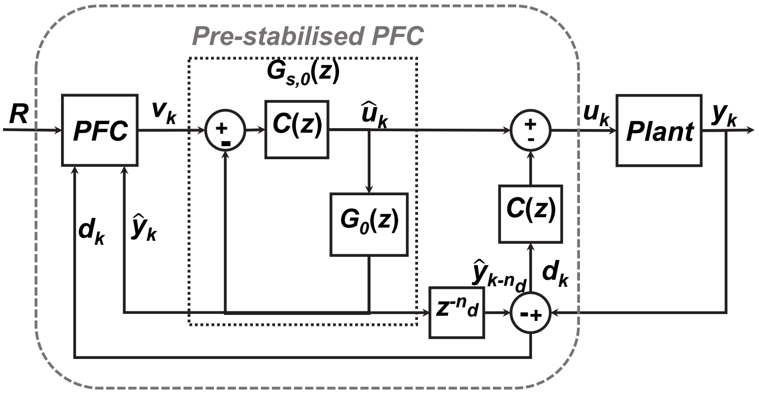


Figure 5.2: Proposed Pre-stabilised PFC control architecture.

Evidently after pre-stabilisation, the degree-of-freedom is reparametrised appropriately, given a suitable inner controller, which can now work easily with the difficult open-loop dynamics.

Remark 5.1. *If $C(z)$ is positioned in the feedback path of the pre-compensation loop rather than the forward path as shown in Figure 5.1, the numerator of the pre-stabilised model $G_{s,0}(z)$ in (5.3) changes to $\beta(z) = p(z)b(z)$. Hence, the computation of $B(z)$ in (5.5) must also be updated accordingly with $B(z) = p(z)a(z)p(z)$. In this case, the coefficient B_0 will be equal to 1 as both $p(z)$ and $a(z)$ are monic polynomials.*

5.2.2 Establishing PPFC Control Law

The PPFC algorithm works similarly to the ordinary PFC but implemented on the pre-stabilised model dynamics. Hence instead of using the difficult open-loop dynamic model $G_0(z)$, output predictions are obtained recursively from $G_{s,0}(z)$ i.e. $\alpha(z)\hat{y}(z) = \beta(z)v(z)$ such that:

$$y_{k+n_y+n_d|k} = \mathbf{H}_{n_y} \underset{\rightarrow}{\mathbf{v}}_k + \mathbf{P}_{n_y} \underset{\leftarrow}{\mathbf{v}}_{k-1} + \mathbf{Q}_{n_y} \underset{\leftarrow}{\hat{\mathbf{y}}}_k + d_k \tag{5.6}$$

where $d_k = y_k - \hat{y}_{k-nd}$ and $\mathbf{H}_{n_y}$, $\mathbf{P}_{n_y}$ and $\mathbf{Q}_{n_y}$ depend on the model parameters $\alpha(z)$ and $\beta(z)$. Furthermore:

$$\underset{\rightarrow}{\mathbf{v}}_k = \begin{bmatrix} v_k \\ v_{k+1} \\ \vdots \\ v_{k+n_y-1} \end{bmatrix} ; \quad \underset{\leftarrow}{\mathbf{v}}_{k-1} = \begin{bmatrix} v_{k-1} \\ v_{k-2} \\ \vdots \\ v_{k-l} \end{bmatrix} ; \quad \underset{\leftarrow}{\hat{\mathbf{y}}}_k = \begin{bmatrix} \hat{y}_k \\ \hat{y}_{k-1} \\ \vdots \\ \hat{y}_{k-l+1} \end{bmatrix} \tag{5.7}$$

where $l = n + m$. The new decision variable v_k remains constant throughout the horizon i.e. $v_{k+i} = v_k$, $\forall i > 0$, which results in the following Pre-stabilised Conventional PFC (PCPFC) control law:

$$v_k = \frac{1}{h_{n_y}} \Big[R - (R - E[y_{k+n_d|k}])\rho^{n_y} - (\mathbf{P}_{n_y} \underleftarrow{\mathbf{v}}_{k-1} + \mathbf{Q}_{n_y} \underleftarrow{\hat{\mathbf{y}}}_{k} + d_k)\Big] \tag{5.8}$$

where as before $h_{n_y} = \mathbf{H}_{n_y}\mathbf{L}_{n_y}$ with $\mathbf{L}_{n_y} = [1 \quad 1 \ldots 1]^T_{1\times n_y}$.

The Pre-stabilised Relative PFC (PRPFC) algorithm can also be deduced in a similar way (see the accompanying UKACC paper [33] in Appendix G for details):

$$v_k = \gamma v_{ss} + \frac{1-\gamma}{h_{n_c}} \Big[R - \Big(\mathbf{P}_{n_c} \underleftarrow{\mathbf{v}}_{k-1} + \mathbf{Q}_{n_c} \underleftarrow{\hat{\mathbf{y}}}_{k} + d_k\Big)\Big] \tag{5.9}$$

where $h_{n_c} = \mathbf{H}_{n_c}\mathbf{L}_{n_c}$ with $\mathbf{L}_{n_c} = [1 \quad 1 \ldots 1]^T_{1\times n_c}$, $v_{ss} = \dfrac{R - d_k}{G_{s,0}(1)}$ and:

$$\gamma = \frac{G_{s,0}(1) - h_{n_c}\theta}{G_{s,0}(1) - h_{n_c}} \tag{5.10}$$

where θ is the chosen input aggression factor, n_c corresponds to 95% settling time of the pre-stabilised dynamic system and:

$$G_{s,0}(1) = \frac{\beta(1)}{\alpha(1)} = \frac{\beta_1 + \beta_2 + \cdots + \beta_l}{1 + \alpha_1 + \alpha_2 + \cdots + \alpha_l} = \frac{\sum_{i=1}^{l} \beta_i}{1 + \sum_{i=1}^{l} \alpha_i}; \quad l = n + m \tag{5.11}$$

Once v_k is known either from (5.8) or (5.9), the actual process input u_k can be computed easily using the expression (5.4).

Remark 5.2. *The computational requirement of* (5.4) *is similar to the open-loop control law* (3.12) *or indeed* (4.8)*, but owing to reparametrisation 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.*

5.2.3 Impact of Pre-stabilisation on Parameter Tuning

Due to reparametrisation of the input function, the nominal initial input u_{1,n_y} is now obtained from the relationship (5.4) as follows:

$$u_{1,n_y} = B_0 v_{1,ny} = \frac{B_0 R}{h_{n_y}}(1 - \rho^{n_y}) \tag{5.12}$$

where h_{n_y} is computed from the pre-stabilised model $G_{s,0}(z)$. Consequently, the input aggression factor θ given by (4.2) becomes:

$$\theta = \frac{u_{1,n_y}}{u_{ss}} = \frac{B_0 v_{1,n_y}}{B_0 v_{ss}} = \frac{G_{s,0}(1)}{h_{n_y}}(1 - \rho^{n_y}) \tag{5.13}$$

which implies that the new tuning methods developed in Chapters 3 and 4, along with the conventional parameter selection guidelines discussed in Section 3.2, are easily applicable to the pre-stabilised prediction model in the standard manner. It is worth emphasising that parameter selection based on the pre-stabilised dynamics yields far more consistent and reliable control performances, which is verified by various simulation examples presented in the attached publications [27–33] in Appendices A-G.

In the following section, we will discuss various methods for pre-stabilising the difficult open-loop dynamics in a straighforward manner.

5.3 Design of Pre-stabilising Compensator

So far we have examined the impact of pre-stabilisation on the core functionality of PFC by assuming a suitable compensator that stabilises the undesirable open-loop dynamics for consistent prediction behaviour. In this section, we will discuss various classical feedback compensation schemes that are simple, well-understood and easily implementable with basic technical know-how and hence without overly complicating the PFC design.

One obvious solution is to utilise proportional (plus derivative) or lead/lag type controller which could possibly stabilise a majority of first- and second-order dynamics using any standard time-domain or frequency-domain tuning method [4]. Nevertheless, there are instances, for example the presence of poorly damped or unstable higher-order poles, which may necessitate relatively sophisticated approaches such as the ones based on pole cancellation or pole placement [5]. These strategies, however, often produce higher-order controllers which, in the context of PFC, may also complicate the computations related to constraint management. Nevertheless, this is an inevitable consequence when simpler alternatives are no longer effective.

Note that the pre-stabilisation strategies discussed in this section have been classified according to the type of dynamics, i.e. first-, second-, or more generic higher-order, for which they are designed.

5.3.1 Pre-compensation of Unstable First-Order Dynamics

The following discussion is based on the proposal presented in the CONTROLO 2020 paper [27] which can be found in Appendix A.

Consider an unstable first-order system given by:

$$G_0(z) = \frac{b_1 z^{-1}}{1 - a_1 z^{-1}}; \quad a_1 \geq 1 \tag{5.14}$$

This system can be stabilised with a simple proportional gain $C(z) = K$ as per the proposed configuration shown in Figure 5.1. The resulting pre-stabilised system therefore has the following transfer function:

$$G_{s,0}(z) = \frac{\beta_1 z^{-1}}{1 - \alpha_1 z^{-1}} = \frac{K b_1 z^{-1}}{1 - (a_1 - K b_1) z^{-1}} \tag{5.15}$$

Theorem 5.3. *The compensated predictions in* (5.15) *are stable and monotonically convergent provided K is selected within the following range:*

$$\frac{1 - a_1}{b_1} < K < -\frac{a_1}{b_1} \tag{5.16}$$

Proof. For convergent predictions, the pre-stabilised pole α_1 must be stable, i.e. lie within the range $0 < \alpha_1 < 1$, implying that:

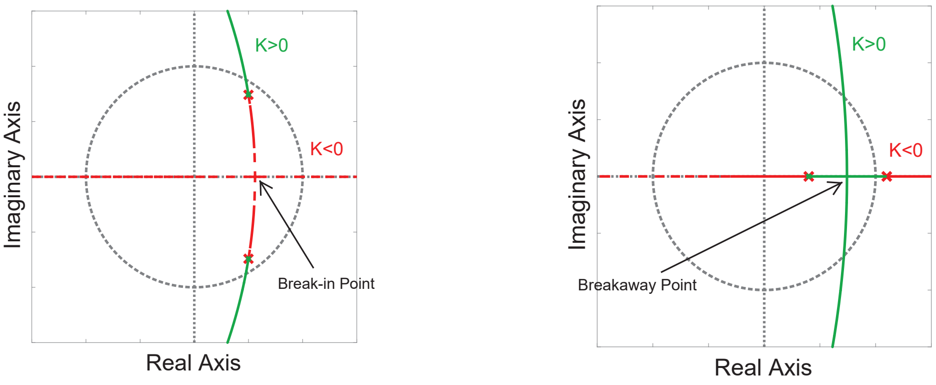
$$0 < a_1 - K b_1 < 1$$

which after simple manipulations leads to (5.16). □

Although Theorem 5.3 in principle defines the upper and lower bounds on K for guaranteed stability, a more systematic choice of the pre-stabilised pole is $\alpha_1 = 1/a_1$ (if $a_1 > 1$) which means the pre-stabilising compensator should be designed with:

$$K = \frac{a_1^2 - 1}{a_1 b_1} \tag{5.17}$$

In case the open-loop system exhibits integrator dynamics, i.e. $a_1 = 1$, then one may simply select $\alpha_1 = 0.5$ by choosing $K = 0.5/b_1$ [85]. This pre-stabilises $G_0(z)$ in a straightforward manner.



(a) (b)

Figure 5.3: Root locus design of a difficult second-order dynamic system with (a) complex pole pair, (b) one unstable pole.

5.3.2 Pre-stabilising Second-Order Dynamics via Root Locus

This proposal is fairly generic and based on the fact that a majority of real-world processes can be adequately represented as dominant second-order dynamics for which simple tailored solutions are well understood.

Assume that a simple proportional controller, i.e. $C(z) = K$, is utilised in the feedback path of the inner loop to compensate a difficult second-order system $G_0(z)$, resulting in the following pre-stabilised transfer function:

$$G_{s,0}(z) = \frac{\beta(z)}{\alpha(z)} = \frac{G_0(z)}{1 + KG_0(z)} = \frac{b(z)}{a(z) + Kb(z)} \tag{5.18}$$

where $a(z) = 1 + a_1 z^{-1} + a_2 z^{-2}$ and $b(z) = b_1 z^{-1} + b_2 z^{-2}$. A simple approach is to design the proportional gain K via *Root Locus* (see the UKACC 2022 paper [32] in Appendix F), which is a powerful graphical tool for control systems analysis and design [110]. Hence, one may analyse the effect of K on the pre-stabilised pole polynomial $\alpha(z)$, with the goal to obtain critically damped poles at the stable break-in/breakaway points, denoted by σ, on the root loci as shown in Figure 5.3. Consequently, at $z = \sigma$, the compensator K must satisfy the following gain condition [111]:

$$K(\sigma) = -\frac{1}{G_0(\sigma)} = -\frac{a(\sigma)}{b(\sigma)} \tag{5.19}$$

where the function $K(\sigma)$ demonstrates local minimum/maximum at the break-in/breakaway points. Thus being stationary, these points can be obtained analytically by taking the first derivative of (5.19) with respect to σ:

$$\frac{dK(\sigma)}{d\sigma} = -\frac{d}{d\sigma}\left[\frac{a(\sigma)}{b(\sigma)}\right] = 0$$

which yields the following relationship:

$$\sigma_i = -\frac{b_2}{b_1} \pm \frac{1}{b_1}\sqrt{a_2{b_1}^2 - a_1 b_1 b_2 + {b_2}^2}, \quad i = 1, 2 \tag{5.20}$$

Depending on the open-loop model parameters, two values of σ are obtained, but only the one within the range $0 < \sigma < 1$ can be implemented for pre-stabilisation. Hence, the pre-stabilised pole polynomial $\alpha(z)$ must be equal to $\alpha(z) = z^2 - 2\sigma z + \sigma^2$ with K evaluated from (5.19). It is noted that one can easily find K using modern software tools and thus avoiding explicit use of the algebra above.

Remark 5.4. *The open-loop zero dynamics of $G_0(z)$ may be significant in some instances, i.e. the root of $b(z)$ may appear in the vicinity of the system poles hindering the successful implementation of this approach. In such cases, it is recommended to design a lead or lag type controller if possible to cancel and replace the problematic zero in order to minimise its undesirable effect. The details to do so are available in the accompanying publication [32] which can be found in Appendix F of this thesis.*

5.3.3 Pre-compensation of Oscillatory Systems via Pole Cancellation

The section summarises the contents of the accompanying ECC 2021 paper [28] which is attached in Appendix B.

Assume that $G_0(z)$ represents a n^{th}-order poorly damped but otherwise stable system such that:

$$G_0(z) = \frac{b(z)}{a(z)}; \quad a(z) = a^-(z)a^+(z) \tag{5.21}$$

where $a^+(z)$ represents p_u complex conjugate poles and $a^-(z)$ factors the remaining $n - p_u$ well damped open-loop poles. As shown in Figure 5.1, the feedback compensation of $G_0(z)$ with $C(z)$ results in the following pre-stabilised transfer function:

$$G_{s,0}(z) = \frac{\beta(z)}{\alpha(z)} = \frac{C(z)G_0(z)}{1 + C(z)G_0(z)} \tag{5.22}$$

which after simple manipulations leads to:

$$C(z) = \frac{\beta(z)a(z)}{b(z)[\alpha(z) - \beta(z)]} \tag{5.23}$$

Evidently, the open-loop zeros $b(z)$ become compensator poles that could destabilise the system especially due to non-minimum phase dynamics. To avoid this issue, we set $\beta(z) = Kb(z)$ with $K \neq 0$ and therefore obtain:

$$C(z) = K\frac{a(z)}{\alpha(z) - Kb(z)} \tag{5.24}$$

where K can be evaluated easily, for example, via root locus. Hence, this compensator actually *cancels* the open-loop poles $a(z)$ and places the *new* poles defined by the polynomial $\alpha(z)$, which means it is necessary to select the pre-stabilised pole polynomial first before designing $C(z)$. Ideally one would want the compensated model to exhibit non-oscillatory behaviour, for which a good starting point is to place the new poles of $G_{s,0}(z)$ at the projection of the dominant oscillatory poles of $G_0(z)$ along the real axis. The resulting pre-stabilised transfer function has the following form:

$$G_{s,0}(z) = K\frac{b(z)}{\alpha(z)}; \quad \alpha(z) = a^-(z)\alpha^+(z) \tag{5.25}$$

where $\alpha^+(z)$ represents the p_u pre-stabilised poles such that $\alpha^+(z) = \prod_{i=1}^{p_u}[z - \Re(z_{p,i})]$ for each open-loop complex conjugate pole $z_{p,i}$ present in $a^+(z)$.

5.3.4 Pre-stabilisation via Pole Placement

The proposed pole placement scheme is based on the analytical approach of feedback compensation presented in the NMPC 2021 paper [29] attached in Appendix C.

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 5.1, resulting in the pre-stabilised transfer function $G_{s,0}(z)$, with a smooth and monotonically convergent prediction behaviour. Then the resulting $(2n-1)^{th}$-order pole polynomial $\alpha(z)$ is given by the following relationship:

$$p(z)a(z) + q(z)b(z) = \alpha(z) \tag{5.26}$$

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

$$\mathbf{M} = \mathbf{S}^{-1}\mathbf{D} \tag{5.27}$$

where $\mathbf{M} = [p_{n-1} \dots 1 \quad q_{n-1} \dots q_0]^T$, $\mathbf{D} = [\alpha_{2n-1} \dots \alpha_0]^T$ and $\mathbf{S}$ is the *Sylvester Matrix* [112] given by:

$$\mathbf{S} = \begin{bmatrix} a_n & 0 & \dots & 0 & b_n & 0 & \dots & 0 \\ a_{n-1} & a_n & \dots & 0 & b_{n-1} & b_n & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & a_1 & \dots & a_{n-1} & 0 & b_1 & \dots & b_{n-1} \\ 0 & 1 & \dots & a_{n-2} & 0 & 0 & \dots & b_{n-2} \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & a_1 & 0 & 0 & \dots & b_1 \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \end{bmatrix} \tag{5.28}$$

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

$$\alpha(z) = o(z)a^-(z)\alpha^+(z) \tag{5.29}$$

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 pre-stabilised 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 $(z-0.5)$ [85].

Proposal for Complex Poles. With $z_{p,i} \in \mathbb{C}$, place the pre-stabilised poles at the real part of the complex open-loop poles, i.e. $\alpha^+(z) = \prod_{i=1}^{p_u}(z - \Re(z_{p,i}))$. This will effectively filter out the undesirable oscillations but without compromising the convergence speed.

This completes the internal feedback loop design via pole placement.

5.3.5 Summary of Pre-stabilised PFC

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

Algorithm 5.1. ***(Unconstrained Pre-Stabilised PFC)*** *First stabilise the difficult open-loop dynamics using PID or any suitable feedback compensation scheme discussed in this section. Next select appropriate tuning parameters (ρ, n_y) or θ as per the methods* *presented in Chapters 3 or 4. Then at each sample k, compute the unconstrained value of* v_k *either using* (5.8) *or* (5.9). *Finally compute the unconstrained value of* u_k *with* (5.4), *and update the plant and the model.*

5.4 Constrained Pre-stabilised PFC

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

The parametrisation of u_k after pre-stabilisation clearly makes the simple saturation policy for constraint handling less straightforward to implement. This is because in addition to the output/state constraints, one must now also ensure input constraint adherence at each future sample over a validation window extending well beyond the coincidence point as any unobserved violation could eventually lead to infeasibility invalidating the current optimisation. Hence, each row of the following vector inequalities must restrict the corresponding prediction such that:

$$\begin{aligned} \mathbf{L}_{n_c}\underline{u} &\leq \underrightarrow{\mathbf{u}}_k \leq \mathbf{L}_{n_c}\bar{u} \\ \mathbf{L}_{n_c}\Delta\underline{u} &\leq \Delta\underrightarrow{\mathbf{u}}_k \leq \mathbf{L}_{n_c}\Delta\bar{u} \\ \mathbf{L}_{n_c}\underline{y} &\leq \underrightarrow{\mathbf{y}}_{k+1} \leq \mathbf{L}_{n_c}\bar{y} \end{aligned} \tag{5.30}$$

It is usually more convenient to represent these inequalities in terms of v_k as it remains constant along n_c, by noting that $\underrightarrow{\mathbf{u}}_k = B_0\mathbf{L}_{n_c}v_k + \underrightarrow{\mathbf{f}}_k$, $\Delta\underrightarrow{\mathbf{u}}_k = \mathbf{C}_{1/\Delta}^{-1}(\underrightarrow{\mathbf{u}}_k - \mathbf{L}_{n_c}u_{k-1})$, and $\underrightarrow{\mathbf{y}}_{k+1} = h_i\mathbf{L}_{n_c}v_k + \mathbf{P}_i\underleftarrow{\mathbf{v}}_{k-1} + \mathbf{Q}_i\underleftarrow{\hat{\mathbf{y}}}_k + \mathbf{L}_{n_c}d_k$ with $i = 1, 2, \ldots, n_c$ [11]. This leads to:

$$\underbrace{\begin{bmatrix} B_0\mathbf{L}_{n_c} \\ -B_0\mathbf{L}_{n_c} \\ B_0\mathbf{C}_{1/\Delta}^{-1}\mathbf{L}_{n_c} \\ -B_0\mathbf{C}_{1/\Delta}^{-1}\mathbf{L}_{n_c} \\ h_i\mathbf{L}_{n_c} \\ -h_i\mathbf{L}_{n_c} \end{bmatrix}}_{\mathbf{X}} v_k \leq \underbrace{\begin{bmatrix} \mathbf{L}_{n_c}\bar{u} - \underrightarrow{\mathbf{f}}_k \\ -\mathbf{L}_{n_c}\underline{u} + \underrightarrow{\mathbf{f}}_k \\ \mathbf{L}_{n_c}\Delta\bar{u} - \mathbf{C}_{1/\Delta}^{-1}\underrightarrow{\mathbf{f}}_k + \mathbf{C}_{1/\Delta}^{-1}\mathbf{L}_{n_c}u_{k-1} \\ -\mathbf{L}_{n_c}\Delta\underline{u} + \mathbf{C}_{1/\Delta}^{-1}\underrightarrow{\mathbf{f}}_k - \mathbf{C}_{1/\Delta}^{-1}\mathbf{L}_{n_c}u_{k-1} \\ \mathbf{L}_{n_c}\bar{y} - \mathbf{P}_i\underleftarrow{\mathbf{v}}_{k-1} - \mathbf{Q}_i\underleftarrow{\hat{\mathbf{y}}}_k - \mathbf{L}_{n_c}d_k \\ -\mathbf{L}_{n_c}\underline{y} + \mathbf{P}_i\underleftarrow{\mathbf{v}}_{k-1} + \mathbf{Q}_i\underleftarrow{\hat{\mathbf{y}}}_k + \mathbf{L}_{n_c}d_k \end{bmatrix}}_{\mathbf{Y}} \tag{5.31}$$

where $\mathbf{C}_{1/\Delta}$ is a lower triangular matrix defined as follows [11]:

$$\mathbf{C}_{1/\Delta} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \tag{5.32}$$

Algorithm 5.2. ***(Constrained Pre-stabilised PFC)*** *After computing the current v_k as per Algorithm 5.1, update the vector $\underset{\rightarrow}{\mathbf{f}}_k$ and verify each row of (5.31) enforcing saturation at $v_k = Y^j/X^j$ for every violation in the j^{th} row. Finally, compute the constraint adhering value of u_k using (5.4).*

Theorem 5.5. *Algorithm 5.2 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 [11]), 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 5.6. *Although recursive feasibility is established in principle for the nominal case, the use of $\underset{\leftarrow}{\mathbf{d}}_k$ in the control law* (5.4) *indeed provides a measure of the disturbance gradient that may allow feasible operation even if $\Delta d_k \neq 0$. Nevertheless, it is noted that rigorous generic recursive feasibility properties require computations, for example see [113–115], which might be considered beyond the price range of PFC. 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 [11].*

5.5 Simulation Results and Discussion

In this section, we investigate the efficacy of various pre-compensation schemes within the proposed PPFC framework using two industrial case studies. It is worth mentioning that this analysis implicitly utilises the PRPFC control law (5.9)-(5.11) while noting that a similar PCPFC controller (5.8) could also be tuned by appropriately selecting ρ and n_y according to the findings of Chapter 4.

5.5.1 Description of Case Studies

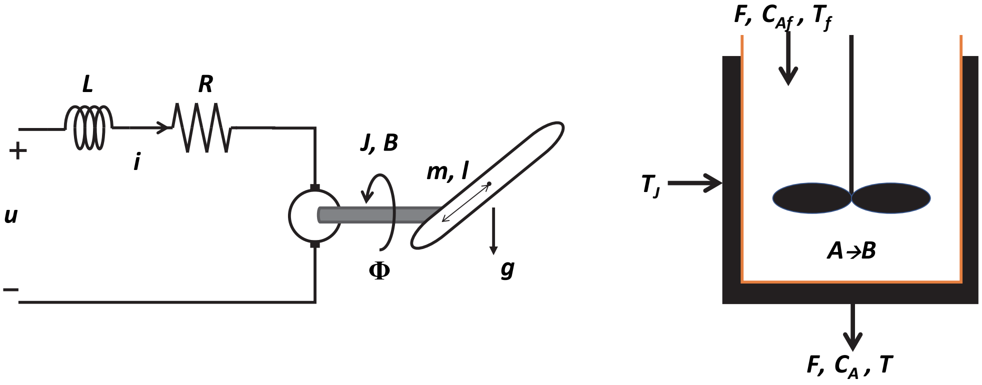
Let us first introduce the open-loop processes that will be used to study the efficacy of the proposal in the later sections:

Position Control of Single Link Robotic Arm. Robotic manipulators have played a vital role in the global industrial revolution leading to better quality product with cheaper manufacturing costs. In this study, we will implement the proposed Pre-stabilised PFC algorithm to control the angular position of a single link robotic arm driven by a brushed DC motor, as shown in Figure 5.4a. The nonlinear model of the manipulator has three coupled states, i.e. the angular position ϕ, the angular velocity $\dot{\phi}$ and the motor current i, related with each other according to the following dynamic relationship [116]:

$$J\frac{d^2\phi}{dt^2} + B\frac{d\phi}{dt} + mglsin\phi = K_\tau i \tag{5.33a}$$

$$L\frac{di}{dt} + Ri + K_e\frac{d\phi}{dt} = u \tag{5.33b}$$

where u is the input voltage (manipulated variable), J is the inertia of the robotic arm, B represents the actuator damping, m and l denote the mass and length of the arm, g is acceleration due to gravity, L and R represent the inductance and resistance of the motor winding, K_τ is the torque constant and K_e represents the voltage constant of the motor. These parameters have the following numerical values: $J = 0.1$ kgm^2, $B = 0.05$ Nms/rad, $m = 2$ kg, $l = 0.75$ m, $g = 9.8$ m/s^2, $K_\tau = 0.768$ Nm/A, $K_e = 0.768$ Nms/rad, $R = 2.6$ Ohm and $L = 25$ mH. Moreover, the motor voltage supply is limited between $0 \leq u \leq 24$ volts with $|\Delta u| \leq 0.5$ volts/s. The model is linearsied, using a sampling period of 10 ms,



(a)

(b)

Figure 5.4: Schematic representation of (a) DC Motor driven Single Link Robot, and (b) Jacketed CSTR process.

around the operating point $\phi_{ss} = 0.314$ rad (18°) and $u_{ss} = 15.4$ volts leading to:

$$G_1 = \frac{\phi'(z)}{u'(z)} = \frac{4.011z^2 + 12.596z + 2.384}{z^3 - 2.320z^2 + 1.681z - 0.352} \times 10^{-5} \tag{5.34}$$

where ϕ' and u' are the output and input deviation variables around their corresponding steady-state values. A second-order model is also constructed using the model reducer app available in MATLAB [117]:

$$G_{1,r} = \frac{\phi'(z)}{u'(z)} = \frac{-1.192z + 4.133}{z^2 - 1.962z + 0.972} \times 10^{-2} \tag{5.35}$$

Note that both linearised models (5.34) and (5.35) exhibit significant oscillations with open-loop poles at $z = 0.362, 0.979 \pm j0.117$ for G_1 and $z = 0.981 \pm j0.101$ for $G_{1,r}$.

Temperature control in Jacketed CSTR. The Continuous Stirred Tank Reactor (CSTR) is a common industrial unit widely employed in different chemical manufacturing processes. In this study, we consider a specific type of CSTR equipped with an outer jacket in which the temperature of a flowing fluid T_J is used to regulate the inside reaction temperature T, as shown in Figure 5.4b. The overall coupled model has two-state nonlinear dynamics as given below [118]:

$$\frac{dC_A}{dt} = \frac{F}{V}\left(C_{A_f} - C_A\right) - k_0 e^{-\frac{E}{R_g T}} C_A \tag{5.36a}$$

$$\frac{dT}{dt} = \frac{F}{V}\left(T_f - T\right) + \left(\frac{-\Delta H}{\rho C_p}\right) k_0 e^{-\frac{E}{R_g T}} C_A - \frac{UA}{V\rho C_p}\left(T - T_J\right) \tag{5.36b}$$

where C_A is the concentration of component A, T is the reaction temperature, C_{A_f} is the feed concentartion, T_f is the feed temperature, T_J is the jacket temperature, F is the input flow rate, V is the reactor volume, k_0 is the frequency factor, E is the activation energy, R_g is the ideal gas constant, $-\Delta H$ is the heat of reaction, U is the heat transfer coefficient, A is the area of heat transfer, ρ is the fluid density and C_p is the fluid heat capacity.

The following parametric values will be used in simulations [119]: $k_0 = 16.96 \times 10^{12}$ h^{-1}, $E = 32400$ Btu/lb mol, $R_g = 1.987$ Btu/lb mol°F, $\rho C_p = 53.25$ Btu/ft^3°F, $UA = 23200$ Btu/h°F, $V = 500$ ft^3, $F = 2000$ ft^3/h, $C_{A_f} = 0.132$ lb mol/ft^3, and $T_f = 60$°F. Furthermore, the process, subject to $T_J \leq 2640$°F, is linearised using a sampling period of 0.01 hours (36 seconds) around the operating point $T_{ss} = 560.8$°F and $T_{J,ss} = 2637.9$°F resulting in an unstable second-order dynamic model (assuming a measurement delay of $n_d = 25$ samples) given by:

$$G_2 = \frac{T'(z)}{T'_J(z)} = \frac{0.00895z - 0.00825}{z^2 - 1.972z + 0.9719}\, z^{-25} \tag{5.37}$$

where both T' and T'_J are deviation variables around the corresponding steady-state values. A first-order model is also constructed using the model reducer app available in MATLAB [117]:

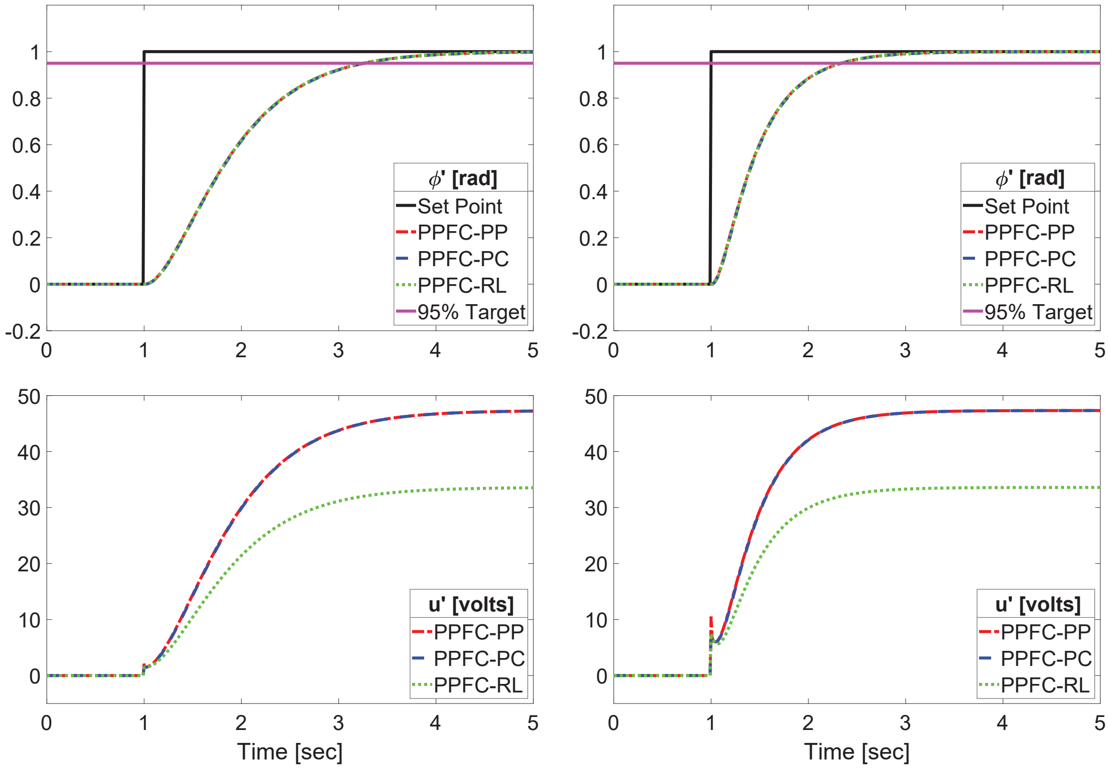
$$G_{2,r} = \frac{T'(z)}{T'_J(z)} = \frac{0.0205}{z - 1.004}\, z^{-25} \tag{5.38}$$

Note that both linearised models (5.37) and (5.38) exhibit output instability with open-loop poles at $z = 0.969, 1.004$ for G_2 and $z = 1.004$ for $G_{2,r}$.

5.5.2 Pre-stabilisation and Parameter Tuning

The third-order model G_1 and the reduced second-order model $G_{1,r}$ of the robotic arm are pre-stabilised using the Pole Placement (PP), Pole Cancellation (PC) and Root Locus (RL) methods with $C_1^{PP} = \dfrac{-66.29z^2 + 18.2z + 2.09}{z^2 + 0.00266z + 0.00014}$, $C_1^{PC} = \dfrac{0.3z^3 - 0.696z^2 + 0.5042z - 0.1055}{z^3 - 2.32z^2 + 1.67z - 0.347}$ and $C_{1,r}^{RL} = -32$. The following pre-compensated transfer function models are obtained:

- $G_1^{PP} = \dfrac{-265.90z^4 - 761.96z^3 + 79.55z^2 + 69.71z + 4.98}{z^5 - 2.32z^4 + 1.67z^3 - 0.346z^2} \times 10^{-5}$
- $G_1^{PC} = \dfrac{1.20z^2 + 3.78z + 0.715}{z^3 - 2.32z^2 + 1.67z - 0.347} \times 10^{-5}$



(a)

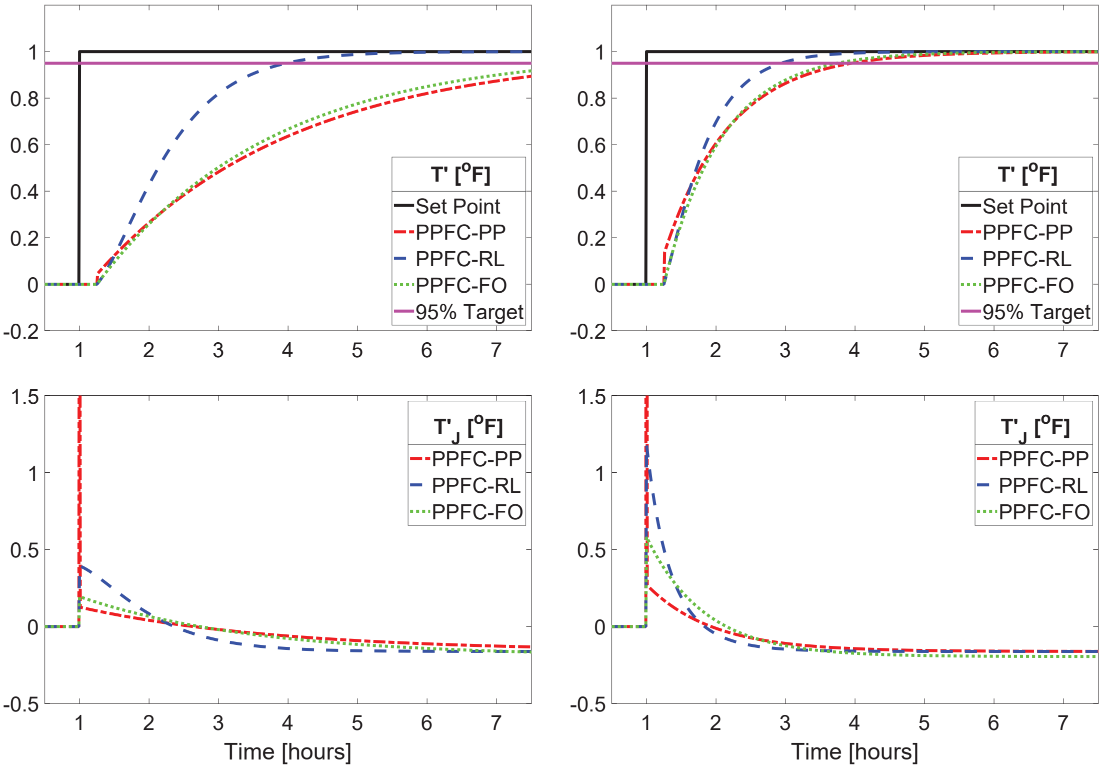
(b)

Figure 5.5: Tuning efficacy of Pre-stabilised PFC for G_1 with Pole Placement, Pole Cancellation and Root Locus compensation schemes using (a) $\theta = 1$ (Benchmark), and (b) $\theta = 5$.

- $G_{1,r}^{RL} = \dfrac{-15z + 44.5}{z^2 - 1.96z + 0.958} \times 10^{-5}$

The dominant pre-stabilised poles positioned at $z = 0.979, 0.979$ in each case warrant a similar closed-loop behaviour in nominal conditions, which is indeed obvious from Figure 5.5a depicting the benchmark (mean-level) system outputs with $\theta = 1$. Furthermore, Figure 5.5b clearly indicates the tuning efficacy of the (relative) PPFC algorithm with $\theta = 5$, as the robotic arm now settles relatively quickly to its target steady-state angular position as compared to the benchmark responses shown in Figure 5.5a.

In the second example, the original second-order model G_2 along with its reduced first-order model $G_{2,r}$ of the Jacketed CSTR process are pre-stabilised using the Pole Placement (PP), Root Locus (RL) and First-Order (FO) methods. Note that due to the presence of open-loop unstable mode, the Pole Cancellation method will not be utilised in this case



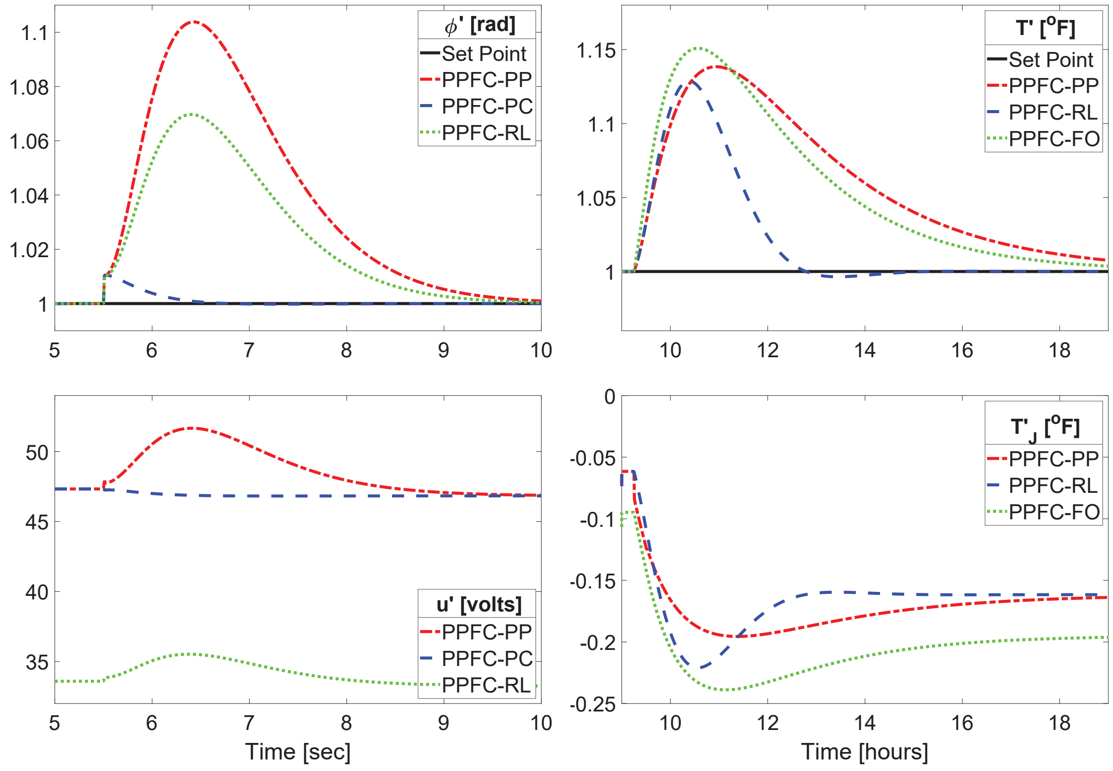
(a) (b)

Figure 5.6: Tuning efficacy of Pre-stabilised PFC for G_2 with Pole Placement, Root Locus and First-Order compensation schemes using (a) $\theta = 1$ (Benchmark), and (b) $\theta = 3$.

for pre-compensation.

- $C_2^{PP} = \dfrac{9.834z - 9.525}{z - 0.0808}$, $G_2^{PP} = \dfrac{0.088z^2 - 0.166z + 0.079}{z^3 - 1.965z^2 + 0.965z}$, poles at $0, 0.997, 0.969$
- $C_2^{RL} = 0.556$, $G_2^{RL} = \dfrac{0.00895z - 0.00825}{z^2 - 1.967z + 0.9673}$, poles at $0.984, 0.984$
- $C_{2,r}^{FO} = 0.389$, $G_{2,r}^{FO} = \dfrac{0.00798}{z - 0.996}$, pole at 0.996

Since the Root Locus method provides faster dominant poles than the other two techniques, its pre-stabilised benchmark response is correspondingly faster as shown in Figure 5.6a. Nevertheless in this case too, the selected tuning parameter $\theta = 3$ works effectively by producing relatively faster closed-loop outputs (Figure 5.6b). However, it is evident that pre-comensation via pole placement in this case may not be a good choice since the sharp initial input peak compared to other methods may cause constraint viola-



(a) (b)

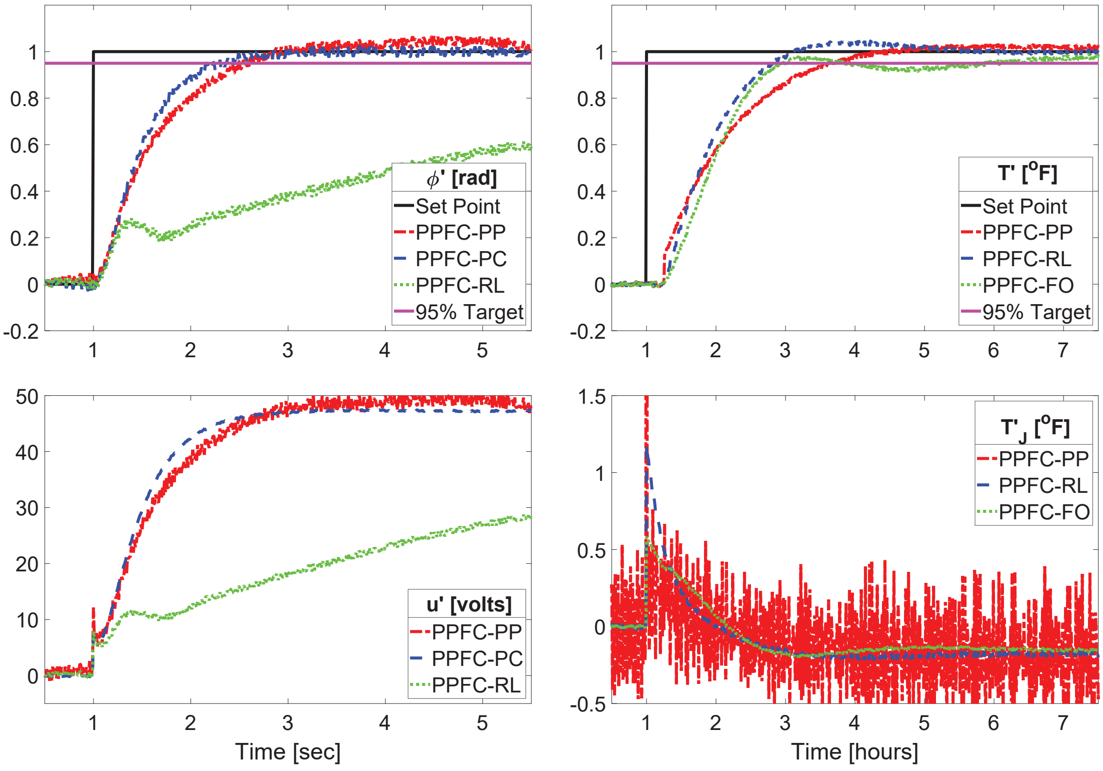
Figure 5.7: Comparison of disturbance rejection between various compensation schemes for (a) G_1 with $\theta = 5$ and a constant output disturbance of 0.1 rad, and (b) G_2 with $\theta = 3$ and a constant input disturbance of 1°F.

tions possibly leading to instability in practice.

5.5.3 Comparison in Practical Scenarios

In this section, we will further analyse the efficacy of these pre-stabilisation methods in more practical scenarios by including the effects of external disturbances, measurement noise and modelling uncertainties on the overall closed-loop performance.

For the single link robot, a constant disturbance of 0.1 rad is introduced at the process output, whereas for the unstable CSTR a constant input disturbance of 1°F is added. The results are shown in Figure 5.7 from which it is evident that the Pole Placement method in both cases demonstrates relatively sluggish disturbance rejection as compared to the other two methods. Notably, Pole Cancellation provides by far the best disturbance

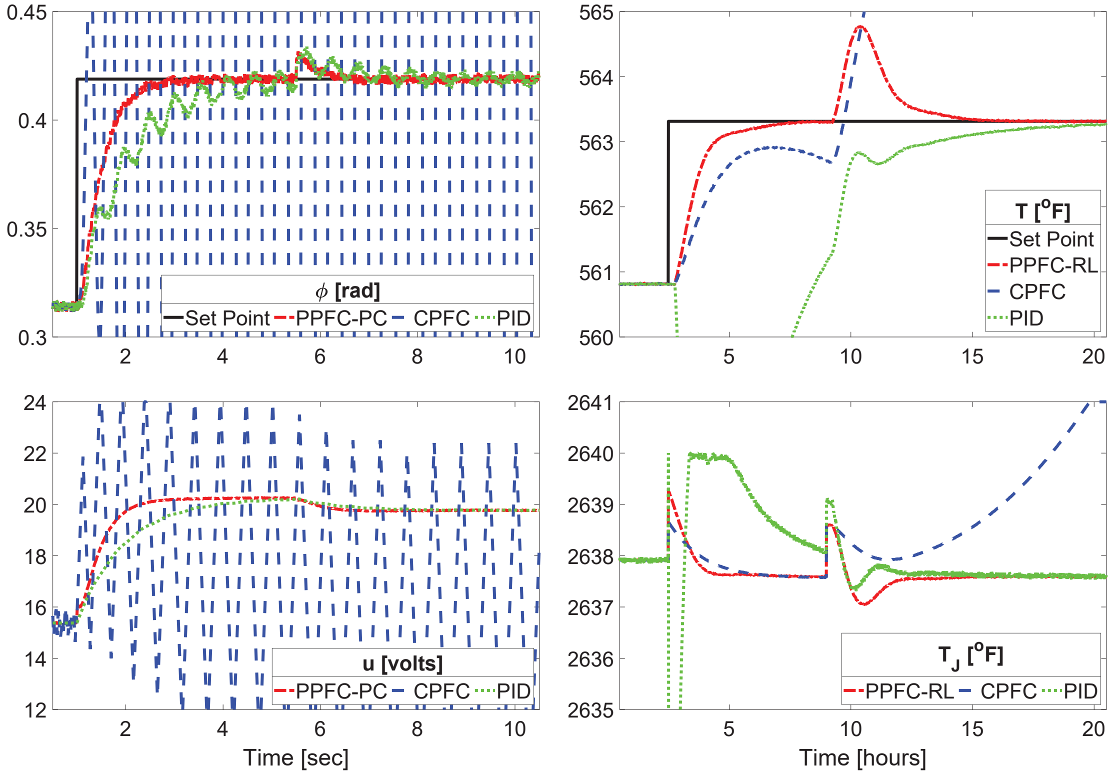


(a) (b)

Figure 5.8: Comparison of tuning efficacy between various compensation schemes along with measurement noise and plant-model mismatches for (a) G_1 with $\theta = 5$ and an unmodelled pole at $z = 0.5$, and (b) G_2 with $\theta = 3$ and a 10% multiplicative uncertainty.

handling with signifiantly quicker output normalisation. A similar trend is observed in the noise management (see Figure 5.8) which shows the sensor noise affecting the PPFC-PP far more adversely than the other methods of pre-stabilisation, especially in the unstable process example.

It is further evident that the presence of plant-model mismatch tends to degrade the closed-loop performance with the second-order model of the oscillatory process (Figure 5.8a), but not so much with the first-order model of the unstable system (Figure 5.8b). Nevertheless due to the inevitability of modelling errors, it generally seems sensible to avoid such reduced order models within the proposed framework, if possible.



(a) (b)

Figure 5.9: Closed-loop performance comparison with constraints and external perturbations for (a) G_1 pre-stabilised via Pole Cancellation using $\theta = 5$ subject to $|\Delta u| \leq 0.5$volts and $0 \leq u \leq 24$volts, and (b) G_2 pre-stabilised via Root Locus scheme with $\theta = 3$ subject to $T_J \leq 2640$°F.

5.5.4 Analysis of Constrained Closed-Loop Performance

Finally, a comparative analysis of the constrained closed-loop performance against the conventional PFC (CPFC) and PID algorithms is presented. For a fair comparison, the CPFC uses the same θ as the corresponding pre-stabilised PFC albeit with difficult open-loop prediction dynamics implemented directly. Furthermore, the PID is synthesised using the robust PID tuning algorithm available in MATLAB [120]. The actual non-linear models (5.33) and (5.36) act as the *plants* for a more realistic evaluation.

Figure 5.9a depicts the scenario for the poorly damped process, where a set point change of 0.105 radians, i.e. 6°, from the initial steady-state is introduced. As evident, both CPFC and PID controllers fail to compensate the open-loop underdamping, though the performance of CPFC is far worse under constraints. On the other hand, the proposed

PPFC not only succesfully filters out the unwanted oscillations, but does so by maintaining feasibility dispite a significant change in both the set point and the disturbance.

For the unstable process, the closed-loop performance is displayed in Figure 5.9b. 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. In this case too, the proposed PPFC 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 predictive functional control law for a superior performance as compared to the direct utilisation of the difficult open-loop dynamics, which clearly fails to fulfil the desired performance specification in an efficient and reliable manner.

Remark 5.7. *Output constraints, if available, may also be implemented easily using system predictions (see Section 5.4). Interested readers are referred to the accompanying ACA publication attached in Appendix E which includes one such simulation example.*

5.6 Chapter Summary

The key highlights of this chapter are summarised below:

- A generic design framework is developed in Section 5.2 which utilises the concept of *pre-stabilisation* to systematically achieve a reliable and consistent control of severely underdamped and/or unstable open-loop dynamics by first compensating the difficult prediction behaviour before implementing PFC in a cascade structure.
- Section 5.3 discusses various pre-compensation schemes that adequately transform the challenging open-loop dynamics into well-damped and monotonically convergent prediction behaviour, i.e. something more conveniently manageable with the standard PFC algorithm.
- While pre-stabilisation improves the tuning efficacy of PFC, it does complicate the application of the standard constraint handling procedure. Nevertheless, Section 5.4 discusses how constraints can be managed more efficiently in these cases where pre-

compensation renders the standard saturation policy far less straightforward to implement within the proposed framework.

Chapter 6

Nonlinear PFC

This chapter presents a simple and low-cost nonlinear predictive control approach using a slightly modified version of the Relative PFC algorithm developed earlier in Chapter 4. Section 6.1 formulates the problem and sets the control objectives. The proposed Nonlinear PFC (NPFC) algorithm is discussed in Sections 6.2 and 6.3 which is based on the accompanying EJC 2022 publication [35] attached in Appendix I. Section 6.4 investigates the efficacy of the proposal with two simulation case studies and the chapter concludes in Section 6.5 with a consice summary of the key developments.

6.1 Problem Statement

Consider a nonlinear system of the form:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), u(t)), \quad \mathbf{x}(0) = \mathbf{x}_0 \quad u(0) = u_0 \tag{6.1}$$

subject to the following physical and operational constraints:

$$\underline{u} \leq u(t) \leq \bar{u}, \quad \Delta\underline{u} \leq \Delta u(t) \leq \Delta\bar{u}, \quad \underline{y} \leq y(t) \leq \bar{y} \tag{6.2}$$

with states $\mathbf{x}(t) \in \mathbb{R}^n$ and input $u(t) \in \mathbb{R}$. Note that the j^{th} system state, denoted by $y(t)$, is designated as the process output (controlled variable) whose current value is measurable. It is further assumed that:

(i) The nonlinear function $\mathbf{f}(\cdot)$ is continuous and differentiable with respect to the states and the input, hence (6.1) can be linearised about a nominal trajectory and represented by a discrete time varying state-space model (details in Section 6.2.1).

(ii) The open-loop behaviour of (6.1) is broadly acceptable, i.e. for a constant input $u(t) = U$, the states are stable and monotonically convergent towards their implied equilibrium positions.

The primary objective in this study is to design a nonlinear PFC (NPFC) controller for the system (6.1) subject to (6.2) in order to satisfactorily track a step target $R = y(\infty)$ as per the required performance specification (for example, a desired closed-loop settling time). Furthermore, the nonlinear controller is expected to exhibit some degree of robustness against external disturbances, measurement noise and/or parametric uncertainties.

6.2 Prediction in Nonlinear PFC

Likewise the Relative PFC algorithm (see Chapter 4), the proposed NPFC too requires three key elements for implementation: a suitable prediction model, a benchmark (mean level) response for performance tuning, and a mechanism to ensure offset-free tracking. Nevertheless, in order to produce accurate predictions, the underlying process model (6.1) must be linearised online at discrete sampling instants [93, 94]. As PFC, by design, is intended to be simple, here we take a very simple approach to this process by approximating the continuous time derivatives in (6.1) using forward Euler differences, accepting that more accurate but equally more demanding numerical integration approaches are possible [121].

6.2.1 Numerical Integration via Explicit Euler Method

Assume that the time difference between the current and the next samples, i.e. $k \to k+1$, is represented by a small quantity $\delta t \to 0$, then according to the first principles, one may approximate the continuous time derivatives in (6.1) as follows [122]:

$$\hat{\dot{\mathbf{x}}}(t) \approx \frac{\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k}{\delta t}; \quad k \geq 0 \tag{6.3}$$

where $\hat{\mathbf{x}}_k$ represents the predicted state at sample k. This relationship is known as the explicit (or forward) Euler difference, the simplest ordinary differential equations (ODE) solver, which gives the following discretised nonlinear prediction model:

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_k + \delta t \mathbf{f}\left(\hat{\mathbf{x}}_k, u_k\right) = \mathbf{F}\left(\hat{\mathbf{x}}_k, u_k\right) \tag{6.4}$$

Thus, the discretised model (6.4) is linearised online about a nominal trajectory $(\mathbf{x}_k^*, u_k^*)$ using first-order Taylor series expansion such that:

$$\hat{\mathbf{x}}_{k+1} \approx \mathbf{F}\left(\mathbf{x}_k^*, u_k^*\right) + \frac{\partial \mathbf{F}}{\partial \hat{\mathbf{x}}_k}\delta\hat{\mathbf{x}}_k + \frac{\partial \mathbf{F}}{\partial u_k}\delta u_k \tag{6.5}$$

where $\mathbf{F}\left(\mathbf{x}_k^*, u_k^*\right) = \mathbf{x}_{k+1}^*$ and the deviation variables are $\delta\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k - \mathbf{x}_k^*$, $\delta u_k = u_k - u_k^*$. Hence, the following linearised prediction model is obtained:

$$\delta\hat{\mathbf{x}}_{k+1} = \mathbf{A}_k\delta\hat{\mathbf{x}}_k + \mathbf{B}_k\delta u_k \tag{6.6}$$

with

$$\mathbf{A}_k = \frac{\partial \mathbf{F}}{\partial \hat{\mathbf{x}}_k} = \mathbf{I}_n + \delta t\frac{\partial \mathbf{f}}{\partial \hat{\mathbf{x}}_k}; \quad \mathbf{B}_k = \frac{\partial \mathbf{F}}{\partial u_k} = \delta t\frac{\partial \mathbf{f}}{\partial u_k} \tag{6.7}$$

where $\mathbf{I}_n$ is the identity matrix of size n.

Remark 6.1. *It is implicit from the use of first-order Taylor series and simple difference equations for the numerical integration that the trajectories do not deviate a long way from the baseline* $(\mathbf{x}_k^*, u_k^*)$*. If they do, then the approximation errors would grow and could impact on behaviour. This means the sample period* δt *should be small enough.*

6.2.2 Prediction using Deviation Variables

Once the linearised model (6.6) has been determined for a notional trajectory, one can easily compute the impact of small deviations in the input, i.e. when $\delta u_k \neq 0$. Predictions can be found by recursive use of (6.6) as follows:

$$\begin{aligned} \delta\hat{\mathbf{x}}_{k+1} &= \mathbf{A}_k\delta\hat{\mathbf{x}}_k + \mathbf{B}_k\delta u_k \\ \delta\hat{\mathbf{x}}_{k+2} &= \mathbf{A}_{k+1}\delta\hat{\mathbf{x}}_{k+1} + \mathbf{B}_{k+1}\delta u_{k+1} \\ \delta\hat{\mathbf{x}}_{k+3} &= \mathbf{A}_{k+2}\delta\hat{\mathbf{x}}_{k+2} + \mathbf{B}_{k+2}\delta u_{k+2} \\ &\vdots \end{aligned} \tag{6.8}$$

Next, making substitutions and assuming that $\delta\hat{\mathbf{x}}_k = 0$:

$$\begin{aligned} \delta\hat{\mathbf{x}}_{k+1} &\approx \mathbf{B}_k\delta u_k \\ \delta\hat{\mathbf{x}}_{k+2} &\approx \mathbf{A}_{k+1}\mathbf{B}_k\delta u_k + \mathbf{B}_{k+1}\delta u_{k+1} \\ \delta\hat{\mathbf{x}}_{k+3} &\approx \mathbf{A}_{k+2}[\mathbf{A}_{k+1}\mathbf{B}_k\delta u_k + \mathbf{B}_{k+1}\delta u_{k+1}] + \mathbf{B}_{k+2}\delta u_{k+2} \\ &\vdots \end{aligned} \tag{6.9}$$

Although somewhat clumsy to use in the current form, the following section utilises these predictions along with an exponential input parametrisation to derive the proposed NPFC control law.

6.3 Nonlinear PFC Algorithm

In this section, we utilise a simple Laguerre input parametrisation (see for example [20, 74]) in the predictions along with the concept of *superposition* to develop the NPFC control law. The detailed proposal is as follows:

6.3.1 Prediction with Exponential Input Parametrisation

As discussed earlier in Section 4.3, a first-order Laguerre function represents an ideal exponential decay which parametrises the future input such that:

$$u_{k+i} = u_{ss} + \lambda^i \eta; \quad \lim_{i \to \infty} u_{k+i} = u_{ss}, \quad i \geq 0 \tag{6.10}$$

where $u_{ss} = u(\infty)$ is the implied steady-state input corresponding to $y(\infty)$, λ is the decay factor to be chosen $(0 < \lambda < 1)$ and η is the new degree-of-freedom that will be computed from the NPFC algorithm. Hence, it is evident that in (6.5)-(6.9):

$$u^*_{k+i} = u_{ss}, \quad \delta u_{k+i} = \lambda^i \eta; \quad i \geq 0 \tag{6.11}$$

which means that the nominal predictions $\mathbf{x}^*_k$ can be easily computed online. This also simplifies the deviations (6.9) as follows:

$$\begin{aligned}
\delta\hat{\mathbf{x}}_{k+1} &= \underbrace{\mathbf{B}_k}_{\Lambda_k} \ \eta \\
\delta\hat{\mathbf{x}}_{k+2} &= \underbrace{[\mathbf{A}_{k+1}\Lambda_k + \mathbf{B}_{k+1}\lambda]}_{\Lambda_{k+1}} \ \eta \\
\delta\hat{\mathbf{x}}_{k+3} &= \underbrace{[\mathbf{A}_{k+2}\Lambda_{k+1} + \mathbf{B}_{k+2}\lambda^2]}_{\Lambda_{k+2}} \ \eta \\
&\vdots \\
\delta\hat{\mathbf{x}}_{k+n_x+1} &= \underbrace{[\mathbf{A}_{k+n_x}\Lambda_{k+n_x-1} + \mathbf{B}_{k+n_x}\lambda^{n_x}]}_{\Lambda_{k+n_x}} \ \eta
\end{aligned} \tag{6.12}$$

It is noted that the main computation here is the simple recursion of:

$$\Lambda_{k+n_x} = \mathbf{A}_{k+n_x}\Lambda_{k+n_x-1} + \mathbf{B}_{k+n_x}\lambda^{n_x} \tag{6.13}$$

where n_x is the prediction horizon usually chosen between $1-2$ time constants of the open-loop step response.

6.3.2 NPFC Control Law

As y_k is the designated output state, the total (n_x+1)-step ahead output prediction is given by:

$$y_{k+n_x+1} = y_k^* + \mathfrak{e}_j^T \Lambda_{k+n_x} \eta + d_k \tag{6.14}$$

where d_k is the offset correction term computed from $d_k = y_k - \hat{y}_k$ and $\mathfrak{e}_j$ is a standard basis vector with a 1 in the j^{th} position corresponding to the state y_k. With $u_k = u_{ss}$, i.e. $\eta = 0$, the predicted error converges as follows:

$$e_{ss}(k+n_x+1) = R - (y_k^* + d_k) \tag{6.15}$$

which compares to the following error convergence when $\eta \neq 0$:

$$e(k+n_x+1) = R - \left(y_k^* + \mathfrak{e}_j^T \Lambda_{k+n_x} \eta + d_k\right) \tag{6.16}$$

Hence, to achieve the desired speed-up, one must compare both (6.15) and (6.16) such that:

$$Se(k+n_x+1) = e_{ss}(k+n_x+1) \tag{6.17}$$

where S is the tuning parameter that speeds-up the closed-loop behaviour relative to the open-loop benchmark response. Note that this is slightly different from the Relative PFC algorithm developed earlier (see Section 4.2), where we implemented parameter γ to achieve speed-up indirectly via the chosen input aggression factor θ.

Theorem 6.2. *The NPFC control law for a target set-point R is given by:*

$$u_k = u_{ss} + \frac{S-1}{S}\left[\frac{R-(y_k^*+d_k)}{\mathfrak{e}_j^T \Lambda_{k+n_x}}\right] \tag{6.18}$$

Proof. Using (6.15) and (6.16) in (6.17) implies:

$$S\left[R - \left(y_k^* + \mathfrak{e}_j^T \Lambda_{k+n_x} \eta + d_k\right)\right] = R - (y_k^* + d_k) \tag{6.19}$$

Hence, the degree-of-freedom η is computed as follows:

$$\eta = \frac{S-1}{S}\left[\frac{R-(y_k^*+d_k)}{\mathfrak{e}_j^T \Lambda_{k+n_x}}\right] \tag{6.20}$$

which leads to the control law (6.18), as $u_k = u_{ss} + \eta$ is implemented at each k. Note that the choice $S = 1$ here is equivalent to the benchmark behaviour providing $\eta = 0$ and $u_k = u_{ss}$. □

Remark 6.3. *Unlike the RPFC control law* (4.8)*, the Nonlinear PFC does not utilise the initial input aggression θ for controller tuning, rather it directly implements S which means the Laguerre pole λ retains its efficacy in this formulation. Nevertheless, in order to ensure practically implementable inputs, S and λ should be selected appropriately; typically S should not be more than $2-3$ and λ should be close to the target closed-loop pole and partially overlaps with the choice of n_x.*

6.3.3 Constraint Handling

One can incorporate constraint handling in a systematic and computationally simple way by comparing system predictions against constraints for a sufficiently large horizon n_c. Note that the exact procedure to do so is very simular to the one already presented in Section 5.4 and therefore discussed very briefly for completeness. With $\underset{\rightarrow}{\mathbf{u}}_k = \mathbf{L}_{n_c} u_{ss} + \mathbf{H}_\lambda \eta$, $\Delta \underset{\rightarrow}{\mathbf{u}}_k = \mathbf{C}_{1/\Delta}^{-1}(\underset{\rightarrow}{\mathbf{u}}_k - \mathbf{L}_{n_c} u_{k-1})$ and $\underset{\rightarrow}{\mathbf{y}}_{k+1} = \underset{\rightarrow}{\mathbf{y}}_k^* + \underset{\longrightarrow}{\mathfrak{e}_j^T \Lambda_k} \eta + \mathbf{L}_{n_c} d_k$ where $i = 1, 2, \dots, n_c$, the following linear inequalities must be verified:

$$\begin{bmatrix} \mathbf{H}_\lambda \\ -\mathbf{H}_\lambda \\ \mathbf{C}_{1/\Delta}^{-1}\mathbf{H}_\lambda \\ -\mathbf{C}_{1/\Delta}^{-1}\mathbf{H}_\lambda \\ \underset{\longrightarrow}{\mathfrak{e}_j^T \Lambda_k} \\ -\underset{\longrightarrow}{\mathfrak{e}_j^T \Lambda_k} \end{bmatrix} \eta \leq \begin{bmatrix} \mathbf{L}_{n_c}\left(\bar{u} - u_{ss}\right) \\ -\mathbf{L}_{n_c}\left(\underline{u} + u_{ss}\right) \\ \mathbf{L}_{n_c}\left[\Delta\bar{u} - \mathbf{C}_{1/\Delta}^{-1}\left(u_{ss} - u_{k-1}\right)\right] \\ -\mathbf{L}_{n_c}\left[\Delta\underline{u} - \mathbf{C}_{1/\Delta}^{-1}\left(u_{ss} - u_{k-1}\right)\right] \\ \mathbf{L}_{n_c}\bar{y} - \underset{\rightarrow}{\mathbf{y}}_k^* - \mathbf{L}_{n_c} d_k \\ -\mathbf{L}_{n_c}\underline{y} + \underset{\rightarrow}{\mathbf{y}}_k^* + \mathbf{L}_{n_c} d_k \end{bmatrix} \tag{6.21}$$

where $\mathbf{C}_{1/\Delta}$ is a lower triangular matrix defined in (5.32), $\mathbf{L}_{n_c} = [1 \quad 1 \quad \dots \quad 1]_{n_c}^T$ and $\mathbf{H}_\lambda = [1 \quad \lambda \quad \dots \quad \lambda^{n_c-1}]_{n_c}^T$. Since the predictions have a single degree-of-freedom η, the selection of η can be determined using a simple *for loop* and thus done very efficiently. One might also note that with (6.10) the maximum input and input rate will occur at the first or second sample, and thus the number of inequalities to be checked for the input constraints is very small.

Remark 6.4. *Constraints limit the input amplitudes available and thus will also impact on the speed-up achievable in some scenarios, especially with large changes in target.*

6.3.4 Summary of Nonlinear PFC

We conclude the discussion of Nonlinear PFC with the following algorithm:

Algorithm 6.1. ***(Constrained Nonlinear PFC)*** *For a given target R, first find the baseline trajectory $(\mathbf{x}_k^*, u_k^*)$ using* (6.11) *and compute the associated state-space matrices* (6.7). *Next find the output prediction* (6.14) *using the recursion Λ_{k+n_x} given in* (6.13) *followed by the computation of η with* (6.20). *Finally, calculate the constraint adhering value of u_k using* (6.21), *and update the plant and the model accordingly.*

6.4 Simulation Results and Discussion

This section investigates the efficacy of the proposed NPFC controller using two real-world case studies. Note that both examples satisfy the assumptions stated in Section 6.1.

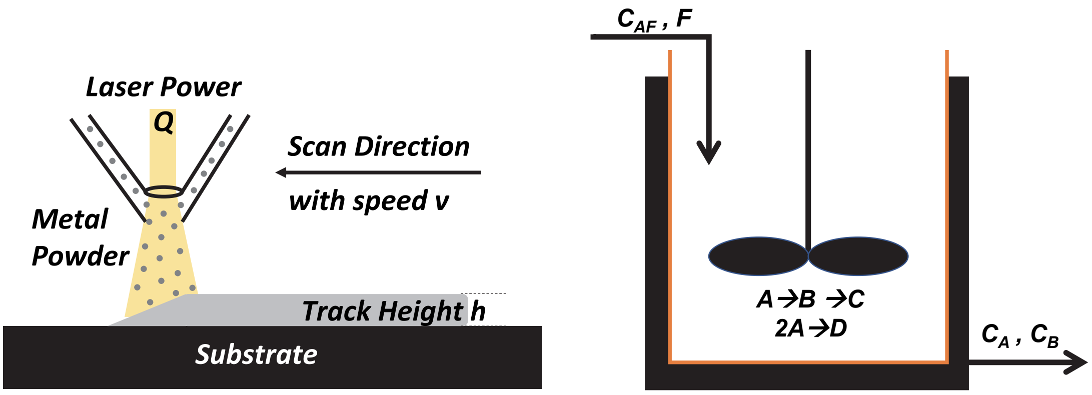
6.4.1 Description of Case Studies

The nonlinear processes are introduced first that will be used in the following analysis:

Laser Metal Deposition Process. Laser Metal Deposition (LMD) is an advanced Additive Manufacturing (AM) technique that guarantees rapid production with cost-effectiveness. The fundamental idea, as shown in Figure 6.1a, is to perform selective melting of metallic powder on a substrate via focused laser heating in a layer-by-layer fashion. Nevertheless, it is complex thermal dominated process with numerous variables, including laser power Q and scan speed v, affecting the melt pool height h. The underlying first-order dynamics are represented by the following Hammerstein type nonlinear model [123]:

$$\dot{h}(t) = \frac{1}{\tau_h}\left(-h(t) + \frac{2}{\pi}\sqrt{\frac{\beta n_Q[Q(t) - Q_c]}{\rho r C_l (T_m - T_0) v}}\right), \quad Q(t) > Q_c \tag{6.22}$$

where τ_h is the open-loop time constant, β is a process coefficient, n_Q is the laser transfer efficiency, Q_c is the critical laser power, ρ is the melt density, r is the melt pool width-height ratio, C_l is the molten material specific heat, and T_m/T_0 are the melting/ambient



(a) (b)

Figure 6.1: Schematic representation of (a) Laser Metal Deposition Process, and (b) Van De Vusse Reactor.

temperatures. The following parametric values are used (Ti-6AL-4V material type [123]): $\tau_h = 0.5$ s, $\beta = 0.3026$, $n_Q = 0.4$, $Q_c = 111.72$ W, $\rho = 4430$ kg/m^3, $1/r = 0.13$, $C_l = 700$ J/kg K, $T_m = 1923$ K, $T_0 = 292$ K and $v = 2$ mm/s.

The process is steadily depositing a constant track height of $h(0) = 50$ μm with $Q(0) = 115.7$ W before a set-point change of $R = h(\infty) = 200$ μm is introduced which requires the following steady-state laser power:

$$Q(\infty) = Q_c + \frac{\pi^2 \rho r C_l (T_m - T_0) v}{4\beta n_Q} h^2(\infty) = 175.17 \text{ W} \tag{6.23}$$

The model is discretised with $\delta t = 1$ ms to obtain a time varying state-space model as follows:

$$A_k = 1 + \delta t \frac{\partial \dot{h}}{\partial h} \quad B_k = \delta t \frac{\partial \dot{h}}{\partial Q} \tag{6.24}$$

where $\frac{\partial \dot{h}}{\partial h} = -\frac{1}{\tau_h}$ and $\frac{\partial \dot{h}}{\partial Q} = \frac{K_Q}{\sqrt{Q_k - Q_c}}$ with:

$$K_Q = \frac{1}{\pi \tau_h} \sqrt{\frac{\beta n_Q}{\rho r C_l (T_m - T_0) v}} \tag{6.25}$$

Thus the recursion (6.13) can be computed easily by substituting the current value of Q_k in the above expressions.

Van De Vusse Reaction. A Van De Vusse Reactor (VDVR) is a complex nonlinear isothermal CSTR in which multiple series-parallel reactions take place simultaneously, as shown in Figure 6.1b, with the only desirable reaction being the production of component B. The governing process dynamics are characterised by the following nonlinear ODEs [124]:

$$\frac{dC_A}{dt} = -k_1 C_A - k_3 C_A^2 + F\left(C_{AF} - C_A\right) \tag{6.26a}$$

$$\frac{dC_B}{dt} = k_1 C_A - k_2 C_B - F C_B \tag{6.26b}$$

where C_A and C_B are the effluent concentrations of components A and B respectively, C_{AF} is the inlet feed concentration, F is the diluration rate (the manipulated variable), and k_1, k_2 and k_3 are process parameters. The following nominal parametric values are assigned: $C_{AF} = 10$ mol/L, $k_1 = 50$ 1/hr, $k_2 = 100$ 1/hr and $k_3 = 10$ L/mol hr [124]. Moreover, the process is operating at the steady states $C_A(0) = 3$ mol/L, $C_B(0) = 1.117$ mol/L and $F(0) = 34.288$ 1/hr before a set point change of $R = C_B(\infty) = 1.25$ mol/L is introduced. In order to implement the proposed NPFC in this situation, one needs to compute the implied $C_A(\infty)$ and $F(\infty)$ using (6.26) as follows:

$$C_A(\infty) = \frac{-K_2 \pm \sqrt{K_2^2 - 4K_1K_3}}{2K_1} \tag{6.27a}$$

$$F(\infty) = \frac{k_1 C_A(\infty) - k_2 C_B(\infty)}{C_B(\infty)} \tag{6.27b}$$

where $K_1 = k_3 C_B(\infty) + k_1$, $K_2 = (k_1 - k_2) C_B(\infty) - k_1 C_{AF}$ and $K_3 = k_2 C_{AF} C_B(\infty)$. Two stable values of $C_A(\infty)$, 4 and 5 mol/L respectively, are obtained but we select $C_A(\infty) = 4$ mol/L which results in a lower diluration rate $F(\infty) = 60$ 1/hr. The model (6.26) is then discretised with $\delta t = 3.6$s to obtain a linear time varying state-space model as follows:

$$\mathbf{A}_k = \mathbf{I}_2 + \delta t \begin{bmatrix} \frac{\partial \dot{C}_A}{\partial C_A} & \frac{\partial \dot{C}_A}{\partial C_B} \\ \frac{\partial \dot{C}_B}{\partial C_A} & \frac{\partial \dot{C}_B}{\partial C_B} \end{bmatrix} \qquad \mathbf{B}_k = \delta t \begin{bmatrix} \frac{\partial \dot{C}_A}{\partial F} \\ \frac{\partial \dot{C}_B}{\partial F} \end{bmatrix} \tag{6.28}$$

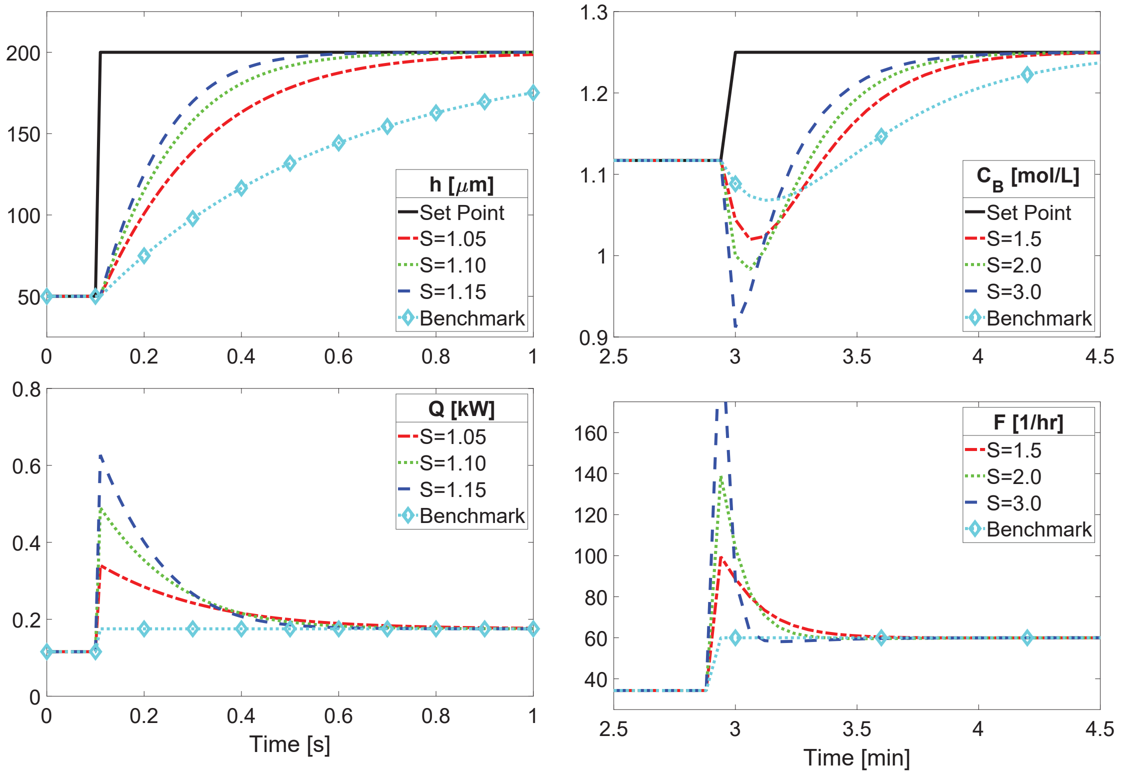
where elements of the state matrix $\mathbf{A}_k$ are:

$$\frac{\partial \dot{C}_A}{\partial C_A} = -k_1 - 2k_3 C_{A,k} \qquad \frac{\partial \dot{C}_A}{\partial C_B} = 0 \qquad \frac{\partial \dot{C}_B}{\partial C_A} = k_1 \qquad \frac{\partial \dot{C}_B}{\partial C_B} = -k_2 - F_k \tag{6.29}$$

and elements of the input matrix $\mathbf{B}_k$ are:

$$\frac{\partial \dot{C}_A}{\partial F} = -C_{A,k} \qquad \frac{\partial \dot{C}_B}{\partial F} = -C_{B,k} \tag{6.30}$$

Thus the recursion (6.13) can be computed easily by substituting the current $C_{A,k}$, $C_{B,k}$ and F_k in the above expressions.



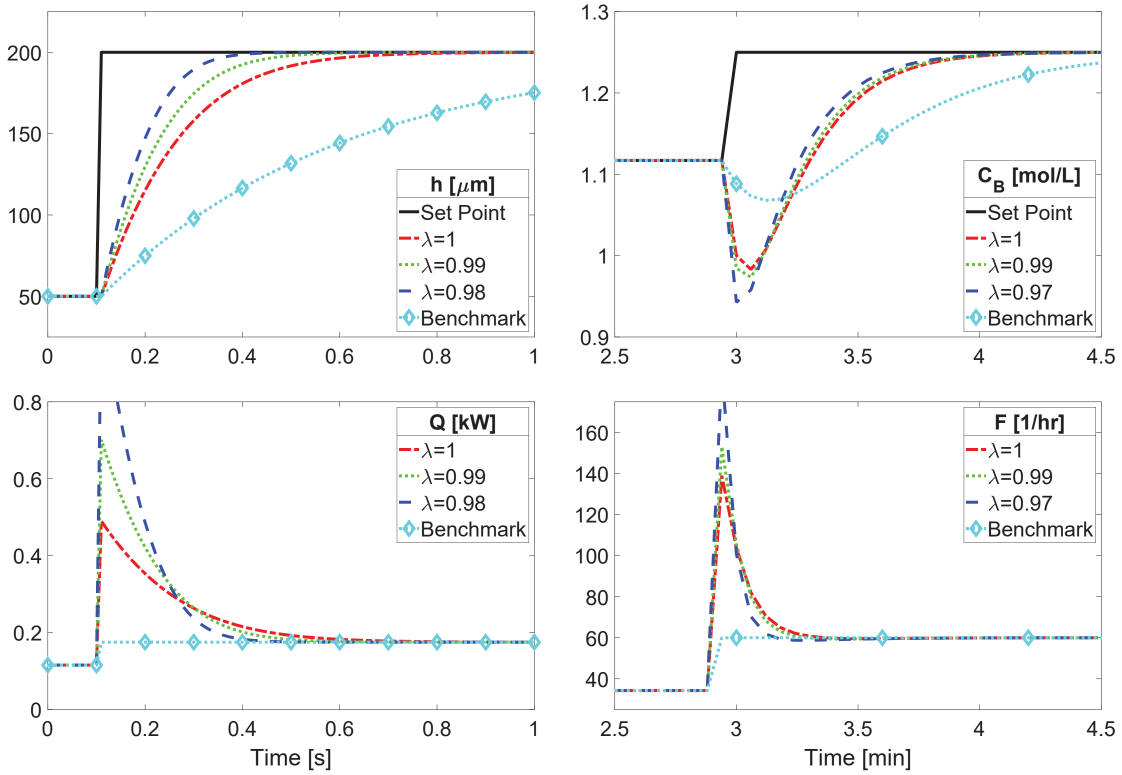
(a) (b)

Figure 6.2: Efficacy of the tuning parameter S with $\lambda = 1$ for (a) LMD Process and (b) VDVR.

Note that the NPFC simulations presented in the following sections utilise a prediction horizon n_x equal to one time constant of the open-loop step response for both case studies.

6.4.2 Analysis of Tuning Efficacy

A core selling point of NPFC is the intuitive nature of controller tuning whereby one can request performance as a relative measure compared against open-loop behaviour, using the so called speed-up factor S; indeed, the simulation results shown in Figure 6.2 demonstrate its significance with improved transient performances. Moreover, the Laguerre pole λ, if chosen carefully, may be useful for fine tuning as shown in Figure 6.3. Nevertheless, a prudent selection of both S and λ is necessary, otherwise the controller could be too aggressive to implement practically.



(a) (b)

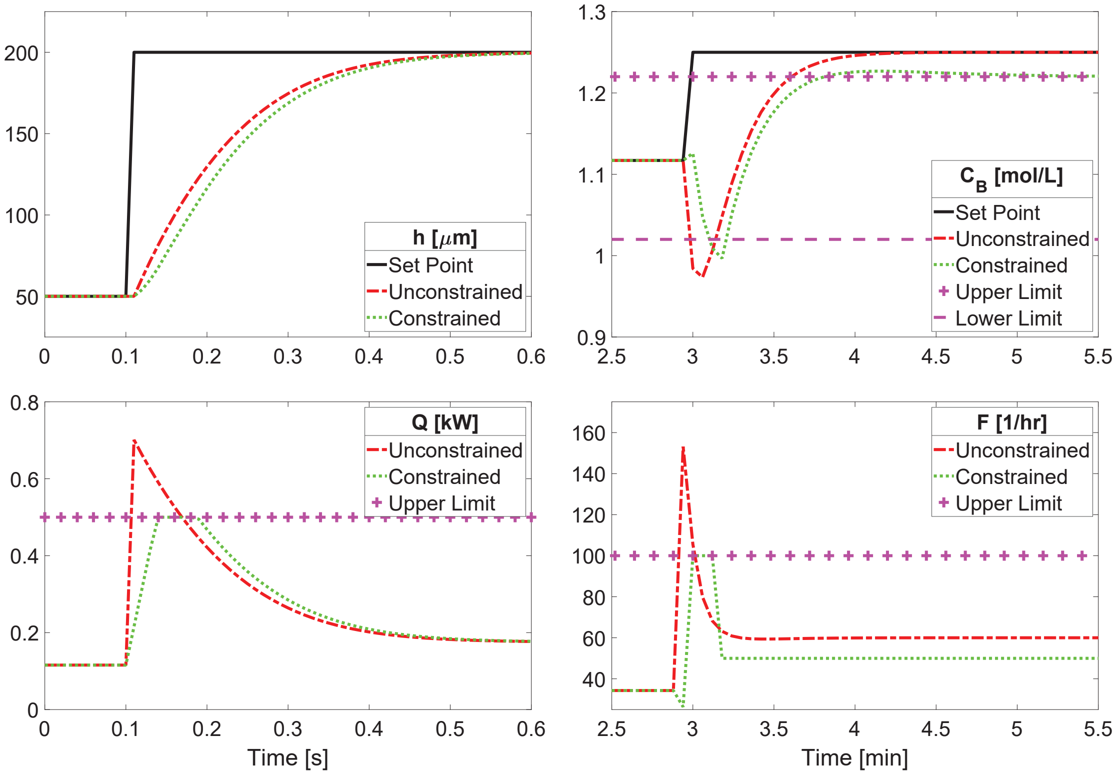
Figure 6.3: Efficacy of the Laguerre pole λ for (a) LMD Process with $S = 1.10$ and (b) VDVR with $S = 2.0$.

6.4.3 Performance with Constraints

Figure 6.4 presents a comparison of unconstrained and constrained output responses by imposing the following process limits: $Q \leq 0.5$ kW and $|\Delta Q| \leq 0.1$ kW/s for LMD, and $0 \leq F \leq 100$ 1/hr and $1.02 \leq C_B \leq 1.22$ mol/L for VDVR. The simulation results demonstrate that while constraint handling with NPFC is straightforward to implement in practice, it unsurprisingly causes a slight slow down in performance. Consequently, both the target S and λ are not achievable up to the desired level (see Figures 6.4a and 6.4b).

6.4.4 Comparisons with RPFC and PID in Practical Scenarios

Finally, we present a comparison between NPFC, RPFC and PID (tuned using the built-in autotuning feature of MATLAB [120]) controllers in the presence of external disturbances,

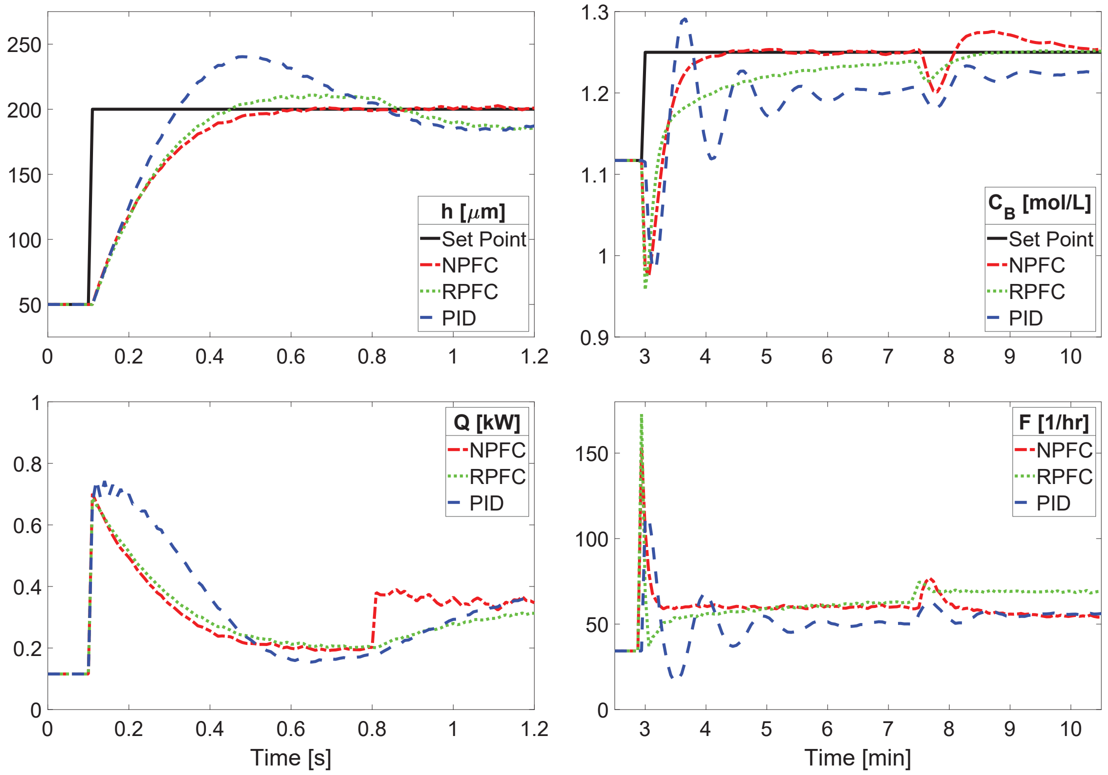


(a) (b)

Figure 6.4: Comparison of constrained and unconstrained performances for (a) LMD Process with $S = 1.10$, $\lambda = 0.99$, $Q \leq 0.5$ kW and $|\Delta Q| \leq 0.1$ kW/s, and (b) VDVR with $S = 2.0$, $\lambda = 0.99$, $0 \leq F \leq 100$ 1/hr and $1.02 \leq C_B \leq 1.22$ mol/L.

measurement noise and parametric uncertainties. A more mainstream NMPC algorithm (such as [121]) is excluded from the analysis as, owing to considerably sophisticated designs, it is expected to outperform these low-cost approaches quite comfortably. Nevertheless, the accompanying EJC 2022 publication [35] contains one such example; interested readers are referred to Appendix I for details.

To demonstrate the impact of uncertainties, the true plant parameters are changed slightly (about 10%) from the nominal values. Moreover, a minute sensor noise is added in the output measurements along with the introduction of following disturbances: a sudden increase in the scan speed v from 2mm/s to 6mm/s around 800ms for the LMD process, and a 5% constant input disturbance around the 8^{th} minute of simulation for the VDVR.



(a) (b)

Figure 6.5: Comparison of (unconstrained) closed-loop performances with uncertainties for (a) LMD Process and (b) VDVR.

The results are shown in Figure 6.5, which unsurprisingly demonstrate the superiority of NPFC in both examples, whereas both linear controllers, i.e. RPFC and PID, suffer due to uncertainties. Specifically, the PID causes large output overshoots whereas the RPFC exhibits relatively slower transients along with sluggish disturbance rejection. This clearly authenticates the efficacy of NPFC which more efficiently approximates the underlying nonlinearities as opposed to the other two alternatives that only implement a fixed process model for control synthesis. Hence, the proposal is expected to be far more effective in applications having wide operating windows, i.e. where large set-point changes may be needed frequently.

6.5 Chapter Summary

The key highlights of this chapter are summarised below:

- The Nonlinear PFC implements a *first-discretise-then-linearise* approach (Section 6.2), wherein the continuous time ODEs are integrated numerically using simple Euler differences (6.3) before obtaining a linear time varying state space model about a nominal trajectory (6.6). This allows manipulating a small input deviation to speed-up the closed-loop behaviour, likewise the RPFC algorithm presented earlier in Chapter 4.

- With the exponential input parametrisation (see Section 6.3), the main computation in NPFC reduces to a simple recursion (6.13), which in terms of modern computing is not significant. Hence, the resulting controller is nearly similar to the standard PFC in cost and complexity, but with the ability to incorporate process nonlinearities far more efficiently, which in turn leads to significanlty superior closed-loop control.

Chapter 7

Conclusions and Future Work

This chapter concludes the thesis by summarising the key findings and contributions in Section 7.1, followed by various recommendations for potential future studies in Section 7.2.

7.1 Final Conclusions

Throughtout Chapters 3-6, this thesis has developed a collection of new and improved PFC strategies with the goal to overcome fundamental tuning weaknesses of the original algorithm and hence extend its usability to a variety of SISO dynamic processes. The key outcomes of this research are summarised in the following sections:

7.1.1 Coping with Tuning Difficulties

The initial part of the study has looked at mitigating tuning challenges associated with well-damped higher-order dynamics via two straightforward proposals.

Firstly, an improved global search algorithm is developed in Chapter 3, which, as opposed to standard methods, performs controller tuning on the basis of predefined input aggression. More specifically, the proposal explicitly limits the expected input activity to a user defined level, thereby restricting the available tuning solutions to only those choices that fulfil this criterion. This by-default leads to more practical designs; the controller automatically adheres to the system's actuation capacity, which in turn minimises the possibility of inadvertent constraint violations due to large set-point changes. Nevertheless, the underlying procedure requires some tedious offline computations that must be carried out in order to reach an appropriate tuning decision.

Secondly, a novel predictive control algorithm, called Relative PFC, is introduced in Chapter 4 which utilises relative statements, rather than absolute measures, to enable straightforward tuning choices with respect to a suitable benchmark behaviour. Consequently, performance tuning reduces to simply one trivial statement, that is how much faster or slower than open-loop do we want the closed-loop system to converge, which significantly simplifies the overall design. As compared to the traditional PFC approaches, the tuning parameter here seems to behave far more consistently so that the user achieves the desired performance to the required degree. Nevertheless, the proposal implicitly assumes satisfactory open-loop behaviour, which means it cannot be applied directly to more challenging process dynamics. For such systems, one possibility is to use pre-stabilisation which is the next contribution of this study.

7.1.2 Systematic Handling of Difficult Prediction Dynamics

The second objective of this research was to overcome the inherent shortcomings of PFC associated with poorly damped and/or unstable process dynamics, for which Chapter 5 has developed a systematic framework utilising the well-known concept of pre-stabilisation from the more mainstream MPC literature.

The proposed Pre-stabilised PFC algorithm is, in essence, a two-stage design process, wherein the designer first employs a well-understood classical feedback control mechanism (such as PID, lead-lag, or indeed more sophisticated approaches based on pole cancellation or pole placement) to modify the difficult open-loop behaviour, thereafter deploying a cascade structure for a reliable PFC implementation. This shows improvements on two main fronts: firstly, the controller tuning after pre-stabilisation becomes far more consistent and meaningful, and secondly, the availability of stable and convergent predictions allows nominal recursive feasibility results under constraints, which is generally not the case with difficult open-loop dynamics. An inevitable consequence of pre-stabilisation, however, is somewhat complicated constraint handling, but given modern computing capacity this is not likely to be a problem.

7.1.3 Efficient Nonlinear Predictive Control

The final contribution of this study is the development of computationally efficient and cheap nonlinear predictive control in Chapter 6.

The proposed Nonlinear PFC is, in principle, based on the novel relative tuning proposal of Chapter 4, which allows relatively low-cost implementations and thus negates the need of expensive consultants for controller deployment and maintenance. This is because, despite being fully nonlinear, the required online computations are relatively minor, akin to the standard PFC, and hence can easily be coded on low-cost processors. Furthermore, the incorporation of systematic constraint handling is straightforward and can be managed with a simple for loop. Another benefit of the proposal is its inherent robustness to parametric uncertainties, which ensures far superior performances than PFC and PID in the presence of undesirable perturbations. Nevertheless, this method is currently restricted to a class of nonlinear systems that mainly exhibit stable and well-damped open-loop behaviour; nonlinear control of more complicated systems has been left as a future work.

7.2 Future Directions

Although this study has provided some significant contributions, there are still opportunities for further improvements that necessitate continued research efforts. Few ideas and suggestions in this regard are listed below:

(i) Tuning ambiguity is one of the core weaknesses of Conventional PFC, and indeed the available tuning guidelines often fail to address this issue satisfactorily. While the new tuning algorithm of Chapter 3 seems to be a meaningful alternative, it is nonetheless a typical global search which may not always be acceptable in practice. Clearly, a method that provides unique tuning choices would be far more practical and is something worth investigating.

(ii) This thesis has investigated the pros and cons of pre-stabilisation with regards to challenging open-loop dynamics. Hence, a possible future work could be to develop a more generic approach, i.e. the one that includes all dynamic types—stable, unstable, oscillatory and non-minimum phase, in order to mitigate the inherent tuning challenges of PFC.

(iii) Another idea is to extend the proposed Nonlinear PFC design to include challenging open-loop dynamics within the framework. Based on the developments in Chapters 5 and 6, it is expected that some form of inner compensation, perhaps using state estimation/feedback, would be needed to develop an effective nonlinear controller in such cases.

(iv) Multivariable PID is quite common in literature that uses simple decoupling strategies (such as, [125, 126]) to transform the Multiple Input Multiple Output (MIMO) problem into mutiple SISO subsytems for straightforward designs. Since PFC is an excellent alternative of PID, a similar decoupling strategy could also be implemented in principle to develop an efficient MIMO PFC controller.

(v) Recent years have witnessed the development of numerous modified PFC variants including Pre-stabilised PFC [31, 84], Laguerre PFC [20], Input Shaping PFC [22, 85], Pole Placement PFC [21, 89], and other PFC designs such as [15, 90]. Hence, it seems sensible to conduct a systematic investigation of these alternatives and consequently develop well-structured guidelines to enable application specific designs in different scenarios and conditions.

References

[1] S. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.

[2] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[3] E. Fernandez-Camacho and C. Bordons-Alba, *Model Predictive Control in the Process Industry.* Springer London, 1995.

[4] R. Vilanova and A. Visioli, Eds., *PID Control in the Third Millennium.* Springer London, 2012.

[5] K. J. Åström and T. Hägglund, *PID controllers: theory, design, and tuning.* Instrument society of America Research Triangle Park, NC, 1995.

[6] V. D. Yurkevich, Ed., *Advances in PID Control.* IntechOpen, 2011. [Online]. Available: https://www.ebook.de/de/product/37157093/advances_in_pid_control.html

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

[8] J. Richalet, A. Rault, J. Testud, and J. Papon, "Model predictive heuristic control," *Automatica (Journal of IFAC)*, vol. 14, no. 5, pp. 413–428, 1978.

[9] J. Richalet, "Pratique de la commande prédictive," *Traité des nouvelles technologies*, 1993.

[10] J. Richalet and D. O'Donovan, *Predictive Functional Control: Principles and Industrial Applications.* Springer Science & Business Media, 2009.

[11] J. A. Rossiter, *A First Course in Predictive Control.* CRC Press, 2018.

[12] M. Bin Abdullah, "Development and analysis of Predictive Functional Control for a range of dynamical processes." Ph.D. dissertation, Department of Automatic Control and Systems Engineering, University of Sheffield, 2018. [Online]. Available: https://etheses.whiterose.ac.uk/23012/

[13] 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.

[14] J. M. Maciejowski, *Predictive Control with Constraints.* Pearson Education, 2002.

[15] M. Khadir and J. Ringwood, "Extension of first order Predictive Functional Controllers to handle higher order internal models," *International Journal of Applied Mathematics and Computer Science*, vol. 18, no. 2, pp. 229–239, 2008.

[16] J. A. Rossiter, "A priori stability results for PFC," *International Journal of Control*, vol. 90, no. 2, pp. 289–297, 2017.

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

[18] J. Richalet, "Industrial applications of model based predictive control," *Automatica*, vol. 29, no. 5, pp. 1251–1274, 1993.

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

[20] M. Abdullah and J. A. Rossiter, "Using Laguerre functions to improve the tuning and performance of predictive functional control," *International Journal of Control*, pp. 1–13, 2019.

[21] J. A. Rossiter, R. Haber, and K. Zabet, "Pole-placement predictive functional control for over-damped systems with real poles," *ISA transactions*, vol. 61, pp. 229–239, 2016.

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

[23] Q. Zhang, X. Lei, and S. Wang, "Nonlinear predictive functional control based on artificial neural network," *IFAC Proceedings Volumes*, vol. 37, no. 1, pp. 797–802, 2004.

[24] B. Zhang and W. Zhang, "Adaptive predictive functional control of a class of nonlinear systems," *ISA Transactions*, vol. 45, no. 2, pp. 175–183, 2006.

[25] X. Liu, L. Ma, X. Kong, and K. Y. Lee, "An efficient iterative learning predictive functional control for nonlinear batch processes," *IEEE Transactions on Cybernetics*, pp. 1–14, 2020.

[26] Y. Wang, S. Li, and B. Zhang, "General regression neural network-based data-driven model-free predictive functional control for a class of discrete-time nonlinear systems," *Nonlinear Dynamics*, vol. 107, no. 1, pp. 953–966, 2021.

[27] M. S. Aftab, J. A. Rossiter, and Z. Zhang, "Predictive Functional Control for unstable first-order dynamic systems," in *Lecture Notes in Electrical Engineering.* Springer International Publishing, 2020, pp. 12–22.

[28] M. S. Aftab and J. A. Rossiter, "Predictive Functional Control with explicit pre-conditioning for oscillatory dynamic systems," in *2021 European Control Conference (ECC).* IEEE, 2021.

[29] M. S. Aftab and J. A. Rossiter, "Pre-stabilised Predictive Functional Control for open-loop unstable dynamic systems," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 147–152, 2021.

[30] J. A. Rossiter and M. S. Aftab, "A comparison of tuning methods for Predictive Functional Control," *Processes*, vol. 9, no. 7, p. 1140, 2021.

[31] M. S. Aftab and J. A. Rossiter, "Predictive functional control for challenging dynamic processes using a simple prestabilization strategy," *Advanced Control for Applications*, 2022.

[32] M. S. Aftab, J. A. Rossiter, and G. Panoutsos, “Predictive functional control for difficult second-order dynamics with a simple pre-compensation strategy,” in *2022 UKACC 13th International Conference on Control (CONTROL).* IEEE, 2022.

[33] M. S. Aftab, J. A. Rossiter, and G. Panoutsos, “Predictive Functional Control for difficult dynamic processes with a simplified tuning mechanism,” in *2022 UKACC 13th International Conference on Control (CONTROL)*, 2022, pp. 130–135.

[34] J. A. Rossiter, M. S. Aftab, and G. Panoutsos, “Exploiting Laguerre polynomials and steady-state estimates to facilitate tuning of PFC,” in *2022 European Control Conference (ECC).* IEEE, 2022.

[35] J. A. Rossiter, M. S. Aftab, G. Panoutsos, and O. Gonzalez-Villarreal, “A novel approach to PFC for nonlinear systems,” *European Journal of Control*, p. 100668, jun 2022.

[36] Y. Ding, L. Wang, Y. Li, and D. Li, “Model predictive control and its application in agriculture: A review,” *Computers and Electronics in Agriculture*, vol. 151, pp. 104–117, 2018.

[37] W. R. Sultana, S. K. Sahoo, S. Sukchai, S. Yamuna, and D. Venkatesh, “A review on state of art development of model predictive control for renewable energy applications,” *Renewable and Sustainable Energy Reviews*, vol. 76, pp. 391–406, 2017.

[38] Y. Huang, H. Wang, A. Khajepour, H. He, and J. Ji, “Model predictive control power management strategies for HEVs: A review,” *Journal of Power Sources*, vol. 341, pp. 91–106, 2017.

[39] A. Mirakhorli and B. Dong, “Occupancy behavior based model predictive control for building indoor climate—A critical review,” *Energy and Buildings*, vol. 129, pp. 499–513, 2016.

[40] A. Afram and F. Janabi-Sharifi, “Theory and applications of HVAC control systems – A review of model predictive control (MPC),” *Building and Environment*, vol. 72, pp. 343–355, 2014.

[41] J. B. Rawlings, “Tutorial overview of model predictive control,” *IEEE Control Systems*, vol. 20, no. 3, pp. 38–52, 2000.

[42] Seborg, Edgar, Mellichamp, and Doyle, *Process Dynamics and Control.* John Wiley and Sons, Inc, 2011.

[43] C. R. Cutler and B. L. Ramaker, “Dynamic matrix control??A computer control algorithm,” *Joint Automatic Control Conference*, vol. 17, p. 72, 1980.

[44] D. Clarke and C. Mohtadi, “Properties of generalized predictive control,” *Automatica*, vol. 25, no. 6, pp. 859–875, 1989.

[45] 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.

[46] J. Lofberg, “Oops! i cannot do it again: Testing for recursive feasibility in MPC,” *Automatica*, vol. 48, no. 3, pp. 550–555, 2012.

[47] E. Mosca and J. Zhang, “Stable redesign of predictive control,” *Automatica*, vol. 28, no. 6, pp. 1229–1233, 1992.

[48] P. Bruijn, L. Bootsma, and H. Verbruggen, “Predictive control using impulse response models,” *IFAC Proceedings Volumes*, vol. 13, no. 9, pp. 315–320, 1980.

[49] R. Rouhani and R. K. Mehra, “Model algorithmic control (MAC); basic theoretical properties,” *Automatica*, vol. 18, no. 4, pp. 401–414, 1982.

[50] J. Richalet and J. Papon, “INDUSTRIAL APPLICATIONS OF INTERNAL MODEL CONTROL,” in *Digital Computer Applications to Process Control.* Elsevier, 1986, pp. 61–67.

[51] H.-B. Kuntze, A. Jacubasch, U. Hirsch, J. Richalet, and C. Arber, “On the application of a new method for fast and robust position control of industrial robots,” in *Proceedings. 1988 IEEE International Conference on Robotics and Automation.* IEEE, 1988.

[52] Compas, Decarreau, Lanquetin, Estival, Fulget, Martin, and Richalet, “Industrial applications of predictive functional control to rolling mill, fast robot, river dam,” in *Proceedings of IEEE International Conference on Control and Applications CCA-94.* IEEE, 1994.

[53] J. Richalet, J. Estival, and P. Fiani, "Industrial applications of predictive functional control to metallurgical industries," in *Proceedings of International Conference on Control Applications.* IEEE, 1995.

[54] A.-D. A. el, P. Fiani, and J. Richalet, "Handling input and state constraints in predictive functional control," in *[1991] Proceedings of the 30th IEEE Conference on Decision and Control.* IEEE, 1991.

[55] M. Abdullah, J. A. Rossiter, and A. F. A. Ghaffar, "IMPROVED CONSTRAINT HANDLING APPROACH FOR PREDICTIVE FUNCTIONAL CONTROL USING AN IMPLIED CLOSED-LOOP PREDICTION," *IIUM Engineering Journal*, vol. 22, no. 1, pp. 323–338, 2021.

[56] M. Abdullah and J. A. Rossiter, "The effect of model structure on the noise and disturbance sensitivity of Predictive Functional Control," in *2018 European Control Conference (ECC).* IEEE, 2018, pp. 1009–1014.

[57] M. Abdullah and J. A. Rossiter, "A formal sensitivity analysis for Laguerre based predictive functional control," in *2018 UKACC 12th International Conference on Control (CONTROL).* IEEE, 2018.

[58] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, 2nd ed. Wiley New York, 2007.

[59] J. M. Diaz, R. Costa-Castello, and S. Dormido, "Closed-loop shaping linear control system design: An interactive teaching/learning approach [focus on education]," *IEEE Control Systems Magazine*, vol. 39, no. 5, pp. 58–74, 2019.

[60] K. Zabet and R. Haber, "Robust tuning of PFC (Predictive Functional Control) based on first- and aperiodic second-order plus time delay models," *Journal of Process Control*, vol. 54, pp. 25–37, 2017.

[61] J. Normey-Rico, C. Bordons, and E. Camacho, "Improving the robustness of dead-time compensating PI controllers," *Control Engineering Practice*, vol. 5, no. 6, pp. 801–810, 1997.

[62] M. Zainuddin, M. Abdullah, S. Ahmad, and K. Tofrowaih, "Performance comparison between Predictive Functional Control and PID algorithms for an automobile cruise control system." *International Journal of Automotive and Mechanical Engineering. ISSN 2229-8649 E-ISSN 2180-1606 (In Press)*, 2022.

[63] R. Haber, U. Schmitz, and K. Zabet, "Implementation of PFC (Predictive Functional Control) in a petrochemical plant," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 5333–5338, 2014.

[64] J. Zhang, "Design of a new PID controller using predictive functional control optimization for chamber pressure in a coke furnace," *ISA Transactions*, vol. 67, pp. 208–214, 2017.

[65] W. X. Jing, L. M. Zhen, C. Shuai, L. Hao, and X. Wang, "PID–PFC control of continuous rotary electro-hydraulic servo motor applied to flight simulator," *The Journal of Engineering*, vol. 2019, no. 13, pp. 138–143, 2019.

[66] M. T. Khadir, "Enthalpy predictive functional control of a pasteurisation plant based on a plate heat exchanger," in *2007 European Control Conference (ECC)*. IEEE, 2007.

[67] M. T. Khadir, "Modelling and predictive control of a milk pasteurisation plant," Ph.D. dissertation, National University of Ireland, Maynooth, 2002.

[68] M. Abdelghani-Idrissi, M. Arbaoui, L. Estel, and J. Richalet, "Predictive functional control of a counter current heat exchanger using convexity property," *Chemical Engineering and Processing: Process Intensification*, vol. 40, no. 5, pp. 449–457, 2001.

[69] J. A. Rossiter and M. Abdullah, "A new paradigm for Predictive Functional Control to enable more consistent tuning," in *2019 American Control Conference (ACC)*. IEEE, 2019.

[70] 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, 2020.

[71] M. T. Khadir and J. Ringwood, “Stability issues for first order predictive functional controllers: Extension to handle higher order internal models,” in *International Conference on Computer Systems and Information Technology*, 2005.

[72] J. A. Rossiter, “Stable prediction for unstable independent models,” *IEEE Transactions on Automatic Control*, vol. 48, no. 11, pp. 2029–2035, 2003.

[73] M. Abdullah and J. A. Rossiter, “Utilising Laguerre function in predictive functional control to ensure prediction consistency,” in *2016 UKACC 11th International Conference on Control (CONTROL)*. IEEE, 2016.

[74] M. Abdullah, J. A. 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.

[75] D. Clarke and R. Scattolini, “Constrained receding-horizon predictive control,” *IEE Proceedings D Control Theory and Applications*, vol. 138, no. 4, p. 347, 1991.

[76] B. Kouvaritakis, J. A. Rossiter, and A. Chang, “Stable generalised predictive control: an algorithm with guaranteed stability,” *IEE Proceedings D Control Theory and Applications*, vol. 139, no. 4, p. 349, 1992.

[77] J. A. Rossiter, J. Gossner, and B. Kouvaritakis, “Infinite horizon stable predictive control,” *IEEE Transactions on Automatic Control*, vol. 41, no. 10, pp. 1522–1527, 1996.

[78] J. R. Gossner, B. Kouvaritakis, and J. A. Rossiter, “Cautious stable predictive control: A guaranteed stable predictive control algorithm with low input activity and good robustness,” *International Journal of Control*, vol. 67, no. 5, pp. 675–698, 1997.

[79] J. A. Rossiter, J. Gossner, and B. Kouvaritakis, “Constrained cautious stable predictive control,” *IEE Proceedings - Control Theory and Applications*, vol. 144, no. 4, pp. 313–323, 1997.

[80] J. A. Rossiter and J. Richalet, *Predictive functional control of unstable processes.* Department of Automatic Control and Systems Engineering, 2001.

[Online]. Available: http://eprints.whiterose.ac.uk/84452/1/acse%20research%20report%20811.pdf

[81] J. A. Rossiter, “Predictive functional control: more than one way to prestabilise,” *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 289–294, 2002.

[82] Z. Zhang, J. A. Rossiter, L. Xie, and H. Su, “Predictive functional control for integral systems,” in *International Symposium on Process System Engineering*, 2018.

[83] M. Abdullah and J. A. Rossiter, “Alternative method for Predictive Functional Control to handle an integrating process,” in *2018 UKACC 12th International Conference on Control (CONTROL).* IEEE, 2018, pp. 26–31.

[84] Z. Zhang, J. A. Rossiter, L. Xie, and H. Su, “Predictive functional control for integrator systems,” *Journal of the Franklin Institute*, vol. 357, no. 7, pp. 4171–4186, 2020.

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

[86] J. A. Rossiter, *Predictive functional control: To pre-stabilise or not?* Department of Automatic Control and Systems Engineering, 2001. [Online]. Available: http://eprints.whiterose.ac.uk/84450/1/acse%20research%20report%20810.pdf

[87] J. A. Rossiter and J. Richalet, “Handling constraints with predictive functional control of unstable processes,” in *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, vol. 6. IEEE, 2002, pp. 4746–4751.

[88] J. A. Rossiter, R. Haber, and K. Zabet, “Pole-placement PFC (Predictive Functional Control) for systems with one oscillatory mode,” in *2016 European Control Conference (ECC).* IEEE, 2016.

[89] K. Zabet, J. A. Rossiter, R. Haber, and M. Abdullah, “Pole-placement predictive functional control for under-damped systems with real numbers algebra,” *ISA transactions*, vol. 71, pp. 403–414, 2017.

[90] Z. Zhang, L. Xie, S. Lu, J. A. Rossiter, and H. Su, "A low-cost pole-placement MPC algorithm for controlling complex dynamic systems," *Journal of Process Control*, vol. 111, pp. 106–116, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959152422000154

[91] J. Rawlings, E. Meadows, and K. Muske, "Nonlinear model predictive control: A tutorial and survey," *IFAC Proceedings Volumes*, vol. 27, no. 2, pp. 185–197, 1994.

[92] M. A. Henson, "Nonlinear model predictive control: current status and future directions," *Computers & Chemical Engineering*, vol. 23, no. 2, pp. 187–202, 1998.

[93] L. Grune and J. Pannek, *Nonlinear Model Predictive Control.* Springer-Verlag GmbH, 2016. [Online]. Available: https://www.ebook.de/de/product/27995742/lars_gruene_juergen_pannek_nonlinear_model_predictive_control.html

[94] O. J. Gonzalez Villarreal, "Efficient Real-Time Solutions for Nonlinear Model Predictive Control with Applications." Ph.D. dissertation, Department of Automatic Control and Systems Engineering, University of Sheffield, 2021. [Online]. Available: https://etheses.whiterose.ac.uk/29336/

[95] L. T. Biegler, "A perspective on nonlinear model predictive control," *Korean Journal of Chemical Engineering*, vol. 38, no. 7, pp. 1317–1332, 2021.

[96] M. Ławryńczuk, *Computationally Efficient Model Predictive Control Algorithms.* Springer-Verlag GmbH, 2014. [Online]. Available: https://www.ebook.de/de/product/22844051/maciej_lawrynczuk_computationally_efficient_model_predictive_control_algorithms.html

[97] M. Ławryńczuk, "Practical nonlinear predictive control algorithms for neural Wiener models," *Journal of Process Control*, vol. 23, no. 5, pp. 696–714, 2013.

[98] M. Ławryńczuk, "Suboptimal nonlinear predictive control based on multivariable neural Hammerstein models," *Applied Intelligence*, vol. 32, no. 2, pp. 173–192, 2010.

[99] R. A. Seyab and Y. Cao, "Nonlinear model predictive control for the ALSTOM gasifier," *Journal of Process Control*, vol. 16, no. 8, pp. 795–808, 2006.

[100] B. Ogunnaike, *Process dynamics, modeling, and control.* New York, NY: Oxford University Press, 1994.

[101] G. R. Sriniwas and Y. Arkun, “A global solution to the nonlinear model predictive control algorithms using polynomial ARX models,” *Computers & Chemical Engineering*, vol. 21, no. 4, pp. 431–439, 1997.

[102] J. K. Gruber, C. Bordons, R. Bars, and R. Haber, “Nonlinear predictive control of smooth nonlinear systems based on Volterra models. Application to a pilot plant,” *International Journal of Robust and Nonlinear Control*, vol. 20, no. 16, pp. 1817–1835, 2009.

[103] P. Tatjewski, *Advanced Control of Industrial Processes.* Springer-Verlag GmbH, 2007. [Online]. Available: https://www.ebook.de/de/product/8896261/piotr_tatjewski_advanced_control_of_industrial_processes.html

[104] M. Norgaard, N. K. Poulsen, O. Ravn, and L. K. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems.* Springer London, 2003. [Online]. Available: https://www.ebook.de/de/product/3764252/m_norgaard_n_k_poulsen_o_ravn_l_k_hansen_neural_networks_for_modelling_and_control_of_dynamic_systems.html

[105] M. Schoukens and K. Tiels, “Identification of block-oriented nonlinear systems starting from linear approximations: A survey,” *Automatica*, vol. 85, pp. 272–292, 2017.

[106] M. Ławryńczuk, “Nonlinear predictive control of dynamic systems represented by Wiener–Hammerstein models,” *Nonlinear Dynamics*, vol. 86, no. 2, pp. 1193–1214, 2016.

[107] M. Ławryńczuk, “Nonlinear predictive control for Hammerstein–Wiener systems,” *ISA Transactions*, vol. 55, pp. 49–62, 2015.

[108] K. Åström, *Introduction to stochastic control theory*, ser. Mathematics in science and engineering. United States: Academic Press, 1970, vol. 70.

[109] 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.

[110] K. Ogata, *Discrete-Time Control Systems*, 2nd ed. Prentice Hall Englewood Cliffs, NJ, 1995.

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

[112] G. Goodwin, *Control system design.* Upper Saddle River, N.J: Prentice Hall, 2001.

[113] E. G. Gilbert, I. Kolmanovsky, and K. T. Tan, "Discrete-time reference governors and the nonlinear control of systems with state and control constraints," *International Journal of Robust and Nonlinear Control*, vol. 5, no. 5, pp. 487–504, 1995.

[114] S. Raković, B. Kouvaritakis, M. Cannon, and C. Panos, "Fully parameterized tube model predictive control," *International Journal of Robust and Nonlinear Control*, vol. 22, no. 12, pp. 1330–1361, 2012.

[115] J. Hu and B. Ding, "An efficient offline implementation for output feedback min-max MPC," *International Journal of Robust and Nonlinear Control*, vol. 29, no. 2, pp. 492–506, 2018.

[116] U. Hasirci, "An engineering education tool for real-time nonlinear control: Stabilization of a single link robot," *The International Journal of Electrical Engineering Education*, vol. 52, no. 4, pp. 320–339, 2015.

[117] MathWorks, "Model Reducer: Reduce complexity of linear time-invariant (LTI) models," https://uk.mathworks.com/help/control/ref/modelreducer-app.html?searchHighlight=model2022. [Online]. Available: https://uk.mathworks.com/help/control/ref/modelreducer-app.html?searchHighlight=modelreducer&s_tid=srchtitle_model%20reducer_1

[118] B. Bequette, "Behavior of a CSTR with a recirculating jacket heat transfer system," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301).* IEEE, 2002.

[119] A. S. Rao and M. Chidambaram, "Analytical design of modified Smith predictor in a two-degrees-of-freedom control scheme for second order unstable processes with time delay," *ISA Transactions*, vol. 47, no. 4, pp. 407–419, 2008.

[120] MathWorks, "PID Tuning Algorithm," https://www.mathworks.com/help/control/getstart/pid-tuning-algorithm.html, 2021. [Online]. Available: https://www.mathworks.com/help/control/getstart/pid-tuning-algorithm.html

[121] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, "From linear to nonlinear MPC: bridging the gap via the real-time iteration," *International Journal of Control*, vol. 93, no. 1, pp. 62–80, 2016.

[122] T. S. A. M. Bayen, *An Introduction to MATLAB® Programming and Numerical Methods for Engineers.* Elsevier, 2015.

[123] Q. Wang, J. Li, M. Gouge, A. R. Nassar, P. P. Michaleris, and E. W. Reutzel, "Physics-based multivariable modeling and feedback linearization control of melt-pool geometry and temperature in Directed Energy Deposition," *Journal of Manufacturing Science and Engineering*, vol. 139, no. 2, 2016.

[124] S. Kuntanapreeda and P. M. Marusak, "Nonlinear extended output feedback control for CSTRs with van de vusse reaction," *Computers & Chemical Engineering*, vol. 41, pp. 10–23, 2012.

[125] L. Liu, S. Tian, D. Xue, T. Zhang, Y. Chen, and S. Zhang, "A review of industrial MIMO decoupling control," *International Journal of Control, Automation and Systems*, vol. 17, no. 5, pp. 1246–1254, 2019.

[126] M. Waller, J. B. Waller, and K. V. Waller, "Decoupling revisited," *Industrial & Engineering Chemistry Research*, vol. 42, no. 20, pp. 4575–4577, 2003.

Appendix A

Predictive Functional Control for Unstable First-Order Dynamic Systems

Muhammad Saleheen Aftab, John Anthony Rossiter, and Zhiming Zhang

This paper has been published in the Proceedings of 14th International Conference on Automatic Control and Soft Computing (CONTROLO 2020), Portugal, 2020

Author Contributions. This paper is a collaborative work between all authors. M. S. Aftab proposed the idea, analysed the concept in case studies, and prepared the initial draft of the paper. J .A. Rossiter provided accurate communication of the earlier PFC and MPC control laws, supervised M. S. Aftab and reviewed the whole project. Z. Zhang reviewed the paper and provided suggestions for improvement.

Predictive Functional Control for Unstable First-Order Dynamic Systems

Muhammad Saleheen Aftab[1](✉), John Anthony Rossiter[1], and Zhiming Zhang[2]

[1] Department of Automatic Control and Systems Engineering,
University of Sheffield, Sheffield, UK
{msaftab1,j.a.rossiter}@sheffield.ac.uk
[2] State Key Laboratory of Industrial Control Technology, Zhejiang University,
Hangzhou, China
zhangzhimingzju@zju.edu.cn

Abstract. Predictive functional control (PFC) has emerged as a popular industrial choice owing to its simplicity and cost-effectiveness. Nevertheless, its efficacy diminishes when dealing with challenging dynamics because of prediction mismatch in such scenarios. This paper presents a proposal for reducing prediction mismatch and thus improving behaviour for simple unstable processes; a two-stage design methodology pre-stabilises predictions via proportional compensation before introducing the PFC component. It is demonstrated that pre-stabilisation reduces the dependency of the closed-loop pole on the coincidence point and also improves robustness to uncertainty. Simulation results verify the improved performance as compared to conventional PFC.

Keywords: PFC · Coincidence horizon · Pre-stabilisation · Proportional compensation

1 Introduction

Predictive functional control (PFC) offers numerous beneficial attributes such as trivial coding, easy implementation and simple handling without needing sophisticated knowledge, software or specialised personnel. These qualities, along with systematic handling of constraints and dead-times compared to other conventional methods, say proportional-integral-derivative (PID) control, make PFC a popular alternative in industry, with numerous successful applications [1].

Conventional PFC [1–3] matches the plant output prediction to a desired first-order target trajectory at only one future point, the so-called coincidence point, by keeping the predicted input constant. One may ask if there exists a reliable criterion for selecting the desired target dynamics and coincidence point? Researchers have established generic guidelines for systems with relatively benign dynamics. For example, it is recommended [2] to use a one-step ahead model prediction for first-order plant as this guarantees target behaviour for first-order systems [4]. Alternatively, one recommendation for higher-order systems is to choose the point of inflection (where the gradient is maximum) on the

J. A. Gonçalves et al. (Eds.): CONTROLO 2020, LNEE 695, pp. 12–22, 2021.
https://doi.org/10.1007/978-3-030-58653-9_2

step response curve as the coincidence point although it is arguable whether this would work well for systems with challenging dynamics. Moreover, for monotonically convergent higher-order systems, a coincidence point where the open-loop step response has risen to approximately 40–80% of the steady-state is often a better choice [4]. Nevertheless, matching underdamped, unstable and non-minimum phase dynamics with target first-order behaviour does not make sense and coincidence point selection for such systems is not straight-forward. Challenging dynamics demand a different parametrisation of the degrees-of-freedom [5], as the typical constant input assumption within the prediction horizon may be inappropriate. One recent attempt [6] parametrised the input with first-order Laguerre polynomial, which improves prediction consistency and convergence rate as compared to the original PFC for systems with simple dynamics; however, this approach is not really tailored to systems with difficult dynamics.

The main objective of this paper is to build on the ideas in [5,7] and indeed conventional wisdom in PFC [2] which is to modify unstable dynamics before applying the PFC design. Accepted practice in the mainstream MPC community uses pre-stabilisation [8,9], so this paper proposes a a two-stage PFC design methodology by integrating pre-stabilised dynamics with PFC decision making. Initially we restrict our study to first-order unstable plants focusing on the effects of a pre-stabilising structure on closed-loop performance, sensitivity and constraint handling. Specifically this paper analyses the relationship between the target pole, pre-stabilising gain and coincidence horizon and establishes guidelines for systematic and effective tuning. Generally a trade-off between closed-loop performance and sensitivity is observed, which signifies the importance of offline sensitivity analysis for proper selection of tuning parameters; something not in the conventional PFC literature. With pre-stabilisation, numerical simulations show improved closed-loop performance as compared to conventional PFC. Extensions for systems with higher-order dynamics constitutes future work.

The remainder of this paper is organised as follows: Sect. 2 succinctly formulates the control problem. Section 3 proposes the two-stage PFC and discusses sensitivity analysis, tuning procedures and constraint handling. Section 4 presents the numerical illustrations. Finally the paper concludes in Sect. 5.

2 Problem Statement

Consider an unstable first-order plant given by:

$$G_p(z) = \frac{b_p z^{-1-w}}{1 + a_p z^{-1}} \tag{1}$$

where a_p and b_p are the plant parameters, w is the system delay and $|a_p| \geq 1$ represents the open-loop unstable pole. The system Eq. (1) is subject to input, input rate and output constraints i.e.

$$u_{\min} \leq u(k) \leq u_{\max} \qquad \Delta u_{\min} \leq \Delta u(k) \leq \Delta u_{\max} \qquad y_{\min} \leq y(k) \leq y_{\max} \tag{2}$$

where $\Delta = 1 - z^{-1}$ is the difference operator. The objective is to design a PFC by first stabilising the prediction dynamics. Furthermore the controller is expected to show some degree of robustness against measurement noise, disturbances and multiplicative uncertainty.

3 Two-Stage Predictive Functional Control

This section proposes a two-stage design approach to controlling the unstable system with PFC. In stage one, the prediction model is stabilised offline through proportional compensation before employing PFC. It should be noted that although open-loop PFC may stabilise unstable systems in an unconstrained environment, pre-stabilisation is necessary for accurate constraint handling. Denote the system model representing (1) as $G_m(z)$, ($a_m = a_p$ and $b_m = b_p$ if $G_m = G_p$):

$$G_m(z) = \frac{b_m z^{-1}}{1 + a_m z^{-1}} \tag{3}$$

The dead-time w is excluded from the prediction model and is added separately in the PFC control law. Next we discuss two alternatives to stabilise system Eq. (3).

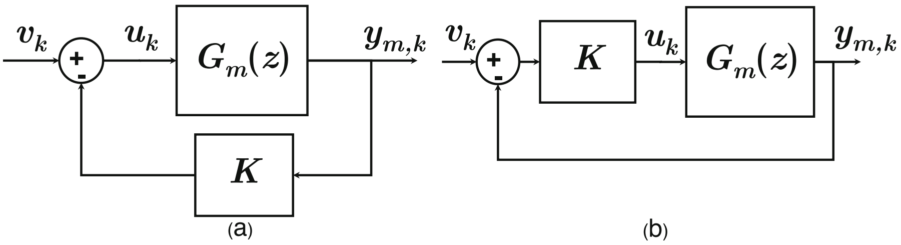


Fig. 1. Pre-stabilisation with proportional compensation

3.1 Stage-1: Model Pre-stabilisation

The delay-free model (3) can be stabilised with proportional compensation either in the feedback path (Fig. 1(a)) or in the forward path (Fig. 1(b)). The closed-loop transfer function for both cases has the form:

$$T_m(z) = \frac{y_m(z)}{v(z)} = \frac{\beta z^{-1}}{1 + \alpha z^{-1}} \tag{4}$$

where $\beta = b_m$ and $\beta = K b_m$ for compensation in feedback and forward paths respectively and $\alpha = a_m + K b_m$. Evidently $T_m(z)$ is stable if $0 \leq |\alpha| < 1$. Moreover, the input u_k for feedback path compensation is parameterised as:

$$u_k = v_k - K y_{m,k} \tag{5}$$

and for forward path compensation as:

$$u_k = K(v_k - y_{m,k}) \tag{6}$$

The implementation of PFC with Fig. 1(a) for integral systems only was reported verbally in [7]. The current study generalises this concept for unstable dynamics and analyses the potential merits and demerits against the structure of Fig. 1(b). The expectation is to gain useful insights for generalising pre-conditioning with more advanced compensation for more complex plants.

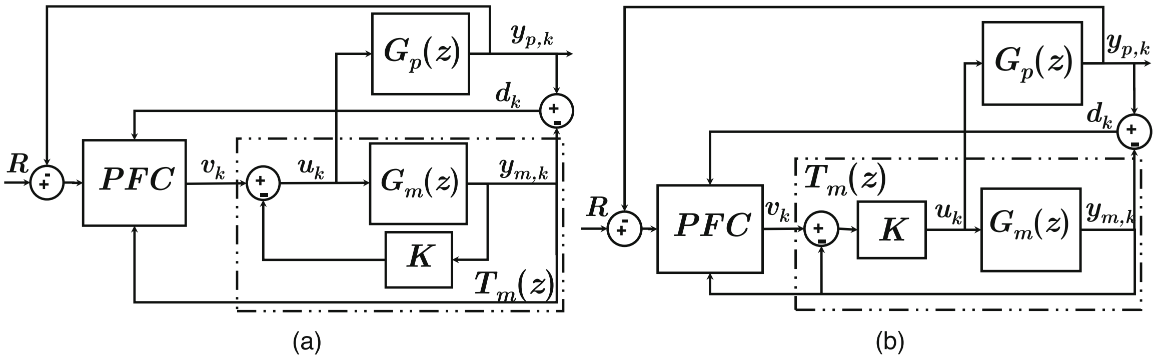


Fig. 2. PPFC structure—PFC on pre-stabilised model with proportional gain in (a) feedback path, and (b) forward path

3.2 Stage-2: PFC Design

The pre-stabilised PFC (PPFC) structure employing a PFC loop on the stabilised model is shown in Fig. 2. In PFC, the output prediction, $y_{p,k}$ is required to follow target first-order dynamics such that:

$$y_{p,k+i} = R - (R - y_{p,k})\rho^i \tag{7}$$

where R is the steady-state set-point value and ρ is the target closed-loop pole. The PFC control law matches the output prediction $y_{p,k+i}$ and target output $R - (R - y_{p,k})\rho^i$ at a single point in future, known as the coincidence point h, while assuming a constant predicted input, i.e. $v_k = v_{k+i|k}$, $\forall i > 0$. Hence, after recursion on model (4), an i-step ahead model prediction is obtained [1,3]:

$$y_{m,k+i} = (-\alpha)^i y_{m,k} + [(-\alpha)^{i-1}\beta + (-\alpha)^{i-2}\beta + \cdots + \beta]v_k \tag{8}$$

The prediction Eq. (8), requires correction from bias due to uncertainties with the offset term d_k where $d_k = y_{p,k} - y_{m,k}$. Thus PFC is defined from:

$$y_{p,k+i} = y_{m,k+i} + d_k = R - (R - y_{p,k})\rho^i \tag{9}$$

Substituting from (8), the solution to (9), or PPFC law, is given as:

$$v_k = \frac{R - (R - y_{p,k})\rho^h - (-\alpha)^h y_{m,k} - d_k}{\sum_{j=1}^{h}(-\alpha)^{h-j}\beta} \tag{10}$$

Theorem 1. *For a given ρ and h either pre-stabilisation technique results in the same control law provided equal proportional gain is used.*

Proof. First using Eq. (10) in Eq. (5) with $\beta = b_m$, gives:

$$u_k^{fback} = \frac{R - (R - y_{p,k})\rho^h - (-\alpha)^h y_{m,k} - d_k}{\sum_{j=1}^{h}(-\alpha)^{h-j} b_m} - K y_{m,k} \tag{11}$$

Now using Eq. (10) in Eq. (6) with $\beta = K b_m$:

$$u_k^{forward} = K\left[\frac{R - (R - y_{p,k})\rho^h - (-\alpha)^h y_{m,k} - d_k}{K\sum_{j=1}^{h}(-\alpha)^{h-j} b_m} - y_{m,k}\right] = u_k^{fback}$$

Thus same control law results irrespective of the pre-stabilisation technique. □

Remark 1. Theorem 1 shows there is no obvious advantage of either pre-stabilisation method. Thus for complex systems, pre-conditioning in the feedback path is expected to give same performance as in the forward path.

Remark 2. System delays can be easily incorporated into PFC control law [3] by noting that $E(y_{p,k+w}) = y_{p,k} + y_{m,k} - y_{m,k-w}$. Therefore Eq. (10) becomes:

$$v_k = \frac{R - [R - E(y_{p,k+w})]\rho^h - (-\alpha)^h y_{m,k} - d_k}{\sum_{j=1}^{h}(-\alpha)^{h-j}\beta} \tag{12}$$

where $d_k = y_{p,k} - y_{m,k-w}$. When $w = 0$, Eq. (10) and Eq. (12) are no different.

3.3 Sensitivity Analysis

The ability of a feedback loop to reject unwanted perturbations in the form of noise, disturbance and multiplicative uncertainty can be assessed with frequency domain sensitivity analysis [10]. Control law (11) can be re-arranged as:

$$u_k = F(z)R - M(z)y_{p,k} - N(z)y_{m,k} \tag{13}$$

where $F(z)$, $M(z)$ and $N(z)$ are appropriate polynomials. Note further:

$$\{y_{m,k} = G_m(z)u_k, \quad (13)\} \quad \Rightarrow \quad D(z)u_k = F(z)R - M(z)y_{p,k} \tag{14}$$

with $D(z) = 1 + N(z)G_m(z)$. Eq. (14) is represented in the block diagram of Fig. 3 where disturbance $d_{y,k}$ and measurement noise n_k are also shown; the effective control law is $C(z) = M(z)D^{-1}(z)$. Consequently, $P_C(z) = 1 + C(z)G_p(z) = D(z)A(z) + M(z)B(z)$ is the closed-loop pole polynomial. From Fig. 3, sensitivity of the plant input to noise is found to be:

$$S_{un}(z) = C(z)[1 + C(z)G_p(z)]^{-1} = M(z)P_C^{-1}(z)A(z) \tag{15}$$

whereas sensitivity of the plant output to disturbance is:

$$S_{yd}(z) = [1 + C(z)G_p(z)]^{-1} = A(z)P_C^{-1}(z)D(z) \tag{16}$$

Sensitivity $S_\delta(z)$ of the closed-loop pole to multiplicative uncertainty uses:

$$\begin{aligned} P_C(z) &= 1 + C(z)[G_p(z) + \delta G_p(z)] \\ &= [1 + C(z)G_p(z)]\left(1 + \delta C(z)G_p(z)[1 + C(z)G_p(z)]^{-1}\right) \end{aligned}$$

where δ is possibly a frequency dependent scalar. Thus:

$$S_\delta(z) = C(z)G_p(z)[1 + C(z)G_p(z)]^{-1} = M(z)P_C^{-1}(z)B(z) \tag{17}$$

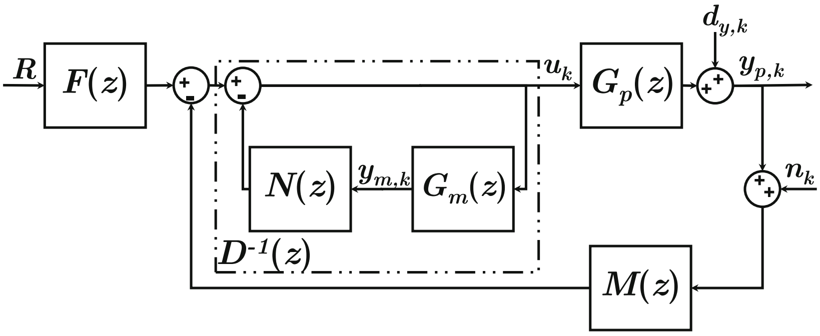


Fig. 3. PPFC block diagram for sensitivity analysis

3.4 Tuning

There are two tuning parameters for a given target pole ρ: the pre-stabilising gain K and the coincidence point h. K determines the position of the pole $|\alpha|$ in z-plane which logically should be restricted between ρ and 1. Therefore K can be tuned within a range where K_U and K_L are upper and lower limits:

$$K_L = -\left(\frac{1 + a_m}{b_m}\right) < K \le -\left(\frac{\rho + a_m}{b_m}\right) = K_U \tag{18}$$

Theorem 2. *Closed-loop pole $z_{CL} = \rho$ is guaranteed if either*

(i) proportional gain $K = K_U$ irrespective of h, or
(ii) coincidence point $h = 1$ irrespective of K.

Proof. We know from [4] that:

$$z_{CL} = -\alpha + \frac{\rho^h - (-\alpha)^h}{\sum_{j=1}^{h}(-\alpha)^{h-j}}$$

(i) selecting $K = -(\rho + a_m)/b_m$ implies:

$$\alpha = a_m - \left(\frac{\rho + a_m}{b_m}\right) b_m = -\rho$$

Consequently $z_{CL} = \rho$ is guaranteed irrespective of h.

(ii) selecting $h = 1$ implies:

$$z_{CL} = -\alpha + \frac{\rho^1 - (-\alpha)^1}{1} = \rho$$

Hence $z_{CL} = \rho$ is guaranteed irrespective of K. □

Corollary 1. *If $K = K_U$ then S_{un} and S_δ are independent of h.*

Proof. $A(z) = 1 + a_p z^{-1}$ and $B(z) = b_p z^{-1}$ do not involve h, plus Theorem 2 proves $P_C(z)$ does not depend on h either. This leaves only $M(z)$ to check. First note that for $K = -(\rho + a_m)/b_m$,

$$\sum_{j=1}^{h} (-\alpha)^{h-j} b_m = \frac{b_m[1 - (-\alpha)^h]}{1 + \alpha} = \frac{b_m(1 - \rho^h)}{1 - \rho} \quad \Rightarrow \quad M(z) = \frac{1 - \rho^h}{\frac{b_m(1-\rho^h)}{1-\rho}} = \frac{1 - \rho}{b_m}$$

which makes $M(z)$ free from h. Thus both S_{un} and S_δ are independent of h. □

Algorithm 1. Select K mid way in its range i.e. $K = (K_U + K_L)/2$. This implies $\rho \leq z_{CL} < (1 + \rho)/2$ for $1 \leq h < \infty$ and also keeps h relevant for tuning offline sensitivity functions.

3.5 Constraint Handling

Another important aspect is the proper handling of constraints. Since pre-stabilisation changes the PFC control variable to v_k, this implies a transfer of constraints from u_k to v_k is necessary, for example via a process of back calculation [2]. For pre-stabilisation with proportional gain in the feedback path:

$$\begin{aligned} u_{min} + K y_{m,k} \leq & v_k \leq u_{max} + K y_{m,k} \\ \Delta u_{min} + K \Delta y_{m,k} \leq & \Delta v_k \leq \Delta u_{max} + K \Delta y_{m,k} \end{aligned} \tag{19}$$

and if proportional gain is placed in the forward path:

$$\begin{aligned} \frac{u_{min}}{K} + y_{m,k} \leq & v_k \leq \frac{u_{max}}{K} + y_{m,k} \\ \frac{\Delta u_{min}}{K} + \Delta y_{m,k} \leq & \Delta v_k \leq \frac{\Delta u_{max}}{K} + \Delta y_{m,k} \end{aligned} \tag{20}$$

Output constraints on the other hand are incorporated through predictions (8). At each time sample k, output constraints have to be satisfied throughout and beyond the coincidence horizon, that is until the predictions have settled [9]. From Eqs. (8)–(9), the predictions for constraint horizon n_c are:

$$y_{p,k+j} = P_j y_{m,k} + H_j v_k + L_j; \quad j = 1, 2, \ldots, n_c, \quad n_c \gg h \tag{21}$$

Therefore the output constraints $y_{min} \leq y_{p,k} \leq y_{max}$ are transferred to:

$$y_{min} \leq P_j y_{m,k} + H_j v_k + L_j \leq y_{max}; \quad j = 1, 2, \ldots, n_c \tag{22}$$

One can utilise a simple loop to test each constraint in turn and select the v_k closest to the nominal value from (10) which satisfies all the constraints [6].

Theorem 3. *Given that v_{k-1} is feasible by assumption and one is able to select $v_k = v_{k-1}$, constraints can always be satisfied in the nominal case as long as n_c is large enough. Proof equivalent to that in [6].*

4 Simulation Results and Discussion

This section examines the performance of the proposed PPFC controller and compares it with conventional PFC. The unstable plant and constraints are:

$$G_1 = \frac{0.2361z^{-6}}{1 - 1.118z^{-1}}; \quad -0.4 \leq u(k) \leq 0.3, \quad -0.1 \leq \Delta u(k) \leq 0.1; \quad 0 \leq y(k) \leq 0.9$$

A disturbance $d_{y,k} = 0.5$ is introduced at the 35th sample and white sensor noise $n_k \epsilon [-0.1, 0.1]$ after the 55th sample; the multiplicative uncertainty is $\delta = 0.5$. The target dynamics are governed by $\rho = 0.75$ and $R = 1$ resulting in $0.4998 < K \leq 1.5587$. We choose the middle value of gain according to Algorithm 1, thus $K = 1.03$ guarantees $0.75 \leq z_{CL} < 0.875$. Note that the output upper limit is intentionally kept below set point to analyse the efficacy of PPFC constraint handling.

4.1 Sensitivity Analysis

A sensitivity analysis is used to select the coincidence horizon h. Figure 4 shows a comparison of sensitivity functions between PPFC and conventional PFC for different h. An worsening trend in sensitivities can be observed for PFC with higher h whereas for PPFC this trend is reversed apart from disturbance rejection that deteriorates slightly. This is expected because with PFC a larger h means the control law is being based on an increasingly large/divergent open-loop prediction and thus is unreliable. The core point is that PPFC clearly outperforms PFC in terms of sensitivity and one can choose h to get some trade off between the different sensitivity functions.

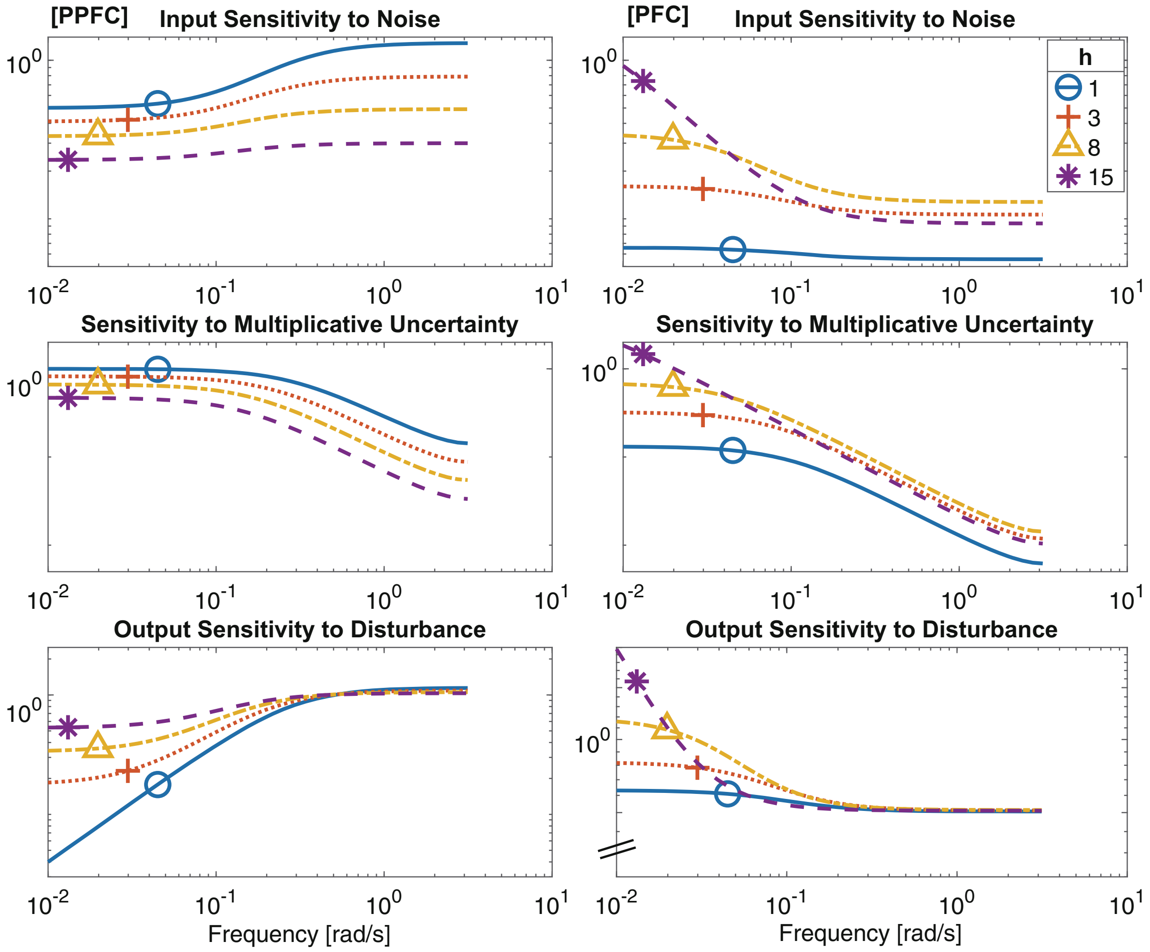


Fig. 4. Comparison of sensitivity plots for G_1 as function of h between PPFC with $K = 1.03$ and conventional PFC (vertical scales for bottom two figures are not equal)

4.2 Closed-Loop Behaviour

The unconstrained time-domain performance shown in Fig. 5 agrees with the results of sensitivity analysis, although nominal performance is affected somewhat by the parameter uncertainty. Again PPFC clearly outperforms PFC and has much more consistent behaviour as h changes. It is particularly notable that PFC begins to fail for large h which is the opposite observation one gets with stable open-loop processes.

When constraints are introduced the advantages of PPFC are even more pronounced as seen in Fig. 6. Notably PPFC performs well notwithstanding the unstable pole and retains feasibility, whereas PFC fails and has an unstable closed-loop.

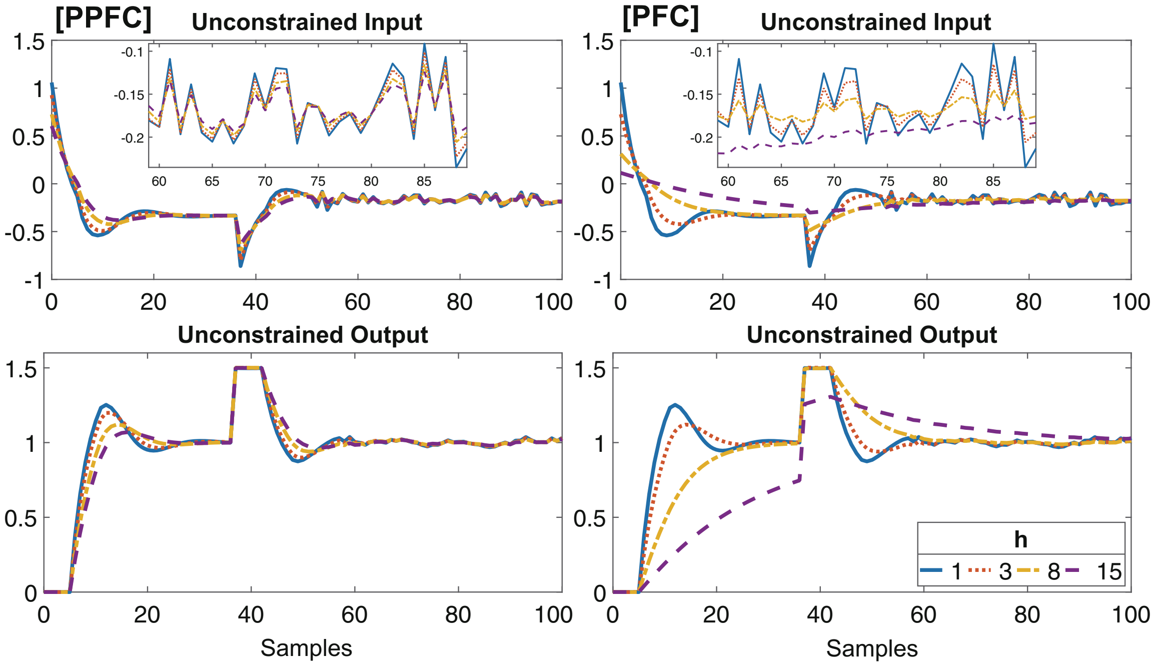


Fig. 5. Comparison of unconstrained input and output for G_1 as function of h between PPFC with $K = 1.03$ and conventional PFC

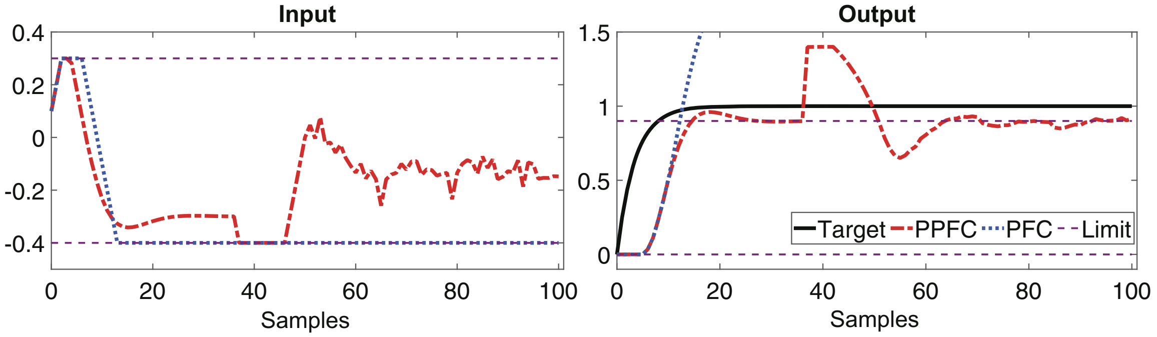


Fig. 6. Comparison of constrained input and output for G_1 between PPFC with $K = 1.03$, $h = 15$ and conventional PFC with $h = 1$

5 Conclusions

This paper proposes a two-stage design approach to controlling unstable first-order plants with PFC. It has been shown that pre-stabilisation with a simple proportional gain improves performance, both with and without constraints. The paper establishes systematic guidelines for selection of both the proportional gain and other tuning parameters and proposes some offline analysis to consider their impact on overall performance. The theoretical aspects of this study have been validated through numerical simulations which demonstrate superior closed-loop control with the proposed scheme.

In future the authors plan to extend this study to more challenging unstable and/or higher-order dynamics. We expect a similar approach to pre-condition oscillatory and non-minimum phase behaviour with PD loops could be exploited which otherwise are difficult to control with conventional PFC alone. It is noted that complex pre-conditioning loops within the PFC framework might involve a slightly more demanding constraint handling procedure, but given modern computing capacity this is not likely to be a problem.

References

1. Richalet, J., O'Donovan, D.: Elementary predictive functional control: a tutorial. In: 2011 International Symposium on Advanced Control of Industrial Processes (ADCONIP), pp. 306–313 (2011)
2. Richalet, J., O'Donovan, D.: Predictive Functional Control: Principles and Industrial Applications. Springer Science and Business Media, London (2009). https://doi.org/10.1007/978-1-84882-493-5
3. Rossiter, J.A.: A First Course in Predictive Control. Second Edition. CRC Press, Boca Raton (2018). https://doi.org/10.1201/9781315104126
4. Rossiter, J., Haber, R.: The effect of coincidence horizon on predictive functional control. Processes **3**(1), 25–45 (2015). https://doi.org/10.3390/pr3010025
5. Rossiter, J.: Input shaping for PFC: how and why? J. Control Decis. **3**(2), 105–118 (2016)
6. Abdullah, M., Rossiter, J.A.: Using Laguerre functions to improve the tuning and performance of predictive functional control. Int. J. Control. **93**(9), 1–13 (2019)
7. Zhang, Z., Rossiter, J.A., Xie, L., Su, H.: Predictive functional control for integral systems. In: 13th International Symposium on Process Systems Engineering (2018)
8. Rossiter, J.A., Kouvaritakis, B., Rice, M.: A numerically robust state-space approach to stable-predictive control strategies. Automatica **34**(1), 65–73 (1998)
9. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.: Constrained model predictive control: stability and optimality. Automatica **36**(6), 789–814 (2000)
10. Abdullah, M., Rossiter, J.A.: The effect of model structure on the noise and disturbance sensitivity of predictive functional control. In: 2018 European Control Conference (ECC). IEEE (2018). https://doi.org/10.23919/ecc.2018.8550374

Appendix B

Predictive Functional Control with Explicit Pre-conditioning for Oscillatory Dynamic Systems

Muhammad Saleheen Aftab, and John Anthony Rossiter

This paper has been published in the Proceedings of 19th European Control Conference (ECC), Netherlands, 2021

Author Contributions. This paper is a collaborative work between both authors. M. S. Aftab proposed the idea, analysed the concept in case studies, and prepared the initial draft of the paper. J .A. Rossiter provided accurate communication of the earlier PFC and MPC control laws, supervised M. S. Aftab and reviewed the whole project.

2021 European Control Conference (ECC)
June 29 - July 2, 2021. Rotterdam, Netherlands

Predictive Functional Control with Explicit Pre-conditioning for Oscillatory Dynamic Systems

Muhammad Saleheen Aftab[1] and John Anthony Rossiter[2]

Abstract— **Predictive functional control (PFC) is a popular industrial process control strategy, but its rather simplistic design renders it less effective in more demanding situations; for example, under-damping, open-loop instability or significant non-minimum phase characteristics have been difficult to control. Devising efficient strategies for such systems remains a topic of interest within the PFC community. This paper shows how a systematic pre-conditioning approach can improve PFC performance for under-damped systems. The proposed pre-conditioning stage is essentially an additional feedback loop whose sole purpose is to provide reliable predictions for PFC decision making. To prevent complicated performance tuning and constraints management procedures, compensator design is kept fairly simple and intuitive. Numerical studies verify the efficacy of the proposal.**

Index Terms—**PFC, coincidence point, under-damping, feedback compensation, pre-conditioning**

I. Introduction

The industrial popularity of predictive functional control (PFC) stems from the simplistic design and development, cost-effective commissioning and maintenance thereafter, and also from the fact that being model-based strategy, it provides better closed-loop control than the obvious alternative of PID, especially in handling large dead-times and constraints [1]. This argument is strongly supported by numerous successful industrial PFC applications [2].

The basic PFC algorithm [2]–[4] matches output predictions with a desired first-order response at only one future point, known as the coincidence point, and with a fixed control action. Intuitively this approach is effective as long as the model behaviour is smooth and monotonically convergent after immediate transients. A prime example is stable first-order plants for which PFC technique is proven to drive the controlled variable to any desirable target trajectory provided "coincidence" occurs exactly one-step ahead [3]. Similar closed-loop performance could be expected with monotonic higher-order dynamics (i.e. dynamics with over-damping), although a coincidence point of one may not suffice due to lag in the predictions [5]. PFC design guidelines for such simple systems are well understood in literature.

What happens when model predictions are oscillatory or, in the worst scenario, completely divergent? Simply put, PFC loses efficacy in such difficult situations [3], [5]. This apparently relates to the fact that constant input within prediction horizon lacks enough flexibility to tackle oscillatory or divergent dynamics and also provide a smooth closed-loop system output [6]. Nevertheless researchers have suggested various modifications in the original PFC to handle difficult dynamics.

One proposal [7] recommends altering the input by separating and subsequently cancelling the un-wanted 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 produces aggressive input activity. Another input shaping proposal [6] 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 some rather less intuitive offline computations. Yet another proposal utilises the partial fraction decomposition of higher-order models [8] into several first-order systems to avail a simple tuning procedure. For oscillatory dynamics [9], however, such a decomposition explicitly embeds complex number algebra into computations limiting its practicality. Suggested modifications (within the same paper) guarantee real number computation but at the price of increased coding requirements.

Designs integrating explicit pre-compensation are fairly common in the mainstream model predictive control literature, whereby one stabilises the unsettled model predictions with some form of feedback compensation [10], [11]. The concept of pre-compensation in PFC, however, is generally restricted to the use of simple proportional gains [12]–[14] to avoid the resultant complex constraints management [4]. Although proportional compensation is usually sufficient for simple systems, challenging dynamics require more involved pre-conditioning strategies. This paper has therefore two major contributions: firstly it proposes an intuitive pre-conditioning technique that relates the compensator parameters to open-loop system dynamics, and secondly guarantees simpler tuning and constraints handling, on par with the standard PFC at best.

The remainder of this paper is organised as follows: Section II defines the problem and sets control objectives. The main methodology is presented in Sections III & IV where the pre-compensator and PFC designs are discussed in detail. Numerical studies follow next in Section V which discuss nominal performance and draw comparisons against standard PFC. Finally the paper concludes in Section VI.

[1],[2]Both authors are associated with the Department of Automatic Control and Systems Engineering, University of Sheffield, Mappin Street, S1 3JD, UK. Email Addresses: [1]`msaftab1@sheffield.ac.uk`, [2]`j.a.rossiter@sheffield.ac.uk`

978-94-6384-236-5 ©2021 EUCA

II. Problem Statement

Consider an n^{th} order stable SISO system

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

where $G(z)$ is strictly proper, $a^{+}(z)$ contains the dominant oscillatory modes $z_a^{+} = p_r \pm jp_i$, and $a^{-}(z)$ represents the remaining poles. The system (1) may also be subject to input (u_k), input-rate (Δu_k) and/or output (y_k) constraints,

$$\begin{aligned} u_{min} \leq & u_k \leq u_{max} \\ \Delta u_{min} \leq & \Delta u_k \leq \Delta u_{max} \\ y_{\min} \leq & y_k \leq y_{\max} \end{aligned} \tag{2}$$

where $\Delta = 1 - z^{-1}$ is the difference operator. The aim is to design a predictive functional controller that operates on pre-conditioned (by using a simple inner feedback loop) model predictions. The pre-compensator is expected to filter out effectively the oscillatory dynamics from (1), while adhering to specified constraints at the same time.

III. Model Pre-conditioning

It is obvious that the constant future input assumption within the PFC framework would fail to damp the oscillatory predictions resulting in relatively poor performance. However one can modify predictions to be smoother with pre-conditioning, and this will enable better-posed PFC decision making. The idea of Pre-conditioned PFC (PPFC) is shown in Fig. 1 where the prediction dynamics (1) are compensated through $C(z)$ via an internal feedback control loop. Next we present the design of $C(z)$ with a pole-placement technique.

A. Simple Pole-Placement Compensator

Assume that feedback compensation of $G(z)$ with $C(z)$, as shown in Fig. 1, results in the transfer function $T(z)$ which provides smooth and monotonically convergent prediction behaviour. Then one may write

$$T(z) = \frac{C(z)G(z)}{1 + C(z)G(z)} = \frac{\beta(z)}{\alpha(z)} \tag{3}$$

After simple manipulations, this leads to

$$C(z) = \frac{\beta(z)a(z)}{b(z)[\alpha(z) - \beta(z)]}$$

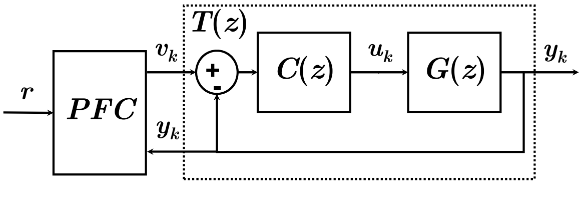


Fig. 1. PPFC structure comprising inner feedback pre-compensation and outer PFC loop.

The open-loop zeros $b(z)$ become compensator poles that could cause stability issues especially with unstable zeros. To avoid this, we set $\beta(z) = Kb(z)$, $K \neq 0$ and get

$$C(z) = K\frac{a(z)}{\alpha(z) - Kb(z)} \tag{4}$$

From (4), it is obvious that the compensator actually *cancels* the open-loop poles $a(z)$ and places new poles given by $\alpha(z)$. Would such a design based on pole-cancellation be acceptable? Let us try to understand the rationale behind pole-placement in the context of PFC.

B. Rationale behind Pole-Cancellation

At this point, readers are reminded of the main design objective, that is to obtain smooth and well-settled model predictions because conventional PFC lacks flexibility to handle oscillatory dynamics efficiently. Traditional PID and lag-lead compensation have been proven ineffective in completely eliminating oscillations especially with higher-order dynamics [15]. The obvious alternative in this case is pole-placement (4).

While it is best to avoid cancelling open-loop unstable poles, researchers report that the decision to either shift or cancel a real stable pole is merely based on design trade-off between either having the shifted pole appear as zero of the sensitivity function to output disturbance, or having the cancelled mode appear as pole of the sensitivity function to input disturbance [16]. A similar argument holds for complex conjugate pole pair, albeit cancellation in this case implies oscillatory input disturbance rejection.

Another concern is related to inexact pole-zero cancellation which is almost always inevitable due to modelling errors. For unstable poles this may lead to output divergence, but for stable open-loop poles the impact depends upon magnitude of the residues from partial fraction expansion and consequently on the design specification for satisfactory performance. Nevertheless, one should not forget that the pole-placement compensator (4) is assisted by an outer PFC loop, which is tuned for performance, robustness and disturbance rejection.

Although a clear understanding would require formal sensitivity analysis, here we demonstrate the performance of a pole-placement technique in combination with PFC, shown in Fig. 2, against a pole-shifting compensator in the presence of output disturbance (constant 0.25 amplitude starting at 25^{th} sample) and input disturbance (constant -0.25 amplitude starting at 60^{th} sample) for G_1 (see Section V-A) with deliberately introduced modelling errors. Evidently pole-placement provides better output disturbance rejection whereas input disturbance rejection is equivalent for both controllers. Moreover, while it may not always be possible to design a pole-shifting controller, the proposed pole-placement compensator always exists for any order dynamic model (1).

From now onwards, we shall focus on the attributes of pole-placement compensator and the design simplicity it brings within the PFC framework.

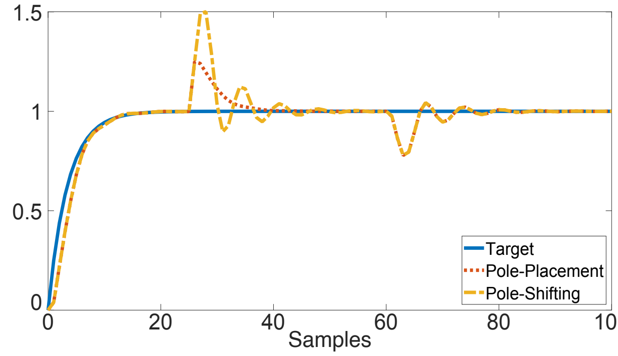


Fig. 2. PFC comparison with pole-cancellation and pole-shifting compensation for G_1 in the presence of input and output disturbances with modelling mismatches

C. Selecting Compensated Dynamics

In order to design a pole-placement compensator, appropriate selection of $\alpha(z)$ is extremely important. Ideally one would want the compensated model $T(z)$ to exhibit non-oscillatory behaviour. A good starting point then is to place the "new" poles of $T(z)$ at the projection of dominant oscillatory poles of $G(z)$ along the real axis. This would effectively filter out the unwanted oscillations without compromising convergence speed [6]. Mathematically

$$T(z) = K\frac{b(z)}{\alpha(z)}; \quad \alpha(z) = a^{-}(z)\alpha^{+}(z) \tag{5}$$

with $\alpha^{+}(z) = (1 - p_r z^{-1})^2$.

It is necessary for internal stability that all poles of $C(z)$ remain inside the unit circle. This can be guaranteed by keeping K within the stable range. However it is quite tedious to obtain an analytical expression for K for higher than second-order systems and therefore we recommend using a graphical tool such as root locus (see, for example [17]) on the denominator of (4) to find the stability margin of $C(z)$.

D. Second-Order Models

Before moving on to the next design stage, it is pertinent to discuss pre-conditioning of under-damped second-order systems exclusively. Assume $a(z) = 1 + a_1 z^{-1} + a_2 z^{-2}$ and $b(z) = b_1 z^{-1} + b_2 z^{-2}$ and also that the oscillatory modes are $z_a = p_r \pm jp_i$. Then according to the preceding discussion $\alpha(z) = 1 - 2p_r z^{-1} + p_r^2 z^{-2}$ is the pole polynomial of the compensated model. Theorem 1 below provides analytical expressions for the stable K range.

Theorem 1: A pre-conditioning compensator designed for second-order model using (4)-(5) is guaranteed stable if:

$$K < \begin{cases} min\left[\dfrac{(1-p_r)^2}{b_1(1-z_z)}, -\dfrac{(1+p_r)^2}{b_1(1+z_z)}\right]; & -\infty < z_z \leq -1 \\ \dfrac{(1-p_r)^2}{b_1(1-z_z)}; & -1 < z_z < p_r \\ \dfrac{1-p_r^2}{|b_1 z_z|}; & z_z > p_r \end{cases}$$

where $z_z = -b_2/b_1$ is the system zero.

Proof: The controller poles are:

$$z_C = \frac{(2p_r + Kb_1) \pm \sqrt{(2p_r + Kb_1)^2 - 4(p_r^2 - Kb_2)}}{2}$$

For convenience, we substitute $x_1 = 0.5(2p_r + Kb_1)$ and $x_2 = 0.5\sqrt{(2p_r + Kb_1)^2 - 4(p_r^2 - Kb_2)}$. Then $z_{C_1} = x_1 - x_2$ and $z_{C_2} = x_1 + x_2$.

(i) if $z_z \leq -1$ then by increasing K, z_{C_1} moves towards z_z whereas z_{C_2} goes towards $+\infty$. For stability we must ensure $z_{C_1} > -1$ and $z_{C_2} < 1$. These two conditions transform into:

$$z_{C_1} > -1 \implies K < -\frac{(1+p_r)^2}{b_1(1+z_z)}$$

$$z_{C_2} < 1 \implies K < \frac{(1-p_r)^2}{b_1(1-z_z)}$$

Depending on the actual position of p_r, one of the controller poles is relatively nearer to the stability boundary. Therefore:

$$K < min\left[\frac{(1-p_r)^2}{b_1(1-z_z)}, -\frac{(1+p_r)^2}{b_1(1+z_z)}\right]; -\infty < z_z \leq -1 \tag{6}$$

(ii) for $-1 < z_z < p_r$, z_{C_1} can never leave the unit circle. Therefore it is sufficient to check only:

$$K < \frac{(1-p_r)^2}{b_1(1-z_z)}; \quad -1 < z_z < p_r \tag{7}$$

(iii) $z_z > p_r$ results in complex conjugate controller poles. In this case, guaranteed stability $|z_C| < 1$ implies:

$$K < \frac{1-p_r^2}{|b_1 z_z|}; \quad z_z > p_r \tag{8}$$

which completes the proof. ■

Remark 1: For some systems one or more zeros might be located at p_r. To prevent inadvertent pole-zero cancellation in such cases, we suggest replacing p_r in the preceding analysis with $p_r + \epsilon$, where $\epsilon \to 0$ and $|p_r + \epsilon| < 1$.

IV. Nominal PFC Design

The design stage implements the pre-conditioned model predictions within a PFC framework, as shown in Fig. 1.

A. The PPFC Control Law

Similar to conventional PFC, the Pre-conditioned PFC (PPFC) drives the output prediction $y_{k+i|k}$ exponentially closer to the set point r with each time step. This convergence mainly depends upon the target pole ρ defined by $\rho = exp(-3T/CLTR)$ where T and $CLTR$ are sampling time and desired closed-loop settling time respectively. Mathematically, the PFC law is derived from the target:

$$y_{k+i|k} = r - (r - y_k)\rho^i; \quad i > 0 \tag{9}$$

On the other hand, eqn. (5) i.e. $\alpha(z)y(z) = Kb(z)v(z)$ provides i-step ahead prediction information as follows:

$$y_{k+i|k} = KH_i \underrightarrow{v}_k + KP_i \underleftarrow{v}_{k-1} + Q_i \underleftarrow{y}_k; \quad i > 0 \tag{10}$$

where H_i, P_i and Q_i depend upon model parameters. For an N^{th} order model:

$$\underrightarrow{v}_k = \begin{bmatrix} v_k \\ v_{k+1} \\ \vdots \\ v_{k+i-1} \end{bmatrix} ; \underleftarrow{v}_{k-1} = \begin{bmatrix} v_{k-1} \\ v_{k-2} \\ \vdots \\ v_{k-N+1} \end{bmatrix} ; \underleftarrow{y}_k = \begin{bmatrix} y_k \\ y_{k-1} \\ \vdots \\ y_{k-N+1} \end{bmatrix}$$

By keeping a constant future input i.e. $v_{k+i} = v_k, \forall i > 0$, the i-step ahead model prediction (10) is matched to the target (9) at one future point called the coincidence point n_y i.e. at $i = n_y$. This results in the PPFC law

$$v_k = \frac{r - (r - y_k)\rho^{n_y} - (KP_{n_y}\underleftarrow{v}_{k-1} + Q_{n_y}\underleftarrow{y}_k)}{Kh} \qquad (11)$$

where $h = \sum_{j=1}^{n_y} H_{n_y}^j$ and $H_{n_y}^j$ is the j^{th} element of H_{n_y}. Naturally the main interest is in finding actual input u_k that drives the plant. It is evident from Fig. 1

$$u(z) = C(z)[v(z) - y(z)] \qquad (12)$$

Lemma 1: Formulation (12) is equivalent to:

$$u(z) = K\frac{a^+(z)}{\alpha^+(z)}v(z)$$

Proof: Substituting $C(z) = Ka(z)/[\alpha(z) - Kb(z)]$ and $y(z) = [b(z)/a(z)]u(z)$ in (12), we get:

$$u(z) = K\frac{a(z)}{\alpha(z)}v(z)$$

further $\alpha(z) = a^-(z)\alpha^+(z)$ and $a(z) = a^-(z)a^+(z)$ imply:

$$u(z) = K\frac{a^+(z)}{\alpha^+(z)}v(z) \qquad (13)$$

which is a simple equivalent of (12). ■

Extracting either u_k or v_k from the other simply requires vector/matrix multiplication as shown below. Hence the associated coding requirement is elementary.

$$\begin{aligned} u_k &= K\hat{a}^+\underleftarrow{v}_k - \hat{\alpha}^+\underleftarrow{u}_{k-1} \\ v_k &= \frac{\hat{\alpha}^+}{K}\underleftarrow{u}_k - \hat{a}^+\underleftarrow{v}_{k-1} \end{aligned} \qquad (14)$$

where vectors $\hat{a}^+$ and $\hat{\alpha}^+$ contain coefficients of polynomials $a^+(z)$ and $\alpha^+(z)$ respectively.

Remark 2: Subtleties related to prediction-bias removal and offset-free tracking have been omitted from (11) as these do not affect the main analysis and results. Numerical examples nonetheless include relevant algebra. Readers are referred to [3] for details.

B. Tuning Procedure

Though PFC tuning has traditionally been heuristic, researchers have managed to establish some generic guidelines for simpler systems [7], [18]. Pre-compensation effectively changes the oscillatory open-loop step response of $G(z)$ to smoother and more settled behaviour, and therefore standard PFC tuning procedures can be fully utilised. In this study, the PPFC control law (11) depends upon three significant parameters: the coincidence horizon n_y, the target pole ρ and the compensator gain K. While a judicious selection of n_y and ρ is of paramount importance, surprisingly K does not affect the closed-loop performance.

Theorem 2: The plant input u_k is independent of compensator gain K.

Proof: In z-domain, eqn. (11) can be written as:

$$v(z) = \frac{(1 - \rho^{n_y})r(z) + [\rho^{n_y} - Q(z)]y(z)}{K[h + P(z)]}$$

where $P(z) = \sum_{j=1}^{N-1} P_{n_y}^j z^{-j}$ and $Q(z) = \sum_{j=0}^{N-1} Q_{n_y}^j z^{-j}$. It follows from Lemma 1:

$$u(z) = \cancel{K}\frac{a^+(z)}{\alpha^+(z)} \cdot \frac{(1 - \rho^{n_y})r(z) + [\rho^{n_y} - Q(z)]y(z)}{\cancel{K}[h + P(z)]}$$

or equivalently in the time-domain:

$$u_k = \frac{\sum_{j=0}^{2} a_j^+(1 - \rho^{n_y})r - (\tilde{P}_{n_y}\underleftarrow{u}_{k-1} + \tilde{Q}_{n_y}\underleftarrow{y}_k)}{h} \qquad (15)$$

for suitable $\tilde{P}_{n_y}$ and $\tilde{Q}_{n_y}$. Hence the plant input is independent of K. Nevertheless, appropriate selection of K is still necessary to maintain internal stability. ■

The recommended tuning procedure [5] suggests choosing n_y within the range $k_L \leq n_y \leq k_U$, where k_L and k_U represent the time samples when the normalised unit-step response of $T(z)$ reaches approximately 0.4 and 0.8 respectively with significant gradient. As for the target pole, one may compare several first-order responses with differing ρ against the normalised step-response to find an intercept within the above-mentioned n_y range. See, for instance, Figs. 3 and 5.

C. Constraint Management

Knowledge of u_k from (14) can facilitate constraint handling in fairly straightforward manner. Instead of adopting computation-intensive algorithms, such as back-calculation [4], one may opt for input and output predictions for constraint assessment. The only caveat, however, is the need to recalculate corrected v_k, using (14), if input violations are detected. Furthermore, output constraints must be validated before inputs, since y_k is based on the pre-conditioned model predictions and hence depend upon v_k. One may use prediction equation (10) by selecting such value of v_k closest to the one obtained with (11) that satisfies all constraints for sufficiently large validation horizon n_c i.e.

$$y_{min} \leq KH_j v_k + KP_j\underleftarrow{v}_{k-1} + Q_j\underleftarrow{y}_k \leq y_{max} \qquad (16)$$

where $j = 1, 2, \ldots, n_c$ and $n_c \gg n_y$.

Remark 3: Constraint handling with the pre-conditioned model predictions (16) is guaranteed recursively feasible as long as n_c is sufficiently large [19]. This, however, may not be true with open-loop oscillatory predictions.

V. Numerical Examples

This section demonstrates the efficacy of PPFC algorithm with two numerical examples. To better understand its advantages, a comparison of closed-loop performance is drawn against the standard PFC for two challenging processes: G_1

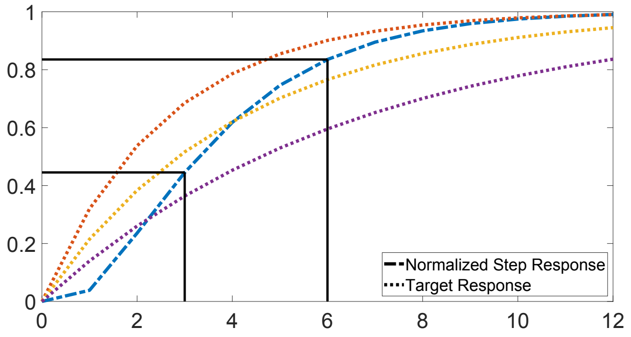


Fig. 3. Target responses with $\rho = [0.6(red), 0.75(yellow), 0.9(purple)]$ overlaying the normalised step response of T_1

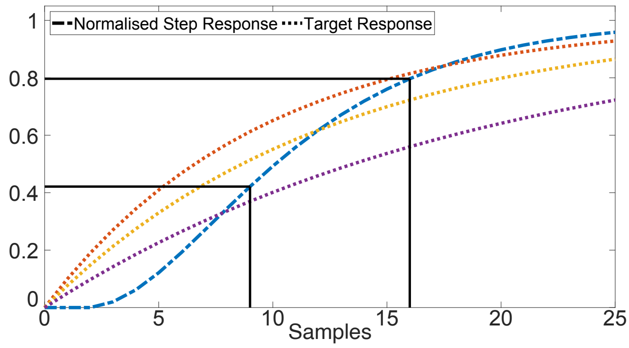


Fig. 5. Targets with $\rho = [0.9(red), 0.92(yellow), 0.95(purple)]$ overlaying the normalised step response of T_2

is a second-order oscillatory system whereas G_2 is slightly non-minimum phase third-order under-damped system.

A. Example-1 (Second-Order Model)

Consider a second-order under-damped system (17) with $|\Delta u_k| \leq 0.45$, $-0.25 \leq u_k \leq 1.75$ and $0 \leq y_k \leq 1.05$. For fair comparison, both PPFC and PFC control laws will be tuned identically.

$$G_1 = \frac{0.1z^{-1} + 0.4z^{-2}}{1 - z^{-1} + 0.8z^{-2}} \tag{17}$$

The oscillatory modes of G_1 are $z_a = 0.5 \pm j0.742$, whereas $z_z = -4$ is the system zero. For stability the compensator gain should be $K < 0.5$ as obtained from (6). Consequently the pre-conditioned prediction model with $K = 0.25$ is:

$$T_1 = \frac{0.025z^{-1} + 0.1z^{-2}}{1 - z^{-1} + 0.25z^{-2}} \tag{18}$$

Next we find n_y and ρ. Fig. 3 shows the pre-conditioned step response overlaid with various first-order target responses and suggests $3 \leq n_y \leq 5$ as a suitable coincidence horizon window. Evidently target dynamics with $\rho = 0.6$ or $\rho = 0.9$ do not match predictions within the desirable n_y range and hence would need over-actuation or under-actuation to enforce an intercept. However, a sensible choice would be $\rho = 0.75$ which gets an exact match at $n_y = 4$.

Efficacy of the PPFC algorithm is obvious with the constrained nominal closed-loop performance shown in Fig. 4. Specifically we observe that the PPFC plant output (upper

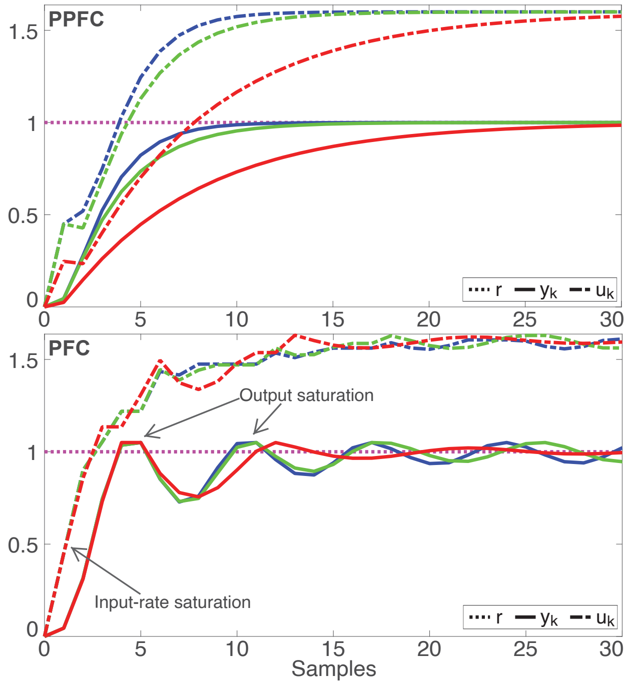


Fig. 4. Nominal constrained performance with G_1, $n_y = 4$ and $\rho = [0.6(blue), 0.75(green), 0.9(red)]$

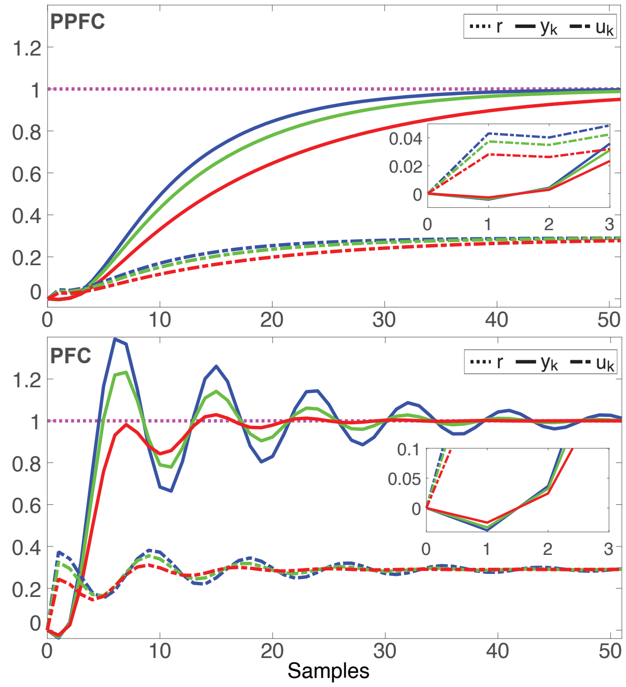


Fig. 6. Nominal unconstrained performance with G_2, $n_y = 12$ and $\rho = [0.9(blue), 0.92(green), 0.95(red)]$

figure) is smooth and oscillation-free, and strongly linked to the corresponding ρ. Additionally no constraint violation is visible, although control input for faster target dynamics is relatively aggressive which reinforces the expectation. The conventional PFC (lower figure) appears incapable within transient period. There seems no obvious link between the output and target dynamics (ρ), and inputs are generally too aggressive. As a result, input-rate and output saturation are seen within first 10 samples.

B. Example-2 (Higher-Order Model)

Consider the model G_2 with *no constraints*

$$G_2 = \frac{-0.1z^{-1} + 0.4z^{-2} + 0.2z^{-3}}{1 - 2.1z^{-2} + 1.69z^{-2} - 0.445z^{-3}} \tag{19}$$

with $a^+(z) = 1 - 1.6z^{-1} + 0.89z^{-2}$ and $a^-(z) = 1 - 0.5z^{-1}$. The oscillatory modes are located at $z_{a^+} = 0.8 \pm j0.5$. Therefore, we define $\alpha^+(z) = 1 - 1.6z^{-1} + 0.64z^{-2}$, $\alpha^-(z) = a^-(z)$ and get the compensated model:

$$T_2 = K\frac{-0.1z^{-1} + 0.4z^{-2} + 0.2z^{-3}}{1 - 2.1z^{-2} + 1.44z^{-2} - 0.32z^{-3}} \tag{20}$$

The root locus plot (not shown) suggests $K < 0.04$ for compensator stability. We choose $K = 0.02$ as its numerical value does not affect the closed-loop performance. Fig. 5 shows the normalised step response of T_2, which clearly recommends a coincidence point within $9 \leq n_y \leq 16$. An intercept between step response and target exists with $\rho = 0.92$ at $n_y = 12$. To enforce an intercept with faster targets, one would need to over-actuate the input which in practice may result in constraint violations.

Nevertheless we examine the closed-loop behaviour with $\rho = [0.9, 0.92, 0.95]$ and $n_y = 12$ in Fig. 6, which presents a rather contrasting display of performance between the PPFC and simple PFC algorithms. Key observations are: Simple PFC with constant future input fails to damp the oscillatory modes of G_2. With the proposed scheme, while v_k remains constant within the coincidence horizon, the plant input u_k is aptly parametrised to overcome the under-damping. This difference is obvious from the varied control dynamics for both controllers. With PPFC, the target pole ρ seems more effective. This is clearly evident in the amount of time the outputs take to settle for both the algorithms. To sum up, while the conventional PFC algorithm may not be suitable for oscillatory dynamics, it is possible to improve its capabilities via pre-conditioning in straightforward fashion.

VI. Conclusion

This paper has proposed a pre-conditioned PFC design methodology for under-damped dynamic systems. The proposed pre-conditioning stage, based on pole-placement, transforms the unsettled open-loop model dynamics into smoother prediction behaviour, known to work well with the standard PFC algorithm. The overall design process is fairly straightforward; one that does not complicate standard constraint handling and controller tuning, and also retains the key attributes of original PFC i.e. simplicity and intuitiveness. We have demonstrated that the PPFC algorithm in essence parametrises control action to efficiently handle oscillations, a quality not present in the conventional PFC. Moreover the main tuning parameter, the target pole, shows improved efficacy in relation to the closed-loop performance as seen in the numerical simulations.

Nevertheless, as discussed in Section III-B, the improved set-point tracking with pole-placement pre-conditioning comes at the price of rather oscillatory input disturbance rejection, but a relatively better response to output disturbance compared to an equivalent pole-shifting compensator balances out this shortcoming. A formal analysis of sensitivity functions is required to fully understand the pros and cons of pole-placement and this constitutes our future research work.

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] 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.

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

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

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

[6] 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.

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

[8] J. Rossiter, R. Haber, and K. Zabet, “Pole-placement predictive functional control for over-damped systems with real poles,” *ISA transactions*, vol. 61, pp. 229–239, 2016.

[9] K. Zabet, J. Rossiter, R. Haber, and M. Abdullah, “Pole-placement predictive functional control for under-damped systems with real numbers algebra,” *ISA transactions*, vol. 71, pp. 403–414, 2017.

[10] 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.

[11] 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.

[12] 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.

[13] Z. Zhang, J. Rossiter, L. Xie, and H. Su, “Predictive functional control for integral systems,,” in *International Symposium on Process System Engineering*, 2018.

[14] M. Abdullah and J. A. Rossiter, “Alternative method for predictive functional control to handle an integrating process,” in *2018 UKACC 12th International Conference on Control (CONTROL)*. IEEE, 2018, pp. 26–31.

[15] K. J. Åström and T. Hägglund, *PID controllers: theory, design, and tuning*. Instrument society of America Research Triangle Park, NC, 1995.

[16] R. H. Middleton and S. F. Graebe, “Slow stable open-loop poles: to cancel or not to cancel,” *Automatica*, vol. 35, no. 5, pp. 877–886, 1999.

[17] B. C. Kuo, *Digital Control Systems*. Oxford University Press Inc, 1995.

[18] J. A. Rossiter, “A priori stability results for pfc,” *International journal of control*, vol. 90, no. 2, pp. 289–297, 2017.

[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.

Appendix C

Pre-stabilised Predictive Functional Control for Open-loop Unstable Dynamic Systems

Muhammad Saleheen Aftab, and John Anthony Rossiter

This paper has been published in the IFAC PapersOnLine Vol. 54, No. 6, 2021

Author Contributions. This paper is a collaborative work between both authors. M. S. Aftab proposed the idea, analysed the concept in case studies, and prepared the initial draft of the paper. J .A. Rossiter provided accurate communication of the earlier PFC and MPC control laws, supervised M. S. Aftab and reviewed the whole project.

Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online
CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 54-6 (2021) 147–152

Pre-stabilised Predictive Functional Control for Open-loop Unstable Dynamic Systems

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

* *Department of Automatic Control and Systems Engineering,
University of Sheffield, Mappin Street, S1 3JD, UK.
(e-mail: msaftab1@sheffield.ac.uk, j.a.rossiter@sheffield.ac.uk)*

Abstract: Predictive functional control (PFC) is the simplest model-based algorithm, equipped with the attributes of a fully fledged predictive controller but at the cost and complexity threshold of a standard PID regulator. It has proven benefits in controlling stable SISO dynamic systems, but similarly to its competitor PID, it loses efficacy when a challenging application is introduced. In this paper, we present a modified PFC approach, especially tailored for open-loop unstable processes, using pre-stabilisation to efficiently control the undesirable dynamics at hand. This is essentially a two-stage design scheme with implications for PFC tuning and constraint handling. The proposal, nevertheless, is straightforward and intuitive, and provides improved closed-loop control in the presence of external perturbations against the standard PFC, and significantly better performance overall compared to the common PID algorithm, as demonstrated in a numerical case-study.

Keywords: PFC; coincidence point; feedback compensation; pre-stabilisation.

1. INTRODUCTION

Model Predictive Control (MPC) is an advanced optimal control strategy with powerful and well-defined procedures for complex multivariate processes (Mayne, 2014). Nevertheless, its computation-heavy nature has traditionally favoured applications with slow dynamics, although the increasing availability of cheap computing resources has significantly widened its scope and utility in recent years (Fernandez-Camacho and Bordons-Alba, 1995; Qin and Badgwell, 2003). But there are areas and applications, for example industrial servo loops, where such an implementation would be logically and financially infeasible and where a cost-effective approach like PID still makes more sense. However, there are scenarios when PID falls short, for instance processes with significant dead-time or tight physical constraints; such cases require additional complexity such as Smith predictors (Skogestad, 2018) and anti-windup algorithms (Visioli, 2006) for improvement. Nonetheless, these solutions are generally ad hoc and, more often than not, degrade other performance attributes; poor robustness to uncertainties is one prominent side-effect of such post-design alterations.

Clearly, there is a need for a systematic yet simpler and cost-effective algorithm, and over the years Predictive Functional Control (PFC) has proved its efficacy as a viable alternative (Richalet et al., 1978). PFC belongs to the family of model-based predictive controllers, and exhibits similar characteristics. As a result, process dead-times and physical constraints are easily integrated in the design, with some degree of robustness owing to the use of a receding horizon (Rossiter, 2018). The main difference, however, arises from the parametrisation of the manipulated variable, which in the case of PFC, is pre-defined as the linear combination of simple polynomial basis functions (Maciejowski, 2002). The optimisation process is further simplified by noting that constant set-point tracking is achievable with constant control moves within the prediction horizon (Khadir and Ringwood, 2008). Although, unlike mainstream MPC, PFC's heuristic nature merely provides a sub-optimal solution, its simplistic design traits have attracted a wide spectrum of applications (Richalet and O'Donovan, 2011; Richalet and O'Donovan, 2009; Richalet, 1993).

Arguably the unique selling point of PFC is its simplicity. Nevertheless, it lacks flexibility to tackle challenging dynamics. For example, open-loop instability, where unreliable predictions cause ill-posed decision-making (Rossiter and Haber, 2015), has been difficult to control with PFC. Previous studies in this area (Rossiter, 2016; Abdullah and Rossiter, 2018a) have proposed algorithmic level modification by shaping the control input, that although they can improve the closed-loop performance, do so with increased computational complexity, thus negating the core notion of simplicity. Pre-stabilisation of dynamics (Rossiter et al., 1998; Mayne et al., 2000) is fairly common in the MPC literature as a means to modify dynamic behaviour of a difficult system to ensure reliable control performance. Surprisingly however, this concept is still relatively unexplored in PFC and largely restricted to first-order (Aftab et al., 2021) and integrator dynamics (Zhang et al., 2018; Abdullah and Rossiter, 2018b). Researchers, in this context, argue that complex pre-stabilisation may also complicate constraint handling (Rossiter, 2018; Richalet and O'Donovan, 2009) which, despite being sub-optimal, is fairly intuitive and powerful in the PFC formulation.

10.1016/j.ifacol.2021.08.537

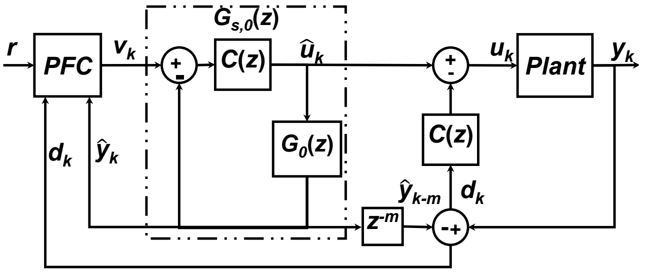


Fig. 1. Proposed PPFC Control Architecture

In this paper, we extend the idea of pre-stabilisation, presented in (Aftab et al., 2021), to higher-order unstable dynamics using the analytical method of internal feedback loop design (Ogata, 1995). It has been found that although the additional control layer burdens the constraint management to some extent, it provides better control of tuning parameters and an impressive overall closed-loop performance, verified by an industrial case study. The remainder of this paper is organised as follows: Section 2 defines the problem and sets control objectives. The main methodology is presented in Section 3 where the Pre-stabilised PFC design is discussed in detail. Next, implications of pre-stabilisation are presented in Section 4. A simulation case study follows next in Section 5 which presents performance comparisons with standard PFC and PI. Finally, the paper concludes in Section 6.

2. PROBLEM STATEMENT

Consider an n^{th}-order coprime delay-free system model

$$G_0(z) = \frac{b(z)}{a(z)} = \frac{b_1 z^{-1} + b_2 z^{-2} + \cdots + b_n z^{-n}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}} \qquad (1)$$

Moreover, $a(z) = a^-(z)a^+(z)$ where the factor $a^+(z)$ contains the p_u open-loop unstable poles. The complete model including time-delay of m samples is $G(z) = z^{-m}G_0(z)$. Moreover, the dynamic plant is subject to actuation limits i.e.

$$\begin{aligned} u_{min} \leq & u_k \leq u_{max} \\ \Delta u_{min} \leq & \Delta u_k \leq \Delta u_{max} \end{aligned} \qquad (2)$$

where $\Delta = 1 - z^{-1}$. The aim is to design a predictive functional controller that operates on pre-stabilised model predictions via internal feedback compensation. The Pre-stabilised PFC (PPFC) is, therefore, expected to perform in the presence of disturbances and modelling uncertainty while adhering to constraints (2).

3. PRE-STABILISED PFC

The fundamental idea behind PPFC, as shown in Fig. 1, is to first stabilise the unstable 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 constraint management. The following two sub-sections explain the proposed design procedure.

3.1 Design of Pre-stabilising Loop

The proposal is based on the analytical approach of feedback compensator design presented in (Ogata, 1995). Assume that a $(n-1)^{th}$-order bi-proper feedback compensator $C(z) = q(z)/p(z)$ is used to modify the open-loop model $G_0(z)$, as shown in Fig. 1, resulting in the pre-stabilised transfer function $G_{s,0}(z)$, with a smooth and monotonically convergent prediction behaviour. Then one may write:

$$G_{s,0}(z) = \frac{\beta(z)}{\alpha(z)} = \frac{q(z)b(z)}{p(z)a(z) + q(z)b(z)} \qquad (3)$$

where $\alpha(z)$ is the $(2n-1)^{th}$-order pre-stabilised pole polynomial, and the underlying relationship,

$$p(z)a(z) + q(z)b(z) = \alpha(z) \qquad (4)$$

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 utilise linear algebra to obtain the coefficients of $p(z)$ and $q(z)$ with,

$$\mathbf{M} = \mathbf{S}^{-1}\mathbf{D} \qquad (5)$$

where $\mathbf{M} = [p_{n-1} \dots p_0 \quad q_{n-1} \dots q_0]^T$, $\mathbf{D} = [\alpha_{2n-1} \dots \alpha_0]^T$ and $\mathbf{S}$ is the *Sylvester Matrix* (Goodwin, 2001) given by:

$$\mathbf{S} = \begin{bmatrix} a_n & 0 & \dots & 0 & b_n & 0 & \dots & 0 \\ a_{n-1} & a_n & \dots & 0 & b_{n-1} & b_n & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & a_1 & \dots & a_{n-1} & 0 & b_1 & \dots & b_{n-1} \\ 0 & 1 & \dots & a_{n-2} & 0 & 0 & \dots & b_{n-2} \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & a_1 & 0 & 0 & \dots & b_1 \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \end{bmatrix} \qquad (6)$$

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

$$\alpha(z) = o(z)a^-(z)\alpha^+(z) \qquad (7)$$

where $o(z)$ is the $(n-1)^{th}$-order observer and $\alpha^+(z)$ represents the p_u pre-stabilised poles. We propose if $a^+(z) = \prod_{i=1}^{p_u}(z - z_{p,i})$ with $z_{p,i} > 1$, then $\alpha^+(z) = \prod_{i=1}^{p_u}(z - 1/z_{p,i})$. In case an integrator factor $(z-1)$ is present, then one may simply replace it with $(z - 0.5)$ (Abdullah and Rossiter, 2018a). Moreover, the minimum order observer is generally selected as $o(z) = z^{n-1}$ (Ogata, 1995). This completes the internal feedback loop design.

Remark 1. For $n = 1$, the compensator reduces to simple proportional gain, i.e. $C(z) = K$. The Pre-stabilised PFC design for first-order unstable systems using proportional compensation has been investigated more extensively (Aftab et al., 2021) so will not be pursued here.

3.2 Pre-stabilised PFC Control Law

The PPFC algorithm works similarly to the original PFC but implemented on the pre-stabilised model dynamics. At each time step, the predicted output y_k is matched to the pre-defined target behaviour at only one coincidence point n_y steps ahead with constant control moves. The process is repeated at the next sample and owing to the receding horizon, an implied feedback is established that moves the plant output closer to the target. The desired behaviour is generally represented as a first-order pole ρ. The ideal n_y-step ahead prediction based on a first-order response is given as:

$$y_{k+n_y+m|k} = r - (r - E[y_{k+m|k}])\rho^{n_y} \qquad (8)$$

where r is the set-point and $E[y_{k+m|k}]$ is the expected m-sample delayed plant output (Rossiter, 2018). Furthermore, $E[y_{k+m|k}] = \hat{y}_k + d_k$ with $d_k = y_k - \hat{y}_{k-m}$, where

d_k accounts for prediction bias due to modelling errors and/or disturbances and $\hat{y}_k$ is the model output. On the other hand, one may obtain the output predictions from $G_{s,0}(z)$ i.e. $\alpha(z)\hat{y}(z) = \beta(z)v(z)$:

$$\hat{y}_{k+n_y|k} = \mathbf{H}\underset{\rightarrow}{\mathbf{v}}_k + \mathbf{P}\underset{\leftarrow}{\mathbf{v}}_{k-1} + \mathbf{Q}\underset{\leftarrow}{\hat{\mathbf{y}}}_k \tag{9}$$

where $\mathbf{H}$, $\mathbf{P}$ and $\mathbf{Q}$ depend on the model parameters. For a generic N^{th} order model:

$$\underset{\rightarrow}{\mathbf{v}}_k = \begin{bmatrix} v_k \\ v_{k+1} \\ \vdots \\ v_{k+n_y-1} \end{bmatrix}; \underset{\leftarrow}{\mathbf{v}}_{k-1} = \begin{bmatrix} v_{k-1} \\ v_{k-2} \\ \vdots \\ v_{k-N+1} \end{bmatrix}; \underset{\leftarrow}{\hat{\mathbf{y}}}_k = \begin{bmatrix} \hat{y}_k \\ \hat{y}_{k-1} \\ \vdots \\ \hat{y}_{k-N+1} \end{bmatrix}$$

The delay-free prediction model $G_{s,0}(z)$ essentially provides m-step ahead estimate of the plant output, which implies:

$$y_{k+n_y+m|k} = \hat{y}_{k+n_y|k} + d_k \tag{10}$$

The decision variable remains constant throughout the horizon i.e. $v_{k+i} = v_k$, $\forall i > 0$, which results in the following PPFC control law:

$$v_k = \frac{r - (r - E[y_{k+m|k}])\rho^{n_y} - (\mathbf{P}\underset{\leftarrow}{\mathbf{v}}_{k-1} + \mathbf{Q}\underset{\leftarrow}{\hat{\mathbf{y}}}_k + d_k)}{h} \tag{11}$$

where $h = \sum_{j=1}^{n_y} H_j$ and H_j is the j^{th} element of $\mathbf{H}$. Next, we will discuss the impact of pre-stabilisation on some key aspects of the proposed PFC approach.

4. IMPLICATIONS OF PRE-STABILISATION

Clearly pre-stabilisation has transformed the decision variable from u_k to v_k, which has significant implications for parameter tuning and constraint handling. Since u_k drives the physical process and the fact that internal compensation is not hard-wired, a direct relationship between both variables must be established for control implementation. Details follow next.

4.1 Relationship between u_k and v_k

If the compensator $C(z)$ were hard-wired, we would have got $u_k = C(z)[v_k - y_{k+m}]$, where y_{k+m} shows the delayed response due to u_k at the current sample. Obviously y_{k+m} is unknown being a future value, but can be replaced with its expected value $E[y_{k+m|k}] = \hat{y}_k + d_k$. Thus,

$$u_k = C(z)[v_k - (\hat{y}_k + d_k)] \tag{12}$$

Furthermore, a similar expression can be written for the pre-stabilised model:

$$\hat{u}_k = C(z)[v_k - \hat{y}_k] \tag{13}$$

Thus after subtracting (13) from (12), the relationship between u_k and $\hat{u}_k$ is established:

$$u_k = \hat{u}_k - C(z)d_k \tag{14}$$

Finding $\hat{u}_k$ is an additional step and adds slightly to the coding complexity. Nevertheless, it is directly related to v_k as shown in the following theorem.

Theorem 1. The control variable $\hat{u}_k$ can be obtained from decision variable v_k using the following expression:

$$\hat{u}_k = \frac{q(z)}{o(z)} \cdot \frac{a^+(z)}{\alpha^+(z)} v_k \tag{15}$$

Proof. Since $\hat{y}_k = G_0(z)\hat{u}_k = G_{s,0}(z)v_k$, eliminating $\hat{y}_k$ results in:

$$\frac{b(z)}{a^-(z)a^+(z)}\hat{u}_k = \frac{q(z)}{o(z)} \cdot \frac{b(z)}{a^-(z)\alpha^+(z)} v_k$$

which simplifies to (15). □

Thus one may replace $\hat{u}_k$ from (15) in (14) to directly evaluate u_k from v_k and vice versa,

$$A(z)u_k = B(z)v_k + E(z)d_k \tag{16}$$

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

$$\begin{aligned} A(z) &= o(z)\alpha^+(z)p(z) = 1 + A_1 z^{-1} + \cdots + A_l z^{-l} \\ B(z) &= q(z)a^+(z)p(z) = B_0 + B_1 z^{-1} + \cdots + B_l z^{-l} \\ E(z) &= -o(z)\alpha^+(z)q(z) = E_0 + E_1 z^{-1} + \cdots + E_l z^{-l} \end{aligned} \tag{17}$$

where $l = p_u + 2n - 2$. Finally,

$$u_k = B_0 v_k + f_k \tag{18}$$

where $f_k = -\mathbf{A}\underset{\leftarrow}{\mathbf{u}}_{k-1} + \mathbf{B}\underset{\leftarrow}{\mathbf{v}}_{k-1} + \mathbf{E}\underset{\leftarrow}{\mathbf{d}}_k$ and the vectors $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{E}$ contain appropriate coefficients of the respective polynomials.

4.2 Controller Tuning

The standard procedure of PFC parameter tuning is based on the conjecture presented in (Rossiter and Haber, 2015), which is mainly applicable to stable dynamics with monotonic steady-state convergence. The recommendation is to select the coincidence horizon n_y preferably within the time window corresponding to 40%-80% rise in the step response with significant gradient. As for finding ρ, one may overlay several first-order responses on the step response to identify which target behaviour coincides within the mentioned n_y range. Evidently this method will not work well with an unstable process, for which a constant input would inevitably lead to divergent output predictions. Parameter selection for such systems is far less consistent and mostly ineffective with no concrete guidelines (Rossiter and Haber, 2015; Rossiter, 2018). Clearly pre-stabilisation makes intuitive sense here, since controller tuning with the stable $G_{s,0}(z)$ in this case will be far more meaningful than the originally unstable process. See, for instance, Fig. 3 that displays PFC parameter selection for a modified system based on the aforementioned procedure.

4.3 Constraint Handling

The standard constraint handling mechanism generally implements simple saturation of the decision variable, which is fairly straightforward with a constant control formulation. Nevertheless, the additional feedback loop in the Pre-stabilised PFC re-parametrises the degree-of-freedom such that u_k no longer remains constant. One possible solution in this case is to transfer the original constraints to the new variable v_k at every sample using a process of back calculation (Richalet and O'Donovan, 2009). Clearly back calculation is computationally intensive; this may work easily with simple feedback designs, for example see (Aftab et al., 2021; Zhang et al., 2020), but with more involved controllers, such as the one in this study, it complicates the validation process. A more efficient approach, however, is to implement constraints on (18) directly, with v_k updated

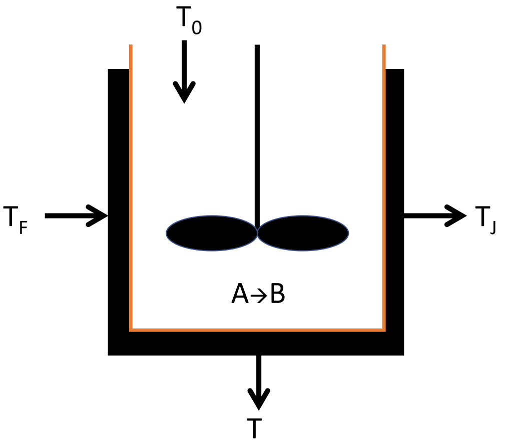


Fig. 2. The Jacketed CSTR Process

only if violation occurs. Each row of the following vector inequalities corresponds to the $(k+i)^{th}$ predicted input (Rossiter, 2018):

$$\mathbf{L}u_{min} \le \underrightarrow{\mathbf{u}}_k \le \mathbf{L}u_{max}$$
$$\mathbf{L}\Delta u_{min} \le \Delta \underrightarrow{\mathbf{u}}_k \le \mathbf{L}\Delta u_{max} \qquad (19)$$

where $i = 0, 1, \ldots, n_c$ and $\mathbf{L} = [1 \quad 1 \ldots]^T$. The validation window n_c (i.e. the length of above inequalities) must extend well beyond the point of coincidence to observe and validate long range adherence. This is crucial because any unobserved input violation could eventually lead to infeasibility, thus invalidating the current input computation. Ideally, n_c should cover the settling period of $G_{s,0}(z)$, i.e. the time to reach and stay within 95% of the implied steady-state, which roughly corresponds to three to five times n_y.

Remark 2. Constraint handling with Pre-stabilised PFC is recursively feasible as long as n_c is sufficiently large (Abdullah and Rossiter, 2018a). This, however, may not be true with open-loop unstable dynamics, for which, in truth, rigorous generic recursive feasibility properties require computations which might be considered beyond the *price range* of PFC.

5. INDUSTRIAL CASE STUDY

This section demonstrates the efficacy of the proposed PPFC algorithm with a case study involving temperature control of a Jacketed CSTR. The Continuous Stirred Tank Reactor (CSTR) is a common industrial unit that is widely employed in different chemical manufacturing processes. The reaction dynamics converting component A into component B in an ideal CSTR has a non-linear first-order behaviour. Nevertheless, many chemical reactions also require a specific temperature to be maintained within the tank for a flawless yield. Therefore, the tank is generally equipped with an outer jacket in which the temperature of a flowing fluid T_J is used as the manipulated variable to regulate the inside reaction temperature T, as shown in Fig. 2. The overall coupled model has two-state nonlinear dynamics with potential for exotic behaviour owing to multiple steady-states (Bequette, 2002). In this study, the linearised model around one operating point depicts unstable second-order dynamics given by (Rao and Chidambaram, 2008):

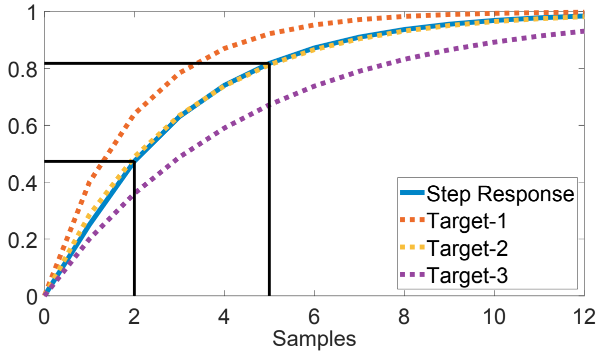


Fig. 3. Target responses with $\rho = [0.6, 0.715, 0.8]$ overlaying the normalised step response of $G_{s,0}(z)$

$$G(z) = \frac{T(z)}{T_J(z)} = \frac{2.102z + 0.4011}{z^2 - 1.465z + 0.058}.z^{-1} \qquad (20)$$

subject to $|T_J| \le 0.21°F$ and $|\Delta T_J| \le 0.075°F$. Note that both T and T_J are deviation variables around the steady-state values $T_{ss} = 101.1°F$ and $T_{J,ss} = 60°F$.

5.1 Pre-stabilisation and Offline Tuning

Next we pre-stabilise the Jacket CSTR model. Here, p_u=1 with one unstable pole $a^+(z) = z - 1.424$, one stable pole $a^-(z) = z - 0.041$ and a delay of $m = 1$ minute in measurement. Since $n = 2$, the pre-stabilised pole polynomial must be third-order with $\alpha(z) = z(z - 0.041)(z - 1/1.424)$. The first-order bi-proper compensator $C(z)$ is then constructed using (4)-(7) resulting in $C(z) = \dfrac{0.303z - 0.0123}{z + 0.0852}$. The following pre-stabilised delay-free model is obtained:

$$G_{s,0}(z) = \frac{T(z)}{v(z)} = \frac{0.637z^2 + 0.096z - 0.005}{z^3 - 0.743z^2 + 0.0288z} \qquad (21)$$

The next step is the controller tuning i.e. finding appropriate n_y and ρ. Fig. 3 shows the pre-stabilised step response curve overlaying various first-order target responses and suggests $2 \le n_y \le 5$ as the suitable coincidence horizon window. Interestingly, the target behaviour with $\rho = 0.715$ almost exactly overlaps the step response, whereas those with $\rho = 0.6$ or $\rho = 0.8$ do not match predictions within the desirable n_y range and hence would need over-actuation or under-actuation to enforce an intercept. In this study, we have selected $\rho = 0.715$ and $n_y = 3$.

In order to assess the performance of PPFC, two more controllers are implemented: the original PFC tuned with the same parameters given above, and a PI controller designed with $K_P = 0.02$ and $K_I = 0.004$. Note that the PI controller operates on the pre-stabilised plant after hardwiring the internal feedback loop with $C(z)$. However, doing so also introduces time-delay in the feedback design, hence a relatively poor PI performance is anticipated.

5.2 Nominal Unconstrained Performance

The unconstrained closed-loop performance in the absence of disturbance and modelling uncertainty is analysed first, with the results shown in Fig. 4. The temperature step response (top figure) achieved with PPFC and standard PFC is smooth and monotonically convergent. Neverthe-

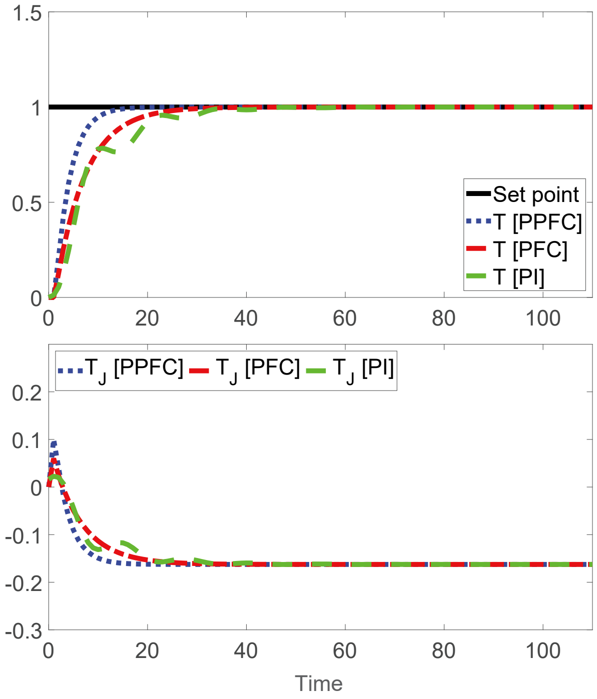


Fig. 4. Unconstrained performance without external perturbations (nominal case)

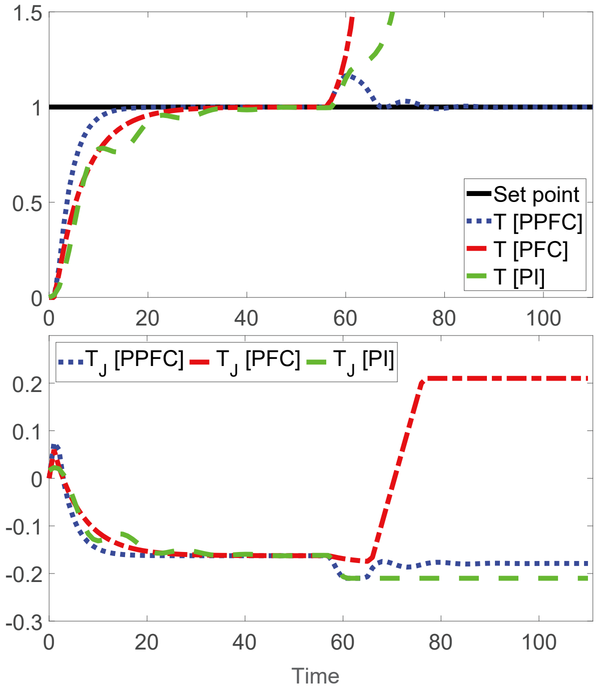


Fig. 5. Constrained performance in the presence of constant disturbance in jacket temperature

less, the tuning parameter ρ appears to have stronger linkage with the output after pre-stabilisation; that is, tuning will be easier and more intuitive in practice with PPFC. On the other hand, the response with the PI controller is rather oscillatory in the beginning, possibly due to the effect of time-delay in the error computation. Overall, the PI controller seems to have the slowest performance amongst the three choices. On the other hand, the jacket temperature control (bottom figure) corresponds to the associated step performance and demonstrates similar behaviour. Evidently, the fastest PPFC response is due to an aggressive control action, peaking at approximately $0.1°F$, as opposed to the other two with slightly lower peak values. While there is no remarkable difference in the nominal performance, it hardly portrays the true picture and the effect of external perturbations must also be considered for a more complete evaluation.

5.3 Constrained Performance in a Practical Situation

In this section, the effects of external perturbations on closed-loop performance are studied. Consider the scenario when a sudden process variation increases the jacket feed temperature by approximately 10% of the planned value. This is simulated as a constant disturbance signal introduced at the plant input around the 55^{th} minute of operation. The simulation results are depicted in Fig. 5. Clearly, both the standard PFC and the PI controllers respond poorly, immediately driving the system into instability. Moreover, the controllers appear highly sensitive as suggested by the aggressive input activity soon after the introduction of disturbance. This inevitably leads to actuator saturation, with possible equipment failure in practice. The Pre-stabilised PFC, on the other hand, displays commendable tracking with far superior disturbance rejection characteristic, providing fast and smooth normalisation of operation. Interestingly, apart from the slight deviation around the 60^{th} minute, the PPFC control performance appears very similar to the nominal behaviour displayed in Fig. 4.

Next an unmodelled pole at $z = 0.1$ is added to analyse the controllers' robustness against modelling mismatch with the results depicted in Fig. 6. While the transient performance of the PPFC is slightly affected (a slow target pole will be better in this case), the benefits of pre-stabilisation are even more pronounced as the constrained performance remains recursively feasible and stable throughout. In comparison, the PI controller clearly fails to accommodate the modelling uncertainty with immediate output divergence along with input and rate constraint violations. Interestingly, the standard PFC also destabilises, although this becomes apparent only around the 100 minute mark, owing to the use of unreliable and numerically infeasible divergent open-loop predictions in the decision making. In practice, this leads to spoiled product and financial loss to the manufacturer.

To conclude, the Pre-stabilised PFC appears to be the most reliable choice for the temperature control of Jacketed CSTR process in the presence of disturbances and modelling uncertainty.

6. CONCLUSION

This paper has presented the concept of pre-stabilised predictive functional control for unstable open-loop dynamic systems. An analytical approach to designing the pre-stabilising compensator is proposed, which is fairly simple and intuitive, and works well in combination with PFC. Specifically, it preserves the simplistic PFC parameter tuning and adds reliability, but at the cost of slightly more onerous constraint management. Nevertheless, the

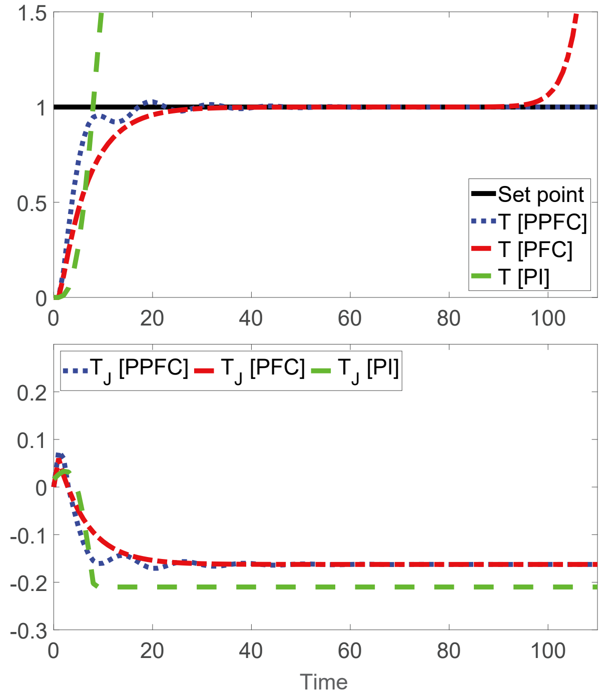


Fig. 6. Constrained performance in the presence of plant-model mismatch

overall advantage of pre-stabilisation in terms of closed-loop performance compared to a standard PFC and PI control has been observed in simulation studies, which also guarantees reliable operation in the presence of external perturbations.

While this study has highlighted the key benefits of pre-stabilisation using one application, in future, the authors plan to extend the scope of validation across a range of case-studies and real-time experiments. The future work will also focus on a more rigorous analysis of loop sensitivity to gain better understanding. Furthermore, a possible extension to accommodate a variety of challenging scenarios, including non-minimum phase and poorly damped dynamics, is also under development.

REFERENCES

Abdullah, M. and Rossiter, J. (2018a). Input shaping predictive functional control for different types of challenging dynamics processes. *Processes*, 6(8), 118.

Abdullah, M. and Rossiter, J.A. (2018b). Alternative method for predictive functional control to handle an integrating process. In *2018 UKACC 12th International Conference on Control (CONTROL)*, 26–31. IEEE.

Aftab, M.S., Rossiter, J.A., and Zhang, Z. (2021). Predictive Functional Control for Unstable First-Order Dynamic Systems. In J.A. Gonçalves, M. Braz-César, and J.P. Coelho (eds.), *CONTROLO 2020*, volume 695 of *Lecture Notes in Electrical Engineering*, 12–22. Springer International Publishing, Cham.

Bequette, B. (2002). Behavior of a CSTR with a recirculating jacket heat transfer system. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*. IEEE. doi:10.1109/acc.2002.1025296.

Fernandez-Camacho, E. and Bordons-Alba, C. (1995). *Model Predictive Control in the Process Industry.* Springer London. doi:10.1007/978-1-4471-3008-6.

Goodwin, G. (2001). *Control system design.* Prentice Hall, Upper Saddle River, N.J.

Khadir, M. and Ringwood, J. (2008). Extension of first order predictive functional controllers to handle higher order internal models. *International Journal of Applied Mathematics and Computer Science*, 18(2), 229–239. doi:10.2478/v10006-008-0021-z.

Maciejowski, J.M. (2002). *Predictive Control with Constraints.* Pearson Education.

Mayne, D.Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, 50(12), 2967–2986. doi:10.1016/j.automatica.2014.10.128.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.

Ogata, K. (1995). *Discrete-Time Control Systems.* Prentice Hall Englewood Cliffs, NJ, 2nd ed. edition.

Qin, S. and Badgwell, T.A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733–764. doi:10.1016/s0967-0661(02)00186-7.

Rao, A.S. and Chidambaram, M. (2008). Analytical design of modified smith predictor in a two-degrees-of-freedom control scheme for second order unstable processes with time delay. *ISA Transactions*, 47(4), 407–419. doi:10.1016/j.isatra.2008.06.005.

Richalet, J. (1993). Industrial applications of model based predictive control. *Automatica*, 29(5), 1251–1274. doi:10.1016/0005-1098(93)90049-y.

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

Richalet, J. and O'Donovan, D. (2009). *Predictive Functional Control: Principles and Industrial Applications.* Springer Science & Business Media.

Richalet, J., Rault, A., Testud, J., and Papon, J. (1978). Model predictive heuristic control. *Automatica (Journal of IFAC)*, 14(5), 413–428.

Rossiter, J.A., Kouvaritakis, B., and Rice, M. (1998). A numerically robust state-space approach to stable-predictive control strategies. *Automatica*, 34(1), 65–73.

Rossiter, J. (2016). Input shaping for pfc: how and why? *Journal of control and decision*, 3(2), 105–118.

Rossiter, J. (2018). *A First Course in Predictive Control.* CRC Press.

Rossiter, J. and Haber, R. (2015). The effect of coincidence horizon on predictive functional control. *Processes*, 3(1), 25–45.

Skogestad, C.G.S. (2018). Should we forget the smith predictor? *IFAC-PapersOnLine*, 51(4), 769–774. doi:10.1016/j.ifacol.2018.06.203.

Visioli, A. (2006). *Practical PID Control.* Springer London. doi:10.1007/1-84628-586-0.

Zhang, Z., Rossiter, J., Xie, L., and Su, H. (2018). Predictive functional control for integral systems. In *International Symposium on Process System Engineering.*

Zhang, Z., Rossiter, J.A., Xie, L., and Su, H. (2020). Predictive functional control for integrator systems. *Journal of the Franklin Institute*, 357(7), 4171–4186. doi:10.1016/j.jfranklin.2020.01.026.

Appendix D

A Comparison of Tuning Methods for Predictive Functional Control

John Anthony Rossiter, and Muhammad Saleheen Aftab

This paper has been published in Processes, Vol 9, 2021

Author Contributions. This paper is a collaborative work between both authors. J. A. Rossiter provided initial proposals and accurate communication of the concepts. M. S. Aftab developed the code, and analysed the concepts in the case studies.

processes

MDPI

Review

A Comparison of Tuning Methods for Predictive Functional Control

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

Department of Automatic Control and System Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, UK; Msaftab1@sheffield.ac.uk

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

Abstract: Predictive functional control (PFC) is a fast and effective controller that is widely used in preference to PID for single-input single-output processes. Nevertheless, the core advantages of simplicity and low cost come alongside weaknesses in tuning efficacy. This paper summarises and consolidates the work of the past decade, which has focused on proposing more effective tuning approaches while retaining the core attributes of simplicity and low cost. The paper finishes with conclusions on the more effective approaches and links to context.

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

check for updates

Citation: Rossiter, J.A.; Aftab, M.S. A Comparison of Tuning Methods for Predictive Functional Control. *Processes* **2021**, *9*, 1140. https://doi.org/10.3390/pr9071140

Academic Editor: Cesar De Prada

Received: 4 June 2021
Accepted: 23 June 2021
Published: 30 June 2021

1. Introduction

Predictive functional control (PFC) is often used in preference to a PID approach on mainly single-input single-output (SISO) applications [1–4]. The reason industrial users might prefer PFC to PID is threefold: (i) being model-based, in theory at least, it can exploit the model information better and thus handle challenging dynamics; (ii) being based on prediction, constraint handling can be embedded in a systematic fashion and (iii) the coding complexity is similar to PID, which is elementary [5,6]; thus, maintenance and implementation are straightforward. A further important point that follows from the above three is that PFC is cheap (similar costs to PID) and is far cheaper than more conventional model predictive control (MPC) schemes, such as DMC and GPC [7].

Despite its widespread adoption in parts of the industry, a traditional PFC algorithm still has a number of weaknesses, with the most important one being that the tuning parameters are effective for only a limited range of dynamical systems and thus critically:

- Links between the tuning parameters and behaviour are not as intuitive as they need to be for many cases [8].
- For systems with challenging dynamics, a traditional PFC algorithm may fail to achieve satisfactory behaviour.

MPC theoreticians may also worry about the lack of a priori stability guarantees [9,10], but given that PFC is a competitor with PID, this issue is not important in practice, and it is common to use a posteriori stability checks.

The originators of PFC [6] proposed a number of ad hoc modifications to the basic algorithm to improve properties and tuning for systems with non-simple dynamics (e.g., open-loop integrators, instability, under-damping, non-minimum phase characteristics). The most popular proposal was to deploy a type of cascade structure, where an inner loop uses proportional only control to improve the dynamics for an outer loop to control with PFC [6,11,12]. This restriction makes it difficult to have a systematic selection of gain, and moreover, the consequent tuning proposals embed dynamics and assumptions that are somewhat contradictory of the original PFC concepts and, for many cases, the back-calculation used for constraint handling is equally ad hoc (suboptimal). In summary, it is difficult to see a systematic design behind these modifications, and the tuning still lacked intuition.

Consequently, in recent years, many authors have proposed a number of more systematic modifications to improve tuning for specific cases, e.g., [9,10,13–17]. Nevertheless, there is overlap in the application and benefits of these approaches alongside an undesirable multiplicity of options. Thus it is timely to write an overview paper that extracts the most useful aspects of these papers and provides some unified and narrower systematic guidance to the user.

Section 2 introduces conventional PFC for completeness before Section 3 summarises concisely the alternative approaches from recent years. The main contribution of this paper is in Section 4, which summarises the strengths and weaknesses of the various proposals before tabulating a recommended generic approach and offering some numerical examples for completeness.

2. Overview of Traditional PFC

This section presents, in brief, the main concepts, notation, and formulation of PFC [5,6,8,18]. As the focus is on the SISO case, transfer function-based models are used; thus, for example, assuming discrete time, the model will take the form:

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

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

Remark 1. *As this paper is focused on tuning and concepts, we will avoid explicit inclusion of disturbance handling in the algebra hereafter to simplify the presentation. The most common way of doing this in PFC (and also common in mainstream MPC) is simulating an independent model in parallel and using the difference between the independent model and the process outputs as a disturbance estimate d_k to correct for any steady-state bias in the predictions. Here, as seen in Figure 1, $d_k = y_p - y_m$, with y_p being the process output measurement and y_m the independent model output.*

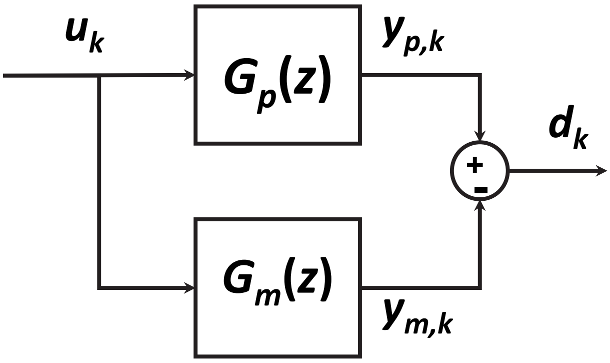


Figure 1. Independent model structure and disturbance estimate.

2.1. Prediction

Finding ouput predictions for model (1) is standard in the literature [7,19], so here we assume the reader is comfortable with the following result based on input increments $\Delta u_{k+i} = u_{k+i} - u_{k+i-1}$:

$$\underset{\rightarrow}{y}_{k+1|k} = H\Delta\underset{\rightarrow}{u}_k + P\Delta\underset{\leftarrow}{u}_k + Q\underset{\leftarrow}{y}_k \tag{2}$$

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

The parameters H, P, Q depend on the model parameters and are easy to determine. It is convenient to define a shorthand to extract individual rows of (2), so:

$$\underset{\rightarrow}{y}_{k+n|k} = H_n \Delta \underset{\rightarrow}{u}_k + P_n \Delta \underset{\leftarrow}{u}_k + Q_n \underset{\leftarrow}{y}_k \tag{3}$$

where $\mathbf{e}_n$ is the nth standard basic vector and $H_n = \mathbf{e}_n^T H, P_n = \mathbf{e}_n^T P, Q_n = \mathbf{e}_n^T Q$.

2.2. Conventional PFC Control Law

PFC design is intuitive in that one uses a first-order dynamic as a model for an ideal closed-loop response [18]; the associated time constant or pole being the most important design parameter. In practice, one aims to achieve this behaviour by ensuring an explicit matching between the predictions and 'ideal behaviour' at a single point, denoted the coincidence point. Hence, a PFC control law is defined by the equality:

$$y_{k+n|k} = (1 - \lambda^n)r + \lambda^n y_k \tag{4}$$

where r is the set point, $y_{k+n_y|k}$ is the n-step ahead system prediction, and λ is the desired closed-loop pole (in effect $\lambda = e^{\frac{-3T}{T_s}}$, where T is the sampling period and T_s is the desired settling time so practitioners could equivalently choose T_s directly).

A core tenant of PFC is computational simplicity [6,18], so within the predictions we assume only a single degree of freedom (d.o.f.). Hence, selecting $\Delta u_{k+i} = 0,\ i > 0$ and combining this with prediction (3) and control law (4) gives:

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

Thus, after rearrangement, the control law (4) is implemented by solving:

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

2.3. Efficacy of Conventional PFC

Control law (6) works well when the open loop system has behaviour that is close to a monotonic step response, such as 1st order systems; for these cases, the tuning parameter λ is then reasonably effective. However, for systems with more complex dynamics or significant lag in the initial response, the tuning parameter is much less effective [8,14,15]. The developments summarised in this paper are focused towards the latter cases.

2.4. Constraint Handling

Constraint handling is not a main discussion point in this paper, so it is included briefly here for completeness. The back calculation [20] favoured in traditional PFC papers is somewhat simplistic and suboptimal, so where constraint handling is required, the authors would recommend users to adopt approaches similar to those in mainstream MPC [14,19,21,22] whereby inequalities are developed to explicitly check every prediction point against the corresponding constraint.

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

For a suitably long horizon, constraints (7) can be captured by the inequalities:

$$C \Delta u_k \le \mathbf{f}_k \tag{8}$$

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

where $\mathbf{f}_k$ depends on past data in $\underset{\leftarrow}{\Delta u}_k, \underset{\leftarrow}{y}_k$ and on the limits.

The input/output predictions will satisfy constraints if inequalities (8) are satisfied, and thus the PFC algorithm should consider these explicitly. Given there is a single d.o.f., a simple *for loop* within the code can find the choice of Δu_k closest to (6) to ensure this very efficiently and, where appropriate, to ensure recursive feasibility properties (assuming convergent predictions).

Remark 2. *Small modifications to the algebra above are needed for the algorithms in the following sections, but given this is straightforward and does not require new concepts, the details are omitted.*

3. Summary of Recent Proposed Enhancements

This section outlines a number of proposals that have appeared in the literature to improve the tuning of efficacy of PFC. The fundamental problem [8,15] is that for many systems there is a poor correspondence between the chosen λ and the resulting closed-loop pole, thus undermining a core selling point of PFC, that is, ease of tuning. Notably, for a system with open-loop unstable poles, significant underdamping, or integrating dynamics, PFC is quite challenging to tune effectively [8], and moreover, the resulting divergent or oscillating predictions may give rise to infeasibility and/or robustness issues.

3.1. Fundamental Weaknesses and Core Conceptual Proposal

A fundamental conceptual weakness of a simplistic PFC approach is that one is basing decisions on an open-loop prediction, which may have undesirable, possibly divergent, dynamics. In addition to creating significant issues with reliable constraint handling, using open-loop predictions can easily lead to ill-posed problems with consequent loss of performance [19]. This is well recognised even in mainstream MPC [23], although to some extent, the issue can be partially side-stepped by having multiple d.o.f. so that the optimised predictions have better dynamics during transients.

Hence, and perhaps ironically, while MPC methods use optimisation of predictions to find a nicely shaped prediction, the optimisation itself is likely to fail unless the parameterisation of the predictions within the optimisation already have appropriate shaping and, surprisingly, the default shapings in many standard MPC laws are poorly chosen for some scenarios. To be more specific, this is a big issue and far more evident where there are low numbers of d.o.f., such as with PFC.

Hence, the underlying philosophy in the proposed modifications to PFC rely on a different parameterisation of the d.o.f. to that shown in (3), that is, the d.o.f. in the predicted future input sequence $\Delta \underset{\rightarrow k}{u}$ are not just the first increment Δu_k. The parameterisation chosen must have two key attributes:

1. It can be reduced to a single d.o.f. so that computation and coding is trivial, in line with a conventional PFC approach.
2. The associated predictions should be better aligned to the desired behaviour of (4) than open-loop predictions.

Hence, the main philosophy deployed in recent work is to pre-stabilise/pre-shape the output predictions so that the effect of unwanted open-loop poles on the predictions are alleviated [24–26].

3.2. Input Shaping PFC

This section summarises the concept of algebraic or explicit input shaping of the open-loop predictions. It relies on explicit pole cancellation within the predictions [22] and some neat algebra, which requires simultaneous equations of a similar complexity to computation of the predictions.

Assume that the system model has some undesirable modes denoted by $a^{+}(z)$, so:

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

The challenge is to determine $\Delta \underset{\rightarrow k}{u}$ such that the implied $u(z)$ (including past behaviour), cancels the modes $a^{+}(z)$ from the future predictions. It can be shown [22] that such a parameterisation takes the form:

$$\Delta \underset{\rightarrow k}{u} = P_1\mathbf{p}; \quad \mathbf{p} = A_1 \underset{\leftarrow k}{y} + A_2 \Delta \underset{\leftarrow k}{u} \tag{10}$$

for suitable P_1, A_1, A_2. One can add future d.o.f. to the predictions with an additional term as follows:

$$\Delta \underset{\rightarrow k}{u} = P_1\mathbf{p} + C_{a^{+}}\phi \tag{11}$$

where the parameter ϕ denotes the degrees of freedom (d.o.f.) and $C_{a^{+}}$ is a Toeplitz matrix of the parameters in $a^{+}(z)$. The corresponding output predictions, from which all the modes linked to $a^{+}(z)$ are now absent, can be deduced using similar algebra to be:

$$\underset{\rightarrow k+1}{y} = P_2\mathbf{p} + H_s\phi \tag{12}$$

for suitable matrices P_2, H_s.

Remark 3. *For PFC, one would choose ϕ to be a simple scalar, that is, with one d.o.f. More complex MPC algorithms may choose this to be a vector or indeed other forms such as $\phi(z)/\alpha(z)$ where $\alpha(z)$ contains some desirable closed-loop dynamics.*

To finish up, it is important to give some reflections on input shaping Algorithm 1 and whether this is an approach worth further investigation.

- On the positive side, the shaping is effective at removing undesirable modes from the predictions, which can be considered essential for reliable constraint handling and recursive feasibility for systems with unstable and/or oscillatory modes.
- On the downside, this approach does not help with the weak links between the tuning parameter λ and the closed-loop behaviour.
- The explicit cancellation used in the predictions can lead to sensitive results [24–26] and potentially aggressive input trajectories. Indeed, this conceptual approach, while interesting, has not been pursued in the mainstream MPC literature. One can mitigate against the aggressive input sightly by incorporating the pole parameter $\alpha(z)$, but, as yet, no systematic guidance exists for this process and, as discussed later, one could argue better alternatives exist.
- The algebra required to produce the parameterisations in (11), (12) are not simple in general to code, albeit the code would be very quick to execute and could be written in about 50 lines of code (e.g., in C, Python). This mitigates against the core selling points of simplicity.

Algorithm 1 The PFC input shaping law is derived by:

```
1. Substitute predictions (12) into (4) to solve for $\phi$.
2. Use (11) to determine the current system input $\Delta u_k$.
```

3.3. Pole Placement PFC

The pole placement technique [10] exploits properties of PFC associated with 1st order systems. It can be shown that for a first order system, the parameter λ is a precise tuning parameter, that is, the nominal closed-loop will have a pole at λ as requested. As before, this section avoids giving the detailed derivations as those are available in the original sources, and here, we focus on core concepts.

Using partial fraction expansions, one can represent a higher order process as a sum of first order processes, for example:

$$\frac{b(z)}{a(z)} = \frac{B_1}{z - A_1} + \frac{B_2}{z - A_2} + \cdots \tag{13}$$

The next trick is to find separate suitable PFC inputs for each of these parallel systems, which would drive the pole, for each system, to λ. A summation of these inputs applied to the actual system should also result in a system closed-loop pole being a λ, from simple superposition arguments. The formal algebra and derivations in the original sources shows that this principle works very well, and one can indeed achieve a closed-loop pole of λ exactly with elementary computations such as (6).

Nevertheless, despite the apparent efficacy for tuning, there are of course some downsides.

- The handling of complex poles requires either complex number algebra or more involved real algebra [27], which could mitigate against easy coding and acceptance.
- Constraint handling was not tackled explicitly and again, could be somewhat more involved than discussed in Section 2.4.
- For higher order systems, there was an implicit need to select all the closed-loop poles, thus rendering this almost equivalent to a standard pole-placement approach. In general, it is not obvious how to select the less dominant poles, which means the approach loses some of its attraction.

3.4. Laguerre PFC

A core weakness of the conventional PFC approach that has been well recognised in the mainstream MPC literature [23,25,28,29] is the restriction, in the predictions, to a finite number of input changes in the immediate horizon. Indeed PFC assumes a constant future input, which is the very extreme case. Such a restriction is reasonable for systems with simple under-damped dynamics and where open-loop speeds of response are satisfactory, but this restriction is a significant problem where the open-loop has more complex dynamics and/or there is a desire for the closed-loop to have faster poles.

Recognising that the input prediction parameterisations for open-loop MPC formulations should have some dynamics was proposed in [30], where Laguerre polynomials were considered. Parameterising input sequences around Laguerre polynomials makes good engineering sense as they are built around a *target pole*, which the user can easily choose and thus, while investigated more widely since the approach was also recently considered for PFC [14].

The core proposal is to replace the parameterisation of future inputs as follows:

$$\Delta \underset{\rightarrow}{u}(z) = \sum_i L_i(z)\eta_i \quad \text{or} \quad \underset{\rightarrow}{u}(z) = u_{ss} + \sum_i L_i(z)\eta_i \tag{14}$$

where $L_i(z)$ are the Laguerre polynomials; for PFC, only the first polynomial is used:

$$L_1(z) = 1 + pz^{-1} + p^2 z^{-2} + \cdots \tag{15}$$

The pole to be selected is clearly seen here as p and would logically match the choice of λ in the PFC target.

Some reflections on this proposal are as follows.

- In general terms, there is a closer (compared to conventional PFC) correspondence between the closed-loop poles and the target pole of λ when the input is parameterised as in (14). Thus, on balance, one might prefer this algorithm.
- It was noted that it was better to use the parameterisation $\underset{\rightarrow}{u}(z) = u_{ss} + L_1(z)\eta$ as otherwise, the transients speed of response was severely impeded by the shape of the first Laguerre polynomial.
- Despite being effective for some systems and an improvement on the conventional PFC approach, there was still poorer consistency between the closed-loop poles and the target pole λ than was satisfactory. Moreover, this approach did not help, or indeed does not have sufficiently tailored shaping, with difficult dynamics such as open-loop unstable poles and oscillatory modes.

3.5. Closed-Loop or Pre-Stabilised PFC

The most recent suggestion for modifying PFC builds on some ideas with the case by case modifications for the original algorithm [6] but implements this in a more systematic and effective manner. The basic idea is to recognise that where a system has poor open-loop dynamics, a simple feedback loop, often based solely on simple proportional control, can improve those dynamics [11,16,17]. As the efficacy of PFC is strongly linked to the nominal dynamics having a shape somewhat close to a well-damped system, this initial pre-stabilisation can be hugely effective in enabling PFC. The ultimate design is now a cascade structure, but given the inner loop is a simple classical design, this additional complexity is a very minor addition and can be viewed as essential for handling challenging dynamics.

A minor consequence of using a cascade structure is that the d.o.f. to be utilised in solving (4) is now the input to the inner loop rather than the input signal (see Figure 2). In the early works, this variable has been parameterised as a single constant, but equally, we could make use of parameterisations such as in Section 3.4. Ironically, the use of a cascade loop has an insignificant impact on the constraint handling aspects as, albeit the parameters in the replacement for (8) are slightly different, the number of rows and treatment is similar.

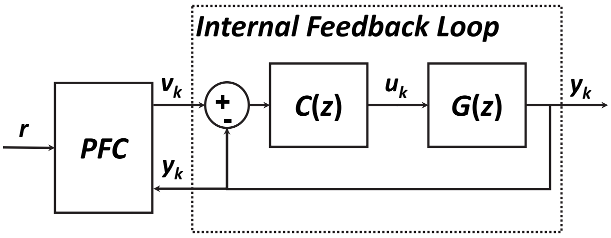


Figure 2. Cascade structure with PFC.

In summary, early reflections on using a simple cascade structure are:

- Final tuning outperforms conventional PFC for a range of dynamics and notably deals effectively with challenging dynamics such as under-damped processes and open-loop instability.
- Enables simple guarantees of recursive feasibility, that is, effective constraint handling; something other methods largely struggle with.
- The main downside is that, for systems with very challenging open-loop dynamics, it may not be straightforward to determine a simple classical control law such as PID for the inner loop. Of course, one could argue that such cases are irrelevant if we are considering PFC as a competitor to simple classical control. Moreover, conventional PFC had no answers to this either, and more expensive alternatives may be required.

3.6. Dealing with Lag in the Closed-Loop Responses

Notwithstanding the desire for the closed-loop pole to match the target pole, it was noted recently that there was a fundamental weakness in the definition of control law (4), which almost inevitably introduces some lag into the closed-loop responses [15]. This paper will not discuss that issue in detail as, in effect, it relates primarily to the feedforward component of the control law and thus could be considered as a separate issue from controller tuning. Feedforward information needs to be integrated carefully or it could cause unexpected distortions in the behaviour.

In simple terms, some form of memory of the target and disturbance estimate information needs to be integrated into (4) and the implied feedforward to ensure consistency of planning from one sample to the next. Where the system response differs from a 1st order response in fast transients, as almost all real systems will, the nominal PFC law of (4) does not recognise this inconsistency adequately and thus may not behave as well as it could.

4. Comparison of Alternative PFC Approaches

This section presents several numerical examples with varying dynamics of the alternative approaches. These are used as evidence for some stronger conclusions given at the end of the section. Two aspects are considered to be at the core for a PFC algorithm to be successful:

1. The tuning needs to be intuitive, that is, not requiring expert involvement. Here, we take that to mean that the user can choose the desired pole λ (in effect a closed-loop settling time) and the algorithm will deliver something close to that.
2. The coding requirements should be simple enough for interrogation and management by local staff, thus again cutting the reliance on expensive consultants. The code must not rely on expensive optimisers and use only the simplest coding constructs.
3. It is accepted that effective constraint handling is non-trivial even for PID approaches, and thus, the code for this part will inevitably be more than a few lines. However, it is similar for all the approaches, as mentioned in Section 2.4 and, thus, not presented.

Due to bullet point 1 above, we will consider three approaches: (i) conventional PFC (Section 2—denoted CPFC; (ii) Laguerre PFC (Section 3.4—denoted LPFC) and (iii) closed-loop PFC (Section 3.5—denoted CLPFC).

4.1. Case Studies

In line with earlier work [22,31], it is appropriate to use some industrially motivated case studies to benchmark the different approaches and numerical comparisons. These are selected to cover a range of challenging dynamics such as integrators, instability and oscillatory modes. These are summarised next.

Boiler Level Control: A typical model for this process [32] is usually a first-order system with an integrator, output water level and the input water flow rate; a representative model is:

$$G_1 = \frac{0.1z^{-1} + 0.4z^{-2}}{(1 - 0.8z^{-1})(1 - z^{-1})} \tag{16}$$

Depth Control of Unmanned Free-Swimming Submersible (UFSS): The depth of an unmanned submarine can be controlled by deflecting its elevator surface, whereby the vehicle will rotate about its pitch axis; the associated vertical forces due to the water flow enable the vehicle to sink or rise. Typical dynamics include one stable pole and two complex poles with stable zeros [33]. Thus, a representative model is given by G_2 below.

$$G_2 = \frac{0.85z^{-1} - 1.5z^{-2} + 0.85z^{-2}}{(1 - 0.6z^{-1})(1 - 1.6z^{-1} + 0.8z^{-2})} \tag{17}$$

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

Temperature Control of Fluidised Bed Reactor: A fluidised bed reactor may involve exothermic reactions and thus open-loop instability. The reactor bed temperature (°C) is controlled with a coolant flow rate $(\mathrm{m}^3\,\mathrm{s}^{-1})$; thus, one obvious risk is that changes in flow rate can easily trigger rapid divergence in the temperature. A representative model to capture the core dynamics for control design includes at least one stable pole and one unstable pole [34]; a typical example is given below as G_3.

$$G_3 = \frac{0.2z^{-1} - 0.26z^{-2}}{(1 - 0.9z^{-1})(1 - 1.5z^{-1})} \tag{18}$$

4.2. Formation of Pre-Compensated Prediction Model for CLPFC

For the integrator system G_1, a proportional controller $C(z) = 0.0461$ implemented as shown in Figure 2 results in the pre-compensated dynamics $T_1 = \frac{0.004615z^{-1} + 0.01846z^{-2}}{1 - 1.795z^{-1} + 0.8185z^{-2}}$ with stable poles at $0.8975 \pm j0.1140$.

For G_2, a PI controller has to be tuned as P and PD controllers are incapable of stabilising the oscillatory open-loop behaviour. Therefore, $C(z) = 0.0038 + \frac{0.0423}{z-1}$ is implemented, providing $T_2 = \frac{0.0032z^3 + 0.027z^{-1} - 0.055z^{-2} + 0.033z^{-3}}{1 - 3.20z^{-1} + 3.98z^{-2} - 2.30z^{-3} + 0.51z^{-4}}$ with stable poles at $0.822 \pm j0.088$ and $0.78 \pm j0.3840$.

For the unstable system G_3, the PID controller fails to stabilise; therefore, the inner loop has been tuned using a pole-placement method discussed in [35]. With $C(z) = \frac{22.914z^{-1} - 20.622z^{-2}}{1 - 4.55z^{-1} + 1.66z^{-2}}$, a pre-stabilised prediction model with stable poles at 0.9, 0.667 and 0.4 is obtained: $T_3 = \frac{4.58z^{-1} - 10.082z^{-2} + 5.36z^{-3}}{1 - 1.967z^{-1} + 1.227z^{-2} - 0.24z^{-3}}$.

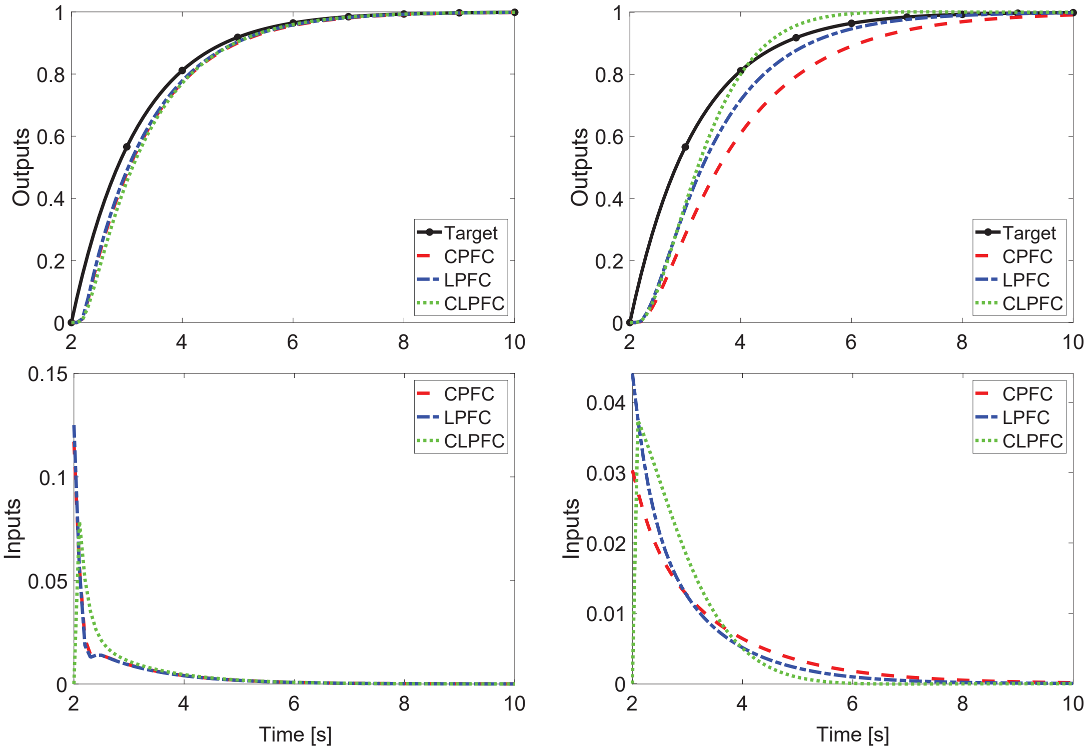
In the following analysis, T_1, T_2 and T_3 designed above will be used with CLPFC, whereas both CPFC and LPFC will utilise the open-loop dynamics G_1, G_2 and G_3 for decision-making.

4.3. Tuning Efficacy and Closed-Loop Performance Comparison

Closed-loop performance of the three PFC approaches is evaluated with regards to the tuning efficacy for both short and long coincidence horizons. For each case study, the selected controller parameters λ and n are tabulated in Table 1. The simulation results are shown in Figures 3–5, and the key observations are:

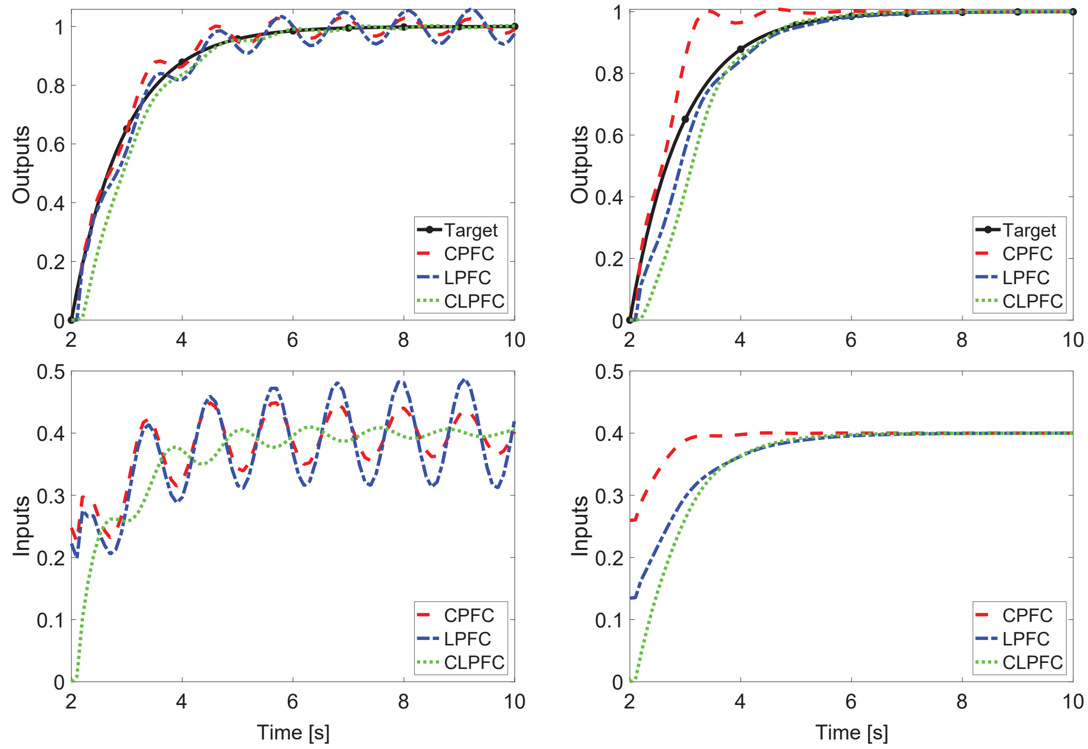
- CLPFC clearly demonstrates a much stronger link between the actual and the desired closed-loop performance, even with longer coincidence horizons. This is important since long horizons may be necessary for better loop shaping against external perturbations, as pointed out in [17]. Furthermore, the control effort with CLPFC is smooth and the least aggressive as opposed to both CPFC and LPFC, which is crucial for constraint adherence in practice.
- The input parametrisation with CPFC and LPFC for small horizon apparently works with the relatively less difficult system G_1; nonetheless, both alternatives fail to handle the more challenging dynamics of G_2 (oscillations) and G_3 (non-minimum phase, instability). In contrast, the re-parametrisation due to the inner loop in CLPFC warrants a faster and smoother output convergence even in difficult applications.
- The dominant closed-loop poles given in Table 1 mirror the closed-loop performance presented graphically. Apart from G_1, the actual poles are far away from the desired λ for CPFC, even the LPFC tuning is less effective for G_2 and G_3. However, the closed-loop poles remain stable and much closer to the target pole with CLPFC.

In short, the CLPFC algorithm overcomes the tuning and performance weaknesses of the CPFC for difficult dynamic problems. Although LPFC has also shown better performance as compared to the conventional algorithm for difficult but stable systems, it appears less effective in controlling non-minimum phase and/or unstable open-loop dynamics.



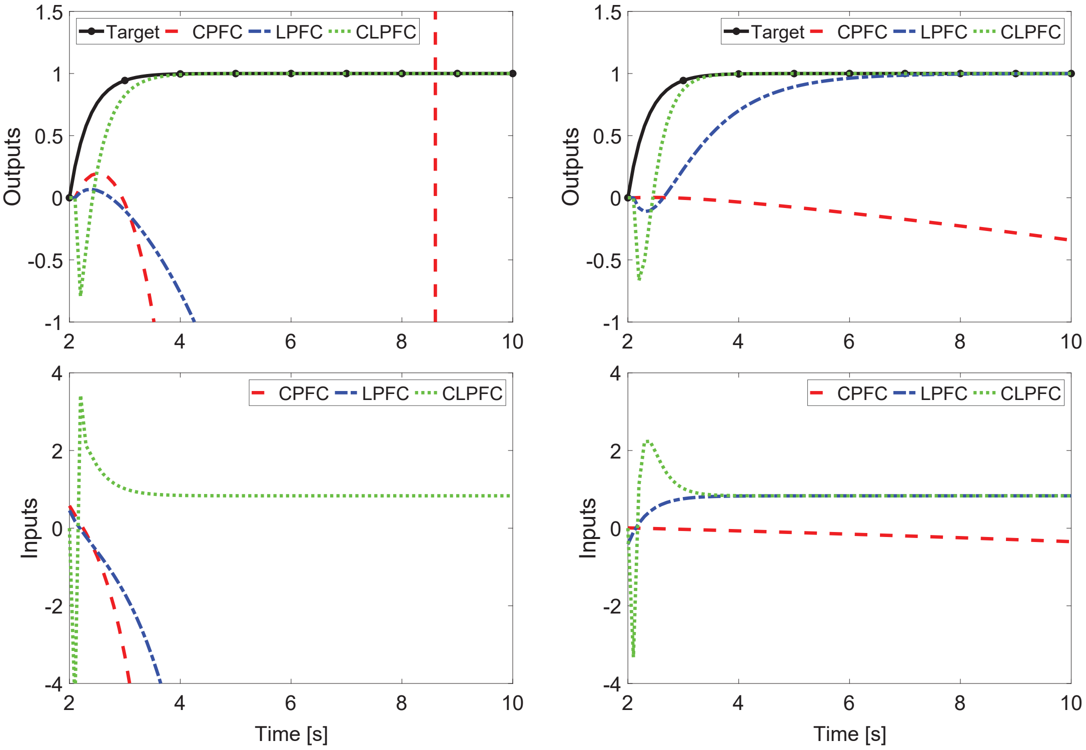
(**a**) $\lambda = 0.92$, $n = 5$ (**b**) $\lambda = 0.92$, $n = 15$

Figure 3. Comparison of closed-loop performance between CPFC, LPFC and CLPFC for G_1.



(**a**) $\lambda = 0.90$, $n = 4$ (**b**) $\lambda = 0.90$, $n = 12$

Figure 4. Comparison of closed-loop performance between CPFC, LPFC and CLPFC for G_2.



(**a**) $\lambda = 0.75, n = 7$

(**b**) $\lambda = 0.75, n = 18$

Figure 5. Comparison of closed-loop performance between CPFC, LPFC and CLPFC for G_3.

Table 1. Dominant closed-loop poles.

			CPFC	LPFC	CLPFC
G_1	$\lambda = 0.92$	$n = 5$ $n = 15$	0.92 0.94, 0.75	0.92 0.92, 0.72	0.92, 0.53 $0.88 \pm$ j0.05
G_2	$\lambda = 0.90$	$n = 4$ $n = 12$	0.88, $0.85 \pm$ j0.51 0.78, $0.73 \pm$ j0.4	0.9, $0.86 \pm$ j0.52 0.9, $0.77 \pm$ j0.4	0.9, $0.85 \pm$ j0.47 0.88, 0.68, $0.78 \pm$ j0.39
G_3	$\lambda = 0.75$	$n = 7$ $n = 18$	1.16, 0.84 1.005, 0.9	1.05, 0.75 0.9, 0.75	0.9, 0.74 0.9, 0.68, 0.39

5. Conclusions

This paper has presented a consolidated review of the recently proposed modified PFC algorithms, specifically focusing on the tuning issues pertaining to challenging dynamic applications. Numerous proposals have surfaced in the past two decades to improve control functionality, at least theoretically, although sometimes at the price of increased complexity resulting in diminished practical appeal. Nevertheless, this paper focuses on two recent approaches, namely Laguerre PFC and closed-loop (or pre-stabilised) PFC, that are well-explored in the mainstream MPC literature with proven efficacy in theory and practice. These modifications mainly work by introducing a different parametrisation of the degree-of-freedom, which is necessary to induce more flexibility to handle difficult dynamics in the control law; such flexibility does not exist in the conventional PFC algorithm, which consequently is difficult to tune effectively for many cases. The industrial case studies presented herein have demonstrated a superior performance and parameter tuning efficacy from the closed-loop PFC, which overcomes the inherent deficiency of the conventional algorithm against difficult dynamic problems to a greater extent while retaining the associated simplicity and intuitiveness. Although the Laguerre PFC alone may not perform as effectively in such applications, as a future consideration, nonetheless, it is expected to yield further performance improvement when utilised in conjunction with the closed-loop PFC formulation.

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

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

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

References

1. Khadir, M.; Ringwood, J. Stability issues for first order predictive functional controllers: Extension to handle higher order internal models. In Proceedings of the International Conference on Computer Systems and Information Technology, Algiers, Algeria, 19–21 July 2005; pp. 174–179.
2. Richalet, J. Industrial applications of model based predictive control. *Automatica* **1993**, *29*, 1251–1274. [CrossRef]
3. Fallasohi, H.; Ligeret, C.; Lin-Shi, X. Predictive Functional Control of an expansion valve for minimizing the superheat of an evaporator. *Int. J. Refrig.* **2010**, *33*, 409–418. [CrossRef]
4. Haber, R.; Rossiter, J.A.; Zabet, K. An alternative for PID control: Predictive functional Control—A tutorial. In Proceedings of the 2016 American Control Conference (ACC), Boston, MA, USA, 6–8 July 2016; pp. 6935–6940.
5. Haber, R.; Bars, R.; Schmitz, U. *Predictive Control in Process Engineering: From the Basics to the Applications*; Wiley-VCH: Weinheim, Germany, 2011.
6. Richalet, J.; O'Donovan, D. *Predictive Functional Control: Principles and Industrial Applications*; Springer: Berlin, Germany, 2009.
7. Clarke, D.W.; Mohtadi, C. Properties of generalized predictive control. *Automatica* **1989**, *25*, 859–875. [CrossRef]
8. Rossiter, J.A.; Haber, R. The effect of coincidence horizon on predictive functional control. *Processes* **2015**, *3*, 25–45. [CrossRef]
9. Rossiter, J.A. A priori stability results for PFC. *Int. J. Control* **2016**, *90*, 305–313. [CrossRef]
10. Rossiter, J.A.; Haber, R.; Zabet, K. Pole-placement Predictive Functional Control for over-damped systems with real poles. *ISA Trans.* **2016**, *61*, 229–239. [CrossRef] [PubMed]
11. Zhang, Z.; Rossiter, J.A.; Xie, L.; Su H. *Predictive Functional Control for Integral Systems*; PSE: Bellevue, WD, USA, 2018.
12. Richalet, J.; Rault, A.; Testud, J.; Papon, J. Model predictive heuristic control: Applications to industrial processes. *Automatica* **1987**, *14*, 413–428. [CrossRef]
13. Khadir, M.; Ringwood, J. Extension of first order predictive functional controllers to handle higher order internal models. *Int. J. Appl. Math. Comput. Sci.* **2008**, *18*, 229–239. [CrossRef]
14. Abdullah, M.; Rossiter, J.A. Using Laguerre functions to improve the tuning and performance of predictive functional control. *Int. J. Control* **2021**, *94*, 202–214. [CrossRef]
15. Rossiter, J.A.; Abdullah, M. Improving the use of feedforward in Predictive Functional Control to improve the impact of tuning. *Int. J. Control* **2020**. [CrossRef]
16. Zhang, Z.; Rossiter, J.A.; Xie, L.; Su, H. Predictive Functional Control for Integrator Systems. *J. Frankl. Inst.* **2020**, *357*, 4171–4186. [CrossRef]
17. Aftab, M.S.; Rossiter, J.A.; Zhang, Z. Predictive functional control for unstable first-order dynamic systems. In *CONTROLO 2020*; Goncalves, J.A., Ed.; Springer Nature Switzerland AG: Cham, Switzerland, 2021; pp. 1–11.
18. Richalet, J.; O'Donovan, D. Elementary predictive functional control: A tutorial. In Proceedings of the 2011 International Symposium on Advanced Control of Industrial Processes (ADCONIP), Hangzhou, China, 23–26 May 2011; pp. 306–313.
19. Rossiter, J.A. *A First Course in Predictive Control*, 2nd ed.; CRC Press: London, UK, 2018.
20. Fiani, P.; Richalet, J. Handling input and state constraints in predictive functional control. In Proceedings of the 30th IEEE Conference on Decision and Control, Brighton, UK, 11–13 December 1991; pp. 985–990.
21. Gilbert, E.; Tan, K. Linear systems with state and control constraints: The theory and application of maximal admissible sets. *IEEE Trans. Autom. Control* **1991**, *36*, 1008–1020. [CrossRef]
22. Abdullah, M.; Rossiter, J.A. Input Shaping Predictive Functional Control for Different Types of Challenging Dynamics Processes. *Processes* **2018**, *6*, 118. [CrossRef]
23. Rossiter, J.A.; Kouvaritakis, B. Numerical robustness and efficiency of generalised predictive control algorithms with guaranted stability. *IEE Proc. D* **1994**, *141*, 154–162.

24. Rossiter, J.A. Input shaping for pfc: How and why? *J. Control Decis.* **2015**, *3*, 105–118. [CrossRef]
25. Mosca, E.; Zhang, J. Stable redesign of predictive control. *Automatica* **1992**, *28*, 1229–1233. [CrossRef]
26. Rossiter, J.A. Predictive functional control: More than one way to pre stabilise. In Proceedings of the 15th Triennial World Congress, Barcelona, Spain, 21–26 July 2002; pp. 289–294.
27. Zabet, K.; Rossiter, J.A.; Haber, R.; Abdullah, M. Pole-placement predictive functional control for under-damped systems with real numbers algebra. *ISA Trans.* **2017**, *71*, 403–414. [CrossRef] [PubMed]
28. Rawlings, J.; Muske, K. The stability of constrained receding horizon control. *IEEE Trans. Autom. Control* **1993**, *38*, 1512–1516. [CrossRef]
29. Scokaert, P.O.; Rawlings, J.B. Constrained linear quadratic regulation. *IEEE Trans. Autom. Control* **1998**, *43*, 1163–1169. [CrossRef]
30. Wang, L. *Model Predictive Control System Design and Implementation Using MATLAB*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009.
31. Aftab, M.S.; Rossiter, J.A. Pre-stabilised predictive functional control for open-loop unstable dynamic systems. In Proceedings of the 7th IFAC Conference on Nonlinear Model Predictive Control, Virtual, 11–14 July 2021.
32. Zhuo, W.; Shichao, W.; Yanyan J. Simulation of control of water level in boiler drum. In Proceedings of the World Automation Congress 2012, Puerto Vallarta, Mexico, 24–28 June 2012.
33. Nise, N.S. *Control System Engineering*; John Wiley & Sons, Inc.: New York, NY, USA, 2011.
34. Kendi, T.A.; Doyle, F.J. III Nonlinear control of a fluidized bed reactor using approximate feedback linearization. *Ind. Eng. Chem. Res.* **1996**, *35*, 746–757. [CrossRef]
35. Ogata, K. *Discrete-Time Control Systems*; Prentice Hall: Englewood Cliffs, NJ, USA, 1995.

Appendix E

Predictive Functional Control for Challenging Dynamic Processes using a Simple Prestabilization Strategy

Muhammad Saleheen Aftab, and John Anthony Rossiter

This paper has been published in Advanced Control for Applications: Engineering and Industrial Systems, Wiley, 2022

Author Contributions. This paper is a collaborative work between both authors. M. S. Aftab proposed the idea, analysed the concept in case studies, and prepared the initial draft of the paper. J .A. Rossiter provided accurate communication of the earlier PFC and MPC control laws, supervised M. S. Aftab and reviewed the whole project.

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:

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

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

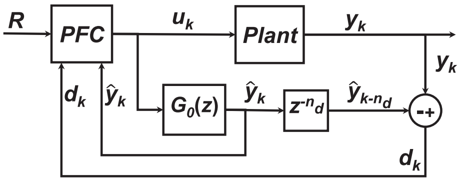


FIGURE 1 The standard predictive functional control control architecture with independent internal model

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]

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

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

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

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

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

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:

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

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.

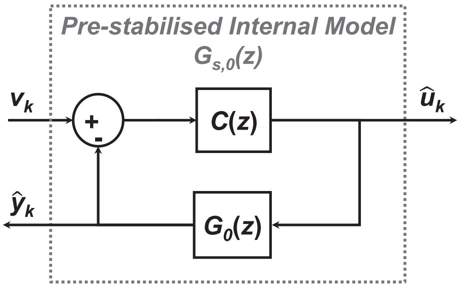


FIGURE 2 Precompensation of internal prediction model $G_0(z)$

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 $p(z) = 1 + p_1z^{-1} + \cdots + p_mz^{-m}$ and $q(z) = q_0 + q_1z^{-1} + \cdots + q_mz^{-m}$. 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 $\beta(z) = \beta_1z^{-1} + \beta_2z^{-2} + \cdots + \beta_lz^{-l}$, $\alpha(z) = \alpha_0 + \alpha_1z^{-1} + \alpha_2z^{-2} + \cdots + \alpha_lz^{-l}$, 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.

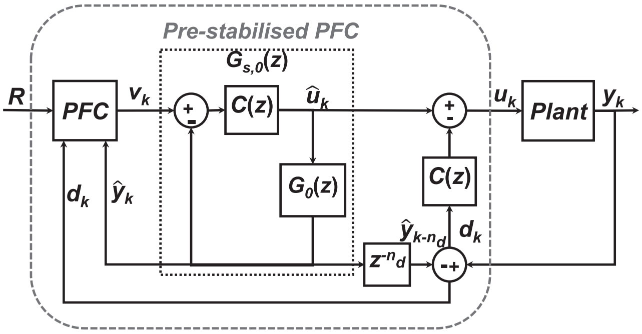


FIGURE 3 Proposed prestabilized predictive functional control control architecture

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:

$$\begin{aligned} A(z) &= \alpha(z)p(z) = 1 + A_1 z^{-1} + A_2 z^{-2} + \cdots \\ B(z) &= q(z)a(z)p(z) = B_0 + B_1 z^{-1} + B_2 z^{-2} + \cdots \\ E(z) &= -\alpha(z)q(z) = E_0 + E_1 z^{-1} + E_2 z^{-2} + \cdots . \end{aligned} \tag{18}$$

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

$$u_k = B_0 v_k + f_k; \quad f_k = -\mathbf{A}\underline{\mathbf{u}}_{\leftarrow k-1} + \mathbf{B}\underline{\mathbf{v}}_{\leftarrow k-1} + \mathbf{E}\underline{\mathbf{d}}_{\leftarrow k}, \tag{19}$$

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:

$$u_{1,n_y} = \frac{B_0 R}{h_{n_y}}(1 - \rho^{n_y}), \tag{20}$$

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:*

$$u_{1,n_y} = \begin{cases} \dfrac{B_0 R}{\beta_1}(1 - \rho); & n_y = 1 \\ \dfrac{B_0 R}{G_{s,0}(1)}; & n_y \to \infty, \end{cases}$$

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:

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

which can be rearranged in the vector form:

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

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

$$u_{1,1} = \frac{B_0 R}{\beta_1}(1-\rho); \quad n_y = 1. \tag{21}$$

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

$$u_{1,\infty} = \frac{B_0 R}{G_{s,0}(1)}; \quad n_y \to \infty. \tag{22}$$

■

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 ρ 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:

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

where $i = 0, 1, \ldots, n_c$ and $\mathbf{L} = [1 \quad 1 \ \ldots]^T$. 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 $\underset{\rightarrow k}{\mathbf{f}}$. 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).

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

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

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

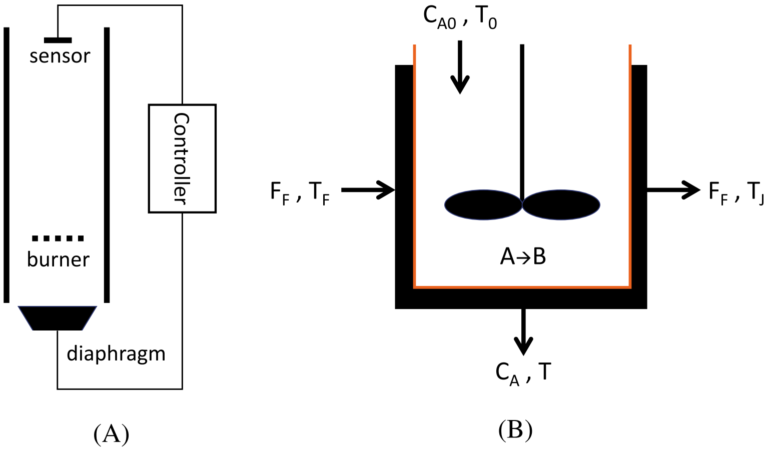


FIGURE 4 (A) Rijke tube apparatus, (B) Jacketed CSTR process

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

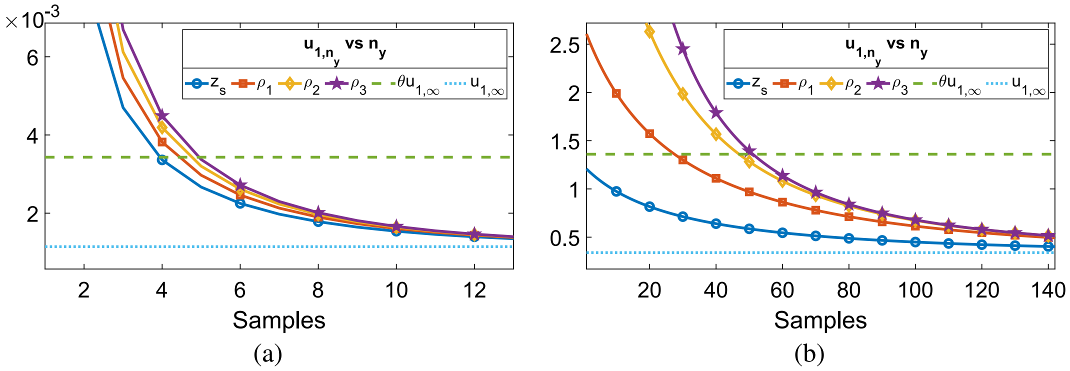


FIGURE 5 Initial input activity for (A) G_1 with $z_s = 0.7715$, $\rho_1 = 0.7215$, $\rho_2 = 0.6715$, $\rho_3 = 0.6215$, $u_{1,\infty} = 0.00124$, $\theta = 3$ and $R = 1$ (B) G_2 with $z_s = 0.9892$, $\rho_1 = 0.9767$, $\rho_2 = 0.9517$, $\rho_3 = 0.9017$, $u_{1,\infty} = 0.3398$, $\theta = 4$ and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

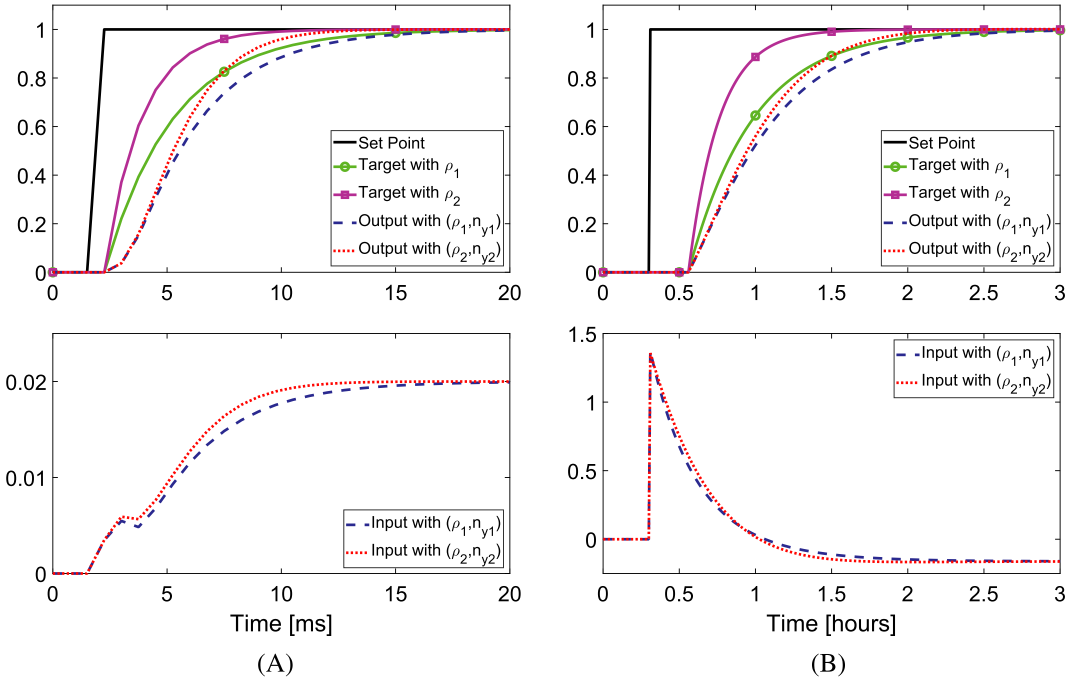


FIGURE 6 Analysis of tuning efficacy with the chosen (ρ, n_y) pairs for (A) G_1 with (0.7215,4) and (0.6715,5), (B) G_2 with (0.9767,27) and (0.9517,47)

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

			$\text{rms}[r_k - y_k]$	$\text{rms}[R - y_k]$
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

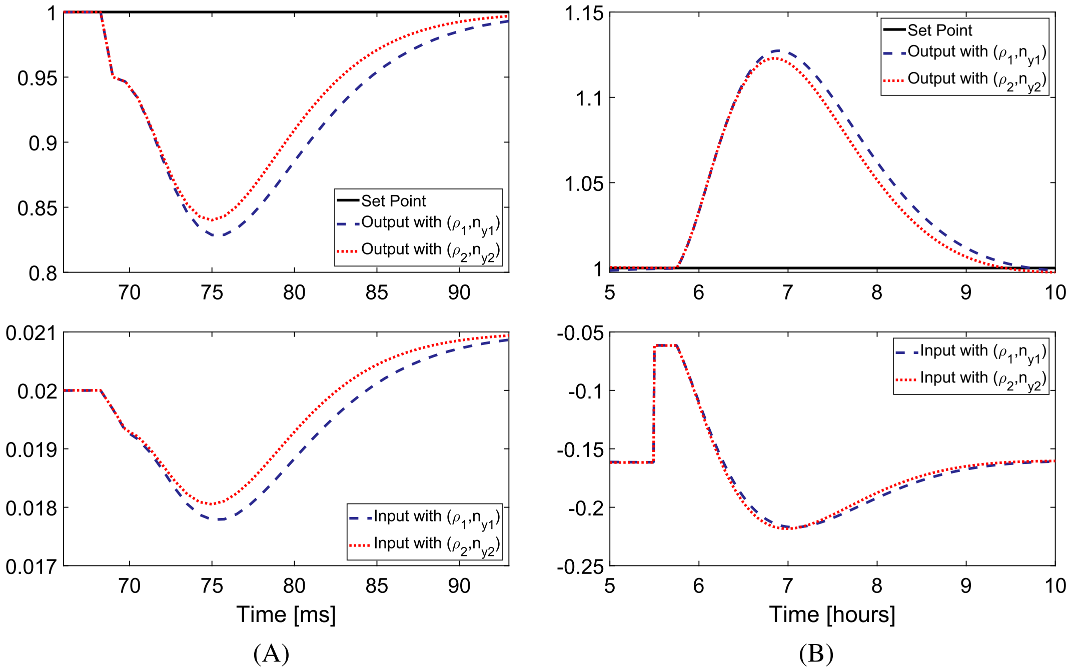


FIGURE 7 Comparison of disturbance rejection with both tuning choices for (A) G_1 with −5% output disturbance, and (B) G_2 with 10% input disturbance

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,

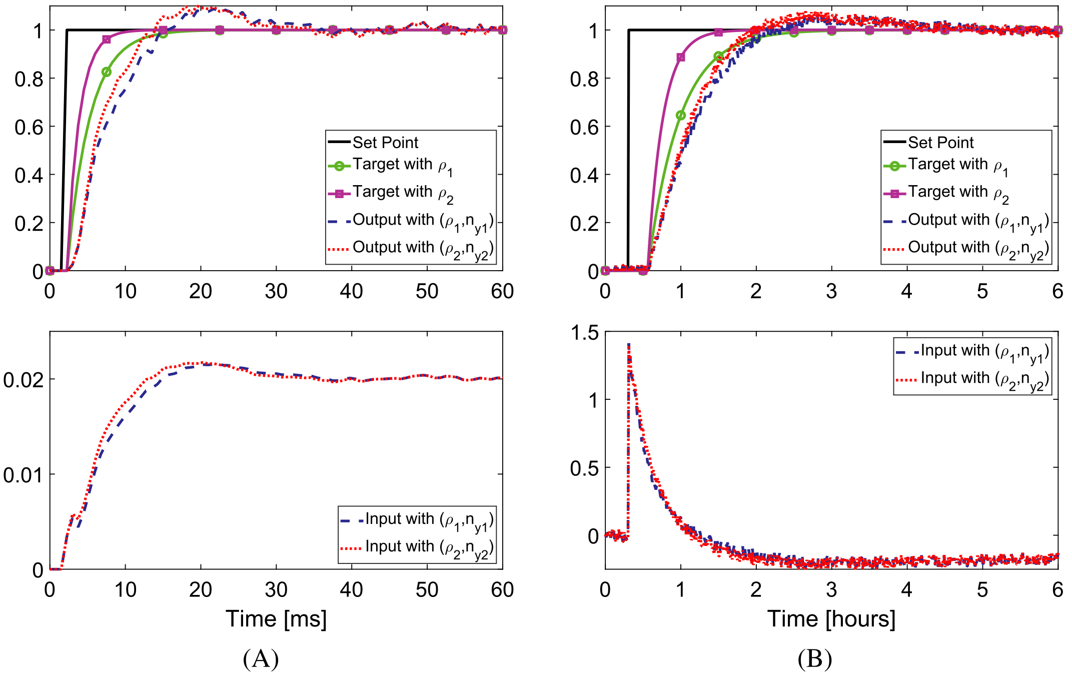


FIGURE 8 Comparison of noise sensitivity (Guassian white measurement noise with $\mu = 0.05$) and modeling mismatches with both tuning choices for (A) G_1 with unmodeled pole at $z = 0.25$, and (B) G_2 with 10% multiplicative uncertainty

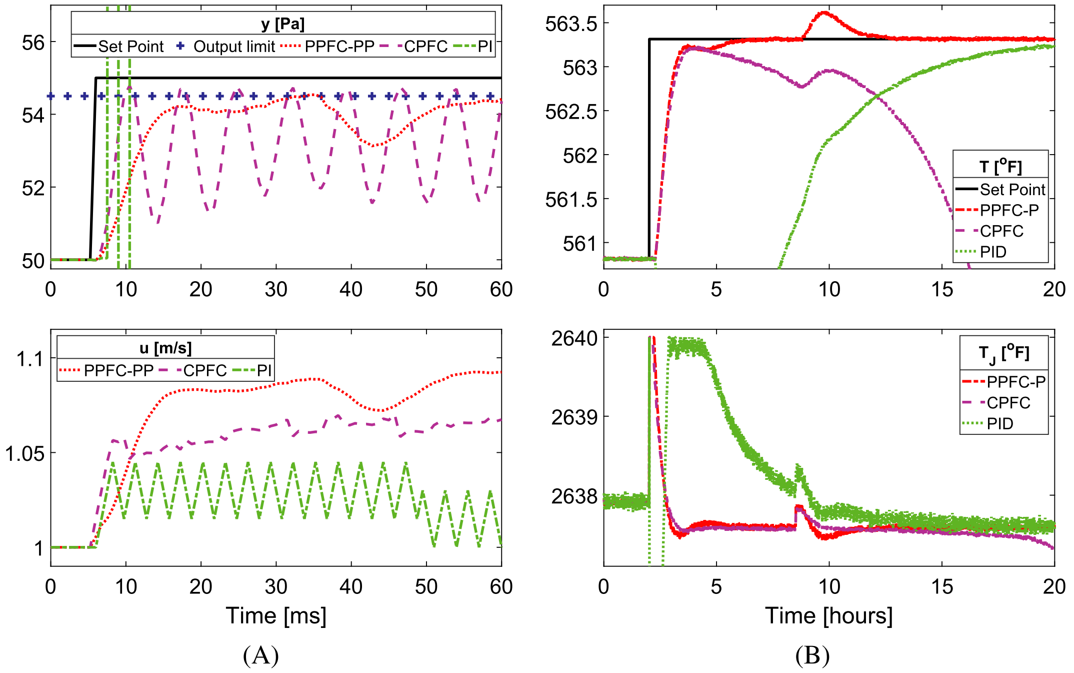


FIGURE 9 Comparison of the constrained closed-loop performance in the presence of external disturbances and measurement noise for (A) the process G_1 subject to $|\Delta u| \leq 0.015$ m/s and $|y| \leq 4.5$ Pa, and (B) the process G_2 subject to $|T_J| \leq 2.1$°F

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

Appendix F

Predictive Functional Control for Difficult Second-Order Dynamics with a Simple Pre-compensation Strategy

Muhammad Saleheen Aftab, John Anthony Rossiter, and George Panoutsos

This paper has been published in the Proceedings of 13th UKACC International Conference on Control (CONTROL 2022), UK, 2022

Author Contributions. This paper is a collaborative work between all authors. M. S. Aftab proposed the idea, analysed the concept in case studies, and prepared the initial draft of the paper. J .A. Rossiter provided accurate communication of the earlier PFC and MPC control laws, supervised M. S. Aftab and reviewed the whole project. G. Panoutsos supervised M. S. Aftab and suggested case studies.

Predictive Functional Control for Difficult Second-Order Dynamics with a Simple Pre-compensation Strategy

Muhammad Saleheen Aftab[1], John Anthony Rossiter[2] and George Panoutsos[3]

Abstract— **Predictive functional control (PFC) is a fairly straightforward model-based technique for controlling stable and monotonically convergent dynamics in a systematic fashion. However, owing to simplified design assumptions, the control performance generally degrades with oscillatory or unstable processes. This paper focuses on pre-stabilising such difficult dynamics, represented as second-order prediction models, before implementing the PFC. In this proposal, the pre-compensator is designed with a root locus method that shifts the undesirable open-loop poles to the stable break-in/breakaway position by varying compensator gain. It has been highlighted that such dynamics transformation enables PFC application in the standard manner by preserving design simplicity and intuitiveness in terms of parameter tuning and constraint handling. Two simulation examples are included to study the pros and cons of the proposal against the conventional PFC algorithm.**

Index Terms—**predictive functional control, root locus, pre-compensation, constraint handling**

I. Introduction

Predictive functional control (PFC), since its introduction in the late 1970's [1], has emerged as the strongest competitor to the widely popular proportional-integral-derivative (PID) algorithm, especially for industrial process control. The advantages of PFC significantly outweigh those of PID in that it systematically handles process dead-times and constraints, which PID cannot without incorporating additional resources such as Smith predictor and anti wind-up techniques [2]. 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 [3], [4].

PFC inherits most design attributes from the mainstream model predictive control (MPC) family [5]. Nevertheless it differs from other predictive control algorithms in the parametrisation of the future input which, in the case of PFC, is assumed as a linear combination of some simple basis functions [3], [6]. A polynomial basis function is usually employed whose order depends upon the characteristics of the set-point trajectory. Thus for a constant set-point, the future input parametrises to just one degree-of-freedom, eliminating the need for the complex optimisation routines generally associated with high-end MPC algorithms. This on one hand simplifies computations, but on the other hand necessitates heuristics to find a sub-optimal solution for the constrained predictive control problems. Unlike mainstream MPC, simple clipping or saturation has been the commonly deployed input constraint management protocol within PFC.

With stable first-order processes, using a constant future input to match the predicted output with the reference trajectory at a single coincidence point is sufficient to achieve any desirable target behaviour, provided the coincidence occurs one time-step ahead in future [4], [5]. Similar results are obtained with well damped higher-order systems although one-step ahead coincidence may not always be appropriate, especially if the prediction dynamics exhibit significant initial lag [7]. The closed-loop performance, however, deteriorates when oscillatory or divergent process dynamics are introduced [5], [7], [8]. This inefficacy relates to the insufficiency of the constant future input assumption alongside a single coincidence point that lacks flexibility to handle such difficult behaviour [9]. Nevertheless, various design modifications have been proposed to handle difficult dynamics with PFC.

One proposal [8] implements input shaping which parameterises the future input so as to cancel the undesirable modes from the model predictions. This modification improves performance, but often results in aggressive control moves, limiting its practicality. A modified input shaping algorithm [9] ensures smooth and less aggressive control action but requires tedious offline computations that negate the core notion of simplicity associated with PFC. Another proposal [10] suggests decomposing the higher-order model into multiple first-order subsystems to benefit from simple tuning procedures. But such decomposition for oscillatory dynamics embeds complex number algebra into the computations which may not work easily with general purpose industrial PLCs [11].

Another alternative is to explicitly pre-stabilise the prediction model with some form of feedback compensation in order to obtain smooth and convergent prediction behaviour [12], [13]. This method is fairly common in mainstream MPC but its application in PFC is generally restricted to pre-stabilising first-order systems with simple proportional gain [14]–[16]. Researchers have pointed out that complex internal feedback compensators may complicate the constraint management process [4], [5].

In this paper, we present a pre-stabilisation technique for challenging dynamic behaviour, represented as open-loop underdamped or unstable second-order models. The proposal implements concepts from root locus theory [17], [18], to shift undesirable open-loop poles to stable break-in/breakaway positions on the root loci, by varying compen-

[1,2,3] All authors are associated with the Department of Automatic Control and Systems Engineering, University of Sheffield, Mappin Street, S1 3JD, UK. Email Addresses: {[1]msaftab1, [2]j.a.rossiter, [3]g.panoutsos}@sheffield.ac.uk

sator gain. Mathematical expressions have been developed that enable pre-compensator design without requiring to plot and analyse root locus paths. Furthermore, it has been shown that the proposal keeps overall design simple and intuitive to benefit from the standard PFC tuning and constraint handling procedures.

The remainder of this paper is organised as follows: Section II formulates the problem and sets control objectives. The main methodology is presented in Sections III & IV where the pre-compensator and PFC designs are discussed in detail. Numerical studies follow next in Section V which discuss closed-loop performance and draw comparisons against the standard PFC. Finally the paper concludes in Section VI.

II. Problem Statement

Consider a difficult real world process characterised by a strictly proper second-order transfer function model

$$G(z) = \frac{b(z)}{a(z)} \tag{1}$$

where $a(z) = 1 + a_1 z^{-1} + a_2 z^{-2}$ and $b(z) = b_1 z^{-1} + b_2 z^{-2}$. It is assumed that the open-loop model shows oscillatory or divergent dynamic behaviour i.e. the open-loop pole polynomial $a(z)$ either comprises a complex conjugate pole pair, see for example Fig. 1(a), or has at least one unstable mode, as shown in Figs. 1(b)-(c).

The problem addressed in this paper deals with designing PFC for the process modelled as $G(z)$. However, as stated earlier, conventional PFC may be less effective with such challenging dynamics. Therefore, the aim is first to stabilise model predictions with a simple internal feedback loop. Furthermore, the controller is expected to exhibit some degree of robustness against parametric uncertainty and/or unmodelled dynamics.

III. Pre-compensator Design

The primary objective of pre-compensation is to transform the undesirable open-loop dynamics into stable and, if possible, monotonically convergent prediction behaviour for straightforward implementation within a PFC framework. The proposed pre-compensation process, illustrated in Fig. 2, employs a simple feedback controller $C(z)$ to stabilise $G(z)$. The controller has the form $C(z) = KC_{in}(z)$, where K is the proportional gain with $K \in (-\infty, +\infty)$ and $C_{in}(z) = 1$ by default but may be designed as a lead or lag compensator if necessary. The internal feedback results in the following pre-compensated model:

$$T(z) = \frac{G(z)}{1 + KC_{in}(z)G(z)} \tag{2}$$

Next we present the design of the pre-compensator based on a root locus technique.

A. Preliminary Design via Root Locus

Root locus is a powerful graphical tool for control systems analysis and design [17]. It is generally used for the assessment of a system's closed-loop performance and relative stability as a function of various parameters, such as system gain and time constant. Here we wish to analyse the effect of varying K on the pre-compensated pole polynomial. Ultimately the goal is to identify such values of K that result in critically damped poles. Note that critical damping in root loci may only occur at the stable break-in/breakaway points.

A point on root locus curve where two poles exit the real axis and diverge to become a complex conjugate pair is known as the breakaway point. Conversely a point on the real axis at which a complex pole pair converges is called the break-in point. Intuitively one may obtain monotonically convergent predictions just by designing K at the break-in/breakaway point, provided it occurs within the stable range $0 < z < 1$. However, the open-loop zero dynamics can be significant in some cases and hence it may not always be possible to achieve a break-in/breakaway at acceptable locations. Let us examine the possible cases in detail.

Underdamped Poles. Consider the case of open-loop underdamped poles as shown in Fig. 1(a). Evidently there are three distinct regions for the system's zero location. It has been found that the poles would break-in within $0 < z < 1$ as long as the zero is either located in R_1 or R_3. However, a zero in R_2 i.e. in the vicinity of the open-loop poles may cause a break-in at $z < 0$. To solve this problem, we propose using a lag type $C_{in}(z)$ as follows:

$$C_{in}(z) = \frac{z + z_n}{z + z_0} \tag{3}$$

where $z_0 = -b_2/b_1$ is the open-loop zero whereas z_n is the new zero deliberately placed in R_1 (within the unit circle) away from the open-loop poles i.e. $z_n < z_0$. Note that this method may not work if the open-loop poles appear in the left half section of the unit circle.

One Unstable Pole. In this case, breakaway can only take place within $0 < z < 1$ if z_0 lies in either R_1 away from the stable pole p_1 or in R_4 away from the unstable pole p_2, see Fig. 1(b). However, a zero in the vicinity of either poles (while remaining in R_1 or R_4) might result in a breakaway outside the desirable range. Moreover, if z_0 lies in either R_2 or R_3 the model cannot be stabilised with simple gain K. A similar procedure as described above with (3) may be employed but only if the open-loop zero is stable i.e. located in R_2.

Two Unstable Poles. With two unstable poles, as shown in Fig. 1(c), the only possibility to get a break-in within the desirable range $0 < z < 1$ is to have the open-loop zero z_0 located within R_2 near the stability boundary. If z_0 is in R_3 then it may or may not be possible to stabilise the model. In all other scenarios, this method would fail to stabilise prediction dynamics. However, if z_0 is located in the stable portion of R_1, then it is possible to employ a lead-type $C_{in}(z)$ similar to (3) but with $z_n > z_0$ in order to replace the open-loop zero with the new one placed in R_2 near $z = 1$.

B. Design Procedure

It should be obvious from the preceding discussion that the efficacy of the preliminary design is strongly linked to the

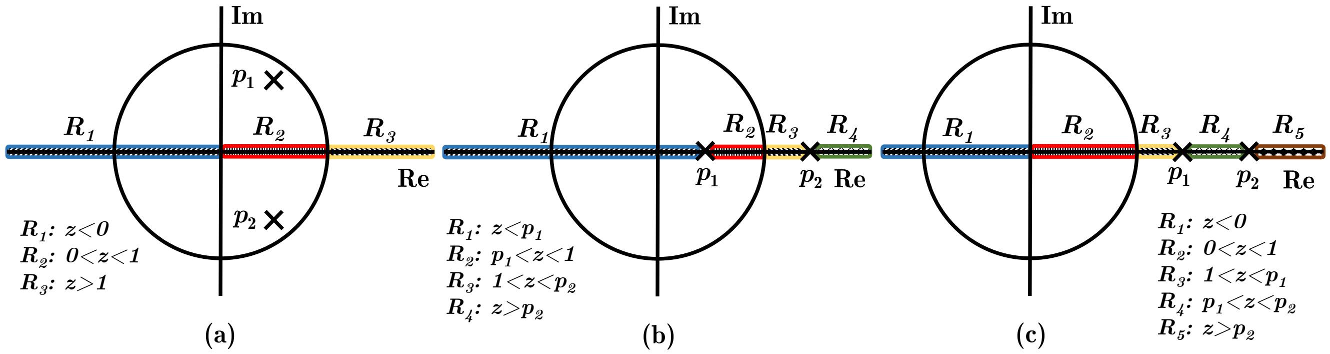


Fig. 1. Pole-zero map of $G(z)$ with (a) complex pole pair, (b) one unstable pole, and (c) two unstable poles; R_i's represent the possible zero positions.

actual pole-zero mapping of $G(z)$. Nevertheless, the concept can be generalised mathematically for any second-order pole-zero configuration as illustrated in Fig. 1. With $C_{in}(z) = 1$, the compensated pole polynomial $1 + KG(z) = 0$ implies:

$$K = -\frac{1}{G(z)} = -\frac{a(z)}{b(z)} \tag{4}$$

The break-in/breakaway points are stationary in nature, thus in order to find them K is differentiated with respect to z and equated to '0' [17]:

$$\left.\frac{dK}{dz}\right|_{z=\sigma} = -\frac{d}{dz}\left[\frac{a(z)}{b(z)}\right]_{z=\sigma} = 0$$

where σ represents the point(s) at which break-in/breakaway occurs. This implies:

$$\begin{aligned} & b(\sigma)a'(\sigma) - b'(\sigma)a(\sigma) = 0 \\ \Longrightarrow\ & (b_1\sigma + b_2)(2\sigma + a_1) - (b_1)(\sigma^2 + a_1\sigma + a_2) = 0 \\ \Longrightarrow\ & b_1\sigma^2 + 2b_2\sigma + (a_1b_2 - a_2b_1) = 0. \end{aligned} \tag{5}$$

Lemma 1: For a given pole polynomial $a(z)$, σ is a non-linear function of the open-loop zero z_0.

Proof: Equation (5) is in the standard quadratic form and can be solved analytically:

$$\begin{aligned} \sigma_{1,2} &= -\frac{b_2}{b_1} \pm \frac{1}{b_1}\sqrt{a_2{b_1}^2 - a_1b_1b_2 + {b_2}^2} \\ &= \left(-\frac{b_2}{b_1}\right) \pm \sqrt{a_2 + a_1\left(-\frac{b_2}{b_1}\right) + \left(-\frac{b_2}{b_1}\right)^2} \\ \Longrightarrow\ \sigma_{1,2} &= z_0 \pm \sqrt{a_2 + a_1z_0 + {z_0}^2}. \end{aligned} \tag{6}$$

Hence for a given $a(z)$, σ is a non-linear function of z_0. ■

Remark 1: When there is no finite zero i.e. $b_1 = 0$, (5) reduces to $\sigma = -0.5a_1$. Thus there can only be either a break-in or breakaway point (depending on the poles) but not both.

If any of the σ_i's from (6) lies within $0 < z < 1$, then one may find the corresponding gain value K using:

$$K = -\frac{a(\sigma_i)}{b(\sigma_i)}; \quad i = 1 \text{ or } 2. \tag{7}$$

This completes the pre-compensator design. But what happens when none of the σ_i's is present within the right half of the unit circle? In this case, one may first check if employing a lead or lag type compensator such as (3) would suffice. Theorem 1 establishes conditions on the usability of $C_{in}(z)$ in such cases.

Theorem 1: The prediction model $G(z)$ can be stabilised with a lead or lag type compensator $C_{in}(z)$, such as (3), if in addition to $|z_0| < 1$,

$$|\sigma_d^2 - a_2| < |2\sigma_d + a_1|$$

where σ_d is the desired break-in/breakaway point such that $0 < \sigma_d < 1$.

Proof: Evidently the purpose of lead/lag compensation here is to replace the open-loop zero z_0 with the new zero z_n at a desirable location within the unit circle. Thus for guaranteed internal stability, $C_{in}(z)$ can only be designed if the old zero z_0 and the new zero z_n both lie inside the unit circle i.e. $|z| < 1$. Subsequently one may re-write (6) as follows:

$$\begin{aligned} \sigma_d &= z_n \pm \sqrt{a_2 + a_1z_n + z_n^2} \\ \Longrightarrow\ (\sigma_d - z_n)^2 &= \left(\pm\sqrt{a_2 + a_1z_n + z_n^2}\right)^2 \end{aligned}$$

which after a few simple manipulations becomes,

$$z_n = \frac{\sigma_d^2 - a_2}{2\sigma_d + a_1}. \tag{8}$$

Since $|z_n| < 1$, this means

$$|\sigma_d^2 - a_2| < |2\sigma_d + a_1| \tag{9}$$

for $0 < \sigma_d < 1$. ■

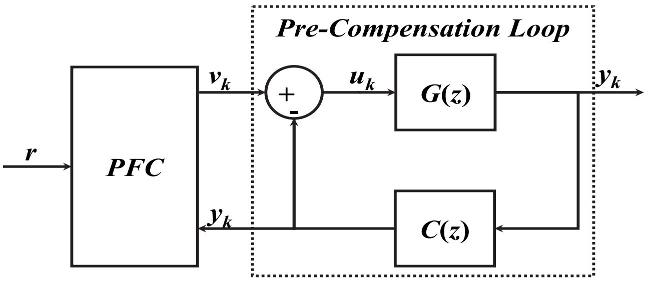


Fig. 2. Pre-conditioned PFC (PPFC) with internal feedback loop.

Equation (8) can validate whether designing $C_{in}(z)$ as lead or lag type compensator would be worthwhile. One may plot z_n as a function of σ_d to find a suitable zero that enforces break-in/breakaway within $0 < z < 1$ (see Fig. 5 for example). If such z_n exists, then K can be evaluated from (4) by replacing $b(z)$ with the new compensated polynomial $\beta(z) = b_1 z^{-1} + b_1 z_n z^{-2}$. This will give the following pre-compensated model.

$$T(z) = \frac{\beta(z)}{\alpha(z)} = \frac{b_1 z^{-1} + b_1 z_n z^{-2}}{1 - 2\sigma_d z^{-1} + \sigma_d^2 z^{-2}} \qquad (10)$$

Remark 2: It should be emphasised that pre-compensation only stabilises the prediction model, whereas attributes such as transient performance, offset-free tracking, dead-time and constraints are managed by the outer PFC loop, as shown in Fig. 2.

IV. Design of Pre-Compensated PFC

The Pre-compensated PFC (PPFC) algorithm, similar to the original PFC [5], attempts to match the predicted response with an ideal (first-order) behaviour at the single coincidence point n_y with constant control moves. This process is repeated at each time step and owing to receding horizon, a virtual feedback is established that moves the plant output closer to the target. This convergence depends upon the desired behaviour and can be implemented as a first-order pole ρ. Assume that the ideal n_y-step ahead prediction based on first-order response is given as:

$$y_{k+n_y|k} = r - (r - y_k)\rho^{n_y} \qquad (11)$$

where r is the constant set-point and y_k is the measured plant output. The n_y steps ahead output predictions are derived from the pre-stabilised model $T(z)$ such that:

$$\hat{y}_{k+n_y|k} = H \underset{\rightarrow}{v}_k + P \underset{\leftarrow}{v}_{k-1} + Q \underset{\leftarrow}{\hat{y}}_k \qquad (12)$$

where H, P and Q depend on the model parameters. For a generic N^{th} order model:

$$\underset{\rightarrow}{v}_k = \begin{bmatrix} v_k \\ v_{k+1} \\ \vdots \\ v_{k+n_y-1} \end{bmatrix}; \underset{\leftarrow}{v}_{k-1} = \begin{bmatrix} v_{k-1} \\ v_{k-2} \\ \vdots \\ v_{k-N+1} \end{bmatrix}; \underset{\leftarrow}{\hat{y}}_k = \begin{bmatrix} \hat{y}_k \\ \hat{y}_{k-1} \\ \vdots \\ \hat{y}_{k-N+1} \end{bmatrix}$$

With constant control values throughout the coincidence horizon i.e. $v_{k+i} = v_k$, $\forall i > 0$, we obtain the PPFC control law from (11)-(12) as follows:

$$v_k = \frac{r - (r - y_k)\rho^{n_y} - (P \underset{\leftarrow}{v}_{k-1} + Q \underset{\leftarrow}{\hat{y}}_k)}{h} \qquad (13)$$

where $h = \sum_{j=1}^{n_y} H_j$ and H_j is the j^{th} element of H.

Remark 3: For clarity of presentation, the PPFC control law derived above does not include algebra relevant to offset-free tracking and dead-times. The numerical examples, nonetheless, include these details. See [5] for further information.

Since v_k is the input to $T(z)$, we must also determine u_k for plant actuation and constraint management. Theorem 2 establishes the relationship between v_k and u_k.

Theorem 2: The PPFC control input v_k and the plant input u_k are related as follows:

$$u_k = \hat{a}\underset{\leftarrow}{v_k} + \hat{\alpha}\underset{\leftarrow}{u}_{k-1}$$

where vectors $\hat{a}$ and $\hat{\alpha}$ contain the suitable coefficients of $a(z)$ and $\alpha(z)$ respectively.

Proof: With reference to Fig. 2, $u_k = v_k - C(z)y_k$ and since $y_k = G(z)u_k$ and $C(z) = KC_{in}(z)$, we get:

$$u_k = v_k - KC_{in}(z)G(z)u_k$$

which implies,

$$[1 + KC_{in}(z)G(z)]\, u_k = v_k$$

But $1 + KC_{in}(z)G(z) = \alpha(z)/a(z)$. Therefore $\alpha(z)u_k = a(z)v_k$ implies:

$$u_k = a(z)v_k + 2\sigma_d z^{-1}u_k - \sigma_d^2 z^{-2}u_k$$

Or equivalently in the time-domain:

$$u_k = \hat{a}\underset{\leftarrow}{v_k} + \hat{\alpha}\underset{\leftarrow}{u}_{k-1} \qquad (14)$$

where $\hat{a} = [1 \quad a_1 \quad a_2]$ and $\hat{\alpha} = [2\sigma_d \quad -\sigma_d^2]$. ■

Remark 4: Constraint handling is one of the key features of conventional PFC, and the techniques to do so are well established in literature [4], [5]. While it is obvious that constraint management after pre-compensation is slightly more expensive, the associated algebra and coding are still fairly benign, see for instance [19], [20], where the impact of pre-stabilisation on constraint validation is discussed in detail. In this paper, we will employ these results directly in the simulation examples.

Remark 5: Pre-stabilisation helps selecting the coincidence point n_y for difficult dynamics in a straightforward manner, based on the conjecture presented in [7]. As per the recommendation, n_y lies within the time range when the pre-stabilised step response rises from 40% to 80% with significant gradient. As for finding ρ, one may overlay several first-order responses on the step response to identify which target behaviour coincides within the mentioned n_y range. See, for instance, Fig. 3.

V. Numerical Examples

This section investigates the efficacy of the proposal with two numerical examples. Example 1 illustrates the case of an oscillatory higher-order process which is modelled as a second-order underdamped system for the Pre-compensated PFC implementation. Example 2, on the other hand, demonstrates the PPFC design for a second-order unstable system when a simple proportional gain alone is ineffective due to dominant open-loop zero dynamics. Details follow next.

A. Example 1

Consider an underdamped process,

$$G_1 = \frac{0.065z^{-1} + 0.26z^{-2}}{1 - 1.35z^{-2} + 1.158z^{-2} - 0.28z^{-3}} \cdot z^{-5} \qquad (15)$$

with an open-loop zero $z_0 = -4$, a real pole at $z = 0.35$, a complex conjugate pole pair $p_{1,2} = 0.5 \pm j0.742$ and a

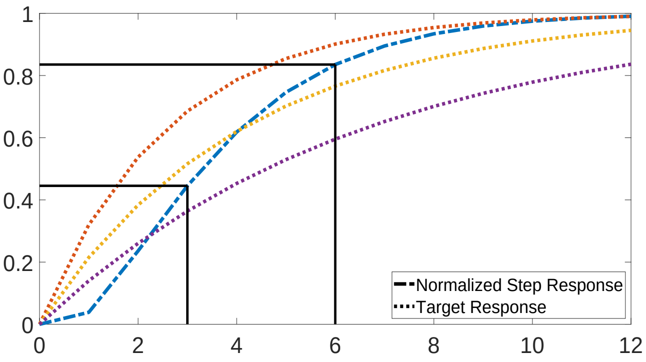


Fig. 3. Target responses with $\rho = [0.68, 0.79, 0.86]$ overlaying the normalised step response of T_1.

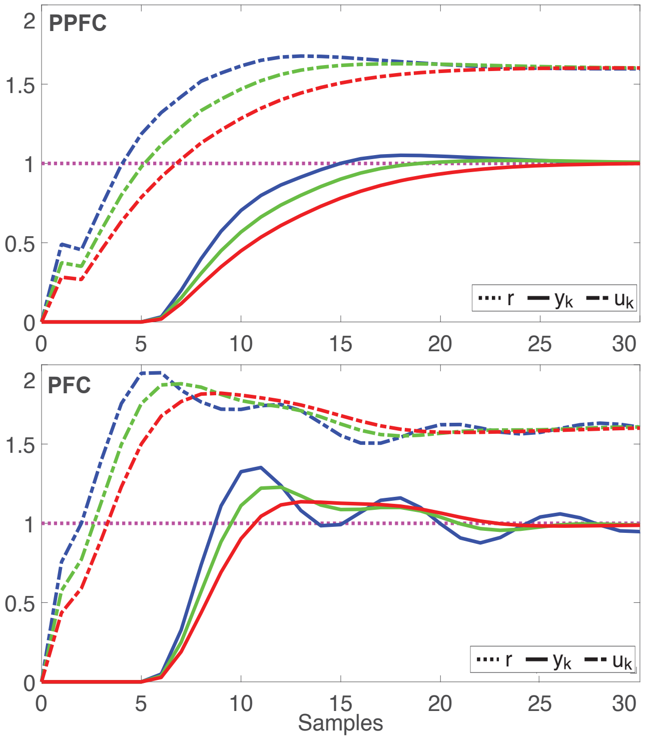


Fig. 4. PPFC vs. PFC with unmodelled pole at $z = 0.35$ for model G_1, $n_y = 4$ and $\rho = [0.68(blue), 0.8(green), 0.86(red)]$.

dead-time of $m = 5$ samples. To apply the proposed pre-compensation, only the dominant second-order dynamics are considered after neglecting the non-dominant pole at $z = 0.35$ from the prediction model. Nevertheless, it is included in the plant simulation to analyse robustness.

The break-in/breakaway point calculated from (6) suggests $\sigma_d = 0.561$ with the corresponding $K = -1.214$. In this example, lag or lead type compensation is not needed i.e. $C_{in}(z) = 1$ and thus $C(z) = -1.214$. This gives the following delay-free second-order pre-compensated model:

$$T_1 = \frac{0.1z^{-1} + 0.4z^{-2}}{1 - 1.121z^{-1} + 0.314z^{-2}} \tag{16}$$

Next we determine the appropriate n_y and ρ by plotting the normalised step response of T_1 overlaying several desired first-order responses with differing ρ's, as shown in Fig. 3. The plot suggests $3 \leq n_y \leq 6$ as a suitable coincidence horizon window. Note that target dynamics with $\rho = 0.68$ or $\rho = 0.86$ do not match predicted behaviour within the desirable n_y range and hence would need over-actuation or under-actuation to enforce an intercept. However, a sensible choice would be $\rho = 0.79$ which gets an exact match at $n_y = 4$.

Efficacy of the PPFC algorithm is obvious with the closed-loop performance shown in Fig. 4. Even in the presence of the unmodelled pole ($z = 0.35$), the PPFC plant output (upper figure) is smooth and oscillation-free, and strongly linked to the corresponding ρ, although control input for faster target dynamics is relatively aggressive as expected. The conventional PFC (lower figure) appears ineffective in damping oscillations even with fairly aggressive control moves in the transient region. In practice, this may lead to actuator saturation resulting in unacceptable control performance.

B. *Example 2*

Consider an open-loop unstable system,

$$G_2 = \frac{1.5z^{-1} - 1.2z^{-2}}{1 - 1.5z^{-1} + 0.44z^{-2}}; \quad |u_k| \leq 0.205 \tag{17}$$

with $z_0 = 0.8$ located between the open-loop poles at $p_1 = 0.4$ and $p_2 = 1.1$. Clearly no break-in/breakaway is possible with simple proportional compensation and therefore $C_{in}(z)$ must be designed to stabilise the model. Fig. 5 plots the new zero z_n as a function of σ_d for G_2. It is evident that a break-in/breakaway within the right half of unit circle can be enforced with a lag-type $C_{in}(z)$. Notice that faster pre-compensated dynamics ($\sigma_d \leq 0.4$) may be obtained with $z_n \approx 0.3$. However, such dynamics may not be appropriate owing to aggressive control action, potentially causing constraint violation. Therefore, we select $z_n = 0$ for $\sigma_d = 0.6633$. The corresponding gain value is then found as $K = 0.1155$. Thus $C(z) = 0.1155z/(z - 0.8)$ and the pre-compensated model is:

$$T_2 = \frac{1.5z^{-1} - 1.2z^{-2}}{1 - 1.327z^{-1} + 0.44z^{-2}}. \tag{18}$$

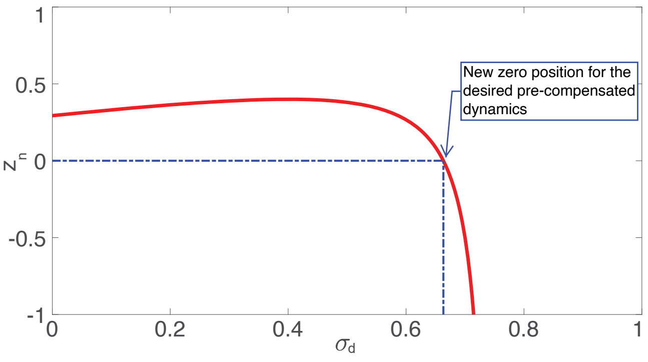


Fig. 5. Plot of z_n ($|z_n| < 1$) vs σ_d ($0 < \sigma_d < 1$) for G_2; $z_n = 0$ with $\sigma_d = 0.6633$ selected for pre-compensation.

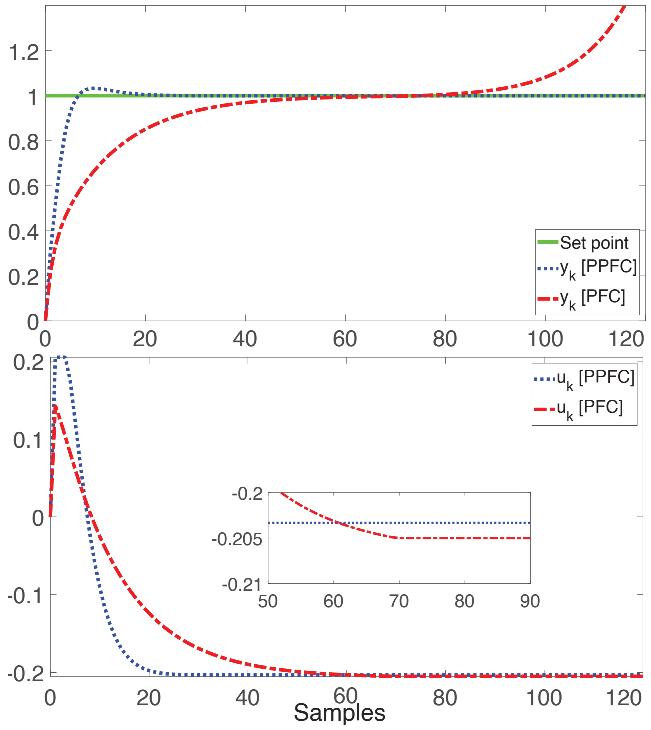


Fig. 6. Closed-loop performance comparison with PPFC ($blue$) and PFC (red) for G_2 including parametric uncertainty (no uncertainty in unstable pole [21]); $n_y = 5$ and $\rho = 0.85$.

Next $n_y = 5$ and $\rho = 0.85$ are obtained with a similar procedure as described in Example 1. The closed-loop performance is compared and contrasted for both PPFC and conventional PFC algorithms in Fig. 6, with deliberately added parametric uncertainty (assuming no uncertainty in the unstable pole [21]). Due to unreliable open-loop divergent predictions, the closed-loop output with PFC destabilises around the 80^{th} sample. Coincidently, the corresponding control input also saturates at $u_{min} = -0.205$ due to constraint violation around that time. On the other hand, the proposed PPFC algorithm keeps the system output smooth and stable while maintaining robustness against uncertainty and without violating input constraints.

VI. Conclusion

A root locus based pre-stabilisation strategy for predictive functional control of difficult dynamic processes is presented. The proposal is fairly generic and based on the fact that a majority of real-world processes can be adequately represented as dominant second-order dynamics for which simple tailored solutions are well understood. The main idea is to form smooth and well-damped predictions with an internal feedback compensator designed to enforce break-in/breakaway at the desired closed-loop poles. We have shown that a simple proportional gain is generally sufficient, however, a lead or lag type compensator may also be needed with some challenging pole-zero configurations. Moreover the proposed PPFC design preserves the inherent simplicity and intuitiveness of the original PFC, including the standard parameter tuning and constraint management procedures.

The proposal has shown promising results for both oscillatory and unstable processes in the presence of uncertainty. Nevertheless, future work will focus on more formal analysis of the closed-loop characteristics against disturbances, sensor noise and modelling errors, along with an analysis of efficacy in real world industrial applications.

References

[1] J. Richalet, A. Rault, J. Testud, and J. Papon, "Model predictive heuristic control," *Automatica (Journal of IFAC)*, vol. 14, no. 5, pp. 413–428, 1978.

[2] 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.

[3] 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.

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

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

[6] M. Khadir and J. Ringwood, "Extension of first order predictive functional controllers to handle higher order internal models," *International Journal of Applied Mathematics and Computer Science*, vol. 18, no. 2, pp. 229–239, jun 2008.

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

[9] 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.

[10] J. Rossiter, R. Haber, and K. Zabet, "Pole-placement predictive functional control for over-damped systems with real poles," *ISA transactions*, vol. 61, pp. 229–239, 2016.

[11] K. Zabet, J. Rossiter, R. Haber, and M. Abdullah, "Pole-placement predictive functional control for under-damped systems with real numbers algebra," *ISA transactions*, vol. 71, pp. 403–414, 2017.

[12] 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.

[13] 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.

[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] M. Abdullah and J. A. Rossiter, "Alternative method for predictive functional control to handle an integrating process," in *2018 UKACC 12th International Conference on Control (CONTROL)*. IEEE, 2018, pp. 26–31.

[16] Z. Zhang, J. Rossiter, L. Xie, and H. Su, "Predictive functional control for integral systems," in *International Symposium on Process System Engineering*, 2018.

[17] K. Ogata *et al.*, *Discrete-time control systems*. Prentice Hall Englewood Cliffs, NJ, 1995, vol. 2.

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

[19] 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.

[20] 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.

[21] S. Skogestad and I. Postlethwaite, *Multivariable feedback control: analysis and design*. Wiley New York, 2007, vol. 2.

Appendix G

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

Muhammad Saleheen Aftab, John Anthony Rossiter, and George Panoutsos

This paper has been published in the Proceedings of 13th UKACC International Conference on Control (CONTROL 2022), UK, 2022

Author Contributions. This paper is a collaborative work between all authors. M. S. Aftab proposed the idea, analysed the concept in case studies, and prepared the initial draft of the paper. J .A. Rossiter provided accurate communication of the earlier PFC and MPC control laws, supervised M. S. Aftab and reviewed the whole project. G. Panoutsos co-supervised M. S. Aftab and suggested possible case study.

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

Muhammad Saleheen Aftab[1], John Anthony Rossiter[2] and George Panoutsos[3]
Department of Automatic Control and Systems Engineering
University of Sheffield
Mappin Street, S1 3JD, Sheffield, UK
[1]msaftab1@sheffield.ac.uk, [2]j.a.rossiter@sheffield.ac.uk, [3]g.panoutsos@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]:

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

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:

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

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:

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

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 $u_k = u_{k+1} = \dots = u_{k+ny}$, 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):

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

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]:*

$$|u_k| > U \Rightarrow |u_k| = U, \quad |\Delta u_k| > D_U \Rightarrow |\Delta u_k| = D_U \tag{5}$$

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).

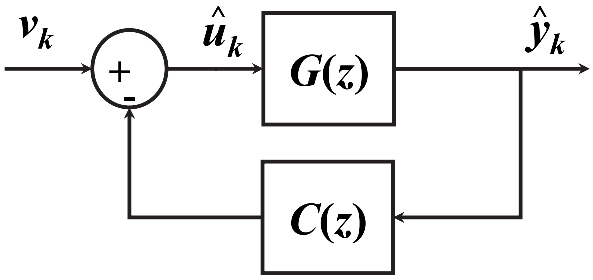


Fig. 1. Pre-stabilisation loop structure.

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.

A. Concept of Pre-stabilisation

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

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

where $a(z) = 1 + a_1 z^{-1} + \dots + a_n z^{-n}$, $b(z) = b_1 z^{-1} + \dots + b_n z^{-n}$ 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:

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

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

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

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:

$$y_{k+n_y|k} = \mathbf{H}\underrightarrow{\mathbf{v}}_k + \mathbf{P}\underleftarrow{\mathbf{v}}_{k-1} + \mathbf{Q}\underleftarrow{\hat{\mathbf{y}}}_k + d_k \quad (10)$$

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 $G_s(z)$:

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

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:

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

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:

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

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:*

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

Proof. With d_k, $\underleftarrow{\mathbf{v}}_{k-1}$ and $\underleftarrow{\hat{\mathbf{y}}}_k$ 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:*

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

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,

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

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 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.

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

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.

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.

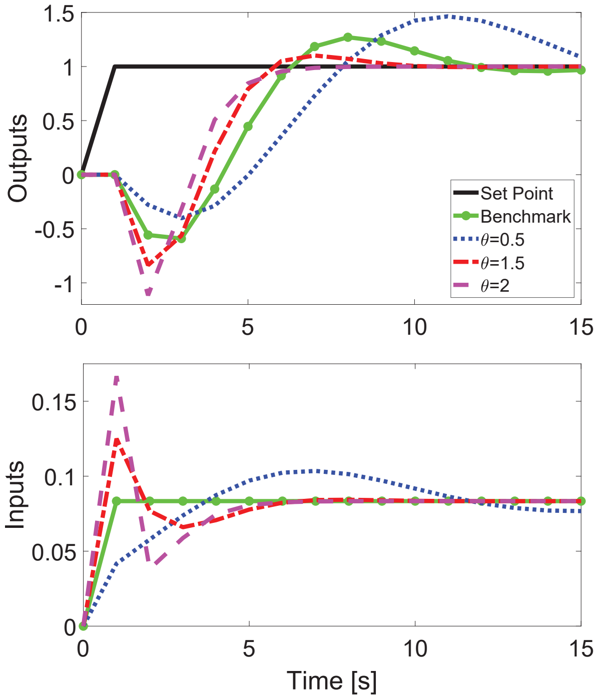


Fig. 2. Tuning efficacy of RPFC for open-loop G_1 in nominal conditions.

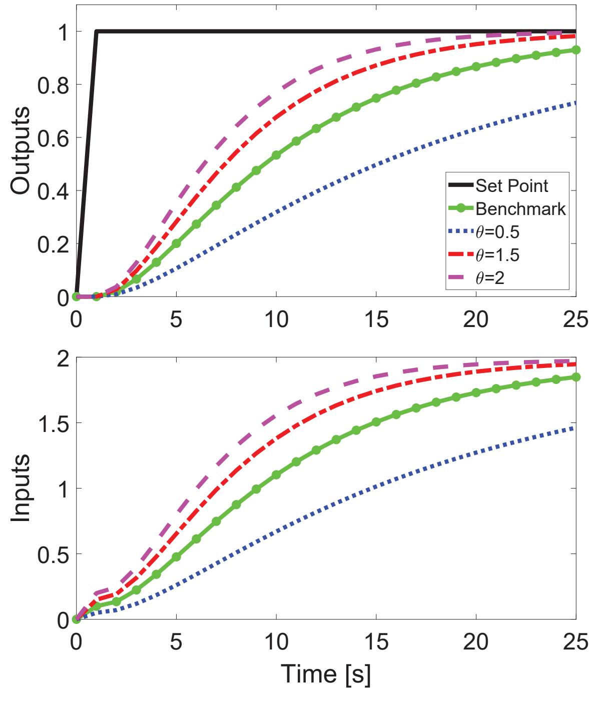


Fig. 3. Tuning efficacy of RPFC for pre-stabilised G_2 in nominal conditions.

A simple proportional compensator $C_2 = -1.88$ suffi-

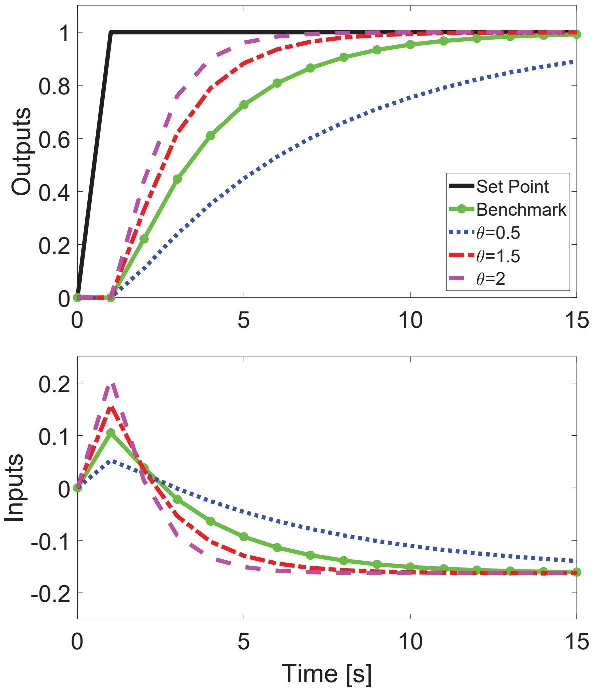


Fig. 4. Tuning efficacy of RPFC for pre-stabilised G_3 in nominal conditions.

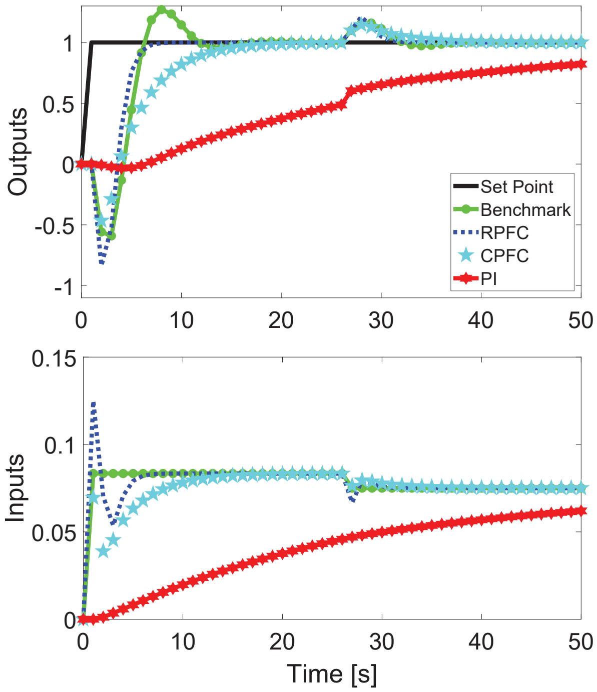


Fig. 5. Comparison of closed-loop performance for G_1 subject to $|\Delta u_k| \leq 0.125$ and 10% output disturbance introduced at 25th second between RPFC ($\theta = 2$), CPFC ($\rho = 0.75$, $n_y = 5$) and PI ($K_p = 0.0012$, $K_i = 0.0023$).

ciently filters out the unwanted oscillations in the open-loop step response of G_2, providing the pre-stabilised prediction model $G_{s,2} = \dfrac{0.19z + 0.18}{z^2 - 1.58z + 0.61}$ with over-damped 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 $2u_{ss}$ 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

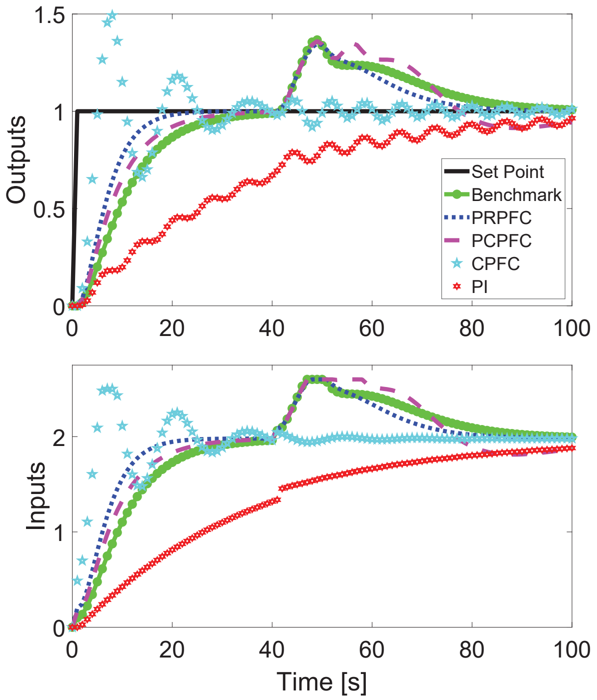


Fig. 6. Comparison of closed-loop performance for G_2 subject to $|u_k| \leq 2.5$ and 10% input disturbance introduced at 40th second between PRPFC ($\theta = 2$), PCPFC/CPFC ($\rho = 0.86$, $n_y = 4$) and PI ($K_p = 0.028$, $K_i = 0.055$).

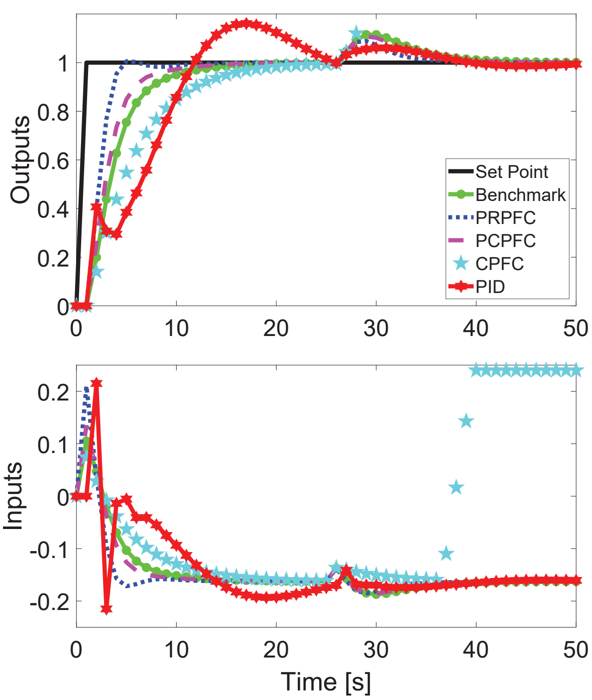


Fig. 7. Comparison of closed-loop performance for G_3 subject to $|u_k| \leq 0.21$ with unmodelled pole at $z = 0.1$ and 15% input disturbance introduced at 25th second between PRPFC ($\theta = 2$), PCPFC/CPFC ($\rho = 0.6$, $n_y = 3$) and PID ($K_p = 0.321$, $K_i = 0.038$, $K_d = 0.323$).

pre-conditioning of difficult open-loop systems, for instance, using 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

Appendix H

Exploiting Laguerre Polynomials and Steady-State Estimates to Facilitate Tuning of PFC

John Anthony Rossiter, Muhammad Saleheen Aftab, and George Panoutsos

This paper has been accepted for publication in the Proceedings of 20th European Control Conference (ECC), UK, 2022

Author Contributions. This paper is a collaborative work between all authors. J. A. Rossiter provided initial proposals and accurate communication of the concepts. M. S. Aftab analysed the concepts in case studies and reviewed/edited the draft. G. Panoutsos co-supervised M. S. Aftab and suggested possible case study.

Exploiting Laguerre polynomials and steady-state estimates to facilitate tuning of PFC

John Anthony Rossiter[1], Muhammad Saleheen Aftab[1] and George Panoutsos[1]

Abstract— **Predictive Functional Control (PFC), a simplified and low-cost MPC algorithm, has gained considerable attention for industrial process control in the last two decades. Although with PFC, controller tuning is relatively simple and more meaningful than a PID controller, its efficacy is poorer for larger prediction horizons–a necessity for stable over-damped and non-minimum phase dynamics. This paper proposes a conceptually novel tuning mechanism based on a single choice which is: how much faster or slower than open-loop would you like the closed-loop to converge? Simulations demonstrate that this is a cheap and simple way of effective tuning by suitably over or under actuating the open-loop control action.**

I. Introduction

The popularity of model predictive control (MPC) is taken for granted these days but most of the focus in the literature is on the more expensive products which require reliable quadratic programming (QP) optimisers for high dimensional optimisations, or indeed even more challenging non-linear optimisations [1], [2]. There is relative little attention given to the other end of the market, that is relatively low cost more akin to PID. There are still many applications where a cheap single-input-single-output (SISO) control law is required, but PID is not as effective as one would like.

A secondary issue which also has gained relatively little interest in the literature is the one of MPC tuning. While it is accepted that the input and output horizons do affect the ultimate tuning, these are not usually considered tuning knobs in themselves as the default position is to take the horizons to be as large as the computing available allows [3], [4]. Consequently the main tuning parameters are the weights in the performance index, but the relationship between the weights and properties such as bandwidth and settling time is not analytical, which means tuning could be considered as much an art form as systematic, or perhaps something amenable to an offline tuning optimisation such as with genetic algorithms [4].

There is one notable exception to the above observations and that is predictive functional control (PFC) [5]. This algorithm is built on some sensible concepts that would appeal to practitioners and thus has found widespread acceptance in industry [6]. Nevertheless, recent literature [7]–[9] has emphasised the theoretical weaknesses in the basic algorithm and thus has sought to produce modifications which retain the appeal of the underlying concepts, but give more rigour and confidence in the final control law. A simple summary of some of the core conclusions of this work is:

1) The use of a constant future input in the predictions in conjunction with a single coincidence point can lead to significant inconsistencies affecting both reliable constraint handling and behaviour [7], [12].
2) The definition of the coincidence point makes inconsistent use of target/disturbance information [18] which often results in additional lag in the responses and thus the tuning is not as intuitive as desired.
3) For systems with undesirable open-loop dynamics, some form of pre-conditionning of the predictions is essential to ensure the PFC implementation is reliable [13]–[15].

This paper is focussed more on the first two points above; the proposals made could be combined with the 3rd point fairly easily but we want a simple focus as befits a short conference paper.

Specifically, this paper explores the role of the input parameterisation within PFC. Recent work, building on insights from the mainstream MPC community [16], has encouraged the use of input prediction parameterisations which converge to the steady-state asymptotically rather than instantly [17]. It has been shown that these improve constraint handling significantly, and also tuning [11]. Nevertheless, one core facet has not yet been explored in the literature and that is the role of *pseudo-open-loop* control, that is one whereby we seek to achieve open-loop dynamics but within a closed-loop including integral action. The advantage of such an approach is that the input is automatically fairly passive which in many scenarios is an advantage.

A second a more significant contribution of this paper is to propose a different flavour of tuning direction to the conventional algorithm, that is, rather than using the desired settling time as the main tuning parameter, instead using something we will call SPEED-UP. In simple terms this means, how much faster than open-loop do we want the closed-loop system to converge. SPEED-UP is a nice tuning factor because it also has a clear relationship with input activity. For example, a SPEED-UP of 2 suggests that the input will over-actuate by roughly double during transients.

Section II will given a brief introduction to classical PFC and some alternative input parameterisations, including the open-loop dynamics option. Section III will introduce the proposed new PFC approach based on SPEED-UP and then section IV will give some simulation comparisons and illustrations.

[1] Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, S1 3JD, UK (e-mail: j.a.rossiter@sheffield.ac.uk, msaftab1@sheffield.ac.uk, g.panoutsos@sheffield.ac.uk)

II. Background on PFC

This section gives an overview of PFC and some simple alternative input parameterisations. This is used as the foundation for the proposal of the following section.

A. System definition

For convenience hereafter, and without loss of generality, take the following *nominal* transfer function model:

$$a(z)y_k = b(z)u_k + d_k \tag{1}$$

so output y_k, input u_k and d_k a disturbance estimate to cater for uncertainty. We assume that true process is similar, for example:

$$a_p(z)y_{p,k} = b_p(z)u_k; \quad d_k = y_{p,k} - y_k \tag{2}$$

Note it is assumed that the input to the process and model are the same. The model used could equally be in state space form and this assumption makes little difference to the control law derivations.

B. System prediction

Prediction is well known [4] so details are omitted here suffice to say one can determine an n-step ahead output predictions as follows, for suitable H, P, Q.

$$\underset{\rightarrow k+1}{y} = H\underset{\rightarrow k}{u} + Q\underset{\leftarrow k}{y} + P\underset{\leftarrow k}{u} + Ld_k \tag{3}$$

where

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

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

and L is a vector of ones.

C. Conventional PFC control law

PFC is based on the premise of matching the output prediction to a first order response with a given time constant. Hence, define a target trajectory r_k as:

$$r_{k+i} = (1 - \lambda^i)R + \lambda^i y_{p,k}, \quad i = 1, 2, \cdots \tag{4}$$

where R is the set-point and λ is the desired closed-loop pole. Note, we ignore details linked to non-zero dead-time examples for simplicity of notation; these are available in many of the references (for example, see [14]).

The PFC law is defined by forcing the prediction of (3) to match the desired trajectory (4) at a specified point n-steps ahead, assuming that the future input is constant, that is, $u_k = u_{k+i}, \forall i > 0$. Hence the PFC law is defined from:

$$e_n^T[HLu_k + Q\underset{\leftarrow k}{y} + P\underset{\leftarrow k}{u} + Ld_k] = (1 - \lambda^n)R + \lambda^n y_{p,k} \tag{5}$$

where e_n is the nth standard basis vector. It is straightforward to determine u_k from (5).

D. Laguerre PFC

It was noted recently [12], [17] that the restriction of the future input to a constant did not match the expected shape of the closed-loop input and thus embedded an inconsistency between predictions and closed-loop, which in turn meant that the tuning was inevitably inconsistent. A simple improvement was to parameterise the future input using a first order Laguerre function, in essence an exponential decay so that:

$$\underset{\rightarrow k}{u} = \underbrace{\begin{bmatrix} 1 \\ \rho \\ \vdots \\ \rho^{n-1} \end{bmatrix}}_{H_\rho} \eta + \begin{bmatrix} u_{ss} \\ u_{ss} \\ \vdots \\ u_{ss} \end{bmatrix}; \tag{6}$$

where ρ is a decay factor to be chosen, η is a degree of freedom (d.o.f.) and u_{ss} is the expected steady-state so that:

$$\{u_{k+i} = u_{ss}, \ \forall i \geq 0\} \quad \Rightarrow \quad \lim_{i \to \infty} E[y_{k+i}] = R \tag{7}$$

For model (1) we can determine that, in steady-state:

$$a(1)y_{ss} = b(1)u_{ss} + d_k \quad \Rightarrow \quad E[u_{ss}] = \frac{a(1)R - d_k}{b(1)} \tag{8}$$

It is straightforward to combine the updated input prediction of (6) with predictions (3) and trajectory (4) to define the modified PFC control law as:

$$e_n^T[H(H_\rho\eta + Lu_{ss}) + Q\underset{\leftarrow k}{y} + P\underset{\leftarrow k}{u} + Ld_k] = (1 - \lambda^n)R + \lambda^n y_{p,k} \tag{9}$$

Hence we solve (9) for η and substitute into (6) to determine u_k.

E. Open-loop dynamics PFC (OL)

A final simple alternative is where one is happy with the open-loop dynamics and the feedback is simply to ensure offset free tracking. Such a control law can be achieved with the simple rule:

$$u_k = E[u_{ss}] \tag{10}$$

where $E[u_{ss}]$ is indicated in (8).

Remark 1: It so happens that one can achieve an open-loop dynamics PFC using control law (9) with $\rho = 0$. This observation will prove useful in the following.

Remark 2: It should be emphasised that the open-loop method avoids use of (4) altogether. This is actually a critical part of the proposal in this paper as this means we avoid the inconsistencies highlighted in [10], [18] whereby the target information is used differently in consequent samples, leading to unexpected lag in the closed-loop behaviour.

F. Constraints

It is possible to incorporate constraint handling into PFC in a systematic and computationally simple way, and while retaining feasibility, as demonstrated in several recent papers [11], [12].

$$\begin{array}{rcl} \underline{u} & \leq \mathbf{u}_k \leq & \overline{u} \\ \Delta\underline{u} & \leq \Delta\mathbf{u}_k \leq & \Delta\overline{u} \\ \underline{y} & \leq \mathbf{y}_k \leq & \overline{y} \end{array} \tag{11}$$

and $\Delta \mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_{k-1}$.

However, as these details are not central to the contribution of this paper they are excluded for clarity and brevity.

III. Proposed PFC control law based on SPEED-UP

The key factor here is transparency of tuning. It is assumed that the operator can view the open-loop speed of response and indeed achieve this with the PFC law given in (10), or indeed equivalently (9) with $\rho = 0$. Hence it is transparent and easy for them to define a closed-loop response as being say, twice as fast, and obviously therefore having input activity twice as big.

A. Increasing speed of target trajectory compared to open-loop benchmark

In order to achieve some faster response, then we need the error convergence of the predicted behaviour of (3) to be appropriately faster for consistency. Hence, one core concept is to choose an appropriate coincidence point that will cause the suitably faster behaviour/convergence.

Begin with a benchmark behaviour that would be achieved with the open-loop method of (10), so that the predictions take the form:

$$y_{k+n|k} = e_n^T[HLu_{ss} + Q\underset{\leftarrow k}{y} + P\underset{\leftarrow k}{u} + Ld_k] \tag{12}$$

The associated n-step ahead prediction error is given as:

$$e_{k+n} = R - e_n^T[HLu_{ss} + Q\underset{\leftarrow k}{y} + P\underset{\leftarrow k}{u} + Ld_k] \tag{13}$$

Next, chose a coincidence point which has faster convergence, so implicitly the associated error is smaller by a factor of β, where β is a factor to be determined.

Lemma 1: For $\beta > 1$, a *relative* PFC control law can be defined as follows. Choose η such that:

$$\begin{aligned} R - e_n^T[HLu_{ss} + Q\underset{\leftarrow k}{y} + P\underset{\leftarrow k}{u} + Ld_k] = \\ \beta[R - e_n^T[HLu_{ss} + HH_\rho\eta + Q\underset{\leftarrow k}{y} + P\underset{\leftarrow k}{u} + Ld_k] \end{aligned} \tag{14}$$

$$\eta = \frac{(\beta - 1)}{e_n^T HH_\rho\beta}[R - e_n^T[HLu_{ss} + Q\underset{\leftarrow k}{y} + P\underset{\leftarrow k}{u} + Ld_k] \tag{15}$$

Then $u_k = u_{ss} + \eta$.

Proof: It is clear from equation (15) that the coincidence point for the predictions with the Laguerre addition, has an associated error which is β times smaller than the error using predictions based on the open-loop approach.

The core point here is that control law (15) gives us a mechanism for achieving faster behaviour using a PFC equivalent statement; this control law is analogous to (5) with the critical exception that now there is no need for the tuning parameter λ. This difference is fundamental to the contribution of the paper as tuning is now based on relative statements rather than absolute ones.

B. Determining a precise PFC law with faster responses

First we establish a common sense observation for faster closed-loop responses, that is, a faster response requires a more aggressive input action.

Lemma 2: In simple terms, for zero initial conditions and a change in the target, a necessary condition for the response to be θ times faster is if the initial input u_k for a step change in the target to be θ times bigger. This lemma is given without proof as self evident.

Next, we look at the impact of requiring a smaller asymptotic error (as in (15)) on the initial input magnitude. The argument is that, from linearity, comparing the input activity with zero initial conditions and zero disturbance is a likely indicator of the resulting closed-loop poles and this simplifies the next stage of the analysis.

The initial input, for a change in target R and coincident point (13) and zero initial conditions, using the open-loop tuning is given in (10). (In this case the input is a constant throughout the predictions.)

Lemma 3: The initial input, for a change in target R and control law (15) is given as follows:

$$u_k = u_{ss} + \eta = \frac{R}{g(1)} + \frac{(\beta - 1)}{e_n^T HH_\rho\beta}[R - e_n^T[HLu_{ss}] \tag{16}$$

This also follows directly from (15).

For convenience hereafter define the following:

$$h_\rho = e_n^T HH_\rho; \quad h = e_n^T HL; \quad E[u_{ss}] = \frac{R}{g(1)} \tag{17}$$

Hence (16) can be simplified to:

$$u_k = \frac{R}{g(1)} + \frac{(\beta - 1)}{h_\rho\beta}[R - h\frac{R}{g(1)})] \tag{18}$$

Theorem 1: The initial input from (18) is θ times faster than (10) if β is chosen as follows:

$$\beta = \frac{h - g(1)}{(\theta - 1)h_\rho - g(1) + h} \tag{19}$$

.

Proof: Placing the two inputs (10), (18) side by side and removing the common factor R, we have:

$$\frac{\theta}{g(1)} = \left[\frac{1}{g(1)} + \frac{(\beta - 1)}{h_\rho\beta}[1 - h\frac{1}{g(1)}]\right] \tag{20}$$

Create a common denominator and match the numerators, hence:

$$h_\rho\beta\theta = h_\rho\beta + (\beta - 1)[g(1) - h] \tag{21}$$

Finally, solving for β gives the result in (19).

Remark 3: The derivation of the value for β was done with the nominal case and zero initial conditions for simplicity. However, as the final control law is in (15), the implied poles will be retained for the closed-loop and moreover, there will be robustness to uncertainty and offset free tracking.

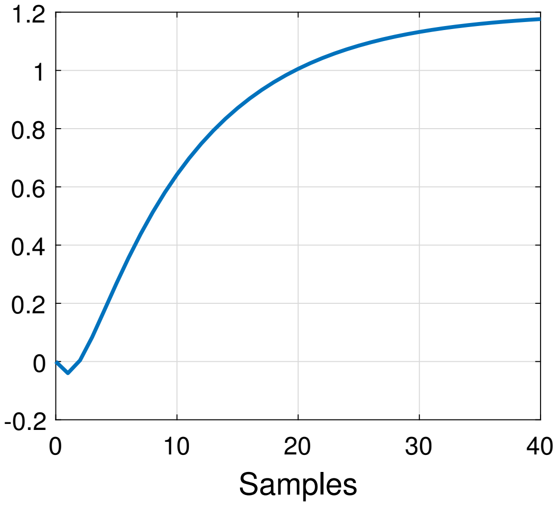


Fig. 1. Open-loop step response for system (22).

C. Summary of proposed algorithm

This section summarises the core conceptual steps and algebra needed to implement the algorithm. it is noticed that the computations are equivalent to a conventional PFC approach and thus neither more nor less complicated to code and implement. The core difference is the approach to tuning where here one adopts relative statements (faster or slower) rather than specifying desired poles/time constants precisely.

1) Verify that the open-loop behaviour is broadly acceptable so can be used as a valid benchmark.
2) Determine the desired speed-up factor θ, that is how much faster than open-loop behaviour do you want the closed-loop to be?
3) Solve for the parameter β using equation (19).
4) Determine the PFC law using equation (15).

Remark 4: Constraint handling can be handled in a conventional PFC manner using a simple *for loop* as discussed in the references. It is reiterated that recursive feasibility is automatic in the nominal case, although of course guarantees in the presence of uncertainty require computational complexity, expense and approaches which exceed the remit of PFC.

IV. Numerical comparisons

This section will demonstrate the efficacy of the proposed PFC approach as an alternative way to tune closed-loop behaviour. It needs to be re-emphasised that the method is based on the assumption that the open-loop dynamics are essentially satisfactory so this method alone may not be appropriate for systems with significant under-damping or open-loop instability. Relevant details can be found in reference [19] which successfully extends this proposal to such difficult systems.

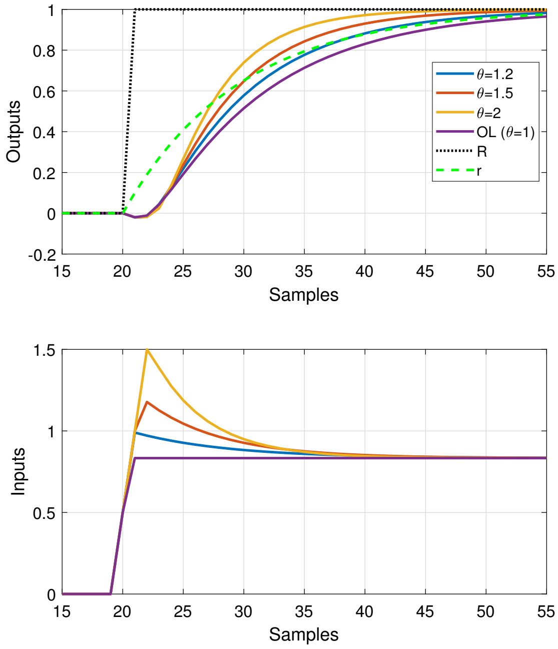


Fig. 2. Closed-loop responses for system (22) with various θ.

A. Example 1

Take the 2nd order, slightly over-damped system, with a non-minimum phase zero:

$$y(z) = \frac{-0.04z^{-1} + 0.1z^{-2}}{1 - 1.4z^{-1} + 0.45z^{-2}} \tag{22}$$

It should be remarked that the presence of the non-minimum phase zero makes a conventional PFC difficult to tune effectively and very difficult to achieve faster than open-loop behaviour!

The coincidence horizon is taken to be 15 in lieu of the slow pole at 0.9. The open-loop response is given in figure 1. The closed-loop responses for different choices of θ are shown in figure 2. It is clear that the required speed up has been achieved accurately and thus the proposed tuning parameter of θ is intuitive and easy to use.

Remark 5: The tuning parameter θ can also be used to achieve performance slower than open-loop, for example where there is a particular desire for the input to be slowly varying. This is illustrated in figure 3.

B. Example 2

Take a 3rd order, system, again with a non-minimum phase zero:

$$y(z) = \frac{0.1z^{-1} - 0.4z^{-2}}{1 - 1.85z^{-1} + 1.035z^{-2} - 0.171z^{-3}} \tag{23}$$

It should be remarked that the presence of the non-minimum phase zero makes a conventional PFC difficult to tune effectively and very difficult to achieve faster than open-loop behaviour!

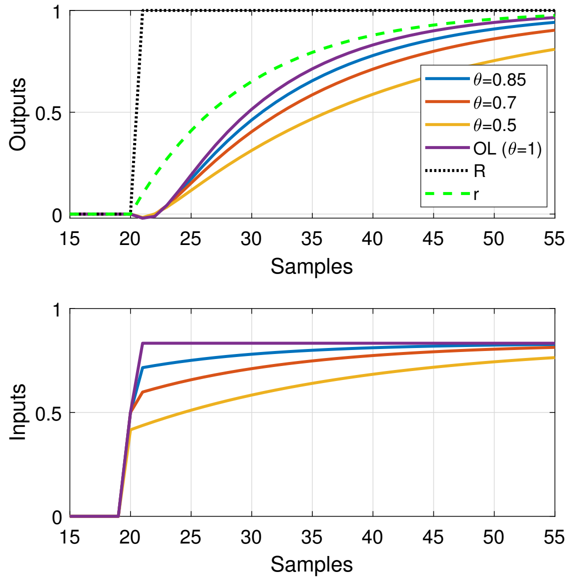


Fig. 3. Closed-loop responses for system (22) with θ chosen to slow behaviour down.

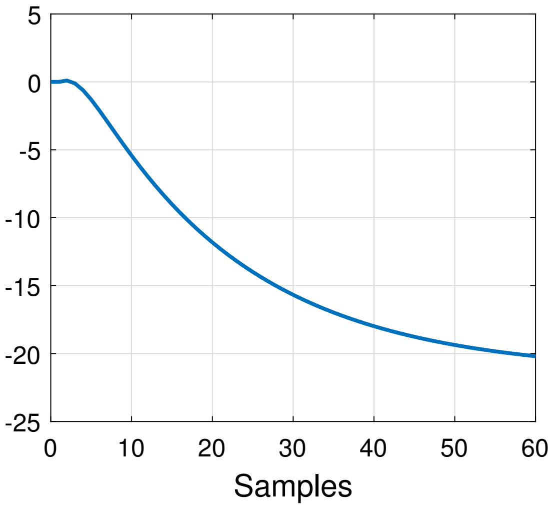


Fig. 4. Open-loop step response for system (23).

The coincidence horizon is taken to be 30 in lieu of the very slow pole at 0.95. The open-loop response is given in figure 4. The closed-loop responses for different choices of θ are shown in figure 5. Once again it is evident that the required speed up has been achieved accurately and thus the proposed tuning parameter of θ is intuitive and easy to use.

C. Disturbance rejection

For completeness, this section illustrates that the benefits are retained by the loop and thus apply, for example during disturbance rejection. Figure 6 shows the disturbance rejection with system (23); it is clear that the SPEED-UP has been retained.

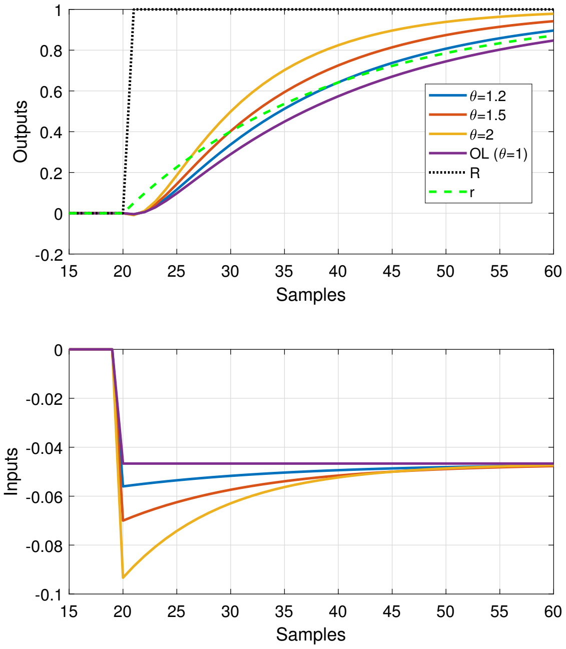


Fig. 5. Closed-loop responses for system (23) with various θ.

V. Conclusions and future work

This paper has proposed a totally different conceptually approach to PFC algorithms, that is where tuning is based on relative rather than absolute statements. The advantage of using relative statements is that it is possible to enable an intuitive tuning parameter, here denoted as SPEED-UP: how much faster, or slower, than open-loop do you want to be? It is also noticeable that the proposed approach moves away form the traditional control law definition around (4) and thus avoids issues linked to inconsistent use of the target information [18].

As compared to traditional PFC approaches and indeed the many modifications proposed in the recent literature, the tuning parameter here seems to behave far more consistently so that the user achieves the desired behaviour; this is evident from figures 2-5 where the initial input over or under actuates to the required degree. It should be emphasised however, that this approach (alone) is not effective with under-damped systems, and may require pre-stabilisation (see [14], [19]) for reliable performance.

A core conceptual point within this paper is that it builds on work [17] which used a Laguerre formulation for the input parameterisation. This is essential as it means that the predicted input moves smoothly from its initial over-actuation to the required steady-state thus giving consistency between predictions and closed-loop behaviour, something that conventional PFC cannot give.

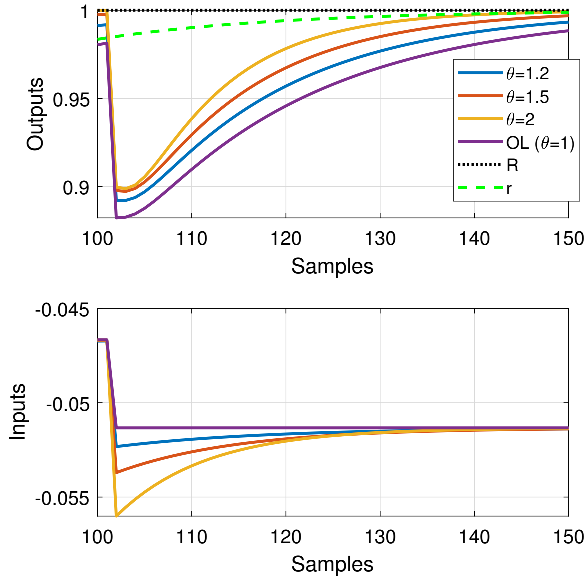


Fig. 6. Closed-loop disturbance rejection for system (23) with various θ.

References

[1] E.F. Camacho and C. Bordons, Model Predictive Control. Springer, 1999.

[2] J.M. Maciejowski, Predictive Control with Constraints. Pearson Education, 2002.

[3] D.Q. Mayne, Model predictive control: Recent developments and future promise, Automatica, vol. 50, no. 12, pp. 2967-2986, Dec 2014.

[4] J. Rossiter, A First Course in Predictive Control. CRC Press, 2018.

[15] Z. Zhang, J. A. Rossiter, L. Xie, and H. Su, Predictive functional control for integrator systems, Journal of the Franklin Institute, Volume 357, Issue 7, 2020, Pages 4171-4186, ISSN 0016-0032, https://doi.org/10.1016/j.jfranklin.2020.01.026.

[5] J. Richalet, A. Rault, J. Testud, and J. Papon, Model predictive heuristic control, Automatica (Journal of IFAC), vol. 14, no. 5, pp. 413-428, 1978.

[6] J. Richalet and D. O'Donovan, Predictive Functional Control: Principles and Indus- trial Applications. Springer Science and Business Media, 2009.

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

[9] J.A. Rossiter and M.S. Aftab, M.S. A Comparison of Tuning Methods for Predictive Functional Control. Processes 2021, 9, 1140. https://doi.org/10.3390/pr9071140.

[10] J. A. Rossiter and M. Abdullah, A new paradigm for predictive functional control to enable more consistent tuning, in 2019 American Control Conference (ACC). IEEE, Jul 2019.

[11] 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.

[12] M. Abdullah and J. A. Rossiter, Using laguerre functions to improve the tuning and performance of predictive functional control, International Journal of Control, pp.1-13, 2019.

[13] M.S. Aftab, and J.A. Rossiter, Pre-stabilised predictive functional control for open-loop unstable dynamic systems, 7th IFAC Conference on Nonlinear Model Predictive Control 2021, 11-14 Jul 2021, Virtual conference.

[14] M. S. Aftab, and J. A. Rossiter, Predictive functional control for challenging dynamic processes using a simple prestabilisation strategy, Advanced Control for Applications, Wiley, 2022, https://doi.org/10.1002/adc2.102.

[16] K.R. Muske and J.B. Rawlings, Model predictive control with linear models, AIChE Journal, 39(2), 262–287, 1993.

[17] M. Abdullah and J. A. Rossiter, Utilising Laguerre function in predictive functional control to ensure prediction consistency, 2016 UKACC 11th International Conference on Control (CONTROL), 2016, pp. 1-6, doi: 10.1109/CONTROL.2016.7737639.

[18] 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, 2020, ISSN 0020-7179.

[19] M. S. Aftab, and J. A. Rossiter, Predictive Functional Control for Difficult Dynamic Processes with a Simplified Tuning Mechanism, 13th UKACC International Conference on Control (CONTROL 2022), IEEE, 2022.

Appendix I

A Novel Approach to PFC for Nonlinear Systems*

John Anthony Rossiter, Muhammad Saleheen Aftab, George Panoutsos, and Oscar Gonzalez-Villarreal

This paper has been published in the European Journal of Control, Elsevier, 2022.

A conference version of this paper has been presented at the 20th European Control Conference (ECC), UK, 2022

Author Contributions. This paper is a collaborative work between all authors. J. A. Rossiter provided initial proposals and accurate communication of the concepts. M. S. Aftab analysed the concepts in case studies and reviewed/edited the draft. G. Panoutsos co-supervised M. S. Aftab and suggested possible case study. O. G. Villarreal provided simulation results with Nonlinear MPC.

European Journal of Control

A novel approach to PFC for nonlinear systems

--Manuscript Draft--

Manuscript Number:	
Article Type:	SI:European Control 2022 (invite only)
Keywords:	Predictive functional control; computational efficiency; transparent tuning; nonlinear systems
Corresponding Author:	John Anthony Rossiter University of Sheffield Sheffield, South Yorkshire United Kingdom
First Author:	John Anthony Rossiter
Order of Authors:	John Anthony Rossiter
	Muhammad Saleheen Aftab
	George Panoutsos
	Oscar Gonzalez-Villarreal
Abstract:	This paper proposes a computationally efficient predictive control law for non-linear systems, that is one that can easily be coded and implemented on low cost hardware. Moreover, it has a secondary core benefit that the core tuning parameter reduces to a single choice which is: how much faster than open-loop would you like the closed-loop to converge? Conceptually the approach builds on the PFC approach but proposes a very different type of coincidence condition which removes the lag associated to the conventional approach. Simulations demonstrate that for some non-linear systems this is a cheap and simple way of ensuring effective feedback, with constraint handling. 1
Suggested Reviewers:	Robert haber robert.haber@th-koeln.de
	Mohammed Abdullah mohd_abdl@iium.edu.my

A novel approach to PFC for nonlinear systems

John Anthony Rossiter [a] Muhammad Saleheen Aftab [a]
George Panoutsos [a] Oscar Gonzalez-Villarreal [b]

[a] *Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, S1 3JD, UK (e-mail: j.a.rossiter@sheffield.ac.uk, msaftab1@sheffield.ac.uk,g.panoutsos@sheffield.ac.uk)*

[b] *School of Aerospace, Transport and Manufacturing Centre for Autonomous and Cyber-Physical Systems, Cranfield University, UK, (email: Oscar.Gonzalez-Villarreal@cranfield.ac.uk)*

Abstract

This paper proposes a computationally efficient predictive control law for non-linear systems, that is one that can easily be coded and implemented on low cost hardware. Moreover, it has a secondary core benefit that the core tuning parameter reduces to a single choice which is: how much faster than open-loop would you like the closed-loop to converge? Conceptually the approach builds on the PFC approach but proposes a very different type of coincidence condition which removes the lag associated to the conventional approach. Simulations demonstrate that for some non-linear systems this is a cheap and simple way of ensuring effective feedback, with constraint handling. [1]

Key words: Predictive functional control, computational efficiency, transparent tuning, nonlinear systems.

1 Introduction

Model predictive control (MPC) [1] is very popular in both the industrial [2,3] and academic communities [4–6]. This is because it makes good intuitive sense combined with delivering reliable results for MIMO (multi-input-multi-output) systems and managed constraint handling. Moreover, although rarely

[1] This is a slightly extended and corrected version of the paper which appeared at the European Control Conference, 2022.

discussed carefully [7], it also has the potential to handle future target information systematically.

Nevertheless, despite the popularity and effectiveness, MPC is still not widely deployed on low level loops where PID continues to dominate. This is as expected. PID tuning [8, 9] is simple enough to be handled without recourse to expensive consultants and moreover, for many practical feedback loops, delivers performance that is adequate. Of course, in addition and critically, PID is much cheaper to purchase, code and implement than MPC in general and thus there needs to be a significant potential benefit before a more expensive alternative would be considered.

This paper focuses on one notable exception to the above observations. There are some SISO (single-input-single-ouput) loops where a simple PID implementation does not deliver adequate performance, perhaps due to challenging dynamics or perhaps due to the need for constraint handling. In such a case, a cheap MPC approach would be competitive in both price and complexity and indeed this is what has been noticed by PFC (predictive functional control) vendors [10, 11] over many years. More specifically:

(1) Being model based, PFC is able, in principle, to exploit model information more systematically than PID and thus improve closed-loop behaviour.
(2) Being prediction based, again in principle, PFC can handle constraints systematically.

Nevertheless, the reader will note the use of words *in principle* to clarify the above statements. A large number of recent works have investigated the tuning [12–15] and constraint handling [6, 17] of PFC and made a number of useful observations and contributions:

- The original PFC algorithm is effective with processes having over-damped behaviour, but tuning is much more difficult with other dynamics.
- Recent work has suggested a number of modified PFC algorithms which are more reliable, consistent and enable better links between the tuning parameters and behaviour.
- The constraint handling in the original PFC algorithm was more akin to approaches used in PID and thus suboptimal at best. Using predictions more systematically is straightforward and enables far better results while still requiring no optimisation.

One of the weaknesses in conventional PFC is the use, in predictions, of a fixed future input whereas it is well known (e.g. [15]) that more nuanced parameterisations of the future input sequence are helpful. A very recent work [18] demonstrated how a very simple PFC algorithm could combine steady-state estimates with a simple exponential parameterisation to give intuitive and effective tuning. This work, for now considered only the linear case whereas, the original creators of PFC saw one huge advantage of the simplicity of the algorithm being in its potential usage with non-linear systems. The reader may note that nonlinear MPC (NLMPC) in the literature is largely both complex and computationally demanding [19]. Hence, the core contribution of this paper is to demonstrate how this recent new PFC approach [18] can be adapted to the nonlinear case and implemented with almost negligible computing and complexity, certainly when compared to more conventional NLMPC algorithms [19].

The paper is organised as follows. Section 2 gives core background on both PFC and the recent proposed algorithm. Section 3 shows how this algorithm can be modified for the nonlinear case. Section 4 presents a case study on a mixing tank with an endothermic reaction and the paper finishes with some numerical results and conclusions.

2 Background on PFC

2.1 System definition

The non-linear system will be taken to be of the form:

$$\dot{x} = f(x, u) \tag{1}$$

with state x and input u (dimensions n_x, n_u respectively) and $f(.)$ is differentiable. This paper assumes this model can be approximated, at individual sample times, by a discrete linear time varying state-space model:

$$x_{k+1} = A_k x_k + B_k u_k \tag{2}$$

More discussion of the linearisation and the use of deviation variables is in section 3.

In practice model (1) is an approximation, so we need to allow for some uncertainty. Here we use standard practice in the literature and define the true process state to be x_p and thus the error term x_e, at each sample, is given as:

$$x_{e,k} = x_{p,k} - x_k \tag{3}$$

As is standard practice in the MPC literature, the error term is used to ensure unbiased prediction and offset free tracking and caters for both parameter uncertainty and disturbances. It assumes that the relevant true state $x_{p,k}$ at the current sample can be measured.

2.2 System prediction

Prediction is well known [6] so details are omitted here. It is sufficient that the reader recognises that with LTV model (2), or indeed similar models, one can easily deduce n-step ahead output predictions as follows, for suitable H, P, L (dimensions implicit from the context).

$$\underset{\rightarrow k+1|k}{x} = H\underset{\rightarrow k}{u} + Px_k + Ld_k \tag{4}$$

where $d_k = x_{e,k}$ and

$$\underset{\rightarrow}{u_k} = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+n-1} \end{bmatrix}; \ \underset{\rightarrow k+1|k}{x} = \begin{bmatrix} x_{k+1|k} \\ x_{k+2|k} \\ \vdots \\ x_{k+n|k} \end{bmatrix}$$

and L is a vector of ones. In the non-linear case, H, P will be time varying and need to be updated every sample, as seen in section 3.

2.3 Conventional PFC control law

This is presented for completeness only and thus very briefly. Conventional PFC is based on the premise of matching the output prediction to a first order response with a given time constant. Hence, assuming the steady-state target is a constant R, define define a target trajectory during transients $r_{k+i|k}$ as:

$$r_{k+i|k} = (1-\lambda^i)R + \lambda^i x_{p,k}, \quad i = 1, 2, \cdots \tag{5}$$

The PFC control law is determined by ensuring that $x_{k+n|k} = r_{k+n|k}$ and thus, in effect, substitution of (4) into (5) and solving for the degree of freedom which typically is the future value of the input (assumed constant).

However, as mentioned in the introduction [12, 14, 18], this algorithm often fails to give reliable behaviour in that the main tuning parameter λ is often ineffective and indeed, for some open-loop dynamics it is difficult to gain satisfactory beahviour. Thus, some simple alternatives have been proposed and hereafter we introduce one of these.

2.4 Open-loop dynamics PFC (OL)

The simplest predictive algorithm is one which makes no attempt to change the dynamics and focuses solely on ensuring offset free tracking. Such an algorithm is summarised as:

$$u_k = E[u_{ss}] \tag{6}$$

where $E[u_{ss}]$ is the expected steady-state input.

This algorithm gives a useful benchmark for more computationally demanding algorithms, and is especially useful when the open-loop dynamics are benign because it gives a very simple and effective control law. Hence, one would embellish this control law if and only if one wanted faster settling times or a slower change of the input.

Remark 1 *This control law is very simple to code and implement as no detailed prediction is needed, rather just a mechanism to estimate u_{ss}. Thus it provides a route to computationally efficient control of non-linear systems*

2.5 Speeding up OL PFC with exponential input parameterisations

In order to speed up the response it is necessary to over actuate during transients. A simple over-actuation strategy [17, 18] is to parameterise the future inputs as follows:

$$u_k = u_{ss} + \lambda^k \eta; \quad \{0 < \lambda < 1\} \Rightarrow \lim_{k\to\infty} u_k = u_{ss} \tag{7}$$

This has a single degree of freedom (d.o.f.), that is η and thus is amenable to simple optimisation. The parameter λ should be chosen sympathetically with the open-loop dynamics and desired closed-loop dynamics, that is, to converge in a roughly equivalent period; typically chosen the same as in (5).

The selection of η is critical, and hence it was proposed [16] to ensure the associated predictions converge a factor S faster than those associated to the use of (6) alone. The conceptual steps are summarised next.

Algorithm 1 *PFC algorithm to speed up predicted convergence by a factor of S.*

(1) Determine the n-step ahead error E_o between the prediction and target using control law (6).
(2) Determine the n-step ahead error $E_\eta(\eta)$ between the prediction and target using control law (7). This depends on η.

(3) Choose η such that $SE_\eta(\eta) = E_o$ where S is a design speed-up factor to be selected.

The algorithm is presented conceptually because, in the non-linear case there will not be fixed algebraic computations for the terms E_o or indeed $E_\eta(\eta)$ and these will need to computed online each sample. However, it is critical to note that the d.o.f. η is a single variable and thus easy to determine efficiently; this will be evident in the numerical examples shown later.

A further important observation is that the tuning is now based on a simple intuitive statement: how much faster than open-loop dynamics would you like to be?

Remark 2 *In the linear case, because explicit and fixed algebraic relationships are possible, it is possible to make the tuning even more precise as discussed in [16]. Here we are extending and applying the concept to the non-linear case where relationships are time varying, and thus those additional steps are not considered for now. Of specific interest one should note that the actual closed-loop speed-up achieved will be different to the ratio of the prediction errors $SE_\eta(\eta) = E_o$ so some offline analysis will be needed.*

2.6 Constraint handling

One can incorporate constraint handling into Algorithm 1 in a systematic and computationally simple way by comparing system predictions against constraints for a sufficiently large horizon; this is standard in the literature [6,18,20].

$$\begin{aligned} \underline{u} \ &\leq \mathbf{u}_k \leq \ \overline{u} \\ \underline{\Delta u} &\leq \Delta\mathbf{u}_k \leq \Delta\overline{u} \\ \underline{x} \ &\leq \mathbf{y}_k \leq \ \overline{x} \end{aligned} \tag{8}$$

Critically it is noted that as the predictions have a single d.o.f. η, the selection of η can be determined using a simple *for loop* and thus done very efficiently. One might also note that with (7) the maximum input and input rate will occur at the first or second sample, and thus the number of inequalities to be checked for the input constraints is very small.

Remark 3 *Constraints limit the input amplitudes available and thus will also impact on the speed-up achievable in some scenarios, especially with large changes in target.*

2.7 Summary of proposed algorithm

This section has summarised the core conceptual steps and algebra needed to implement the proposed algorithm 1. The user needs to define the following design parameters.

(1) What is the prediction horizon n? Good practice [12] suggests something like 2 time constants.
(2) What is the speed up factor S? Clearly this depends entirely on what the user wants but we would not expect much bigger than 2-3 or significant over actuation is inevitable and this is rarely implementable in practice.
(3) The parameter λ used in (7) is needed. Typically this should be close to the target closed-loop pole and partially overlaps with the choice of n.

Having defined the core parameters, the remaining steps are linked to computation of the expected errors which is discussed in the following sections.

The reader should be reminded however that a core requirement for the efficacy of the proposed approach is that the open-loop behaviour is broadly acceptable (that is almost meets the performance requirements) so can be used as a valid benchmark.

3 Background on linearisation and prediction with non-linear models

In NLMPC it is necessary to form predictions for a non-linear model. As PFC, by design, is intended to be simple, here we take a very simple approach to this process, accepting that more accurate but also more demanding numerical integration approaches are possible.

3.1 Linearisation about a trajectory

Hence, we use superposition to find predictions by separating the nominal trajectory (x_k, u_k) from the deviations part. It is implicit hereafter that the nominal or baseline trajectory is that associated to input prediction (6).

(1) Simple difference equations are used to simulate the non-linear model and thus to form a baseline prediction based on some assumed future input. Let these values be: x_k, u_k for states x_k and inputs u_k and k the sample number.

(2) The model is linearised about all points x_k, u_k on the baseline prediction to form state-space models of the form:

$$\delta\dot{x}_k = A_k\delta x_k + B_k\delta u_k \tag{9}$$

where $\delta x_k = \hat{x}_k - x_k, \delta u_k = \hat{u}_k - u_k$ are deviations relative to the baseline prediction and A_k, B_k are the linearised model parameters at the kth sample of the baseline prediction. The full predicted state and input values are $\hat{x}_k = x_k + \delta x_k, \hat{u}_k = u_k + \delta u_k$.

Remark 4 *In order to derive the matrices A_k, B_k, we need to undertake partial differentiation of a model which is based on the first derivative, for example, assume that:*

$$\dot{\hat{x}} = f(\hat{x}, \hat{u}) \approx f(x, u) + \frac{\partial f}{\partial x}\delta x + \frac{\partial f}{\partial u}\delta u \tag{10}$$

$$\dot{\hat{x}} = f(\hat{x}, \hat{u}); \quad \dot{\hat{x}} = \dot{x} + \dot{\delta x} = f(x, u) + \dot{\delta x} \tag{11}$$

$$\delta\dot{x} = \frac{\partial f}{\partial x}\delta x_k + \frac{\partial f}{\partial u}\delta u_k = A_k\delta x_k + B_k\delta u_k \tag{12}$$

3.2 Prediction using deviation variables

Once one has determined the models (9) for a notional trajectory, one can easily determine the impact of small deviations in the input, that is $\delta u_k \neq 0$. Predictions can be found by recursive use of (9) as follows:

$$\delta\dot{x}_k \approx \frac{\delta x_{k+1}}{\delta t} = A_k\delta x_k + B_k\delta u_k \tag{13}$$

Summarising one deduces (for suitable period δt):

$$\begin{aligned} \delta x_{k+1} &\approx [A_k\delta x_k + B_k\delta u_k]\delta t \\ \delta x_{k+2} &\approx [A_{k+1}\delta x_{k+1} + B_{k+1}\delta u_{k+1}]\delta t \\ \delta x_{k+3} &\approx [A_{k+2}\delta x_{k+2} + B_{k+2}\delta u_{k+2}]\delta t \\ &\vdots \end{aligned} \tag{14}$$

Next, making substitutions and assuming that $\delta x_k = 0$:

$$\begin{aligned} \delta x_{k+1} &\approx [B_k\delta u_k]\delta t \\ \delta x_{k+2} &\approx [A_{k+1}B_k\delta u_k\delta t + B_{k+1}\delta u_{k+1}]\delta t \\ \delta x_{k+3} &\approx [A_{k+2}[A_{k+1}B_k\delta u_k\delta t + B_{k+1}\delta u_{k+1}]\delta t \\ &\qquad + B_{k+2}\delta u_{k+2}]\delta t \\ &\vdots \end{aligned} \tag{15}$$

3.3 Prediction with input parameterised using an exponential

The predictions of (15) are somewhat clumsy to use, but in the context of the PFC algorithm to be used here, the predicted input (7) has been parameterised as follows:

$$\delta u_{k+i} = \lambda^i \eta \tag{16}$$

for a given λ and η the d.o.f. to be selected online. Substitute (16) into (15) then:

$$\begin{aligned}
\delta x_{k+1} &= \underbrace{B_k \delta t}_{\alpha_k}\ \eta \\
\delta x_{k+2} &= \underbrace{[A_{k+1}\alpha_k + B_{k+1}\lambda]\delta t}_{\alpha_{k+1}}\ \eta \\
\delta x_{k+3} &= \underbrace{[A_{k+2}\alpha_{k+1} + B_{k+2}\lambda^2]\delta t}_{\alpha_{k+2}}\ \eta \\
&\vdots \\
\delta x_{k+n+1} &= \underbrace{[A_{k+n}\alpha_{k+n-1} + B_{k+n}\lambda^n]\delta t}_{\alpha_{k+n}}\ \eta
\end{aligned} \tag{17}$$

It is noted that the main computation here is the simple recursion of:

$$\alpha_{k+n} = [A_{k+n}\alpha_{k+n-1} + B_{k+n}\lambda^n]\delta t \tag{18}$$

Algorithm 2 *Predictions for the non-linear model $\dot{x} = f(x, u)$ with input parameterisation* (16) *are computed as follows:*

(1) Estimate the required steady-state input u_{ss} and simulate the model $\dot{x} = f(x, u)$ forward (using numerical integration) over the required horizon using $u_k = u_{ss}, \forall k > 0$.

(2) For the nominal trajectory (x_k, u_k) determined in step 1, form the matrices A_k, B_k at every sample using:

$$A_k = \frac{\partial f}{\partial x}; \quad B_k = \frac{\partial f}{\partial u}$$

(3) Use recursion (17) *to determine α_{k+n} for the required horizon n. The n-step ahead predictions for the state are now given as:*

$$\hat{x}_{k+n} = x_{k+n} + \alpha_{k+n}\eta \tag{19}$$

Remark 5 *It is implicit from the use of first order Taylor series and simple difference equations for the numerical integration that the trajectories do not deviate a long way from the baseline. If they do, then the approximation errors would grow and could impact on behaviour. This means the sample period δt should be small enough.*

We can define the final algorithm more precisely.

Algorithm 3 *The PFC algorithm 1 can be combined with Algorithm 2 as follows.*

(1) Find the baseline trajectory (x_k, u_k) *using* (6) *and also the associated state space matrices* (12) *and prediction* (19).
(2) Determine the n-step ahead error $E_o = R - x_{k+n}$.
(3) Determine the n-step ahead error $E_\eta(\eta) = R - \hat{x}_{k+n}$.
(4) Choose η *such that* $SE\eta(\eta) = E_o$*:*

$$S[R - x_{k+n} - \alpha_{k+n}\eta] = R - x_{k+n} \tag{20}$$

(5) The control value to be implemented at the current sample is: $u_k = u_{ss} + \lambda^0\eta$.

4 Description of case study

This section describes a simple mixing tank with an endothermic reaction. Chemical A is produced by a reaction in the tank, but this reaction is endothermic thus cooling down the tank contents. The rate of reaction is also temperature dependent (the main non-linear characteristic), so to maximise the reaction rate, the temperature needs to be maintained and thus heat must be supplied. Consequently, the tank can be described by two equations, one for the concentration and a second for the temperature. The objective is to control the concentration (output C_A) by manipulation of supplied heating (input W).

4.1 Core model equations

The concentration model depends on the flow rates into and out of the tank (assumed equal) and the reaction rate:

$$V\frac{dC_A}{dt} = \gamma C_A V e^{0.05(T-T_i)} + F(C_{A0} - C_A) \tag{21}$$

where V is tank volume, F is the flow rate, T is the temperature in the tank (assume well mixed), T_i is the temperature of the in flow, C_{A0} is the concentration of the inflow and C_A is the concentration in the tank. The variable γ is linked to the reaction rate.

The basic heat equation is:

$$V\rho C_p \frac{dT}{dt} = F\rho C_p(T_i - T) - \beta[\gamma C_A V e^{0.05(T-T_i)}] + W \tag{22}$$

where ρ is fluid density, C_p is fluid heat capacity, W the heat supply and β a variable linked to the rate of reaction and thus how much heat is absorbed by the reaction.

For convenience, the model equations (21),(22) can be re-arranged as follows.

$$\frac{dC_A}{dt} = \gamma C_A e^{0.05(T-T_i)} + \frac{F}{V}(C_{A0} - C_A) \tag{23}$$

$$\frac{dT}{dt} = \frac{F}{V}(T_i - T) - \beta[\frac{\gamma}{\rho C_p} C_A e^{0.05(T-T_i)}] + \frac{W}{V\rho C_p} \tag{24}$$

For the purposes of this paper the following values were used: $\rho = 10^3 kgm^{-3}$, $C_p = 4000J/kgdeg$, $V = 5m^3$, $F = 0.01m^3s^{-1}$, $\beta = 10^7 deg$, $\gamma = 0.005s^{-1}, T_i = 20deg$. It is also noted that the inlet concentration and inlet temperature are not considered to be degrees of freedom in this paper.

4.2 Steady-state estimates

For the proposed algorithm, we need to determine an estimate of the steady-state, assuming that the provided heating is constant, that is $W = W_{ss}$. Moreover, assume that the required steady-state concentration is known as $C_{A,ss}$. There will be an implied steady-state temperature T_{ss}.

A steady-state exists if the derivatives in (23),(24) are zero:

$$0 = \gamma C_{A,ss} e^{0.05(T_{ss}-T_i)} + \frac{F}{V}(C_{A0} - C_{A,ss}) \tag{25}$$

$$0 = \frac{F}{V}(T_{ss} - T_i) - \beta[\frac{\gamma}{\rho C_p} C_{A,ss} e^{0.05(T_{ss}-T_i)}] + \frac{W_{ss}}{V\rho C_p} \tag{26}$$

Using (25) to solve for the steady-state temperature gives:

$$e^{0.05(T_{ss}-T_i)} = -\frac{F}{V}\frac{(C_{A0} - C_{A,ss})}{\gamma C_{A,ss}} \tag{27}$$

$$(T_{ss} - T_i) = 20 \log\left(\frac{F}{V}\frac{(C_{A,ss} - C_{A0})}{\gamma C_{A,ss}}\right) \tag{28}$$

Now we can use (26) and (28) to find the required power input to maintain this temperature.

$$\beta[\gamma V C_{A,ss} e^{0.05(T_{ss}-T_i)}] - F\rho C_p(T_{ss}-T_i) = W_{ss} \tag{29}$$

4.3 Linearisation of case study

It is clear that both model equations (23), (24) take the form:

$$\dot{x} = f(x,u); \quad x = \begin{bmatrix} C_A \\ T \end{bmatrix}; \quad u = W \tag{30}$$

Hence we can linearise as in section 3. The partial derivatives can be computed as follows:

$$\frac{\partial}{\partial C_A}(\dot{C}_A) = \gamma e^{0.05(T-T_i)} - \frac{F}{V} \tag{31}$$

$$\frac{\partial}{\partial T}(\dot{C}_A) = 0.05\gamma C_A e^{0.05(T-T_i)} \tag{32}$$

$$\frac{\partial}{\partial C_A}(\dot{T}) = -\beta[\frac{\gamma}{\rho C_p} e^{0.05(T-T_i)}] \tag{33}$$

$$\frac{\partial}{\partial T}(\dot{T}) = -\frac{F}{V} - 0.05\beta[\frac{\gamma}{\rho C_p} C_A e^{0.05(T-T_i)}] \tag{34}$$

The corresponding linearised state-space model is given as:

$$A_k = \begin{bmatrix} \frac{\partial}{\partial C_A}(\dot{C}_A) & \frac{\partial}{\partial T}(\dot{C}_A) \\ \frac{\partial}{\partial C_A}(\dot{T}) & \frac{\partial}{\partial T}(\dot{T}) \end{bmatrix}; \quad B_k = \begin{bmatrix} 0 \\ \frac{1}{V\rho C_p} \end{bmatrix} \tag{35}$$

In summary, the parameters needed for (17) depend upon the current values of C_A, T as evident from (31)-(34) and thus can easily and quickly be updated with the values in (x_k, u_k) as required. Consequently the prediction equations outlined in Algorithm 2 and used in (20) of Algorithm 3 can be determined.

5 Simulation results

This section demonstrates the efficacy of the proposed Algorithm 3 on the case study given in section 4. A core selling point is the intuitive nature of the tuning whereby one can request performance as a relative measure compared against open-loop behaviour, the so called speed up factor. A second selling point is the computational simplicity; the main computing requirement is the recursion in (17) which, in terms of modern computing, is not significant.

This section will present results for the nominal case and also with significant parameter uncertainty to demonstrate that, as expected with most MPC approaches, the algorithm is robust to some uncertainty. We also include some comparisons with a conventional NMPC approach to highlight some of the differences.

5.1 Open-loop behaviour

In order to form a benchmark, this section begins by demonstrating the performance achieveable with control law (6) which ensures offset free tracking with open-loop dynamics. The corresponding behaviour is shown in Figure 1 where it is clear that:

- The settling time is around 2000 sec.
- There is no offset in C_A.
- The closed-loop input signal W is constant.

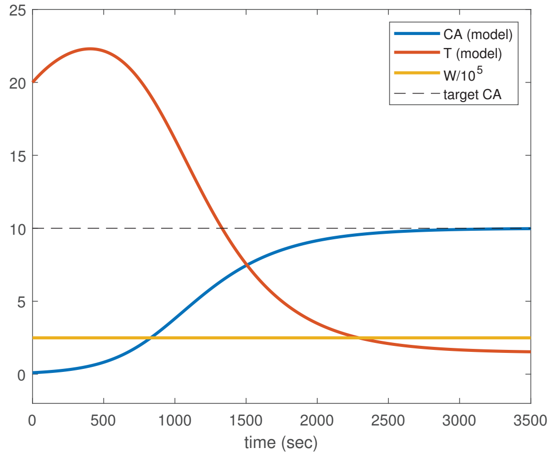


Fig. 1. Closed-loop responses for the nominal case using control law (6).

To demonstrate the impact of uncertainty, the true plant parameters are changed slightly from those in subsection 4.1 to: $V = 4.9m^3,\ F = 0.012m^3s^{-1},\ \beta = 1.1 \times 10^7 deg,\ ,T_i = 21deg$. The corresponding simulation is given in Figure 2. Unsurprisingly the behaviour is slightly different, but again there is no offset in the steady-state but now the feedback takes a while to determine the correct steady-state value for W which slows down the overall settling time.

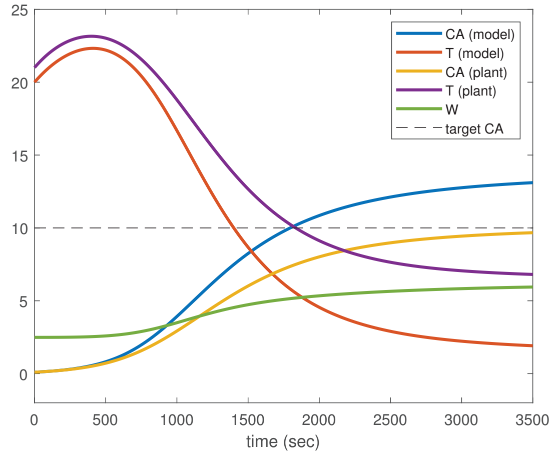


Fig. 2. Closed-loop responses for the robust case using control law (6).

5.2 Speeding up behaviour

Next we deploy Algorithm 3 and use a target speed up of a factor of $\beta = 2$ with $n = 15, \lambda = 0.9$ and sampling period of $5s$. The corresponding behaviour for the nominal case is shown in Figure 3 alongside the results for $\beta = 1$ where it is clear that:

- The Settling time is closer to 1000 sec and thus nearly twice as fast as in Figure 1.
- There is no offset.
- The closed-loop input signal is very aggressive (off the scale) and indeed the transient temperature has risen by nearly 20 degrees to facilitate the faster rise in the reaction rate.

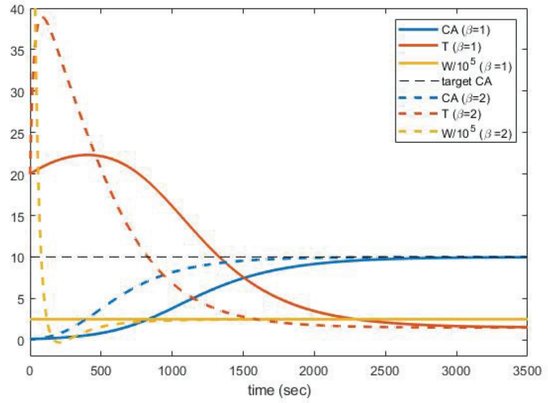


Fig. 3. Closed-loop responses for the nominal case using control law (20) alongside the open-loop behaviour.

Similar observations arise in the uncertain case as seen in Figure 4. The speed up is achieved as requested, alongside offset free tracking, but at the expense of aggressive heating during transients.

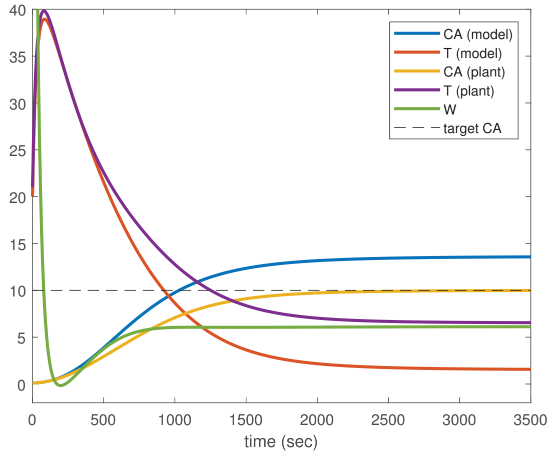


Fig. 4. Closed-loop responses for the uncertain case using control law (6).

5.3 Constraint handling

For completeness, Figure 5 gives a simulation for the uncertain case and with some input constraint handling to demonstrate that this is straightforward to implement. Here we implement a reasonable upper limit on the heating available to be not significantly bigger than the steady-state requirements. Unsurprisingly this results in a slight slow down in performance, that is, the target speed up of a factor of 2 is not achievable in this case.

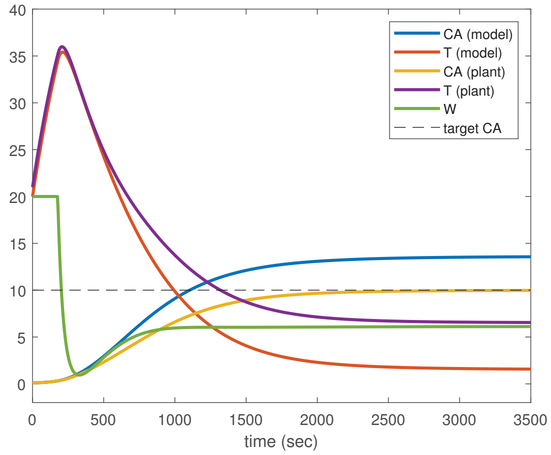


Fig. 5. Closed-loop responses for the uncertain case using control law (6) and with input saturation.

5.4 Off-the shelf NMPC

For completeness we illustrate the differences with a much more expensive and complicated off-the-shelf NMPC approach in Figure 6. In this case the user is able to change the performance index weights to achieve different performances, so we show how simple changes to the input weighting lead to different closed-loop behaviours. Nevertheless, what is most interesting here is that: i) the overall speed of response and smoothness of the output behaviour is similar to PFC and ii) NMPC is much less aggressive in its use of the heating and indeed more flexible in general to tune and trade-off the input and output behaviour.

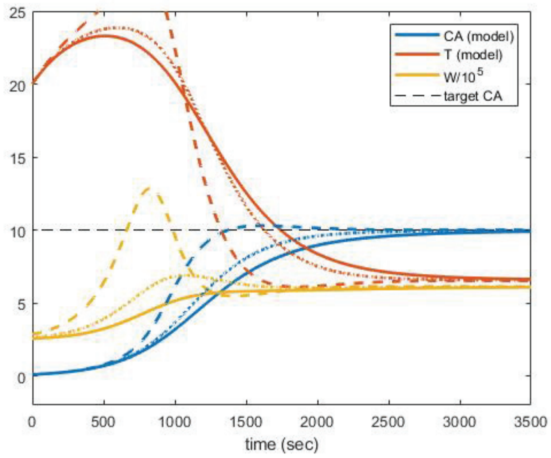


Fig. 6. Closed-loop responses for the uncertain case using NMPC and three alternative choices of weights (solid, dashed and dotted refer to different weighting choices).

6 Conclusions and future work

This paper has modified a recently proposed PFC Algorithm for the non-linear case and demonstrated that it can be applied affectively. The algorithm deploys an intuitive tuning factor, denoted speed-up, which is easy for workers to relate to and thus negates the need for experts to manage the implementation. The user can easily explore the impact of different speed-up choices on other aspects of behaviour such as input activity and decide upon the desired trade-off.

Other core benefits are that, despite being a full non-linear control law, the required on-line computations are relatively minor and thus can easily be coded in low cost processors. Also, in line with other predictive control laws, the incorporation of systematic constraint handling is straightforward and can be managed with a simple for loop. Moreover, the algorithm demonstrates the expected robustness to some parameter uncertainty similarly to other MPC approaches.

Future work will look at whether this concept can be usefully applied to scenarios where the model and measurement information is less precise, or fuzzy, such as where feedback is based on images rather than specific numerical values. It is also important to present a more complete and balanced comparison with tuning using alternative and conventional PFC approaches and this constitutes work in progress.

References

[1] D. Q. Mayne, Model predictive control: Recent developments and future promise," Automatica, vol. 50, no. 12, pp. 2967-2986, Dec 2014.

[2] E. Fernandez-Camacho and C. Bordons-Alba, Model Predictive Control in the Process Industry. Springer London, 1995.

[3] S. Qin and T. A. Badgwell, A survey of industrial model predictive control technology," Control Engineering Practice, vol. 11, no. 7, pp. 733-764, Jul 2003.

[4] E. F. Camacho and C. Bordons, Model Predictive Control. Springer, 1999.

[5] J. M. Maciejowski, Predictive Control with Constraints. Pearson Education, 2002.

[6] J. Rossiter, A First Course in Predictive Control. CRC Press, 2018.

[7] S. S. Dughman and J. A. Rossiter (2020) Systematic and effective embedding of feedforward of target information into MPC, International Journal of Control, 93:1, 98-112.

[8] K. J. Astrom and T. Hägglund, PID controllers: theory, design, and tuning, Instrument society of America Research Triangle Park, NC, 1995.

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

[10] J. Richalet, A. Rault, J. Testud, and J. Papon, Model predictive heuristic control," Automatica (Journal of IFAC), vol. 14, no. 5, pp. 413-428, 1978.

[11] J. Richalet and D. O'Donovan, Predictive Functional Control: Principles and Indus- trial Applications. Springer Science and Business Media, 2009.

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

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

[14] J. A. Rossiter and M. Abdullah, A new paradigm for predictive functional control to enable more consistent tuning," in 2019 American Control Conference (ACC). IEEE, Jul 2019.

[15] J.A. Rossiter and M.S. Aftab, A Comparison of Tuning Methods for Predictive Functional Control. Processes 2021, 9, 1140. https://doi.org/10.3390/pr9071140.

[16] J.A. Rossiter and M.S. Aftab, Exploiting Laguerre polynomials and steady-state estimates to facilitate tuning of PFC. ECC 2022.

[17] 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.

[18] M. Abdullah and J. A. Rossiter, Using Laguerre functions to improve the tuning and performance of predictive functional control," International Journal of Control, pp. 1-13, 2019.

[19] J.B. Rawlings, E.S. Meadows, K.R. Muske, Nonlinear Model Predictive Control: A Tutorial and Survey, IFAC Proceedings Volumes, Volume 27, Issue 2, 1994, Pages 185-197, ISSN 1474-6670, https://doi.org/10.1016/S1474-6670(17)48151-1.

[20] 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, 11-14 Jul 2021.