



The
University
Of
Sheffield.

Development and Analysis of Predictive Functional
Control for a Range of Dynamical Processes

Muhammad Bin Abdullah

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

Doctor of Philosophy

Automatic Control and Systems Engineering

University of Sheffield

November 2018

*Dedicated to my beloved parents, wife and son
for their continuous loves and support.*

ABSTRACT

This thesis presents a development and analysis of a low computation Model Predictive Control (MPC) that is known explicitly as Predictive Functional Control (PFC) for different types of dynamical processes. Since the current concept of PFC suffers from several issues such as the weak efficacy of tuning parameter, conservative constrained solution and poor handling of challenging dynamical systems, the prime objective of this research is to develop novel approaches to tackle these limitations while retaining the simplicity of formulation, coding and tuning of PFC. For the first contribution, a Laguerre based PFC (LPFC) is proposed to handle a system with stable and straightforward dynamics where it can provide better consistency between model predictions and actual system behaviour. Consequently, LPFC also improves the overall closed-loop performance including the efficacy of tuning parameter while providing more accurate and less conservative constrained solutions compared to the conventional PFC. For the second contribution, a Pole Shaping PFC (PS-PFC) is developed based on the original idea of the traditional pole cancellation technique to alleviate the effect of undesirable poles when dealing with a more challenging dynamical processes such as those with open-loop integrating, oscillating and unstable modes. This approach provides a stable response with less aggressive input demand compared to the pole cancellation method while retaining a similar recursive feasibility property of constraint handling. The third contribution of this research is on the development of an off-line sensitivity analysis to measure the robustness and possible sensitivity trade-off for different PFC structures in response to noise, disturbance and parameter uncertainty. The findings show that although LPFC provides better closed-loop performance than PFC, yet the control loop may become more sensitive to noise. On the other hand, for PS-PFC, since the conventional Independent Model (IM) structure is unable to handle a divergent prediction, a T-filter is proposed where it manages to recover the sensitivity to noise by sacrificing its sensitivity to disturbance and parameter uncertainty.

ACKNOWLEDGMENTS

I am highly indebted to my beloved father, Abdullah Bin Hasan, mother, Hashidah Binti Abdul Razak, and wife, Wan Nadzlia Shazwanie for all their support, sacrifice, encouragement, patience and prayer that have made me what I am today.

I would like to express my sincere sense of gratitude to my supervisor Dr John Anthony Rossiter for the continuous guidance, suggestions, encouragement and support throughout this research work. I also would like to extend my appreciation to my employer International Islamic University Malaysia and Ministry of Education Malaysia for the scholarship.

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 Bin Abdullah

Supervisor: _____

John Anthony Rossiter

CONTENTS

List of Figures	xi
List of Tables	xiv
List of Acronyms	xv
Chapter 1: INTRODUCTION	1
1.1 MOTIVATION	1
1.2 PROBLEM STATEMENT	3
1.3 RESEARCH OBJECTIVES	5
1.4 RESEARCH FLOW WITH MAIN CONTRIBUTIONS	6
1.5 THESIS ORGANISATION	7
Chapter 2: TECHNICAL DETAILS OF PFC	10
2.1 NOMINAL PFC CONCEPT	10
2.2 PFC WITH CASCADE STRUCTURE	18
2.3 SUMMARY	20
Chapter 3: LITERATURE REVIEW	22
3.1 MODEL PREDICTIVE CONTROL	22

3.2	CLASSICAL PID	25
3.3	PREDICTIVE FUNCTIONAL CONTROL	27
3.4	TUNING EFFICACY AND CONSTRAINT HANDLING OF PFC	30
3.5	HANDLING CHALLENGING DYNAMICAL PROCESSES	34
3.6	ROBUSTNESS OF PFC	37
3.7	SUMMARY OF LITERATURE REVIEW	41
Chapter 4:	POLE PLACEMENT PFC	44
4.1	THE CONCEPT OF PP-PFC	44
4.2	EXTENSION OF PP-PFC TO SYSTEMS WITH COMPLEX POLES	48
4.3	DEVELOPMENT OF PP-PFC USING REAL NUMBERS ALGEBRA	49
4.4	RESULTS	52
4.5	SUMMARY	56
Chapter 5:	LAGUERRE BASED PFC	57
5.1	TUNING WEAKNESS OF PFC	57
5.2	LAGUERRE BASED PFC	58
5.3	SYSTEMATIC CONSTRAINED HANDLING WITH LPFC	64
5.4	DIFFERENT PARAMETERISATIONS OF LPFC	71
5.5	LPFC FOR AN INTEGRATING PROCESS	77
5.6	SUMMARY	81

Chapter 6:	POLE SHAPING PFC	83
6.1	POLE CANCELLATION PFC (PC-PFC)	83
6.2	THE CONCEPT OF POLE SHAPING PFC (PS-PFC)	86
6.3	RESULTS	91
6.4	SUMMARY	92
Chapter 7:	SENSITIVITY ANALYSIS	96
7.1	SENSITIVITY ANALYSIS OF DIFFERENT PFC STRUCTURES	96
7.2	SENSITIVITY ANALYSIS OF LAGUERRE PFC	104
7.3	SENSITIVITY ANALYSIS OF PS-PFC WITH T-FILTER	111
7.4	SUMMARY	118
Chapter 8:	CONCLUSIONS AND FUTURE WORKS	119
8.1	FINAL CONCLUSIONS	119
8.2	RECOMMENDATION FOR FUTURE WORK	121
Bibliography		123
Appendix A:	Pole-placement Predictive Functional Control for Under-damped Systems with Real Numbers Algebra	134
Appendix B:	Utilising Laguerre Function in Predictive Functional Control to Ensure Prediction Consistency	135
Appendix C:	Development of Constrained Predictive Functional Control	

	using Laguerre Function Based Prediction	136
Appendix D:	Using Laguerre Functions to Improve the Tuning and Performance of Predictive Functional Control	137
Appendix E:	Alternative Method for Predictive Functional Control to Handle an Integrating Process	138
Appendix F:	Input Shaping Predictive Functional Control for Different Types of Challenging Dynamics Processes	139
Appendix G:	The Effect of Model Structure on the Noise and Disturbance Sensitivity of Predictive Functional Control	140
Appendix H:	A Formal Sensitivity Alysis for Laguerre Based Predictive Functional Control	141
Appendix I:	Sensitivity Analysis for an Input Shaping Predictive Functional Control for Challenging Dynamics Process	142

LIST OF FIGURES

2.1	Independent Model (IM) structure.	13
2.2	System step response overlaid with the target trajectory.	15
2.3	System prediction at the first sample instant.	15
2.4	System closed-loop response.	16
2.5	Schematic of PFC considering state or output constraints.	18
2.6	Transparent PFC structure.	19
4.1	Parallel model format alongside the actual process G_p	46
4.2	Unconstrained and constrained performances of PP-PFC with different ρ . . .	53
4.3	The experimental plant.	54
4.4	PP-PFC performance of the under-damped Quanser servo with flexible joint.	55
5.1	Comparison between the open-loop prediction at a first sample with the implied closed-loop behaviour of PFC for the process G_2 , G_3 and G_4 with varying n_y	59
5.2	The response of open-loop prediction at a first sample and the implied closed-loop behaviour of the process G_2 , G_3 and G_4 for nominal PFC and LPFC. . .	63
5.3	Tuning LPFC with different Laguerre pole a_l for the process G_3 and G_4	64

5.4	Constrained performance of PFC and LPFC for process G_2 with $\lambda = 0.7$ and $n_y = 1$	69
5.5	Performance of constrained CPFC and LPFC for process G_5	70
5.6	CPFC responses with different poles $\lambda_x = 0.984$ and $\lambda_x = 0.963$ for process G_5	71
5.7	Input and output prediction of PFC, LPFC ,LPFC2 for process G_6	75
5.8	Closed-loop response of PFC, LPFC ,LPFC2 for process G_6	76
5.9	Constrained performance of of PFC, LPFC ,LPFC2 for process G_6	77
5.10	Open-loop and closed-loop performance of TPFC and LPFC with different tuning for process G_7	80
5.11	Constrained performance of TPFC and LPFC with bounded output $\bar{y} = 0.8$	81
6.1	Input and output predictions with PFC, PC-PFC and PS-PFC for processes G_8, G_9, G_{10}	93
6.2	Closed-loop input and output behaviour of PC-PFC and PS-PFC for processes G_8, G_9, G_{10}	94
6.3	Constrained input and output behaviour of PFC, PC-PFC and PS-PFC for processes G_8, G_9, G_{10}	95
7.1	PFC with RM structure equivalent block diagram.	98
7.2	PFC prediction structure with T-filter.	99
7.3	PFC with T-filter equivalent block diagram.	100
7.4	PFC with IM structure equivalent block diagram.	102
7.5	Sensitivity analysis for process G_{11}	103

7.6	Sensitivity analysis for G_{12}	104
7.7	IM structure equivalent block diagram for PFC and LPFC.	105
7.8	Sensitivity plot for process G_{13} with $\lambda = 0.7$ and $n = 7$	107
7.9	Closed-loop response of process G_{13} with $\lambda = 0.7$ and $n = 7$ in the presence of disturbance, noise, and uncertainty.	108
7.10	Sensitivity plot for process G_{13} with $\lambda = 0.92$ and $n = 9$	109
7.11	Closed-loop response of process G_{13} with $\lambda = 0.92$ and $n = 9$ in the presence of disturbance, noise and uncertainty.	110
7.12	Sensitivity plots for G_{14} with different T-filters.	115
7.13	Sensitivity plots for G_{15} with different T-filters.	115
7.14	Closed-loop responses of G_{14} in the presence of noise, output disturbance and parameter uncertainty.	116
7.15	Closed-loop responses of G_{15} in the presence of noise, output disturbance and parameter uncertainty.	117

LIST OF TABLES

4.1	Computational loading for different realizations of PP-PFC for second order system.	52
4.2	Relative simulation times of the different realizations of PP-PFC for process G_1	54
4.3	Quantify PP-PFC performance for Quanser servo with flexible joint	55

LIST OF ACRONYMS

CARIMA Controlled Auto Regressive Integral Moving Average

CLTR Closed-loop Time Response

D.O.F Degrees Of Freedom

DMC Dynamic Matrix Control

FIR Finite Impulse Response

FOPDT First Order Plus Dead Time

GPC Generalised Predictive Control

IM Independent Model

LPFC Laguerre Predictive Functional Control

LQR Linear Quadratic Regulator

MIMO Multi Input Multi Output

MPC Model Predictive Control

ONEDOF One Degree of Freedom

PC-PFC Pole Cancellation Predictive Functional Control

PFC Predictive Functional Control

PID Proportional Integral Derivative

PLC Programmable Logic Control

PP-PFC Pole Placement Predictive Functional Control

PS-PFC Pole Shaping Predictive Functional Control

RM Realigned Model

SR Step Response

TPFC Transparent Predictive Functional Control

SISO Single Input Single Output

Chapter 1

INTRODUCTION

This chapter presents an overview of the research topic where Section 1.1 starts with the motivation behind this work and Section 1.2 introduces the problem statement to set up the prime research objectives, which are given in Section 1.3. The following Section 1.4 provides the research flow with a summary of contributions and Section 1.5 describes the thesis organisation together with the list of related publications.

1.1 MOTIVATION

Advancement in modern computation has triggered the use of Model Predictive Control (MPC) in widespread applications including automotive (Hrovat et al., 2012), chemical (Farina et al., 2016), alternative energy (Abdullah & Idres, 2014b) and others. The underlying concept is that at each sampling time, MPC will minimise a quadratic cost function and provide an optimal control solution. This controller offers systematic frameworks to handle systems with constraints, delays, many outputs and challenging dynamics (J. A. Rossiter, 2018; L. Wang, 2009). Because of its appealing characteristics, MPC has become one of the major subjects in the current scientific research. This past decade has seen the rapid development of MPC control theories in many areas such as how to improve the controller stability (Mayne et al., 2000), feasibility (Giselsson & Rantzer, 2010), robustness (Bemporad & Morari, 1999), and also accuracy (Qin & Badgwell, 2000).

Nevertheless, the implementation of MPC in real applications is more expensive compared to other traditional simple controllers, considering it requires a higher computation demand and expensive computer hardware for modelling and optimisation processes

(J. A. Rossiter & Haber, 2015; Vazquez et al., 2014; Jones & Kerrigan, 2015; Dovžan & Škrjanc, 2010). This scenario becomes worse if the number of constraints within a system increase as it needs to ensure recursive feasibility throughout a validation horizon (J. A. Rossiter, 2018; Mayne et al., 2000; Rawlings & Muske, 1993). Although the tuning of MPC is easier than other classical controllers, yet the selection of weights in the cost function still requires trial and error procedure, which is not entirely transparent especially to the less experienced users. Hence, proper training to understand the effect of tuning parameters on the system behaviour and overall controller frameworks are often needed (Qin & Badgwell, 1997) and these requirements are indeed time consuming. Despite its vast theoretical developments, MPC applications are still largely restricted to high-end, complex or large unit operations (Qin & A. Badgwell, 2003; Mayne, 2014) where the optimality of solutions are necessary.

Conversely, in many small-scale industries, the majority of staff prefer a simpler controller such as Proportional-Integral-Derivative (PID) (Åström & Hägglund, 2001) although there exist other advanced controllers such as Linear Quadratic Regulator (LQR) (Kwakernaak & Sivan, 1972), Sliding Mode Control (SMC) (Utkin, 1993) and others. The main factor behind this is because of their limited understanding and experiences of other controller's principle (J. A. Rossiter & Haber, 2015). Although PID frameworks are well established, the users may face difficulties in handling constrained systems. For example, the usage of an integrator during constraint violations can produce wind-up and/or saturation (J. A. Rossiter, 2018). Some may argue that anti-wind-up techniques can prevent this limitation (Visioli, 2006), but it often requires a well-trained user to implement this ad-hoc tuning rule (De Doná et al., 2000) while these are difficult to design and manage for different combinations of system dynamics and constraints.

From the practical perspective, it is clear that many industrial end-users are willing to trade between optimality of control solutions with ease/cost of implementation, especially for a simple application (Richalet & O'Donovan, 2009). This preference has triggered the widespread acceptance of Predictive Functional Control (PFC), which was developed by Richalet around the 1960s and early 1970s (Richalet et al., 1978). The main difference between PFC and MPC is on the minimisation process, where the objective is to track a

desired target trajectory at a specific point instead of finding an optimal solution. Besides, a constant future input assumption of PFC (Richalet & O'Donovan, 2009) reduces the overall complexity of coding, and consequently, it only uses minimal computation, and indeed, for low order models, the coding is almost trivial and can be implemented on a modest processor, including a Programmable Logic Controller (PLC) (Khadir & Ringwood, 2008). The use of settling time or closed-loop time constant as a tuning parameter makes the designing process more transparent, especially to the operating technician (Hashizume, 2015). PFC also retains some of the advantages of MPC namely the constraints and delays managements (Richalet & O'Donovan, 2009). Taken together, these advantages highlight the potential of PFC to replace PID controller, especially for Single Input Single Output (SISO) applications (J. A. Rossiter, 2018; Richalet & O'Donovan, 2009; Hashizume, 2015).

Surprisingly, PFC only receives comparatively little attention in the academic literature despite its wide acceptance in the industry due to its difficulty in providing rigorous properties like stability assurance and robust feasibility except for a few cases (J. A. Rossiter, 2017). As far as the author is concerned, there is limited formal analysis of PFC performance in the literature except for few papers (J. A. Rossiter & Haber, 2015; J. A. Rossiter, 2016; J. Rossiter et al., 2016) and books (Richalet & O'Donovan, 2009; J. A. Rossiter, 2018; Haber et al., 2012). General studies in this field are mainly focused on the implementation aspect in specific applications (Skrjanc & Matko, 2000; Yang et al., 2005; Yiming & Bin, 2012) rather than improving the control law itself. Besides, most of the conventional methods that used to improve the reliability of PFC (Richalet & O'Donovan, 2009) often lack academic rigour (J. A. Rossiter, 2016; Valencia-Palomo & Rossiter, 2011). This inadequacy indicates a need for more systematic analysis and development of PFC, which can provide useful contributions to both theoretical frameworks and practical implementations.

1.2 PROBLEM STATEMENT

From the previous discussion, it is clear that PFC concept can provide a simple approach along with other potential benefits. Besides, this controller also has been successfully implemented in various small-scale industrial applications as reported in Richalet & O'Donovan (2009). Nevertheless, this thesis would like to point out that the current PFC

concept also may suffer from several weaknesses in providing a satisfactory control performance. For example:

- There is a tuning issue with PFC where the use of constant future input assumption could lead to ill-posed solutions. Specifically, the open loop prediction and closed-loop behaviour are inconsistent due to insufficient degrees of freedom (d.o.f) especially when using a small coincidence horizon n_y (J. A. Rossiter & Haber, 2015; J. A. Rossiter, 2016). This situation may reduce the accuracy of constrained solutions while making it very conservative when satisfying the output or state constraints. Conversely, if a larger n_y is used, better prediction consistency can be obtained, but the effectiveness of desired closed-loop pole λ as the tuning parameter becomes less significant. Hence, this scenario triggers the need for an alternative future input parameterisation that can provide more d.o.f to eliminate this limitation. Besides, as will be discussed in the following chapter, the current PFC approach for handling soft constraints by switching the input between multiple regulators (Richalet & O'Donovan, 2009) is obsolete and not systematic, as most of the modern constraint approaches are often embedded inside the controller. It is notable that the concept was developed back in the 70s where the computation ability was not as advanced as today. However, as time passed, PFC practitioners are still comfortable in using this old method. Theoretically, there is a possibility to adopt a similar systematic constraint approach as in the modern MPC.
- Although the implementation of PFC is straightforward for simple dynamic systems (J. A. Rossiter & Haber, 2015), a user may face a difficulty in tuning a more challenging dynamic process such as those with integrating, oscillating or unstable mode mainly because of its poor open-loop prediction (J. A. Rossiter & Haber, 2015; J. A. Rossiter, 2016, 2002; Rawlings & Muske, 1993). The default PFC concept is unable to cope with these dynamics and often requires some modification in its control law. In practice, a cascade-like structure or decomposition model as suggested in Richalet & O'Donovan (2009) are widely used, where it will stabilise the open-loop process before implementing the standard control law. However, there are several limitations for these methods and in fact, there is no clear explanation on how to systematically tune the controller which leads to a counter-intuitive implementation of PFC. Thus, a simple concept of

PFC that can adapt to these challenges while retaining its simplicity of tuning and other advantages would be a worthwhile contribution.

- Another issue that needs attention is regarding the sensitivity of the PFC prediction structure. Embedding a robust formal design as in MPC will require a more complex algorithm and implementation which runs against the core principle of PFC. The common practice is to employ an Independent Model (IM) structure (Richalet & O'Donovan, 2009) for unbiased prediction. However, this structure is not implementable with an open loop divergent process (J. Rossiter et al., 2016) unless the cascade structure is implemented as it can retain the stability of IM. Conversely, it is also worth to consider other prediction structures such as a Realigned Model (RM) (similar to a CARIMA model in GPC) while improving its robustness via a simple T-filter. Thus, a formal off-line sensitivity analysis for PFC is essential to get an insight into what sort of sensitivity trade-offs one should expect when using different prediction structures. Besides, the same analysis can be used to compare and measure the robustness of other proposed modifications to the PFC control law.

1.3 RESEARCH OBJECTIVES

Based on the problem statements, the objectives of this work are established as :

1. To design a novel PFC method that can improve the prediction consistency, tuning process, and constraint handling.
2. To develop novel PFC approaches which can deal with challenging dynamic systems while retaining the ease of coding, implementation and also recursive feasibility constrained solutions.
3. To provide an off-line systematic sensitivity analysis for different PFC structures and explore the possibility to adopt other alternative structures to improve its robustness.

1.4 RESEARCH FLOW WITH MAIN CONTRIBUTIONS

After a comprehensive literature review which will be presented in Section 3, this work tackles each of the listed objectives separately. For the first objective, this work only focus on a simple and straightforward dynamical system, where the main contributions are:

1. Development of Laguerre based Predictive Functional Control (LPFC) specifically for a straightforward and stable process, which provides better prediction consistency and tuning efficacy compared to the conventional PFC while retaining the simplicity of coding and implementation.
2. Development of a more systematic and cost-effective constraint handling method for LPFC that provides more accurate and less conservative constrained solutions compared to the traditional multiple regulators approach.
3. A formal comparison of closed-loop and constraint handling performances between the traditional PFC concept and LPFC with different input parametrisations (through inputs and input increments) while identifying the potential weaknesses for these approaches.

For the second objective, this work develops and analyses the performance of three different alternative PFC structures specifically to handle challenging dynamic systems and the contributions are:

1. Extending the concept of LPFC to handle an integrating process and comparing the tuning, performance and constraint handling ability with the traditional cascade structure while identifying the possible limitations for both approaches.
2. Development of the newly introduced concept of Pole Placement PFC by J. Rossiter et al. (2016) for handling an oscillating process without the use of imaginary number via simple algebraic manipulation while discovering its advantages and disadvantages.

3. Development of a Pole Shaping PFC (PS-PFC) by adopting the concept of Pole Cancellation PFC of J. A. Rossiter (2016) to handle different types of challenging dynamical systems where it can provide a stable performance with less aggressive input demand while retaining a similar recursive feasible constrained property.

In order to achieve the third objective, this work provides a systematic sensitivity analysis for PFC, LPFC and PS-PFC. Another three contributions can be listed as follow:

1. Derivation of an off-line sensitivity analysis for different types of PFC structures for comparing the robustness between IM and RM predictions in the presence of disturbance and noise while introducing the use of T-filter with the RM to get the best trade-off between the two structures.
2. A formal robustness analysis for the proposed LPFC and traditional PFC to understand the possible sensitivity trade-off in the presence of noise, disturbance and parameter uncertainty.
3. Development of a robust prediction structure for PS-PFC by utilising a T-filter to improve its robustness against noise while analysing the possible sensitivity trade-off with disturbance and parameter uncertainty.

1.5 THESIS ORGANISATION

This thesis consists of eight main chapters. Chapter 1 introduces an overview of this work wherein motivation, problem statement, objectives and main contributions are explained. Chapter 2 provides the technical details of PFC and continues with Chapter 3, which reviews the works of literature that are related to this research. For clarity of presentation, Chapter 4, 5, and 6 are divided according to the types of PFC that have been developed rather than following the research flow (as discussed in Section 1.4) since different types of formulations are used for different approaches. Thus, each of these chapters present the development of Pole Placement PFC, Laguerre based PFC and Pole Shaping PFC, respectively. The following Chapter 7 analyses the sensitivity of different PFC prediction

structures and Chapter 8 provides the conclusion with possible future works. Since most of the contributions in this thesis are already published or accepted in journals or conference proceedings, only a summary and the main highlights are presented in Chapter 4, 5, 6 and 7. More detail descriptions and derivations of the proposed concept can be found from these papers which have been attached in the Appendix section.

Appendix A (Zabet et al., 2017) (published),

Zabet, K., Rossiter, J. A., Haber, R. & **Abdullah, M.** (2017). Pole-placement Predictive Functional Control for under-damped systems with real numbers algebra, *ISA Transaction*, 71, 403–414.

Appendix B (Abdullah & Rossiter, 2016) (published),

Abdullah, M., & Rossiter, J. A. (2016). Utilising Laguerre function in Predictive Functional Control to ensure prediction consistency. In *2016 11th UKACC International Conference on Control* (pp. 1–6).

Appendix C (Abdullah et al., 2017) (published),

Abdullah, M., Rossiter, J. A., & Haber, R. (2017). Development of constrained Predictive Functional Control using Laguerre function based prediction. *IFAC-PapersOnLine*, 50(1) (pp. 10705–10710).

Appendix D (Abdullah & Rossiter, 2019b) (accepted),

Abdullah, M., & Rossiter, J. A. (2019b). Using Laguerre functions to improve the tuning and performance of Predictive Functional Control. *Accepted by International Journal of Control*.

Appendix E (Abdullah & Rossiter, 2018a) (accepted),

Abdullah, M., & Rossiter, J. A. (2018a). Alternative method for Predictive Functional Control to handle an integrating process. in *Proceedings of 2018 12th UKACC International Conference on Control*.

Appendix F (Abdullah & Rossiter, 2018d) (published),

Abdullah, M., & Rossiter, J. A. (2018d). Input shaping Predictive Functional Control for different types of challenging dynamics processes,” *Processes*, 6(8),118.

Appendix G (Abdullah & Rossiter, 2018b) (in press),

Abdullah, M., & Rossiter, J. A. (2018b). The effect of model structure on the noise and disturbance sensitivity of Predictive Functional Control,” in *Proceedings of 2018 European Control Conference*.

Appendix H (Abdullah & Rossiter, 2018c) (in press),

Abdullah, M., & Rossiter, J. A. (2018c). A formal sensitivity analysis for Laguerre based Predictive Functional Control,” in *Proceedings of 2018 12th UKACC International Conference on Control*.

Appendix I (Abdullah & Rossiter, 2019a) (submitted),

Abdullah, M., & Rossiter, J. A. (2019a). Sensitivity analysis for an input shaping Predictive Functional Control for processes with challenging dynamics,” in *Submission to 2019 3rd IEEE Conference on Control Technology and Applications*.

Remark 1.1. *It should be noted that most of the results in this thesis will use a qualitative analysis instead of quantitative for ease of presentation and explanation. Adding qualitative analysis will not give any extra information to the reader since most of the simulations are using arbitrary systems in a discrete domain except for a few cases where it seems appropriate.*

Chapter 2

TECHNICAL DETAILS OF PFC

Before presenting the literature review, it is better for readers to have some technical insight into the traditional PFC framework. Hence, this chapter provides two brief formulations of PFC that will be used or benchmarked in this work. Section 2.1 introduces the nominal PFC concept together with its traditional constraint handling technique. Section 2.2 describes the cascade structure of PFC that is specifically known as transparent control and often used for handling integrating systems or other challenging dynamical processes. The final Section 2.3 summarises the main highlights of this chapter.

2.1 NOMINAL PFC CONCEPT

A fundamental concept in any predictive controller follows a simple human intuition, where one would need to anticipate a future behaviour far in advance to select a required control action to achieve the desired goal while taking into account the associated limitations and other uncertainties. In general, PFC has different formulations and representations that depends on what sort of objective that it wants to achieve or types of a prediction model that is used. For example, the formulation for tracking a ramp or parabola set-point is different from tracking a constant set-point. Thus, the nominal formulation that will be presented in this section is considered as the most basic form of PFC and only can be used explicitly for tracking a constant set point. This formulation is divided into several major components namely the target trajectory, system prediction, control law, and constraint handling.

2.1.1 Target Trajectory

In principle, the target trajectory is a time-based variable that describes the desired closed-loop behaviour before it converges to the final set-point. Although there is no restriction on the choice of a function for this trajectory, in practice a user often utilises a simple exponential due to its simplicity of calculation in real-time and predictable convergence rate (Richalet & O'Donovan, 2009). This preference keeps the coding and implementation of PFC straightforward and easy to understand which leads to its well-accepted concept in the industry.

For transparency of presentation, consider the first-order system with constant set point R and without delay:

$$r(s) = \frac{1}{T_r s + 1} R; \quad r(z) = \frac{(1 - \lambda)}{1 - \lambda z^{-1}} R \quad (2.1)$$

where $r(s)$ and $r(z)$ are the continuous and discrete form, respectively. From here, a user can easily select a desired convergent rate for the response based on the time constant T_r . In industrial practice, a Closed-loop Time Response (CLTR) is often used to define T_r where technically, it represents the time taken to reach 5% of the desired set-point ($T_r = CLTR/3$). Notably, the parameter T_r also can be represented in the discrete form as the desired closed-loop pole $\lambda = e^{-\frac{T_s}{T_r}}$, where T_s is the sampling time. For clarity of presentation, this work uses λ as the tuning parameter instead of CLTR to avoid the conversion.

Defining $y_{p,k}$ as the current output measurement from a plant, the prime control objective of PFC is to force its prediction at k sample into the future, $y_{p,k+n_y|k}$ to match the target response at a single sample instant in the future, which is known as the coincidence horizon n_y (the second tuning parameter). Conceptually, the idea is equivalent to enforcing the following discrete equality:

$$y_{p,k+n_y|k} = r_{k+n_y|k} = (1 - \lambda^{n_y})R + \lambda^{n_y}y_{p,k} \quad (2.2)$$

where the reference trajectory is re-initialised at each sample to embed the feedback loop.

2.1.2 Model Prediction and Structure

Notably, the PFC concept is quite flexible and can take any form of prediction model ranging from a simple Finite Impulse Response (FIR), transfer function, or state space, where it is well known that each of these models has its pros and cons (J. A. Rossiter, 2018). The formulation in this thesis is constructed with a general transfer function as this model is commonly used in many PFC applications. Besides, it only requires a small number of parameters compared to the FIR and is suitable enough to represent a SISO process (where generally PFC concept will work well) without the need for an observer as with a state space model.

For a comprehensive representation of different order process dynamics, the derivation of n_y -step ahead prediction can be formed based on Toeplitz-Hankel forms (J. A. Rossiter, 2018). For a general polynomial $f(z) = f_0 + f_1z^{-1} + \dots + f_nz^{-n_y}$, it can be defined via future matrix C_f and past matrix H_f as:

$$C_f = \begin{bmatrix} f_0 & 0 & 0 & \cdots \\ f_1 & f_0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ f_{n_y} & f_{n_y-1} & f_{n_y-2} & \cdots \end{bmatrix}, H_f = \begin{bmatrix} f_1 & f_2 & \cdots & f_{n_y} \\ f_2 & f_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad (2.3)$$

Given a general transfer function model with dimension of n_b (numerator) and n_a (denominator):

$$y_m(z) = \frac{b(z)}{a(z)}u(z) \quad (2.4)$$

the simplified prediction structure for inputs u_k and outputs $y_{m,k}$ at sample k can be formulated as:

$$y_{m,k+n_y|k} = \underbrace{C_a^{-1}C_b}_{H} u_{\rightarrow k} + \underbrace{C_a^{-1}H_b}_{P} u_{\leftarrow k} + \underbrace{C_a^{-1}H_a}_{Q} y_{m,\leftarrow k} \quad (2.5)$$

where matrix H , P , Q depend on model parameters and the definition of $u_{\rightarrow k}$, $u_{\leftarrow k}$, $y_{m,\leftarrow k}$ are

given as:

$$\vec{u}_k = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+n_y-1} \end{bmatrix}; \overleftarrow{u}_k = \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \\ u_{k-n_b} \end{bmatrix}; \overleftarrow{y}_{m,k} = \begin{bmatrix} y_{m,k} \\ y_{m,k-1} \\ \vdots \\ y_{m,k-n_a} \end{bmatrix} \quad (2.6)$$

Remark 2.1. *There are some subtleties to ensure offset-free tracking in the presence of measurement noise, disturbance and parameter uncertainty which can be deployed either via Realigned Model (RM) or Independent Model (IM) structure (Richalet & O'Donovan, 2009). RM structure measures the past output directly from a plant through the stored memory in a processor, while IM calculates the same parameter independently through the model computation. For introducing the concept, the formulation used here is the IM structure as it is considered as a standard design in many PFC applications (Richalet et al., 1978; Richalet & O'Donovan, 2009).*

Figure 2.1 illustrates the schematic structure of IM where the process model G_m and plant G_p run in parallel. The correction term d_k is calculated at each time step as:

$$d_k = y_{p,k} - y_{m,k} \quad (2.7)$$

where $y_{m,k}$ and $y_{p,k}$ denote the output from model and plant, respectively. Hence, the plant prediction which in general is explicitly unknown can now be represented with the unbiased model prediction by adding the correction term d_k into the prediction (2.5) as:

$$y_{p,k+n_y|k} = H\vec{u}_k + P\overleftarrow{u}_k + Q\overleftarrow{y}_{m,k} + Ld_k \quad (2.8)$$

where L is a unit vector as d_k is assumed to be constant throughout the prediction window.

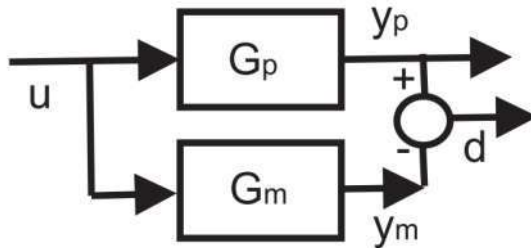


Figure 2.1: Independent Model (IM) structure.

Remark 2.2. *The use of IM structure in predictive control is originated from the concept of Smith predictor and Independent Model Control (IMC) (Garcia & Morari, 1982). The main advantage is that it can take any form of prediction model without the need of an observer or filter. The term d_k in (2.7) is representing the Process Model Error (PME) to ensure that the internal model of a system provides non-biased predictions while avoiding offset error (Richalet & O'Donovan, 2009; J. A. Rossiter, 2018).*

2.1.3 PFC Control Law

After defining the target trajectory and system prediction, the next step is to derive the control law. As pointed out before, there are only two tuning parameters in PFC which are the desired closed-loop pole λ and the coincidence horizon n_y . The λ is selected between the range of $0 < \lambda < 1$ to determine the convergence rate to the steady state input R . Smaller value provides faster convergence and vice versa. For the coincidence horizon, ideally one may desire a smaller n_y to drive the output response as close as possible to the target trajectory. However, this is not always the best tuning practice as the user also needs to weigh other factors such as the aggressiveness of input and stability of performance, especially when dealing with a higher order or challenging dynamical processes (J. A. Rossiter, 2002; J. A. Rossiter & Haber, 2015).

With suitable tuning parameters, the control law is then computed by forcing the system prediction to match the target trajectory specifically at n_y step ahead. Extracting the n_y^{th} row from matrix H, P, Q , the control input is solved by substituting (2.8) into (2.2). In order to simplify the calculation process without the matrix inversion, PFC assumes a constant future input, namely $u_{k+i|k} = u_k, i = 0, \dots, n_y$. In consequence, it simplifies H_{n_y} to single parameter $h_{n_y} = \sum H_{n_y}$ and (2.2) becomes:

$$h_{n_y} u_k + P_{n_y} \underline{u}_k + Q_{n_y} \underline{y}_{m,k} + d_k = (1 - \lambda^{n_y})R + \lambda^{n_y} y_{p,k} \quad (2.9)$$

The required control input can be easily solved as:

$$u_k = \frac{(1 - \lambda^{n_y})R + \lambda^{n_y} y_{p,k} - (P_{n_y} \underline{u}_k + Q_{n_y} \underline{y}_{m,k} + d_k)}{h_{n_y}} \quad (2.10)$$

then, the same process is repeated to update the value of u_k at each time step.

2.1.4 Simple Illustration

In order to illustrate the PFC concept, Figure 2.2 shows the step response of an arbitrary second-order over-damped system overlaid with the desired target trajectory r with $\lambda = 0.7$ (for this case the final set point is 1).

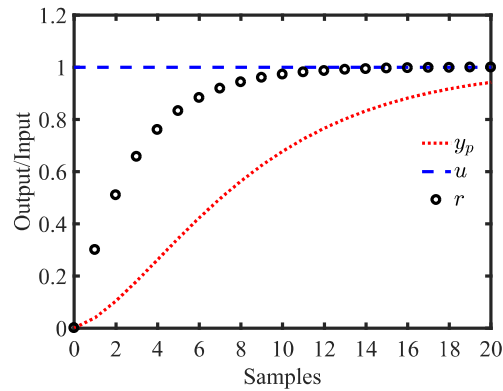


Figure 2.2: System step response overlaid with the target trajectory.

Figure 2.3 demonstrates that the implied output predictions at the first sample $y_{p|1}$ is forced to coincide with the trajectory r at a selected horizon $n_y = 4$ with the assumed constant future input at the first sample $u_{|1} = 2.8$. Based on a receding horizon principle as in MPC (L. Wang, 2009; J. A. Rossiter, 2018), PFC only send the first computed input to the plant and the same process is repeated at the next time step.

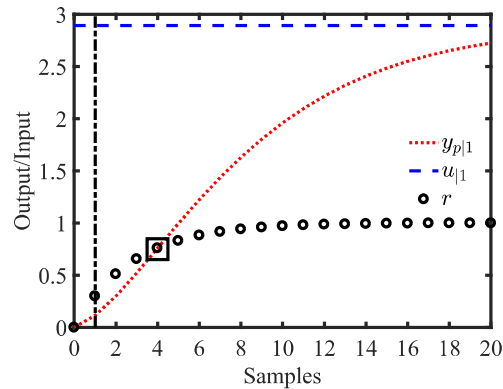


Figure 2.3: System prediction at the first sample instant.

Figure 2.4 presents the closed-loop performance of the system where the output y_p converges to the desired steady-state set-point with a close enough dynamics compared to the desired trajectory r .

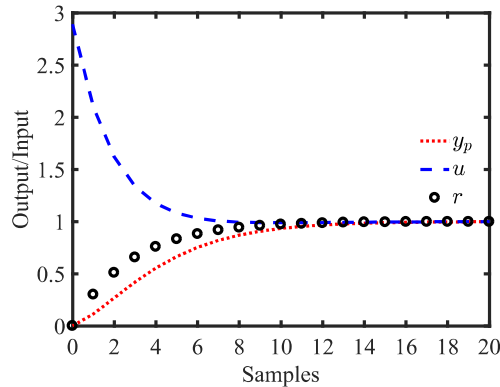


Figure 2.4: System closed-loop response.

2.1.5 Constraints Handling

One of the key selling points of PFC is the constraint handling ability (Hashizume, 2015) where generally, the constraints can be implemented to the input, rate, state and output. The first two constraints are known as hard constraints that often employed due to the physical limitations of the process hardware. Violating these limits may reduce the component lifespan or at worst damaging the process itself. The other two constraints are defined as soft constraints which can be ignored sometimes but may affect the controller performance and economic profit. Effectively satisfying these constraints may offer many attractive benefits such as higher production profit, better control performance, lower maintenance cost and safer control environment (J. A. Rossiter, 2018; Richalet & O'Donovan, 2009; L. Wang, 2009; Abdullah & Idres, 2014a).

The traditional PFC handles the hard and soft constraints using different approaches. For input and rate limits, the inequalities can be represented as:

$$\underline{u} \leq u_k \leq \bar{u} \quad (2.11)$$

$$\underline{\Delta u} + u_{k-1} \leq u_k \leq \overline{\Delta u} + u_{k-1} \quad (2.12)$$

where $\underline{\Delta u}$ and $\overline{\Delta u}$ are the minimum and the maximum rate, while \underline{u} and \bar{u} denote the minimum and maximum input, respectively. Without explicitly including these constraints in the control computation, a clipping method can be utilised (Abu et al., 1991). When the limit in (2.11) or (2.12) is violated, the controller will treat it as an equality constraint (Richalet & O'Donovan, 2011).

Remark 2.3. *The input and rate constraint need only be implemented on the current input within conventional PFC. Although the constrained solutions are not optimum, the recursive feasibility property can be guaranteed due to the constant future input assumption (Richalet & O'Donovan, 2009). This type of solution is sufficient enough particularly for stable and straightforward dynamical systems.*

For state and output constraints, PFC employs multiple regulators, which run in parallel as shown in Figure 2.5. Each of this regulators uses different tuning parameters of λ and n_y (Richalet & O'Donovan, 2009; Abu et al., 1991).

- The first regulator PFC_1 is the preferred control law and produces input $u_{1,k}$ using (2.10) to track the set point while satisfying the hard constraints. Within some validation horizon n_i to be defined, the supervisor uses input $u_{1,k}$ to predict the future state behaviour using a prediction model such as (2.5). If the state predictions are within their limit, then use $u_k = u_{1,k}$.
- The second regulator PFC_2 is more conservatively tuned to track the state limit by manipulating input $u_{2,k}$. When the state limit is expected to be violated using PFC_1 , then use $u_k = u_{2,k}$.
- An advanced decision-making method such as fuzzy logic, look-up table, or artificial neural network may be utilised for a smoother transition (Richalet & O'Donovan, 2009).

For ease of presentation, this formulation considers state constraints although similar approach can be used when satisfying output constraints. The second controller PFC_2

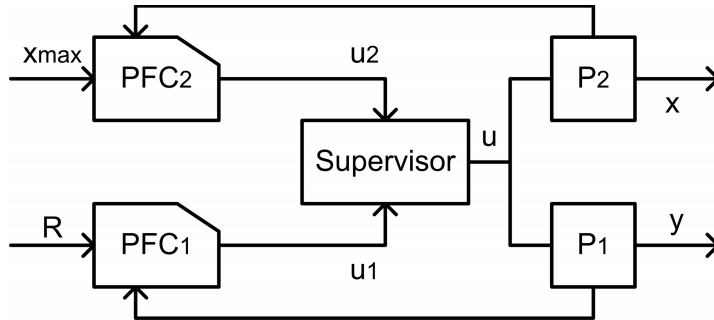


Figure 2.5: Schematic of PFC considering state or output constraints.

regulates the second input $u_{2,k}$ based on a state prediction equation:

$$x_{k+n_x|k} = h_{n_x} u_k + P_{n_x} \underline{u}_k + Q_{n_x} \underline{x}_k + d_k \quad (2.13)$$

where $h_{n_x}, P_{n_x}, Q_{n_x}$ denote the state model parameters and in this case, the correction term $d_k = x - x_m$ (x_m denotes the model used to calculate the state). The maximum state limit \bar{x} is a set as a target instead of R . With a suitable coincidence horizon n_x and the desired closed loop pole λ_x , input $u_{2,k}$ is computed as:

$$u_{2,k} = \frac{(1 - \lambda_x^{n_x})x_{max} - \lambda_x^{n_x} x_k - (P_{n_x} \underline{u}_k + Q_{n_x} \underline{x}_k + d_k)}{h_{n_x}} \quad (2.14)$$

It is also important that the associated PFC is tuned, if possible, to avoid oscillations or any overshoot in the predictions for satisfying the limits.

Remark 2.4. *A suitable validation horizon n_i for checking the predictions associated to PFC_1 should be used since the projection of $u_{1,k}$ must include the open loop time response of PFC_2 . In addition, the target pole λ_x of PFC_2 must be compatible with the need to satisfy the internal constraints of PFC_1 . Choosing a fast pole to improve the overall system response may decrease the controller robustness and introduce conflicts with the hard constraints (Richalet & O'Donovan, 2009).*

2.2 PFC WITH CASCADE STRUCTURE

In order to handle a system with marginally stable or oscillating dynamics, PFC practitioners often employ a two level cascade structure that is known as transparent control

(see Figure 2.6). The inner loop employs a proportional gain with negative feedback to stabilise the open loop prediction, while nominal PFC controls the outer loop and eliminates any offset due to disturbance while enhancing the overall dynamic performance (Richalet & O'Donovan, 2009).

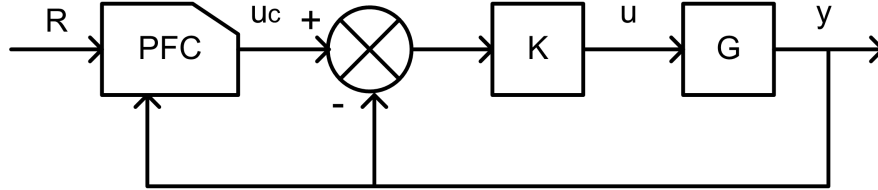


Figure 2.6: Transparent PFC structure.

The inner loop with gain K will be used as a prediction model by PFC instead of the original plant model G to compute the manipulated input u_c . Given the pre-stabilised model in continuous form as:

$$y(s) = \frac{GK}{1 + GK} u_c(s) \quad (2.15)$$

PFC in the outer loop will compute the input $u_{c,k}$ using the standard formulation as discussed in Section 2.1.3. Then, the actual input u that will be sent to the plant is defined as:

$$u_k = K(u_{c,k} - y_k) \quad (2.16)$$

With this technique, the controlled system is able to maintain regulation during set-point changes by introducing a temporary over-compensated set-point (Richalet & O'Donovan, 2009). Besides, due to the pre-stabilisation, this approach can use the standard IM structure effectively to handle noise, disturbance and parameter uncertainty even for an open-loop divergent process.

Remark 2.5. *Transparent PFC (TPFC) only accepts proportional gain rather than the combination with integral and/or derivative to keep the constraint implementation purely algebraic (Richalet & O'Donovan, 2009). To implement input or rate constraints, it is crucial for the model to detect possible constraint violations a priori. Thus, a back calculation procedure is needed to transfer the information from the inner loop to the outer loop as:*

$$y_k + \frac{u_{min}}{K} \leq u_{c,k} \leq y_k + \frac{u_{max}}{K} \quad (2.17)$$

$$y_k + \frac{\Delta u_{min} + u_{k-1}}{K} \leq u_{c,k} \leq y_k + \frac{\Delta u_{max} + u_{k-1}}{K} \quad (2.18)$$

Failure to do this could introduce an overshoot in the input (and/or output) due to a mismatch between the predicted model behaviour and the actual system behaviour.

Remark 2.6. *For output or state constraints, the multiple-regulators approach as discussed in Section 2.1.5 is used to satisfy the limits (Richalet & O'Donovan, 2009). For this case, the second controller needs to be tuned more carefully to avoid conflict with the hard constraints.*

2.3 SUMMARY

Based on the presentation of nominal and cascade formulations of PFC, there are several points that worth to highlight:

1. It is clear that the nominal control law of PFC is straightforward and can be coded in just a few lines while only requires low computation demand.
2. The use of λ as one of the tuning parameters of PFC makes the concept intuitive and transparent especially to the industrial user since this parameter has a direct relationship with the desired closed-loop time repose that one would like to achieve.
3. Generally, PFC can use any form of prediction model. Since this controller is mostly effective in controlling SISO system instead of MIMO, this work adopts a transfer function model where it requires fewer parameters than the FIR model (Ljung, 1998) and can avoid the use of an observer as with the state space model. The advantages and disadvantages of these models will be further discussed in the next chapter.
4. Although PFC uses IM as a standard prediction structure, this work will also consider the use of a RM structure for a specific case, especially when dealing with challenging dynamical applications which will be discussed later on in the upcoming chapter.
5. The constraint handling approach of PFC that is based on multiple regulators is obsolete and may become very conservative since the second regulator is tuned with a slower pole for avoiding conflict with the hard constraints.

6. The concept of transparent control to handle a marginally stable or oscillating process seems promising. Nevertheless, the restriction in using a proportional only controller in the inner loop to simplify the constraint handling may limit its reliability to pre-stabilised an open-loop unstable process.

Overall, this chapter provides a brief insight into the technical details of PFC formulations namely the nominal and transparent PFC. A more detailed review of the tuning procedure, alternative structures and issues faced by PFC will be discussed in the next chapter.

Chapter 3

LITERATURE REVIEW

This chapter starts with a general review of Model Predictive Control in Section 3.1. The following Section 3.2 is devoted to the classical concept of PID controllers that are often used in many industrial applications. Section 3.3 discusses the benefits and limitations of PFC along with its latest development in various types of applications. Next, significant issues of PFC in managing the tuning parameters, challenging dynamical processes, and robustness are reviewed in Section 3.4, 3.5 and 3.6, respectively. The final Section 3.7 provides a summary of this chapter to consolidate the findings and suggestions on how to tackle the listed issues.

3.1 MODEL PREDICTIVE CONTROL

Nowadays, automatic control has become a necessary tool in most of the industrial applications in order to keep all the processes running smoothly according to the desired specifications within a safe working environment. Besides, implementing self-control machines can increase the production rate while reducing human resources and the operation cost. There are many types of controllers available in the market and literature that may be generally classified either via linear or non-linear, stochastic or robust, discrete or continuous, modern or classical and others. Each types of these controllers has their advantages and disadvantages depending on how or where it is implemented. Selecting the right type of controller for a specific application is an essential task for an engineer, where they need to weigh several aspects such as computation time, tuning process, process dynamics, constraint handling and others. One of the preferred types of controllers that can satisfy most of the mentioned requirements is a Model Predictive Controller.

Model Predictive controller (MPC) can be referred to as a type of controller that uses an implicit model for prediction, solving the control action based on a specific objective function and applying a receding horizon concept where only the first computed output is sent to the plant while the whole process is repeated at each time step (Camacho & Bordons, 2012). This controller offers many attractive benefits to a user, for example, the ability to handle constraints in a systematic fashion (Garcia et al., 1989) where an optimal input can be computed to satisfy the set point while considering the associated physical and performance limitations by solving a specific cost function. Other than that, MPC also can handle different types of systems including those with long time delay, non-minimum phase or unstable mode due to its flexibility in selecting different prediction model and objective function (L. Wang, 2009). Besides, its flexibility also provides a natural extension to non-linear control or hybrid concept that can utilise a modern approach such as fuzzy logic and artificial neural network (Tatjewski, 2007). The tuning procedure can be considered more straightforward compared to the classical PID, which is based on the selection of weights in the cost function rather than requiring a complete off-line frequency analysis.

There are many variants in the MPC framework depending on what type of model and objective function are used. Most of the early MPCs employed a discrete linear model and specially designed for SISO applications such as Finite Impulse Response (FIR) or step response (SR) as in Dynamic Matrix Control (DMC) (Cutler & Ramaker, 1980) and transfer function as in Generalized Predictive Control (GPC) (Clarke & Mohtadi, 1989). The modelling process of DMC is very straightforward, yet it may require a high number of parameters. Conversely, GPC only needs a small number of parameters for the modelling process, but it is less straightforward to extend the concept for handling MIMO processes. Nevertheless, these two classic MPCs have been successfully implemented in various industrial applications as reported in many references (Camacho & Bordons, 2012; J. A. Rossiter, 2018; Tatjewski, 2007).

With the advancement in computation power together with its significant drop in the price and size, academics started to implement a state space model in MPC, where it is more convenient to generalise its overall concept for formal analysis and extension to MIMO system. Other models including FIR, SR, and transfer function can be easily converted

into the state space format. This model is also capable of forming a non-linear future prediction. Besides, a feed-forward concept can be naturally embedded in the formulation to improve its robustness against a measured disturbance. However, this model may provide numerical ill-conditioning when dealing with a large prediction horizon or dimension of state (L. Wang, 2009; Tatjewski, 2007). In addition, the identification process of state space in a real application can be challenging if the full state variables are not measurable. For this case, an estimator such as the Kalman filter is often used to estimate the unmeasurable states with a careful tuning process (Bishop et al., 2001).

Notwithstanding those facts, MPC also has undergone various types of development through this century. In early development, many works tried to come out with numerous heuristic tuning methods (Garcia et al., 1989). Later on, this issue becomes clear that if the controller is solved based on an ill-posed decision where there is prediction mismatch between the open-loop and actual behaviour, the optimisation process becomes ineffective. The leading solution at that time is to use large enough prediction horizon with the similar number of control horizon (Mosca & Zhang, 1992). However, this can lead to a substantial computation expense in addition to the numerically ill-conditioned problem (J. A. Rossiter, 2018). In order to avoid this problem, L. Wang (2009) proposed the use of Laguerre functions to parameterise the control horizon, so that a small number of parameter can be used to get an effect of a larger horizon. Besides, an exponentially weighted function is embedded inside the prediction horizon to improve the ill-conditioning problem. Consequently, this concept can provide a prescribed degree of stability, which can be extended to the constrained cases by modifying the weights in the cost function. A similar outcome can be obtained with the dual mode MPC (Sckaert & Rawlings, 1998), which utilises an infinite prediction based on LQR with the concept of a tail. The stability of this controller can be proved using a Lyapunov equation with the condition that the computed solutions must be feasible. On the opposite side, Valencia-Palomo & Rossiter (2011) developed an auto-tuned MPC that can be easily embedded into a standard PLC with systematic constraint handling. Although this concept is simpler than the common MPC, yet it can be considered as more realistic than the classical PID. These works are only a few examples of the more extensive developments of MPC in the literature.

Although MPC may provide optimal control and other associated benefits, yet it has several well known drawbacks. First, the derivation of MPC concept and the identification of its prediction model are considerably more complex than the classical PID controller. An accurate model is necessary because if there is a significant mismatch between the predicted behaviour and actual response, the optimisation process becomes meaningless (J. Rossiter & Kouvaritakis, 2001). Additionally, MPC often needs a higher computation load and time for the optimisation procedure. Although this requirement is not a severe problem considering the current computation power, yet most of the process industries are still relying on the use of a simple processor such as PLC where other tasks such as measuring and sending a signal are also carried out simultaneously. Besides, the processor may require more computation when the number of variables and prediction horizon increases (Valencia-Palomo & Rossiter, 2011). The similar requirement is required when satisfying constraints as it needs to retain optimality and recursive feasibility of the solutions. Due to these situations, in most cases, MPC is only used to handle more complex processes rather than simple ones.

3.2 CLASSICAL PID

PID is the most popular controller, which is well established and has become a standard approach in handling the closed-loop control problem within numerous small-scale industrial applications. The main reason behind this is because of its simplicity in concept, low implementation cost and well-explained theory in the academic literature. This controller is well known to be very useful in handling a simple and straightforward linear control problem and highly compatible to deploy in a basic PLC processor. Currently, many engineering software such as Matlab, LabView and others provide a built-in PID toolbox. This controller consists of three optional regulators, which are a proportional, integral and derivative. The objective is to find a suitable gain for individual or combination of more than one regulator to obtain the desired closed-loop response. Its traditional tuning rule as developed by Ziegler & Nichols (1942) to get the best behaviour while retaining stability is well recognised and accepted among control community and has become a standard syllabus in most of the control system undergraduate courses.

According to a report by Åström & Hägglund (2001), almost 90% of the control loop in the process industry are PID, particularly with PI since the derivative term is occasionally used. This situation is not surprising since most PID users are quite comfortable in using its concept even when the performance is poor compared to what could be achieved (J. A. Rossiter & Haber, 2015). Although the tuning process of PID controller seem to be intuitive, yet in a real process it may be not as straightforward as one imagines, especially when multiple objectives are conflicting with each other in the presence of strong non-linearities, disturbance or environmental uncertainties (Kumar et al., 2011). In order to avoid this issue, a careful off-line tuning is often employed through several trials until the controller manages to obtain the desired closed-loop response. A knowledge related to root locus, Bode plot, Nyquist plot to assess the closed-loop stability deems necessary to implement and tune this controller effectively.

By default, the concept of PID does not provide an optimal solution where the computation solely relies on the constant parameter of the gains in the feedback loop without having direct knowledge of the process. Consequently, it may encounter difficulties when handling a complex system with constraints or long time delay. To overcome this limitation, PID also has undergone various kind of developments. For handling physical limitations without saturation and oscillation in the response, an anti-wind-up with a proper tuning is frequently utilised (Åström & Hägglund, 2001). For controlling a process with long time delay, the PID can be implemented with an active dead-time compensator known as Smith predictor (Astrom et al., 1994). These methods can be considered as a traditional modification of PID to improve its reliability to handle those processes.

The advancement of modern computation also has triggered numerous development of PID in many areas. For instance, several works had proposed a hybridisation between PID concept with other advanced strategies such as fuzzy logic, neural network or MPC to name a few. For example, Xu et al. (2005) proposed to use GPC to optimally tune the PID parameters offline and obtained a similar performance as using GPC alone. Diordiev et al. (2003) developed a hybrid PID-Fuzzy for a simple DC/DC converter where the original PID concept was used as a based controller, and fuzzy was utilised to provide the desired gain. This controller managed to provide better response in handling a highly non-linear

process compared to the conventional PID. In order to increase the adaptability of this controller, Yongquan et al. (2003) utilised a neural network to train the PID-Fuzzy gain using back-propagation-algorithm. From all these listed works, it is clear that the concept of PID can be extensively explored with other methods to increase its reliability, especially when dealing with highly non-linear processes.

Nevertheless, there are also some drawbacks that should be perceived when using PID controller. The proposed constraint handling approach of PID via anti-wind-up is still posterior where a user will ignore all the limits during the designing process and only consider it at the implementation stage (Garcia et al., 1989). Consequently, it is quite difficult to generalise this ad hoc strategy to other applications. Besides, most of the output or state constraints are not considered in PID where it can only be avoided through a proper tuning (Valencia-Palomo & Rossiter, 2011). Evidently, the hybrid PID concept can provide a better performance compared the conventional one, yet the complexity in implementation, tuning and designing process also need to be weighed by a user.

3.3 PREDICTIVE FUNCTIONAL CONTROL

From the brief review of MPC and PID in the previous two sections, it is clear that each of these controllers has their speciality. MPC is suitable for a complex system while PID is useful for a simple process. Nevertheless, a user also should consider another potential alternative to PID which is the Predictive Functional Control (PFC). This controller can be considered as one of the simplest forms of MPC, which uses a similar concept except that it tracks the target trajectory at specific points instead of finding the optimal solution via quadratic cost functions. This simplification makes it attractive to the industrial practitioner to implement it in a small scale application that has limited resources. Since PFC can be naturally embedded in a standard PLC, the implementation can be extensive, easily accessible and cheap. Richalet (2007) pointed out that in early industrial work, technical staff become confident to use PFC as an alternative to PID when given a strong promotion and practical support in a systematic training programme. Richalet et al. (1995) also reported several early implementations of PFC in the metallurgical industry such as continuous casting, rolling mills and roll eccentricity. Besides, numerous successful applications

of PFC in other fields can be found in these references (Richalet, 2007; Hashizume, 2015; Richalet & O'Donovan, 2009; Pedersen et al., 2017; Škrjanc, 2007).

PFC also inherits some of the key features of MPC such as the handling of constraints and delay control problems. Indeed, these properties elevates the key selling points of PFC as it is quite difficult to obtain these futures with the default PID regulator. Besides, the tuning process of PFC is very simple and straightforward, where the use of CLTR as the tuning parameter (as discussed in Section 2.1.1) provides valuable information to a user (Hashizume, 2015). In addition, PFC also has a notable performance in tracking ramp and parabolic set points, which is beneficial especially when implemented in the aerospace applications (Richalet & O'Donovan, 2009). The simplicity in its coding and implementation also make it possible to extend its usage to non-linear control (Yang et al., 2006). Similar to MPC, the PFC concept also is quite flexible and can take any type of internal model, target trajectory, prediction structure. This flexibility makes it easy to combine the concept with other techniques to further improve its utility.

3.3.1 Recent Developments

The recent developments of PFC are mostly focused on improving its performance for a specific application. Similar to the review of MPC and PID, the traditional PFC concept was often combined with other advanced approaches to obtain the advantages from both methods. For instances, in heat exchanger control problem, Škrjanc & Matko (2000) implemented PFC with Fuzzy logic to manage a highly non-linear behaviour of the process. An almost similar concept is adopted by Yang et al. (2005) in the fossil power plant application, where PFC was employed with customised Elman neural network to improve the control behaviour of steam temperature. In order to handle a highly uncertain batch reactor, a cascade PFC structure with an adaptive internal model was utilised by Yiming & Bin (2012) to increase the controller robustness.

There are also several works that were looking into the combination between PID with PFC concept. In a marine engine application, R. Wang et al. (2018) used a PID controller to track the engine speed where the gains are tuned by PFC based on a first-order plus dead

time (FOPDT) reference model. This controller managed to provide a better transient behaviour, steady-state performance, adaptation and robustness compared to classical PID regulator. A similar concept was adopted by J. Zhang (2017) in the control of chamber pressure in a coke furnace where a similar conclusion was obtained.

Muñoz et al. (2018) presented the first application of PFC in autonomous underwater vehicles where they compared its performance with other controllers based on fuzzy and gain scheduling. It was shown that PFC provides better performance in both high and intermediate level implementations with the smoothest behaviour and minimum oscillation in the presence of sensor noise. Besides, PFC provided a safer control law where the limits of actuators were considered and easily programmed into Guanay's embedded systems. Pedersen et al. (2017) compared the performance of three control methods namely a traditional gain scheduled PI-based controller, PFC and PFC with a neural network on a laboratory set-up for handling the superheat in a refrigeration system. It was found out that the performance of these controllers are almost similar and can only be compared in term of their tuning parameters and computational load, where PFC provided the best package.

Guo et al. (2017) proposed an augmented PFC in an autopilot missile application. A novel performance index that depends on the reference trajectory, output prediction, and the set-point was designed to improve the closed-loop dynamics. The simulation showed that the proposed controller managed to produce better response while satisfying the implied input constraints in the presence of an abrupt disturbance. This work is an excellent example to highlight the ability of PFC in tracking different types of set-points such as ramps and parabola. Other than that, R. Zhang & Jin (2018) extended the concept of PFC to a non-linear control for handling oxygen content regulation in a coke furnace application by utilising a hybrid model based on a linear iterative design for catering the non-linearity. The implementation of this concept was straightforward and can provide better response compared to the conventional one.

3.3.2 Limitations

From the discussion in section 3.3.1, it is apparent that PFC has undergone various kind of developments and can provide many attractive benefits with such a simple approach that are worth considering as an alternative controller to the classical PID. Nevertheless, it is also observed that the current research works in PFC are mostly focussing towards the practical improvement in a specific application by adopting a more advanced approach rather than improving the underlying control law itself. Hence, this thesis is solely interested in proposing a new development within this area so that a user may get some insight into how to effectively used the PFC controller before extending its concept to a more sophisticated design.

The main problem with the PFC concept can be related to its simplicity. The mathematical foundations of PFC are not systematic and rigorous compared to other MPC approaches (Valencia-Palomo & Rossiter, 2011). For instance, the performance or stability analysis is primarily a posteriori and very difficult to prove except for the specific case with a first order process as reported by J. A. Rossiter (2017). Hence, this scenario provides an opportunity for the academic community to propose a more rigorous yet intuitive concept of PFC that can be easily embedded in a simple PLC processor while retaining its simplicity of formulation. Based on a comprehensive PFC analysis presented by J. A. Rossiter & Haber (2015), three crucial issues should be considered to improve its underlying control law namely the tuning efficacy, challenging dynamic applications and robustness to uncertain cases. The next three sections will review these issues together with the default and recently developed methods to overcome it.

3.4 TUNING EFFICACY AND CONSTRAINT HANDLING OF PFC

In general, PFC can be tuned to get better accuracy, robustness and transient dynamics (Richalet & O'Donovan, 2009). Accuracy of the controller defines its ability to provide the desired set-point without offset error in the presence of uncertainties. This tuning criterion can be more difficult to achieve if the desired set-point is not constant such as tracking ramps or parabola. Moreover, it also has a link with the second tuning criteria which is

robustness that describes the deterioration of performance in the presence of uncertainties. Conversely, the transient dynamics represents how fast or slow the system will converge to the set-point or rejecting the disturbance. Since the scope of this research is only focused on tracking a constant set-point, only the dynamics and robustness become the main interest. These requirements have a direct relationship with the choice of coincidence horizon n_y and desired closed-loop pole λ .

As discussed before, the core objective of PFC is to drive the closed-loop performance to behave like a pre-specified first order system, which is the target trajectory (2.2) (Richalet & O'Donovan, 2009). Theoretically, this concept is valid if the selected coincidence horizon n_y closely represents the overall system response (J. A. Rossiter & Haber, 2015). However, it is often impossible to achieve this condition with a non-first order process considering its closed-loop response can not exactly approximate the first-order target trajectory. In order to overcome this issue, a systematic tuning procedure is required.

3.4.1 Tuning Procedure

An ideal tuning is to select the desired closed-loop pole λ similar to the open-loop characteristic, yet this is not entirely practical since most of the applications desire to modify its convergent speed. For a first-order system, it has been proven mathematically that the link between the desired pole and the actual closed-loop is weak and only effective if the coincidence horizon $n_y = 1$ (Richalet & O'Donovan, 2009; J. A. Rossiter & Haber, 2015; J. A. Rossiter, 2017). Choosing $n_y > 1$ will deviate the closed loop dynamics further from the target trajectory. However, this selection is not applicable with a higher order model, where larger coincidence horizons are usually needed to retain the robustness for compensating a lag in its step response and avoiding any over aggressive input activity or instability. Earlier works by Khadir & Ringwood (2008) analysed the use of a reduced first order model to represent a higher order system. Although, $n_y = 1$ can be used with this approach, yet the response is quite poor compared to the use of a full-scale model with a well tuned n_y . Thus, a proper section of n_y is necessary to retain the efficacy of λ as the tuning parameter as it also can affect the robustness and computed input activity (Richalet & O'Donovan, 2011).

For a higher order process, several tuning rules have been proposed in the literature. According to the PFC tuning manual given by Richalet & O'Donovan (2009), the coincidence horizon should be selected based on the inflexion point of the step response where the gradient is maximum. This point varies with the section of λ and may work only for a specific dynamical system. For a non-minimum phase process, the range of inflexion point itself may not be enough to represent the overall dynamics. For this case, Khadir & Ringwood (2008) suggested that n_y should be selected beyond the inflexion point to avoid any instabilities. However, a user should also note that if n_y is too large, it can reduce the effectiveness of λ as a tuning parameter and becoming closer to the open-loop dynamics of a system. J. A. Rossiter & Haber (2015) recommended a safe range for selecting n_y , which is in between 40% to 80% rise of the step input response to the steady state. Although there are no formal or generic proofs for this conjecture, yet it still helps a user to get a general idea on selecting a suitable n_y to get better closed-loop performance while retaining the stability.

There is also an alternative structure of PFC that can provide better tuning guide-line as proposed by J. Rossiter et al. (2016). In their work, a different concept of PFC from the conventional one known as Pole-Placement PFC (PP-PFC) was developed where a higher order process including non-minimum-phase dynamics can be effectively tuned with $n_y = 1$. The central concept is to decompose a higher order model into a group of individual first order models that operate in parallel. Then, a separate PFC control law is deployed for each of these submodels. By utilising the concept of linearity, the algorithm combines partial contributions of each individual inputs sum to unity for achieving the desired dynamics. This modification eliminates the previous trial and error tuning procedure and provides more intuitive design while retaining the simplicity of coding and implementation. Nevertheless, the proposed controller may require a complex number when dealing with an under-damped process. This requirement will limit its application to a simple dynamical system as most of the available processor in the industry can only cope with real numbers.

With regards to all the listed tuning procedures or alternative approaches, there still exists an inconsistency issue between the open-loop predictions used for decision making and the actual closed-loop behaviour that results, especially when using smaller n_y

(J. A. Rossiter & Haber, 2015). Conversely, a large n_y may provide better prediction consistency, but the effectiveness of λ as the tuning parameter becomes less significant. The main reason behind this is because of the constant future input assumption of PFC where there is insufficient d.o.f to ensure prediction consistency when anything other than open-loop behaviour is wanted (J. A. Rossiter, 2016). Some may argue that PFC is never design to compute an input based on long term predictions as eventually it will still converge to the steady state due to the receding horizon principle. Nevertheless, this inconsistency issue may affect the constraint handling performance of PFC especially when dealing with output and state constraints where long-term predictions are often required for checking and enforcing the limits.

3.4.2 Constraint handling

As discussed in Section 2.1.5, the traditional PFC handles input and rate constraints using a simple saturation approach, which is feasible enough to implement only on the first sample due to the constant future input assumption. As for the output and state constraints, a user employs multiple PFCs with the different tuning of λ and n_y that work in parallel either to track the set point or satisfy the constraints using a logical supervisor (Richalet & O'Donovan, 2009; Abu et al., 1991). Although, there are many pieces of evidence that this approach can work in real applications, yet it has a potential weakness when all the constraints are activated. Since there is no interaction among the regulators, a conflict between constraints may occur. In order to avoid this issue, a user needs to tune the second regulator using a slower pole as pointed out in Remark 2.4. This preference, however, may lead to very conservative constrained solutions.

With the advancement in computation, it is now possible to formulate a more systematic constraint handling strategy using a single set of linear inequalities that can surpass the limitation of the conventional approach (J. A. Rossiter & Richalet, 2002). This concept is a common practice in a standard MPC framework and can be formed similarly as ONEDOF (J. A. Rossiter et al., 2001) and reference governor approaches (Gilbert & Tan, 1991). It also worth to highlight that the implementation of this concept with MPC may require more computation even with relatively small horizons since all the constraints need to be

satisfied simultaneously at each iteration (Tatjewski, 2007; L. Wang, 2009). Conversely, the implementation with PFC is expected to consume less computation due to its simpler control law, albeit providing suboptimal constrained solutions.

Nevertheless, this systematic constraint strategy requires consistent open-loop prediction to avoid long-term violation effectively. With the current ill-posed decision, the implementation of this strategy will give less accurate and more conservative constrained solution especially when the validation horizon of the constraint is bigger than the coincidence horizon n_y (J. A. Rossiter, 2018; L. Wang, 2009; Mayne et al., 2000; J. A. Rossiter & Richalet, 2002). In reality, the underlying PFC constant input assumption which restricts the prediction to a single d.o.f is indeed counter-productive (J. A. Rossiter, 2016). Instead, it is more desirable to utilise a dynamic that evolves over many more samples (J. A. Rossiter et al., 2010) to further improve the tuning and constraint performance of PFC while logically retaining the simplicity of formulation.

3.5 HANDLING CHALLENGING DYNAMICAL PROCESSES

For systems with close to monotonic step responses such as first order or over-damped second order dynamics, a typical PFC can perform well with a proper selection of tuning parameters λ and n_y (J. A. Rossiter & Haber, 2015). Conversely, the default PFC framework may not be adequate for handling less desirable dynamics such as oscillatory, unstable, under-damped, and significant non-minimum phase systems or any combination of these dynamics. Again, the main reason is that the constant future input dynamics of PFC does not have enough d.o.f where any non-zero input will provide divergent or oscillatory predictions. (J. A. Rossiter & Haber, 2015; J. A. Rossiter, 2016, 2002; Rawlings & Muske, 1993). In order to have a good prediction, these undesirable open-loop poles need to be altered before implementing the standard control law. If not, it would lead to instability or infeasibility when dealing with constraints (Kouvaritakis et al., 1996). There are several approaches available in the literature to fix this issue such as decomposition model, cascade structure and pole cancellation. This section will review the advantages and disadvantages of these methods.

3.5.1 Decomposition Model

Decomposing a model is the process where the original model representing a system is modified to provide a stable open-loop response. Specifically, it is decomposed into two stable models by compromising the approach between RM and IM structure (Richalet & O'Donovan, 2009). The first model takes the control input, while the second model is fed by the process output which can take a form as compensated input. The outputs summation from both models will lead to stable prediction. This structure gives a simple solution to handle integrative or unstable processes where a simple PID controller is not effective. For instance, in an aerospace application, this model manages to improve the control performance of rocket pitch-angle, which has a fast and under-damped dynamics, while reducing its vibrations in the transient period (Škrjanc, 2007).

Although these methods often work for handling integrating, oscillating and unstable process, yet a more nuanced implementation is required. Specifically, the tuning process is less simple and intuitive, which undermines the critical selling point of PFC where there is a specific rule that needs to be satisfied for ensuring stability (Richalet & O'Donovan, 2009). For some cases such as when dealing with a process that has both unstable and non-minimum phase dynamics, the decomposed model has a limit to stabilise this system where the time constant of the numerator must be less than the denominator. Besides, the identification process of this model can be quite challenging for a less trained user.

3.5.2 Cascade Structure

As discussed in Section 2.2, PFC practitioners also deploy a form of cascade PFC design or known as a Transparent control to handle integrating process (Richalet & O'Donovan, 2009). It has become a normal practice to form a two-stage design where the inner design is utilised to stabilise the open-loop dynamics by manually selecting a suitable stabilising gain using a Proportional only controller, while the external loop deployed the conventional PFC to control and achieve the dynamic performance. The same method also can be extended to handle an oscillatory process and some of unstable dynamics systems. Indeed, the underlying concept is well received and has a similarity with the concept of dual-mode

MPC (J. A. Rossiter, 2018; J. A. Rossiter et al., 1998). It is shown in the coking furnace application that the cascade structure also can improve the process sensitivity to a set-point change and disturbance, by carefully tuned both of the regulators to alleviate the offset term (R. Zhang et al., 2009).

Nevertheless, the main drawback of this structure is that it can only accept a proportional controller to keep the constraint implementation purely algebraic which is not practical with a full PID regulator. Consequently, a user may face difficulty when handling a more complex unstable process that cannot be stabilised by merely adding the proportional gain in the control loop. It is also noted that the cascade structure needs a back calculation procedure when implementing hard constraints to transfer the information from the inner loop to the outer loop. Such an approach can be considered as a standard practice and has a similarity with the concept of anti-windup as in the classical PID controller. However, if the input limits are only checked at the first sample, a recursive infeasibility may arise as the condition is now changed compared to the nominal PFC that based on the constant input assumption. Hence, the implementation of PFC becomes more complicated as one needs to carefully tune the stabilising gain to pre-stabilise the system while ensuring recursive feasibility when satisfying constraints. Besides, there is no systematic procedure and clear explanations in the literature on how to effectively pre-stabilise the inner loop and to handle constraints effectively within a nested loop where most of the implementations are ad hoc with little if any rigour.

3.5.3 Pole Cancellation

Pole Cancellation PFC can be considered as a more systematic approach compared to the two previous methods, which has clear design choices and simple coding to manage the open-loop challenging dynamics processes. The core concept is to shape or constrain the future input predictions of a model so that the output predictions will converge to a steady state value (J. A. Rossiter et al., 1997). By utilising a linear combination principle, the sequence of predicted input is selected to cancel out the undesirable poles. This parametrisation approach is originally derived from the conventional MPC framework for unstable dynamics (J. A. Rossiter, 2018). It can act as an excellent alternative to the previous

decomposition model and eliminates the tuning difficulties associated with that structure (J. A. Rossiter, 2002). In addition, this input parametrisation concept can also increase the efficacy of the closed-loop time constant as PFC tuning parameter when implemented with a CARIMA model (J. A. Rossiter, 2016). This method also enables easier constraints implementation with guaranteed recursive feasibility solutions which is generally not applicable to other conventional approaches.

Theoretically, this concept offers a more general control law to handle a different kind of challenging dynamics system compared to the previous two conventional approaches. However, there are also several main drawbacks that should be noted by a user. For example, the control input change from the shaping procedure can be very precise and quite aggressive which may not be practical to implement in a real applications (J. A. Rossiter, 2016). Also, the use of the realigned CARIMA model can lead to a more sensitive response to measurement noise especially when dealing with a higher order system where more measurements are needed to compute the control law. Hence, a user may require a suitable low pass filter to recover the controller robustness against the measurement noise.

3.6 ROBUSTNESS OF PFC

Robustness is a measure of how well a system can maintain its performance and stability in the presence of uncertainties that can be in the form of measurement noise, disturbance or plant model mismatch. If uncertainty is not handled carefully, it may cause unstable and/or infeasible control solution (Bemporad & Morari, 1999; Mayne et al., 2000). As discussed before, it is quite difficult for PFC to adopt a robust formal design as in MPC (Mayne et al., 2005) due to its simplicity requirement (Khadir & Ringwood, 2008). Besides, the main purpose of PFC is to provide a sub-optimal control solution with a moderate enough performance and robustness specifically for a simple system. Hence, the logical alternative is to derive its control law using a method that can provide a robust design (Zabet & Haber, 2017; J. A. Rossiter, 2018). Besides, there is a direct link between the robustness of PFC control law with its prediction model and structure.

3.6.1 Types of Model

In any predictive control, the accuracy of prediction is fundamental to the efficacy of the control design. Also, different prediction models will give different sensitivity to uncertainties. This relationship can affect the consequent behaviour of the controllers whether it can retain the steady-state behaviour and stability of the controlled performance. Similar to MPC, PFC also can take several types of models to make a future prediction such as:

1. **Finite Impulse Response (FIR):** This structure is constructed based on the impulse response of a plant. The traditional MPC known as Dynamic matrix control uses this model and has been widely received in the industry. The design and implementation are simple and easy to understand (Qin & Badgwell, 1997) due to its clear description of process time delay, response time, and gain of the system (L. Wang, 2009). The other advantage is that it has lower variance prediction errors without the need for robust design (J. A. Rossiter, 2018). This structure also is easily extended for adaptation (Han et al., 2003). The main drawback is that it requires large numbers of coefficients, which require extra space and memory in the program (L. Wang, 2009).
2. **Transfer function:** It is constructed based on the input and output relationship of the process. There are many types of transfer functions, which depends on the added term. Common PFC practice is to use a first-order transfer function (Richalet & O'Donovan, 2009). For higher order model, it can be represented by several sub models constructed in parallel using a partial fraction. In MPC, the well-known example of prediction using a transfer function is in GPC. This structure has good compatibility with the popular black-box identification technique (Ljung, 1998). However, one may face a problem when extending it to multi-variable case (L. Wang, 2009; J. A. Rossiter, 2018). Since most of PFC application is focus on SISO process, this model can be considered as a suitable option.
3. **State space:** This structure is very systematic and can be used to represent multi-variable systems. It is constructed based on a set of input, output and state variables. State space model is more compact and has fewer parameters compared to the other

three models. This structure makes the designing and analysing processes of the controller become more transparent (L. Wang, 2009; Bemporad & Morari, 1999) since the previous two models also can be formed into the state-space structure. One of the limitations using state space model is that, if the state of the plant is not measurable, an observer or estimator is needed to complete the prediction, where the tuning of an estimator is not often straightforward and mostly based on trial and error.

Each of the presented model structures has their benefits and limitations when implemented with PFC which is analogous to the previous discussion in MPC. For general predictive control application, a typical academic prefers to use transfer function and state space models, while most of the industrial practitioners are favoured toward the FIR. A comprehensive sensitivity analysis for these types of models has been made and presented by (J. A. Rossiter, 2018). It can be concluded that the use of state-space or transfer function without an estimator (realigned models) in predictive control can lead to poor sensitivity concerning measurement noise. This weakness is the main reason why the industrial users have favoured a FIR model. However, the FIR structure often needs a high number of coefficients to represent the plant (L. Wang, 2009). In general, PFC can adopt any model, but in order to keep the presentation transparent, the author used mainly a transfer function as it is more convenient to link the system poles to the target pole for further modifications or analysis. Besides, it is suitable enough to represent SISO applications without the need for an observer.

3.6.2 Prediction Structure

Prediction structure has a direct impact on the loop sensitivity. There are two typical types of prediction structure which are the Realigned Model (RM) and Independent Model (IM) as discussed before in Remark 2.1 in Section 2.1.2. The RM structure is more sensitive to measurement noise compared to IM structure, yet better in rejecting the disturbance. Hence, a user needs to select a suitable prediction structure to get the best performance. There are several possible ways to handle and improve robustness in PFC such as:

1. **Independent model:** The common practice in the industry is to employ the independent model approach that is claimed to have a low sensitivity to noise and disturbance. The concept is very straightforward as discussed in Section 2.1.2. This method can take any models without the need of observer or filter. The concept is equivalent to a more efficient implementation of popular step response models. However, the usage is only limited to a stable prediction where for the process with unstable poles need to be pre-stabilised first before implementing the structure.
2. **T-filter (J. A. Rossiter, 2018):** This approach is often used with an RM structure of the transfer function that is known as CARIMA model as in GPC to improve the noise sensitivity. As far as the author is concerned, no work has tried to implement this filter in PFC framework. T-filter acts as a low-pass filter by reducing the transference of high-frequency noise. The benefits that a user can gain is that it can improve the sensitivity to the parameter or signal uncertainty without any impact on the nominal tracking. This method can also handle the unstable process. Although, this approach does not have a systematic design where it is difficult to tune for a specified effect, rather than trial and error, yet there is a possibility to tune it systematically using Youla parametrisation (J. A. Rossiter, 2018).
3. **Kalman filter:** Constructed based on an optimal framework to improve prediction accuracy (Grewal, 2011). It can be used to estimate both state and disturbance in a system. However, this filter assumes rather than estimates the knowledge of signal characteristics. It also does not cater for all the sensitivity functions rather than gives optimal state estimation for a given signal uncertainty (J. A. Rossiter, 2018).

Other than the listed methods, there are also several ad hoc approaches that can be used to improve PFC robustness. For example, Satoh et al. (2012) designed a cascade-like structure where the inner loop consists of a disturbance observer and outer the loop employs the nominal PFC. This structure managed to provide better robustness against higher order disturbance and such as those with ramp and parabolic dynamics. Zabet & Haber (2017) proposed to use a feed-forward structure with a Smith predictor. With this structure, PFC can be tuned systematically for robustness. Azira et al. (2018) showed that the use of state

space with the reduced observer (where only a partial of the states are measured) provide less steady state error compared to the full estimated observer in a pneumatic positioning system application. However, there is no systematic analysis was done where the conclusion was purely made based on observation. Li et al. (2017) designed an improved PFC prediction structure by embedding disturbance information into the system model instead of using a feed-forward or observer approach. The simulation and experimental results of permanent magnet synchronous motor showed that the proposed method is more robust than PFC with the conventional observer. Nevertheless, a user should note that the identification of disturbance is not a simple task, and it can be different for other applications.

Although there are many possible ways to improve the robustness of PFC either via implementing different types of a prediction model, structure or estimator, there is a lack of formal comparison between these approaches. Besides, it is good for a user to have a solid understanding of how these structures can affect the loop sensitivity before using it. Since the sensitivity is a process dependent, an off-line analysis is desirable to identify the trade-off between sensitivity and also to measure the robust performance of the controller. However, most of the existing sensitivity analysis in the literature (J. A. Rossiter, 2018) are mainly focused on the MPC algorithm . Whereas for PFC, different results may occur since the control law is more straightforward.

3.7 SUMMARY OF LITERATURE REVIEW

This chapter has provided a brief review of the MPC, PID and PFC control framework. From the survey, it is found out that although MPC is a good controller and can provide many attractive benefits, yet its application is more suitable for complex applications. As for a simple application, PFC seems more suitable to be implemented when compared to both MPC and traditional PID controllers. The review also pointed out that PFC has been widely used in many industrial applications and undergone several developments to improve its control performance further. Nevertheless, there are still some issues that are worth highlighting concerning with its default control law such as the efficacy of tuning, prediction inconsistency, handling constraints, challenging dynamics applications and also robustness to uncertainty. Hence, the primary objective for this research to fill in these gaps

by developing several approaches to tackle the issues.

Firstly, it is noted that there is a potential development that can be done with the PP-PFC. The ability to tune the coincidence horizon with $n_y = 1$ for a higher order process may provide various benefits such as simpler tuning rule and implementation. However, the current processor may require to implement a complex number when controlling a system with complex poles. Since most of the simple processors in the industry are only capable of processing a real number, this work will propose an alternative way in Chapter 4 to avoid the use of complex numbers in the calculation process while retaining its computation simplicity.

Secondly, it is also pointed out that the prediction inconsistency issue may affect the constraint handling performance of PFC. The main reason behind this is because of the artificial constraint of PFC in using the constant future input assumption that has insufficient d.o.f. In Chapter 5, a novel PFC algorithm based on Laguerre polynomial will be proposed to enlarge the d.o.f with a single parametrisation for improving the prediction consistency. Besides, a more systematic constraint handling method as commonly used in MPC will be embedded inside this algorithm for replacing the traditional multi-regulator approach to provide more accurate and less conservative constrained solutions.

For handling a challenging dynamics process that contains either integrating, oscillating or unstable poles, the concept of PC-PFC will be developed to include the effect of the tail in Chapter 6. Consequently, instead of cancelling those undesired poles, the control law will shape it according to a user specification. This control law is expected to give a less aggressive input activity while retaining the valuable property of PC-PFC that is the guaranteed of recursive feasibility constrained solutions.

As for the robustness issue, there is a clear need for off-line sensitivity analysis for PFC to provide some insight on what sort of sensitivity trade-off that one should expect when using different types of prediction structures. Besides, the same analysis can be carried out to assess the robustness of any proposed control law of PFC. This development will be covered in Chapter 7 along with another proposal of using the T-filter with RM structure in the PFC framework.

Finally, it is also should be highlighted that the scope of this research is limited to a constant set point target. In addition, the delay control problem will be ignored for clarity the presentation since implementing PFC for this process is straightforward (J. A. Rossiter & Haber, 2015; Richalet & O'Donovan, 2009). Also, the off-line sensitivity analysis will only cover a transfer function model since only this model is used throughout the work. Other issues such as tracking ramp or parabolic set points or the different selection of prediction model will be denoted as future work.

Chapter 4

POLE PLACEMENT PFC

This chapter provides a summary of the ISA Transaction paper which is attached in Appendix A (Zabet et al., 2017) that develops a novel method for Pole Placement PFC (PP-PFC) specifically for handling a system with complex poles by using real numbers algebra. Section 4.1 starts a brief review on a general concept of PP-PFC that was originally developed by J. Rossiter et al. (2016) to handle over-damped systems together with a short discussion regarding its advantages. Section 4.2 introduces the extension work of this paper where the default concept of PP-PFC is formulated for handling systems with complex poles. Section 4.3 presents the original contributions of this work where new algorithms of PP-PFC are developed to avoid the use of complex numbers. Section 4.4 demonstrates some of the simulation results from the paper and Section compares the performance of PP-PFC with the traditional pole-placement technique (Aström & Murray, 2010).

4.1 THE CONCEPT OF PP-PFC

It is noted from the previous discussion that a user should be more cautious in tuning the conventional PFC for a higher order system where the main parameter λ , is often ineffective and not a good representation of the closed-loop dynamic that results when using $n_y > 1$ (J. A. Rossiter & Haber, 2015). One way to avoid this issue is via implementing the PP-PFC where the main motivation is to exploit the efficacy of PFC for first-order systems in order to propose an equally simple tuning process that will work on higher-order systems. Also, this control law can provide a better link between the closed-loop performance and the target trajectory compared to the nominal PFC. Since the prediction model structure of PP-PFC is a bit different from the one that is given in Section 2.1.2, a basic formulation for

a first order system using nominal PFC will be presented first before deriving the PP-PFC concept.

4.1.1 Tuning PFC for First Order System

PFC has been particularly effective in industry partially because many real systems have dynamics which are close to first-order and it is easy to show that for a first-order system, the PFC tuning parameters work perfectly, as long as one uses a coincidence horizon of one (J. A. Rossiter & Haber, 2015). In other words, the target pole λ becomes the closed-loop pole exactly in the nominal case $y_m = y_p = y$.

- For a first-order model with $n_y = 1$, the control law is given as follows:

$$\left. \begin{aligned} y_{k+1} &= bu_k - ay_k, \\ y_{k+1} &= (1 - \lambda)r_k + \lambda y_k, \end{aligned} \right\} \Rightarrow u_k = \frac{(1 - \lambda)r_k + (a + \lambda)y_k}{b} \quad (4.1)$$

- Rearranging and substituting the corresponding control action back into the system dynamics gives:

$$y_{k+1} = b \frac{(1 - \lambda)r_k + (a + \lambda)y_k}{b} - ay_k = (1 - \lambda)r_k + \lambda y_k \quad (4.2)$$

From which it is clear that the closed-loop behavior is represented by a first-order model with unity gain (no steady-state offset) and the desired pole λ .

4.1.2 Pole Placement PFC Control Law

In order to maintain simple coding, PFC overcomes the complexity of prediction algebra by using partial fractions to express the n^{th} -order model $G_m(z)$ as a sum of first-order models (Richalet & O'Donovan, 2009; Khadir & Ringwood, 2008) and hence:

$$\left. \begin{aligned} y_{m,k} &= G_m u_k, \\ G_m &= \sum_{i=1}^n G_i, \end{aligned} \right\} \Rightarrow y_{m,k} = \sum_{i=1}^n G_i u_k = \sum_{i=1}^n \frac{b_i z^{-1}}{1 + a_i z^{-1}} u_k \quad (4.3)$$

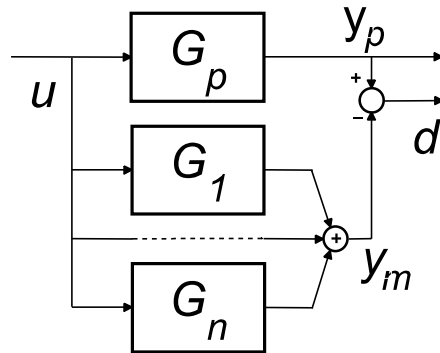


Figure 4.1: Parallel model format alongside the actual process G_p .

The effective structure of the model is illustrated in Figure 4.1 where G_p represents the real (unknown) process and G_i denote the partial fraction expansion of the assumed model G_m . In practice this means that the independent model deployed in PFC code comprises a number of first-order independent models running in parallel; clearly the coding and computation requirement for each is trivial.

The advantage of this parallel formation is that n_y steps ahead predictions can be defined explicitly and without the need for costly or cumbersome prediction algebra. To be precise, the predictions for the model can be expressed as the sum of the predictions of a number of first-order models with component outputs $y_m^{(i)}$, that is:

$$y_{m,k+n_y} = \sum_{i=1}^n \left[b_i \frac{1 - (-a_i)^{n_y}}{1 + a_i} u_k + (-a_i)^{n_y} y_{m,k}^{(i)} \right] \quad (4.4)$$

The key concept of PP-PFC is to treat each sub-model G_i as if it had an independent input and then deploy a nominal PFC algorithm to compute what that input should be in order to achieve some specified dynamic, say pole ρ_1 . The next core concept is to exploit linearity and linear combinations. The algorithm takes a linear combination of all the proposed inputs to determine the desired input to the real system. By utilizing a sensible constraint (that the partial contributions of each individual inputs sum to unity), it is easy to show that the desired dynamic is then achieved in the nominal case $d_k = y_{p,k} - y_{m,k} = 0$.

Algorithm 4.1. (PP-PFC): The PP-PFC algorithm for achieving a target closed-loop pole comprises the following steps.

1. Define targets for each individual sub-model G_i based on the model steady-state gains of output $y_m^{(i)}$ (4.3) using the formulae:

$$r_k^{(i)} = \frac{\gamma_i}{\sum_{j=1}^n \gamma_j} r_k; \quad \gamma_i = \frac{b_i}{1 + a_i} \quad (4.5)$$

2. Identify proposed inputs for each sub-model ($i = 1, \dots, n$) using the control law (4.1):

$$u_k^{(i)} = \frac{(1 - \rho_1)[r_k^{(i)} - d_k^{(i)}] + (a_i + \rho_1)y_{m,k}^{(i)}}{b_i}; \quad d_k^{(i)} = \frac{\gamma_i d_k}{\sum_{j=1}^n \gamma_j} \quad (4.6)$$

3. Form a linear combination of these inputs to determine the process input as:

$$u_k = \sum_{i=1}^n \beta_i u_k^{(i)}; \quad \sum_{i=1}^n \beta_i = 1 \quad (4.7)$$

It is important that a sensible choice is made for the values of β_i as, while any choice satisfying (4.7) will give the desired closed-loop pole, the choice made also has an impact on the other closed-loop poles. Indeed, the remaining flexibility in the values of β_i can be used to assign the other closed-loop poles at values $\rho_i, i = 2, \dots, n$ using a partial fraction by the following definitions (J. Rossiter et al., 2016):

$$\beta_j = \frac{\prod_{i=2}^n (a_j + \rho_i)}{\prod_{i=1, i \neq j}^n (a_j - a_i)}, \quad \forall j = 1, 2, \dots, n \quad (4.8)$$

Remark 4.1. It has been proven in the paper that the control law of (4.6), (4.7) ensures that the target pole ρ_1 becomes a closed-loop pole in the nominal case (where $d_k = 0$). Besides, using the choice of β_i as in (4.8) results in all the poles $\rho_i, i = 2, \dots, n$ becoming closed-loop poles. The stability of PP-PFC also is guaranteed in the nominal case as the positions of the poles are all known and have to be selected to be inside the unit circle.

Hence, the overall implied control law can be formed as:

$$u_k = \sum_{i=1}^n \beta_i u_k^{(i)} = (1 - \rho_1) \sum_{i=1}^n \frac{\beta_i r_k^{(i)}}{b_i} + \sum_{i=1}^n \beta_i \frac{(a_i + \rho_1) y_{k,m}^{(i)}}{b_i} \quad (4.9)$$

4.2 EXTENSION OF PP-PFC TO SYSTEMS WITH COMPLEX POLES

For simplicity the presentation will assume just a single pair of complex poles; this is reasonable as PFC would rarely be used on very high-order models given that low-order models usually capture the core dynamics. Moreover, notwithstanding this, the results will automatically carry over anyway. Consider a model $G_m(z)$ which has roots at $-a_1, -a_2, \dots, -a_n$ with a_1, a_2 a complex conjugate pair. A partial fraction expansion of $G_m(z)$ into first-order terms is:

$$G_m(z) = \frac{n(z)}{(1 + a_1 z^{-1})(1 + a_2 z^{-1}) \dots (1 + a_n z^{-1})} = \sum_{i=1}^n \frac{b_i z^{-1}}{1 + a_i z^{-1}} \quad (4.10)$$

It is noted that the residues b_1, b_2 will be complex conjugates.

A quick review of the previous section will reveal that none of the algebra required numbers to be purely real and the algebra and pole computations should equally apply for complex numbers. The obvious consequence is that a system with complex coefficients should still be amenable to the PP-PFC control law of (4.7). In fact, the only requirement that needs careful checking is that the input $u(k)$ to be implemented to the real process must be real. Applying a control law which utilises $\beta_1 u^{(1)}(k) + \beta_2 u^{(2)}(k)$ as defined in (4.7) will result in a real input as long as $\beta_2 = \beta_1^*$ (means complex conjugate). The overall implied control law associated to a pair of complex poles is given as:

$$\begin{aligned} u(k) &= \beta_1 u^{(1)}(k) + \beta_1^* u^{(1)*}(k) \\ &= (1 - \rho_1) \left[\frac{\beta_1 r^{(1)}(k)}{b_1} + \frac{\beta_1^* r^{(1)*}(k)}{b_1^*} \right] + \beta_1 \frac{(a_1 + \rho_1) y_m^{(1)}(k)}{b_1} + \beta_1^* \frac{(a_1^* + \rho_1) y_m^{(1)*}(k)}{b_1^*} \end{aligned} \quad (4.11)$$

It should also be remarked that the condition of $\sum \beta_i = 1$ implies that $\beta_2 = \beta_1^*$.

Remark 4.2. *Several proofs had been provided in the paper that the desired closed-loop poles of ρ_i are achieved for any choices of desired poles and any open-loop poles, irrespective of whether they are complex or real. In all cases, the proposed control law of (4.7) produces a real input. However, it is emphasised that the underlying signals implied in the independent model of Figure 4.1 will be complex numbers and as this model is retained in the control law implementation, it assumes that complex number algebra is supported by the operating system.*

4.3 DEVELOPMENT OF PP-PFC USING REAL NUMBERS ALGEBRA

Notably, the previous concept of PP-PFC needs to exploit complex number algebra and linear combinations in order to deliver guarantees of stability and performance for handling underdamped systems, and this has some possible negative consequences. Firstly, the computational effort is slightly greater, although that could be considered trivial in practice. Secondly however, the requirement for complex number algebra in itself could be a problem as many low level process control units (where PFC would be applied alongside competitor approaches such as PID) do not support complex number algebra. In view of these observations, practical implementation is easier by avoiding complex number algebra and hence a modified formulation of the PP-PFC algorithm is developed which utilises just real numbers while retaining the key attributes of simple algebra, coding and tuning.

Two alternative implementations are developed in this section: i) handling the real and imaginary components explicitly and ii) a formulation of the algorithm avoiding complex numbers altogether. Readers should note that the case of target poles ρ_i being complex is also included as this gives the designer extra flexibility which can be useful, and this is a novel contribution to the PFC field.

4.3.1 Formulation with Explicit Real and Imaginary components

Based on a complex numbers representation in Cartesian coordinates, the real and imaginary parts of IM prediction can be handled with real number algebra where each component of the complex numbers (real and imaginary part) is calculated separately. Consider $x = \text{Re}\{x\} + j \text{Im}\{x\}$ and $y = \text{Re}\{y\} + j \text{Im}\{y\}$, then:

$$xy = \underbrace{\left[\text{Re}\{y\} \text{Re}\{x\} - \text{Im}\{y\} \text{Im}\{x\} \right]}_{\text{real part}} + j \underbrace{\left[\text{Re}\{y\} \text{Im}\{x\} + \text{Im}\{y\} \text{Re}\{x\} \right]}_{\text{imaginary part}} \quad (4.12)$$

The update equation of independent model $G_m \Rightarrow y_m^{(i)}(k) = b_i u(k-1) - a_i y_m^{(i)}(k-1)$ can be handled using the following two separate computations.

$$\begin{aligned} \operatorname{Re}\{y_m^{(i)}(k)\} &= \operatorname{Re}\{-a_i\} \operatorname{Re}\{y_m^{(i)}(k-1)\} - \operatorname{Im}\{-a_i\} \operatorname{Im}\{y_m^{(i)}(k-1)\} + \operatorname{Re}\{b_i\}u(k-1) \\ \operatorname{Im}\{y_m^{(i)}(k)\} &= \operatorname{Re}\{-a_i\} \operatorname{Im}\{y_m^{(i)}(k-1)\} + \operatorname{Im}\{-a_i\} \operatorname{Re}\{y_m^{(i)}(k-1)\} + \operatorname{Im}\{b_i\}u(k-1) \end{aligned} \quad (4.13)$$

Since $u(k)$ need to be real while all the complex number appears to be in the conjugates, all the imaginary terms must cancel out and only the real part of the term $\beta_i u^{(i)}(k)$ needs to be computed as:

$$\begin{aligned} \operatorname{Re}\{\beta_i u^{(i)}(k)\} &= \operatorname{Re}\left\{ (1 - \rho_1) \frac{\beta_i \gamma_i}{b_i \sum_{j=1}^n \gamma_j} \right\} r(k) + \operatorname{Re}\left\{ (a_i + \rho_1) \frac{\beta_i}{b_i} \right\} \operatorname{Re}\{y_m^{(i)}(k)\} \\ &\quad - \operatorname{Im}\left\{ (a_i + \rho_1) \frac{\beta_i}{b_i} \right\} \operatorname{Im}\{y_m^{(i)}(k)\} \end{aligned} \quad (4.14)$$

Remark 4.3. *The paper (Zabet et al., 2017) also pointed out that compared to PP-PFC using complex algebra, the increase in computational demand using real number algebra is inconsequential although the coding is slightly more involved.*

4.3.2 Formulation with Real Numbers

The main concept deployed next is to exploit the structure in the Independent Model of Figure 4.1 in order to reduce the control law to an even simpler final form. Ironically, there is a partial move away from the partial fraction expansion in first-order terms to the final implementation so that the implied partial fractions are all real, although the full decomposition structure is still implicit in the control law design.

Considers the parts of (4.11, 4.14) linked to complex conjugate pairs of poles in $G(z)$. The one-step-ahead prediction models for the summed outputs of G_1 , G_2 and the output of $G_{1,2} = G_1 + G_2$ must match, assuming the inputs into each are the same. This means the complex states of G_1 , G_2 can be inferred from the real states of $G_{1,2}$:

$$\begin{aligned} G_{1,2} = G_1 + G_2 &= \frac{b_1 z^{-1}}{1 + a_1 z^{-1}} + \frac{b_2 z^{-1}}{1 + a_2 z^{-1}} = \frac{B_1 z^{-1} + B_2 z^{-2}}{1 + (a_1 + a_2)z^{-1} + a_1 a_2 z^{-2}} \quad (4.15) \\ y_m^{(1,2)}(k+1) &= B_1 u(k) + B_2 u(k-1) - (a_1 + a_2) y_m^{(1,2)}(k) - (a_1 a_2) y_m^{(1,2)}(k-1) \end{aligned}$$

$$\left. \begin{aligned} y_m^{(1)}(k+1) &= b_1 u(k) - a_1 y_m^{(1)}(k) \\ y_m^{(2)}(k+1) &= b_2 u(k) - a_2 y_m^{(2)}(k) \end{aligned} \right\} \Rightarrow y_m^{(1,2)}(k+1) = y_m^{(1)}(k+1) + y_m^{(2)}(k+1) \quad (4.16)$$

In consequence, ignoring the dependence on the term $u(k)$ which is yet to be determined, one can write that:

$$\begin{aligned} -a_1 y_m^{(1)}(k) - a_2 y_m^{(2)}(k) &= B_2 u(k-1) - (a_1 + a_2) y_m^{(1,2)}(k) - (a_1 a_2) y_m^{(1,2)}(k-1) \\ y_m^{(1)}(k) + y_m^{(2)}(k) &= y_m^{(1,2)}(k) \end{aligned} \quad (4.17)$$

Therefore, given they are conjugates, the values $y_m^{(1)}$, $y_m^{(2)}$ can be inferred from these simultaneous equations (noting that in both the imaginary parts are zero by definition).

$$\begin{aligned} 2 \begin{bmatrix} -\operatorname{Re}\{a_1\} & \operatorname{Im}\{a_1\} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \operatorname{Re}\{y_m^{(1)}(k)\} \\ \operatorname{Im}\{y_m^{(1)}(k)\} \end{bmatrix} \\ = \begin{bmatrix} B_2 u(k-1) - 2 \operatorname{Re}\{a_1\} y_m^{(1,2)}(k) - a_1 a_1^* y_m^{(1,2)}(k-1) \\ y_m^{(1,2)}(k) \end{bmatrix} \Rightarrow \\ \begin{bmatrix} \operatorname{Re}\{y_m^{(1)}(k)\} \\ \operatorname{Im}\{y_m^{(1)}(k)\} \end{bmatrix} = \frac{\begin{bmatrix} 0 & \operatorname{Im}\{a_1\} \\ 1 & \operatorname{Re}\{a_1\} \end{bmatrix}}{2 \operatorname{Im}\{a_1\}} \begin{bmatrix} B_2 u(k-1) - 2 \operatorname{Re}\{a_1\} y_m^{(1,2)}(k) - a_1 a_1^* y_m^{(1,2)}(k-1) \\ y_m^{(1,2)}(k) \end{bmatrix} \end{aligned} \quad (4.18)$$

In addition, the real value of the proposed weighted input signal $\operatorname{Re}\{\beta_i u_i(k) + \beta_{i+1} u_{i+1}(k)\}$ for the two sub-models having complex conjugated poles a_i and a_{i+1} comprises numerous components ($K_{0,i}$, $K_{1,i}$, $K_{2,i}$) which can be computed off-line and stored as:

$$\begin{aligned} \operatorname{Re}\{\beta_i u^{(i)}(k) + \beta_{i+1} u^{(i+1)}(k)\} &= \operatorname{Re} \left\{ \underbrace{\frac{1 - \rho_1}{\sum_{j=1}^n \gamma_j} \left[\frac{\beta_i \gamma_i}{b_i} + \frac{\beta_{i+1} \gamma_{i+1}}{b_{i+1}} \right]}_{K_{0,i}} \right\} r(k) \\ &+ 2 \operatorname{Re} \left\{ \underbrace{\left(a_i + \rho_1 \right) \frac{\beta_i}{b_i}}_{K_{1,i}} \operatorname{Re}\{y_m^{(i)}(k)\} - 2 \operatorname{Im} \left\{ \underbrace{\left(a_i + \rho_1 \right) \frac{\beta_i}{b_i}}_{K_{2,i}} \operatorname{Im}\{y_m^{(i)}(k)\} \right\} \right\} \end{aligned} \quad (4.19)$$

Thus, the proposed common input signal $\operatorname{Re}\{\beta_i u^{(i)}(k) + \beta_{i+1} u^{(i+1)}(k)\}$ for the two sub-models having complex conjugate poles can be simplified to a second-order control law which is based solely on real number algebra and using the states of the second-order model $G_{i,i+1}$ by substitution of (4.18) into (4.19) as:

$$\begin{aligned} \operatorname{Re}\{\beta_i u^{(i)}(k) + \beta_{i+1} u^{(i+1)}(k)\} &= K_0^{(i,i+1)} r(k) + K_1^{(i,i+1)} y_m^{(i,i+1)}(k) \\ &+ K_2^{(i,i+1)} y_m^{(i,i+1)}(k-1) + K_3^{(i,i+1)} u(k-1) \end{aligned} \quad (4.20)$$

where:

$$\begin{aligned} K_0^{(i,i+1)} &= K_{0,i}; & K_1^{(i,i+1)} &= \frac{K_{1,i} \operatorname{Im}\{a_i\} - K_{2,i} \operatorname{Re}\{a_i\}}{2 \operatorname{Im}\{a_i\}}; \\ K_2^{(i,i+1)} &= -\frac{K_{2,i} a_i a_i^*}{2 \operatorname{Im}\{a_i\}}; & K_3^{(i,i+1)} &= \frac{K_{2,i} B_{2,i}}{2 \operatorname{Im}\{a_i\}} \end{aligned} \quad (4.21)$$

Remark 4.4. Only the component of the control law corresponding to pairs of complex poles needs to use the formulation of (4.20). The contribution of submodels with real poles can use the simpler formulation. A comparison of computation loading between the alternative approaches is given in Table 4.1.

Table 4.1: Computational loading for different realizations of PP-PFC for second order system.

PP-PFC with complex algebra	PP-PFC with calculating real/imag. parts	PP-PFC with real algebra
13 operations	11 operations	7 operations

Remark 4.5. There is typo error in the original paper (Zabet et al., 2017) for Table 4.1. In order to compute a control input specifically for a second order system $\beta_i u^{(i)}(k) + \beta_{i+1} u^{(i+1)}(k)$, the PP-PFC with complex algebra (4.11) requires 13 operations (summation and multiplication), while PP-PFC with real/imaginary part (4.14) needs 11 operations and PP-PFC with real number algebra (4.20) only uses 7 operations.

4.4 RESULTS

This section provides some of the numerical examples from the paper (Zabet et al., 2017) that compare the simulation times of the control (as an indicator to the simplicity of the control action calculation) using classical PP-PFC, PP-PFC with real and imaginary parts calculation, and the new formulated PP-PFC algorithm, for various choices of ρ . Consider an arbitrary third order under-damped process G_1 with poles at $-0.9, -0.9 \pm 0.4j$:

$$G_1 = \frac{-0.66z^{-1} + 0.08z^{-2} + 0.6z^{-3}}{1 - 2.72z^{-1} + 2.626z^{-2} - 0.8924z^{-3}} \quad (4.22)$$

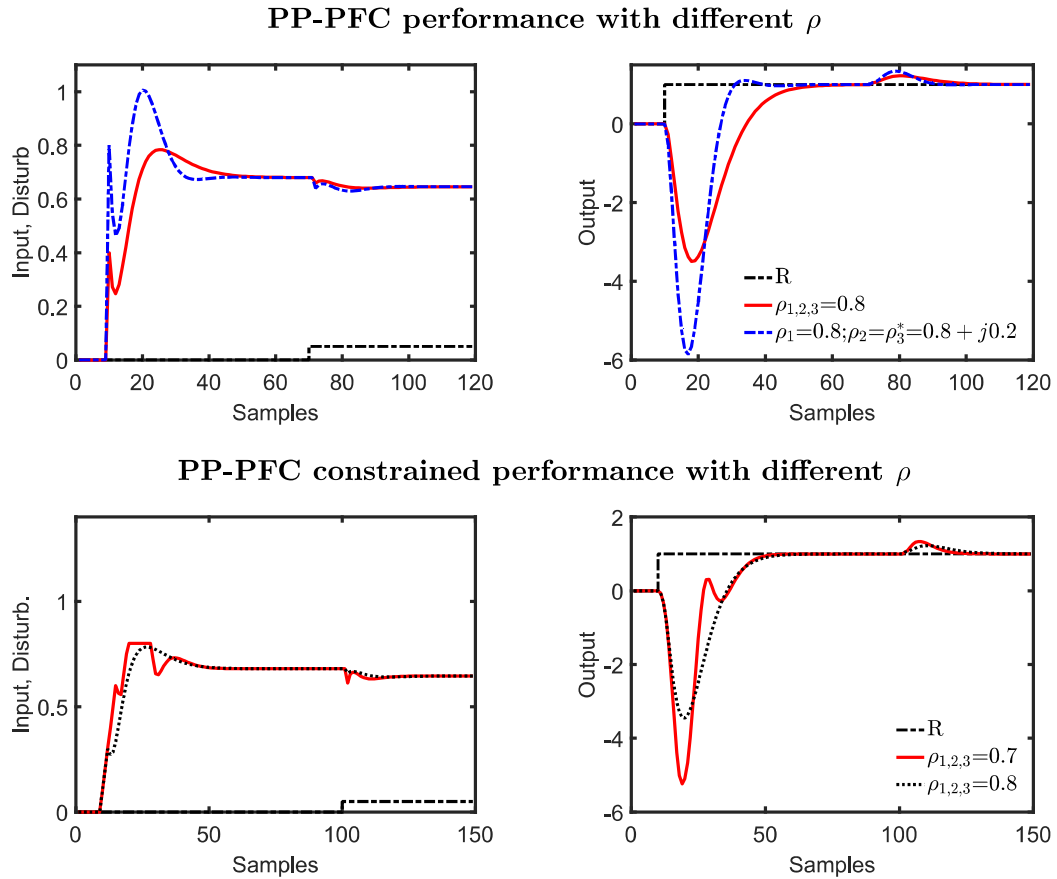


Figure 4.2: Unconstrained and constrained performances of PP-PFC with different ρ .

Figure 4.2 demonstrates the response of PP-PFC with and without constraints for various choices of ρ . An output disturbance is added around the 70th sample to demonstrate the disturbance compensating ability of the approach. These results highlight that:

1. The proposed PP-PFC gives effective control while retaining an intuitive link to the resulting closed-loop behavior with the tuning parameter ρ . Moreover, a user can select the target pole as being complex, which is not applicable for conventional PFC. When the controller is tuned with $\rho_{1,2,3} = 0.8$, the response converges within 50 samples. Conversely, with $\rho_1 = 0.8; \rho_2 = \rho_3^* = 0.8 + j0.2$, a faster response is obtained where it settle around 38 samples with maximum overshoot of 0.1.
2. Table 4.2 provides the simulation times comparison for different formulation of PP-PFC. it is shown that the new formulated PP-PFC provides the fastest control action

Table 4.2: Relative simulation times of the different realizations of PP-PFC for process G_1

ρ	PP-PFC with complex algebra	PP-PFC with calculating real/imag. parts	PP-PFC with real algebra
$\rho_{1,2,3} = 0.8$	100%	45%	42%
$\rho_1 = 0.8$ $\rho_{2,3} = 0.8 \pm j0.2$	100%	41%	37%

calculations compared to the one with explicit calculation of real and imaginary part, and the PP-PFC using complex algebra.

3. It is also clear that PP-PFC manages to satisfy the implied input limit $\bar{x} = 0.8$ and rate limit $\overline{\Delta u} = 0.1$ effectively without detriment to the closed-loop performance.

It is also shown in Zabet et al. (2017) that PP-PFC based on real numbers algebra managed to control real laboratory hardware that is the Quanser servo with a flexible link (see Figure 4.3) with the different selection of poles according to the desired settling time. The same performance as in the previous simulation is obtained. Figure 4.4 shows the controller managed to provide a smooth tracking to the desired target and while, retaining the intuitive link between the target dynamic ρ and the closed-loop convergence speed. A detail quantify performance for the responses are provided in Table 4.3.



Figure 4.3: The experimental plant.

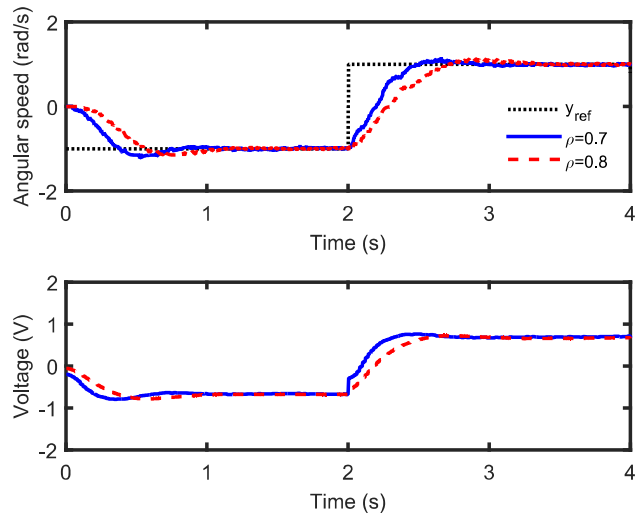


Figure 4.4: PP-PFC performance of the under-damped Quanser servo with flexible joint.

Table 4.3: Quantify PP-PFC performance for Quanser servo with flexible joint

ρ	Rise time	Settling time	Maximum overshoot
0.7	0.4s	0.85s	0.16
0.8	0.57s	1.1s	0.12

Remark 4.6. *The concept of PP-PFC has a similarity as the traditional pole-placement method (Åström & Murray, 2010). Both methods will produce a same response if the poles are placed in the same position. Nevertheless, the PP-PFC will provide the advantage in predictions and constraints implementation as it is well noted that it is quite challenging to satisfy any limits with the traditional pole-placement method. As discussed before in Chapter 3, a special tools such as anti-wind up is needed (Åström & Häggglund, 2001) which often requires an ad-hoc tuning procedure. Future work will look at a more systematic comparison between these two methods since designing the traditional pole-placement is non-simple and beyond the remit of this work.*

4.5 SUMMARY

In summary, this work has proposed a new PFC approach to handle systems with under-damped open-loop dynamics where in many cases, a conventional PFC approach is difficult to tune. Critically, the overall coding complexity and requirements are similar to the code of the conventional PFC, but a core advantage is that the tuning options are now more straightforward than with a conventional algorithm. In fact, it is shown additionally that one is now able to select the target closed-loop pole to be complex and this is often advantageous compared to the restriction to real poles with conventional PFC. In addition, the new formulation of the PP-PFC algorithm (for systems with under-damped open-loop dynamics) reduces the calculation efforts in comparison to the conventional PP-PFC formulation because of dealing with real numbers only. A further advantage is that PP-PFC can be used in PLC or decentralized control systems which usually do not support complex algebra.

Nevertheless, there are also several limitations and drawbacks that should be noted when using the current concept of PP-PFC. Firstly, the controller can only work with a stable and converging system since the IM structure is not able to handle a diverging process. Secondly, it also may face difficulty in satisfying long-range constraints which are often required in the industry since its underlying framework is mainly designed for a single horizon $n_y = 1$. Consequently for an under-damped process, PP-PFC may produce an oscillatory prediction when satisfying output limits if the selected validation horizon is greater than one. Due to these circumstances, a user may desire a more generalised approach that can handle different types of challenging dynamics system while satisfying the implied constraints effectively. This alternative method will be discussed later in Chapter 6.

Chapter 5

LAGUERRE BASED PFC

This chapter presents a summary of the author's contributions from four different papers (Abdullah & Rossiter, 2016; Abdullah et al., 2017; Abdullah & Rossiter, 2019b, 2018a) which are related to the development of PFC using Laguerre based prediction to achieve the first research objective. Section 5.1 starts with some illustrations, which demonstrate the prediction inconsistency of PFC and its impact on the tuning efficacy. In order to overcome these issues, Section 5.2 introduces the concept of Laguerre based PFC (LPFC) and continues with the development of a more systematic constraint handling approach in Section 5.3. The next Section 5.4 provides a formal comparison of the closed-loop performance and efficacy of constraints handling between the traditional PFC and LPFC control law with different parameterisations of Laguerre polynomial. Section 5.5 extends the use of LPFC to handle integrating processes and Section 5.6 provides the overall summary for this chapter.

5.1 TUNING WEAKNESS OF PFC

Before introducing the concept of LPFC, it is important first to understand the current tuning weaknesses of traditional PFC where its simplification using a constant future input assumption may lead to ill-posed decision making. Figure 5.1 illustrates the comparison between the open-loop behaviour and the closed-loop response together with the expected target trajectory $\lambda = 0.7$ for arbitrary first-order G_2 , second-order G_3 and third order non-minimum phase G_4 systems with varying selection of n_y .

$$G_2 = \frac{0.25z^{-1}}{1 - 0.8z^{-1}} \quad (5.1)$$

$$G_3 = \frac{0.04z^{-1}}{1 - 1.6z^{-1} + 0.64z^{-2}} \quad (5.2)$$

$$G_4 = \frac{-0.0569z^{-1} + 0.0514z^{-2} + 0.0502z^{-3}}{1 - 1.9842z^{-1} + 1.3301z^{-2} - 0.3012z^{-3}} \quad (5.3)$$

From the observation, a smaller n_y provides an output response (y_p red solid line) closer to the target trajectory (r black circle) compared to the output of larger n_y (y_p blue solid line) for processes G_2, G_3, G_4 . However, the consistency between an open-loop output prediction at the first sample ($y_{p|1}$ red dotted line) and the actual behaviour (y_p red solid line) is poor. Conversely, larger n_y provides better prediction consistency but the efficacy of λ as a tuning parameter becomes less significant, where the actual response (y_p blue solid line) converges far away from the target trajectory (r black circle). Indeed, a user may desire to use a smaller n_y for the best closed-loop performance, yet the prediction consistency issue also should be considered, since an ill-posed solution may affect the constrained performance of PFC if the validation horizon n_i (used for checking any future violation) is larger than the coincidence horizon n_y . Thus, a user needs to select the coincidence horizon n_y carefully to get the best trade-off between the two outcomes (J. A. Rossiter & Haber, 2015).

In order to improve this trade-off, the future input dynamics may require extra d.o.f to ensure consistency between open-loop prediction and closed-loop behaviour while retaining the effect of λ . This work proposes a simple modification to the conventional control law by utilising Laguerre polynomials to replace the future input assumption of PFC to overcome this issue.

5.2 LAGUERRE BASED PFC

The Laguerre function is a powerful tool to represent d.o.f and usually is used in the system identification field (Škultéty et al., 2013; Sarah et al., 2014; Wahlberg, 1991). A higher order process can be modelled with a few low order Laguerre networks to simplify its coding and implementation. In MPC, this orthonormal function is used to enhance the horizon effect, either in the continuous (L. Wang, 2001) or discrete (L. Wang, 2004) time domain. In essence, the effect of a large input horizon can be obtained with a small number of parameters/d.o.f.. This simplification reduces the implied computational burden dramatically while improving the controller performance when longer prediction horizons are needed (Abdullah & Idres, 2014b; L. Wang, 2004). Moreover, the Laguerre function can

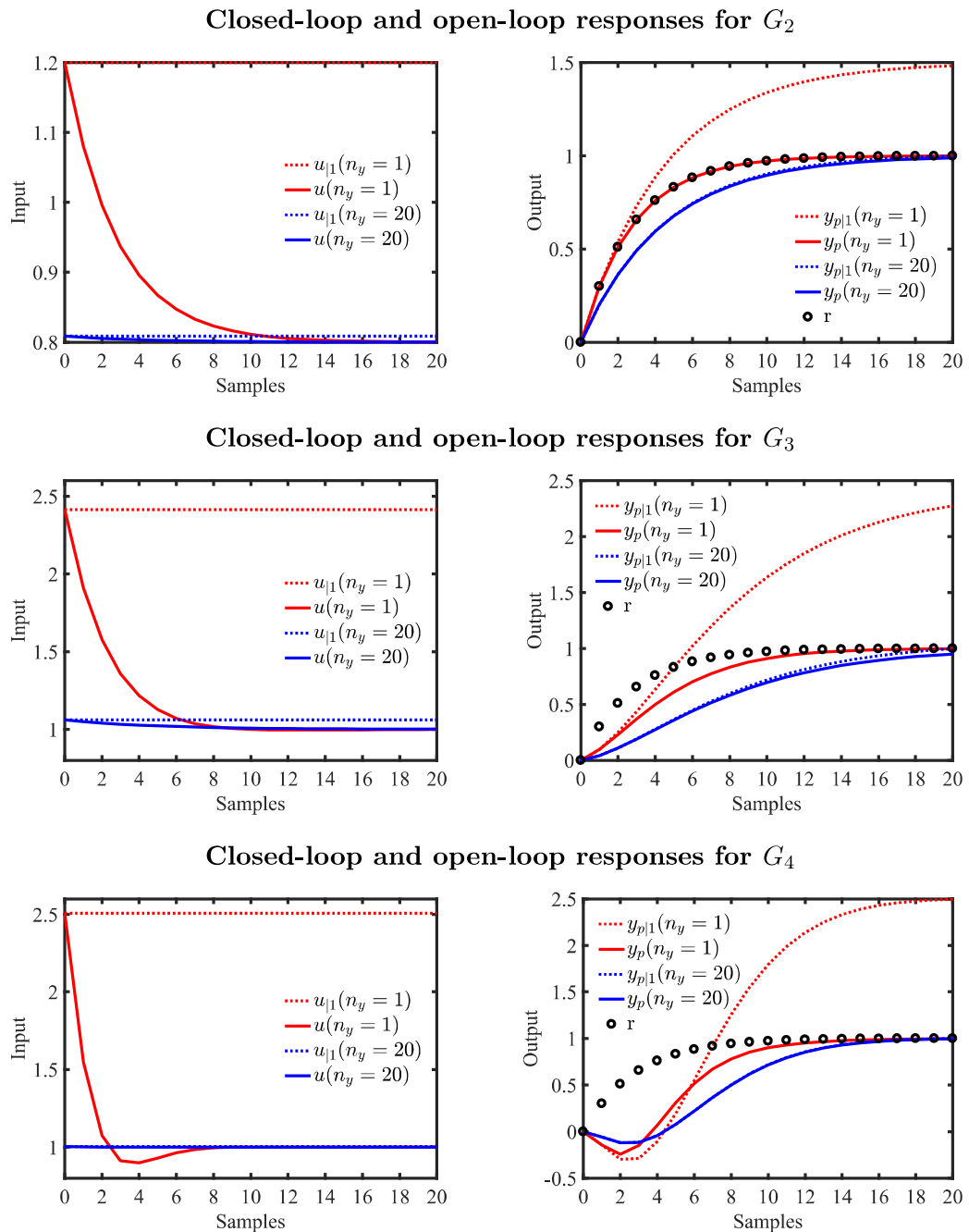


Figure 5.1: Comparison between the open-loop prediction at a first sample with the implied closed-loop behaviour of PFC for the process G_2 , G_3 and G_4 with varying n_y .

capture the desired system dynamics to some extent in that its convergence is linked to its pole which is a design parameter. This feature makes it an attractive option to parametrise d.o.f for prediction, while improving the feasibility and performance of predictive controllers (J. A. Rossiter et al., 2010; J. A. Rossiter & Wang, 2008; Khan & Rossiter, 2011).

Nevertheless, this work adopts a different approach, where the main idea is to replace the PFC constant input prediction with an alternative dynamics assumption based on a Laguerre polynomial. This modification is expected to give a well-posed solution while enhancing the effectiveness of λ as the tuning parameter.

5.2.1 Formulation of LPFC

The core concept of LPFC is to embed two dynamics within the predicted input:

1. The expected steady-state input.
2. The d.o.f which converges to the steady state value with dynamics according to the selected Laguerre pole.

Although Laguerre polynomials of high order have been used in MPC (Abdullah & Idres, 2014b; L. Wang, 2009), this work employs a first-order Laguerre polynomial for shaping the input prediction to ensure a smooth convergence while retaining the simplicity of the PFC concept. The first order Laguerre function, with modified scaling for simplicity, can be expressed in vector form as:

$$L(z) = \frac{1}{1 - a_l z^{-1}} \equiv 1 + a_l z^{-1} + a_l^2 z^{-2} + \dots \quad (5.4)$$

where a_l is the Laguerre pole. Define $L = [1, a_l, a_l^2, \dots, a_l^{n_y-1}]^T$. Now we are in a position to define the input prediction to be deployed in PFC.

Theorem 5.1. *A future input parameterised as*

$$u(z) = \frac{u_{ss}}{1 - z^{-1}} + \frac{\eta}{1 - a_l z^{-1}} \quad (5.5)$$

will give output predictions which settle at the desired steady-state, where η represents a degree of freedom and u_{ss} denotes the expected steady state input to remove offset.

Proof. The signal defined in (5.5) has the property that:

$$\lim_{k \rightarrow \infty} u_k = u_{ss} \quad (5.6)$$

This is obvious as the definition of $L(z)$ shows that the components converge to zero asymptotically. By definition u_{ss} is:

$$\lim_{k \rightarrow \infty} u_k = u_{ss} \quad \Rightarrow \quad \lim_{k \rightarrow \infty} y_k = R \quad (5.7)$$

where R is the set target. As the Laguerre polynomial evolves over the horizon, it will converge to the steady state input with respect to its pole a_l . \square

Noting that (5.5) is equivalent to $\underline{u}_k = L\eta_k + u_{ss}$, the output prediction is modified by substituting this into the model prediction of (2.8):

$$y_{p,k+n_y|k} = H(u_{ss} + L\eta_k) + P\underline{u}_k + Qy_{m,k} + d_k \quad (5.8)$$

Algorithm 5.1. Define the output prediction n_y -steps ahead using (5.8). The LPFC law is defined by substituting this prediction into target trajectory (2.2) and solving for the d.o.f. η .

$$\eta_k = \frac{(1 - \lambda^{n_y})R + \lambda^{n_y}y_{p,k} - (h_{n_y}u_{ss} + P_{n_y}\underline{u}_k + Q_{n_y}y_{m,k} + d_k)}{H_{n_y}L} \quad (5.9)$$

Due to the receding horizon principle (L. Wang, 2009) and the definition of $L(z)$, the current input is defined as:

$$u_k = u_{ss} + \eta_k \quad (5.10)$$

Remark 5.1. The input steady state value u_{ss} can be estimated as:

$$u_{ss} = G_m(z)^{-1}(R - d_k) \quad (5.11)$$

The inclusion of error term d_k in (5.11) is to ensure an unbiased estimation.

Remark 5.2. For a first-order system, a_l should be equal to λ to ensure consistent dynamics with the target trajectory. Although for higher-order systems, tuning $a_l < \lambda$ can provide faster convergence (Abdullah & Rossiter, 2016), yet to keep the implementation simple, the decay rate of Laguerre is set to $a_l = \lambda$ which is good enough for most of the stable and straightforward dynamics processes.

5.2.2 Results

The proposed control law is implemented in three different processes, which are G_2 , G_3 and G_4 . Figure 5.2 shows that LPFC provides better prediction consistency between the open-loop prediction at the first sample ($y_{p|1}$ blue dotted line) and closed loop response (y_p blue solid line) compared to PFC ($y_{p|1}$ red dotted line and y_p red solid line). Besides, LPFC also improves the overall closed-loop performance, where its actual output converges closer to the desired set point than PFC (except for the first-order process, where the performance is equivalent).

Another interesting observation is that when implementing LPFC with the non-first order processes, Figure 5.3 demonstrates that using smaller a_l provides a faster output response, yet with a more aggressive input demand. Conversely, larger a_l produces slower convergence to the set-point, but with a less aggressive input activity. Nevertheless, these changes still provide a well-posed decision making, where the future input trajectory will eventually converge to its steady-state value.

In essence, LPFC can improve the prediction consistency and the overall closed-loop response of the traditional PFC while retaining the simplicity of tuning and implementation. The UKACC 2018 conference paper (Abdullah & Rossiter, 2016), which is attached in Appendix B provides a more detailed discussion and analysis for this work. Acknowledging the advantages of LPFC, the next section will demonstrate its benefit when handling constraints.

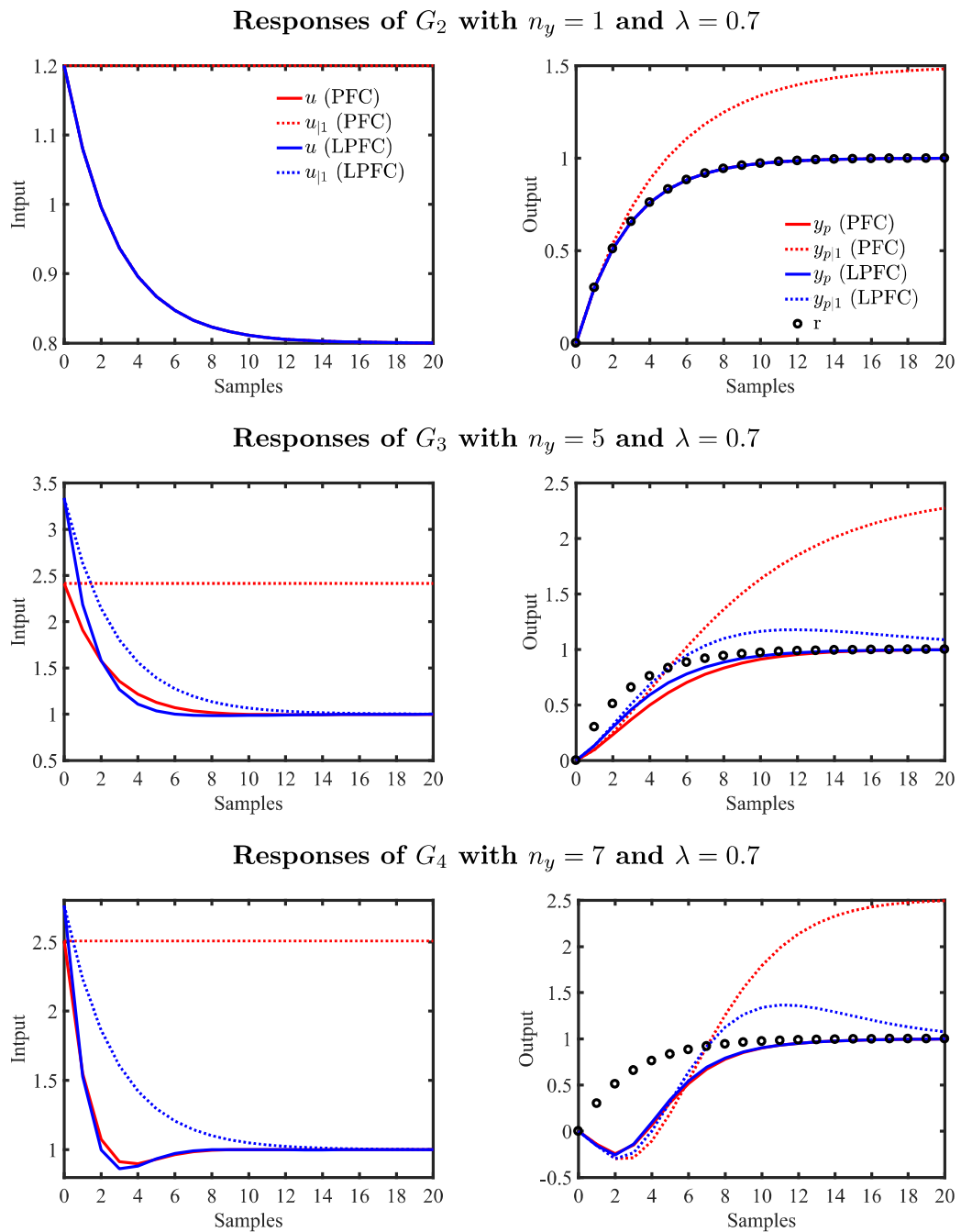


Figure 5.2: The response of open-loop prediction at a first sample and the implied closed-loop behaviour of the process G_2 , G_3 and G_4 for nominal PFC and LPFC.

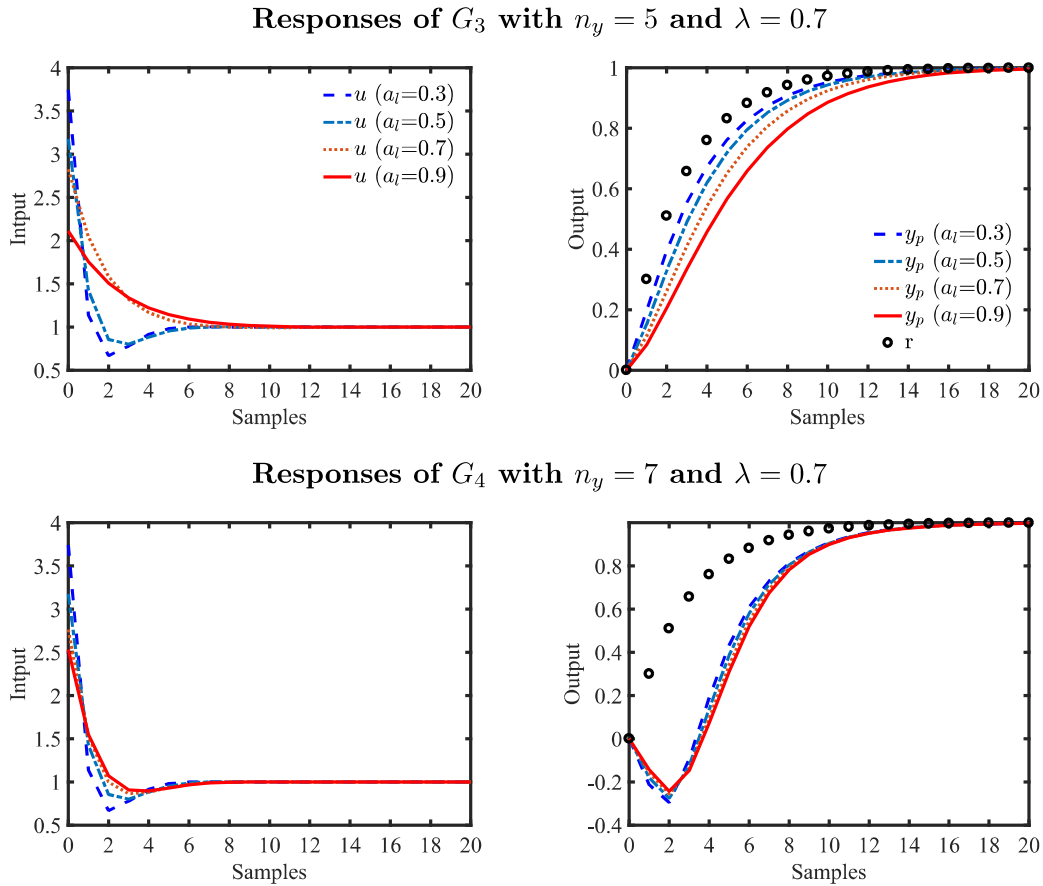


Figure 5.3: Tuning LPFC with different Laguerre pole a_l for the process G_3 and G_4 .

5.3 SYSTEMATIC CONSTRAINED HANDLING WITH LPFC

As discussed in the previous two chapters, the traditional PFC utilises a multiple regulators approach to handle the soft constraints while employing a simple clipping approach for the hard constraints. This suboptimal framework often works in real processes, but the performance may be very conservative since the second regulator needs to use a slower λ to avoid conflict with the hard constraints in the primary regulator. Besides, the constant input assumption also may detect an early violation due to the prediction mismatch especially when the validation horizon $n_i > n_y$. Although this method uses a real prediction, the second regulator is not aware of the other controller behaviour. Hence, it all comes down to the experience or understanding of a user to tune this controller effectively.

Since PFC utilises the model prediction, it is possible to adopt a similar constraint handling concept as in the modern MPC, which provides a more systematic tuning and implementation compared to the multiple-regulators approach. Nevertheless, this strategy also needs a good prediction consistency or well-posed decision to compute an effective constrained solution for satisfying the limits within a long validation horizon. Noting that the previous LPFC concept manages to improve the prediction consistency issue, the next paper (Abdullah et al., 2017) which is attached in Appendix C explores its capability to improve the constraints handling of PFC. There are two main contributions, which are:

1. The substitution of the typical PFC multiple-regulators scheme (Richalet & O'Donovan, 2009; Abu et al., 1991) with a vector approach as in MPC, which is more cost effective, systematic and easy to tune.
2. The improvement in constrained performance with LPFC, where it provides more precise and less conservative solutions compared to the traditional approach.

5.3.1 Constraint Handling via Vector Approach for PFC

Given that PFC deploys only simple coding to enable use on low-level processors, the constraint handling is defined to be simple and thus avoids the optimisers common in more mainstream algorithms (Abu et al., 1991; Gilbert & Tan, 1991). Assume constraints, at every sample, on input and states as follows:

$$\underline{\Delta u} \leq \Delta u_k \leq \overline{\Delta u}; \quad \underline{u} \leq u_k \leq \overline{u}; \quad \underline{y} \leq y_k \leq \overline{y} \quad (5.12)$$

where $\Delta u_k = u_k - u_{k-1}$ is the input increment or rate.

For input and rate constraints, this strategy uses a similar saturation approach as in Section 2.1.5, that is, if the proposed u_k violates (5.12), then move to the nearest value which does not. The modernising part will be on the implementation of output and state constraints by adopting a standard core concept as in the MPC literature the so-called ONEDOOF and reference governor (J. A. Rossiter et al., 2001).

1. For a suitable validation horizon n_i , compute the entire set of future predictions $y_{p,k+i|k} = H_i u_{\underline{k}} + P_i u_{\underline{k}} + Q_i y_{\underline{m},k} + d_k$, $i = 1, \dots, n_i$. Use the compact notation $y_{p,\underline{k}+1} = H u_{\underline{k}} + P u_{\underline{k}} + Q y_{\underline{m},k} + d_k$ to capture the output predictions in a single vector where $y_{p,\underline{k}+1} = [y_{p,k+1|k}, y_{p,k+2|k}, \dots, y_{p,k+n_i|k}]^T$.
2. Combine the input, rate and output constraints along with the output predictions into a single set of linear inequalities of the form:

$$C u_k \leq f_k \quad (5.13)$$

$$C = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ H \\ -H \end{bmatrix}; \quad f = \begin{bmatrix} \bar{u} \\ -u \\ \frac{\Delta u}{\Delta u} \\ -\frac{\Delta u}{\Delta u} \\ \bar{y} \\ -y \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ -u_{k-1} \\ u_{k-1} \\ P u_{\underline{k}} + Q y_{\underline{m},k} + d_k \\ -P u_{\underline{k}} - Q y_{\underline{m},k} - d_k \end{bmatrix}$$

where f_k depends on past data in $u_{\underline{k}}, y_{\underline{m},k}$ and on the limits. The horizon for the output predictions $y_{\underline{k}+1}$, and thus the row dimension of H , should be long enough to capture all core dynamics!

3. The predictions satisfy constraints iff (5.13) is satisfied and thus a conventional MPC algorithm will ensure this occurs and that is consider inequalities $C u_k \leq f_k$ explicitly rather than an alternative constraint representation which may be suboptimal or approximate.

The proposed PFC constraint handling algorithm is summarised next. This uses a single simple loop to select the u_k closest to the unconstrained solution of (2.10) which satisfies (5.13).

Algorithm 5.2. *At each sample:*

1. Define the unconstrained value for u_k from (2.10).
2. Define the vector f_k of (5.13) (it is noted that C does not change).

3. Use a simple loop covering all the rows of C as follows:

(a) Check the i th constraint that is the i th row of $Cu_k \leq f_k$ using $a_i = C_i u_k - f_{k,i}$.

(b) If $a_i > 0$, then set $u_k = (f_{k,i})/C_i$, else leave u_k unchanged.

Remark 5.3. *Recursive feasibility is defined as the ability of controller to ensure that given a current feasible solution, it is possible to guarantee feasibility at the next sample and thus at all future samples (J. A. Rossiter, 2018). Failure to ensure this condition can allow infeasibility at some point in the future where the control may become undefined. In other term, ones needs to ensure that the control input:*

$$u_{k+1|k+1} = u_{k+1|k} \quad (5.14)$$

Theorem 5.2. *In the nominal case and for stable open-loop processes, Algorithm 5.2 is guaranteed to be recursively feasible and moreover converge to a feasible value for u_k that is closest to the unconstrained choice.*

Proof. Assume feasibility at initiation and also note that for stable open-loop processes the predicted outputs are convergent for constant future inputs $u_{k+i} = u_k, \forall i > 0$. Consequently, if one has feasibility at sample $k - 1$, then the choice $u_k = u_{k-1}$ must be feasible, that is satisfy (5.13). Hence, as long as u_{k-1} is a possible choice (which it must be as all constraints must satisfy $C_i u_{k-1} \leq f_{k,i}$), recursive feasibility is assured and a feasible solution will lie between u_{k-1} and the unconstrained u_k . Each constraint $C_i u_k \leq f_{k,i}$ will either lower or upper bound u_k ; if $u_k < u_{k-1}$ then only the lower bounds can be active and if $u_k > u_{k-1}$ only the upper bounds. Hence, an active constraint $C_i u_k \leq f_{k,i}$ will bring u_k closer to u_{k-1} if violated by the unconstrained u_k but otherwise will have no affect. In consequence, the final u_k will be only as close to u_k as it needs to be to satisfy all the active constraints and thus, is also as close to the original unconstrained u_k as possible. \square

Remark 5.4. *Because this approach (Algorithm 5.2) deploys a very simple for-loop, coding is simple and very fast and certainly far more simple than traditional MPC approaches which often use a quadratic program albeit potentially suboptimal but equally, more systematic and probably quicker than the ad-hoc approaches common with PID.*

5.3.2 Constraint Handling via Vector Approach for LPFC

The procedure for constraint handling for LPFC is analogous to that discussed in the previous subsection. The core conceptual difference is that the d.o.f is now η_k . In principle, the input constraints need to be checked along the entire prediction horizon. However, given the maximum magnitude increments occur at the first sample, only Δu_k needs to be checked, and similarly, the maximum/minimum of u_{k+i} has a simple dependence on η_k, u_{ss} so again only one value needs to be checked.

As in Algorithm 5.2, the aim is to choose the d.o.f. as close as possible to their unconstrained values, and subject to (5.12). Inclusion of Laguerre based input dynamics $\underline{u}_k = L\eta + u_{ss}$ changes the vector of linear inequality into:

$$C\eta_k \leq f_k \quad (5.15)$$

$$C = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ HL \\ -HL \end{bmatrix}; \quad f = \begin{bmatrix} \bar{u} \\ -\underline{u} \\ \frac{\Delta u}{\Delta u} \\ -\underline{\Delta u} \\ \bar{y} \\ -\underline{y} \end{bmatrix} - \begin{bmatrix} u_{ss} \\ u_{ss} \\ u_{ss} - u_{k-1} \\ -u_{ss} + u_{k-1} \\ Hu_{ss} + P\underline{u}_k + Qy_{\leftarrow m,k} + d_k \\ -Hu_{ss} - P\underline{u}_k - Qy_{\leftarrow m,k} - d_k \end{bmatrix}$$

Then an approach similar to Algorithm 5.2 will solve for η_k .

Corollary 5.1. *In the absence of uncertainty, the inequalities implied in (5.15) are always feasible, assuming feasibility at the previous sample, no changes in the target and a long enough horizon.*

Proof. The structure of the input prediction based on (5.5) is such that, as long as u_{ss} does not change from one sample to the next, then one can always choose η so that the predicted input trajectory is unchanged; this is obvious from the simple exponential structure. Consequently, if there exists an η to satisfy constraints at the previous sample, there must exist a valid value at the current sample. \square

Remark 5.5. *Infeasibility can arise due to too fast or too large changes in the target (or disturbances) as this causes large changes in the value of u_{ss} . However, Laguerre PFC helps*

enormously in this case because the exponential structure embedded into the input prediction automatically slows down any over aggressive input responses and thus significantly increases the likelihood of feasibility being retained. In the worst case, set point changes need to be moderated (as in reference governor approaches (Gilbert & Tan, 1991)).

5.3.3 Results

Two numerical examples are considered as case studies for output and state constraints, respectively. The first example compares the performance of LPFC and PFC using a systematic constraints approach (Algorithm 5.2) for process G_2 . Figure 5.4 demonstrates in the presence of 0.2 input disturbance from 20s to 25s, the constrained solution of LPFC (blue line) satisfies the output limit $\bar{y} = 1.05$ more precisely and less conservatively compared to PFC (red line) when using the validation horizon $n_i = 10$.

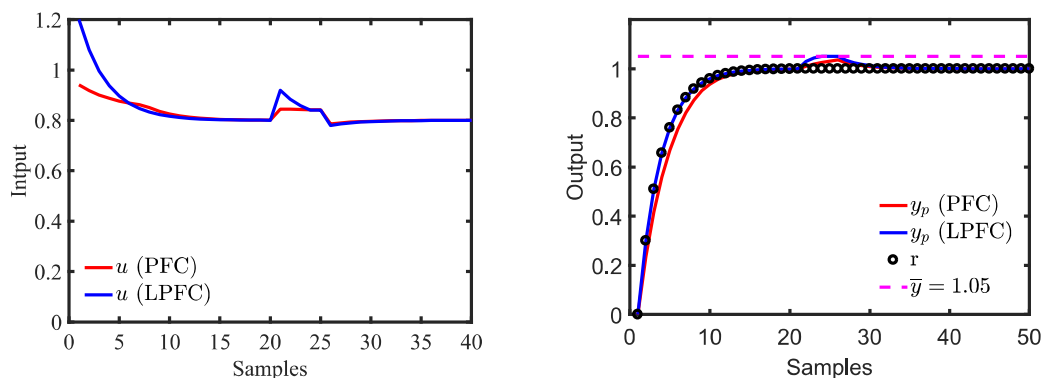


Figure 5.4: Constrained performance of PFC and LPFC for process G_2 with $\lambda = 0.7$ and $n_y = 1$.

The second example considers two processes that run in parallel where the main process $G_{5,1}$ and state process $G_{5,2}$ receive a similar manipulated input u from the regulator. Here, the performance of LPFC using Algorithm 5.2 will be compared with the conventional multiple-regulator scheme (CPFC) as discussed in Section 2.1.5.

$$G_{5,1} = \frac{0.0164z^{-1}}{1 - 0.9835z^{-1}}; \quad G_{5,2} = \frac{0.08914z^{-1} - 0.08674z^{-2}}{1 - 1.918z^{-1} + 0.92z^{-2}} \quad (5.16)$$

For safety and economic reasons, the state is constrained at $\bar{x} = 127$ with a limited input $\bar{u} = 160$, and speed $\overline{\Delta u} = 4$. For a fair comparison, LPFC and the first regulator of

state constrained PFC (CPFC) will both use $n_y = 1$, validation horizon ($i = 68$) and pole ($\lambda = 0.975$) to track the set point ($R = 100$). Since CPFC treats the maximum state as a second target (2.14), the coincidence horizon ($n_x = 30$) and desired pole ($\lambda_x = 0.984$) of the second constraining regulator are selected carefully to satisfy the internal constraints.

Figure 5.5 shows that LPFC outperforms CPFC while satisfying the state constraints. Although the state behaviours (dotted line) of both $x(\text{CPFC})$ and $x(\text{LPFC})$ are within the limits, the output settling time of $y(\text{LPFC})$ 200 samples is almost twice as fast as $y(\text{CPFC})$ (300+ samples) and closer to the target trajectory R .

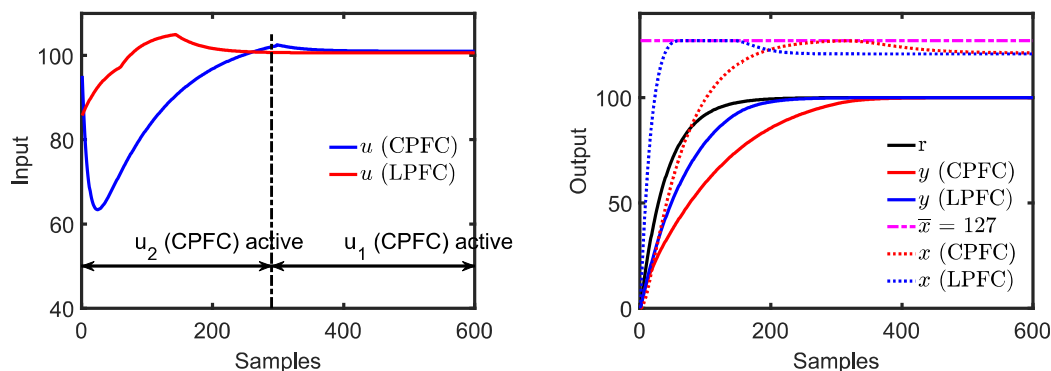


Figure 5.5: Performance of constrained CPFC and LPFC for process G_5 .

Conversely, CPFC requires a careful tuning process and a higher operation cost as two regulators are used simultaneously. Besides, it also needs a large pole to slow down the control response for avoiding any conflict with the internal constraints. Figure 5.6 demonstrates the effect of poor tuning decision with a smaller pole $\lambda_x = 0.963$, where CPFC (blue line) violates the maximum input $\bar{u} = 160$.

In summary, the proposed LPFC control law with the systematic constraints handling approach eliminates the careful tuning process of multiple-regulators since the constraint is now explicitly included in the control computation. Moreover, the algebra for computing the vectors C, f as in (5.15) is the same as that required for computing the predictions and thus is unavoidable where constraint handling is desired and specifically, needs no input or tuning choices from the designer. This work has not investigated the implications of infeasibility due to large disturbances or setpoint changes any further than insisting on

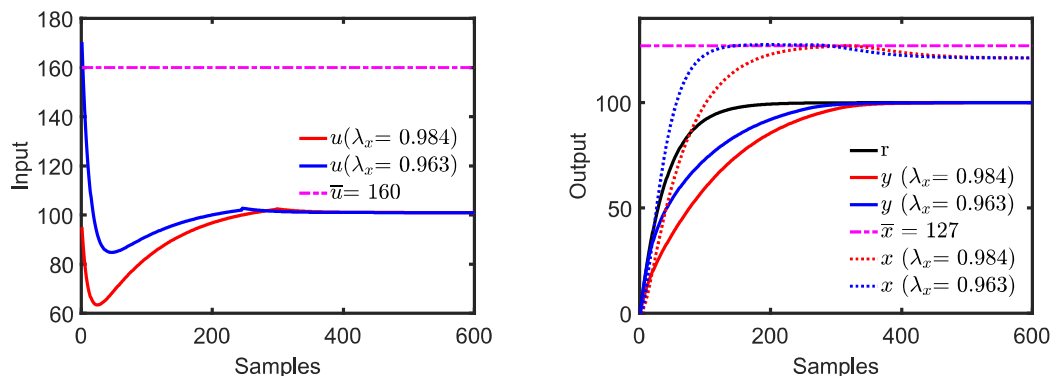


Figure 5.6: CPFC responses with different poles $\lambda_x = 0.984$ and $\lambda_x = 0.963$ for process G_5 .

sensible limits to changes in u_{ss} as that is a more challenging scenario and requires a priori trade-off decisions such as which constraints or requirements to sacrifice during transients. The details formulation and discussion of this work are available in 2017 IFAC conference paper (Abdullah et al., 2017), which is attached in Appendix C.

5.4 DIFFERENT PARAMETERISATIONS OF LPFC

As shown in the previous section, LPFC offers some benefits in providing better prediction consistency, tuning effect and constrained performance compared to the traditional PFC approach. Nevertheless, there is a tendency that the constraint handling of LPFC may face an infeasibility issue especially when there is a significant change in the u_{ss} . Hence, the IJC 2018 journal paper (Abdullah & Rossiter, 2019b) which is attached in Appendix D explores another possibility to use an alternative parameterisation of LPFC explicitly using an input increment for avoiding this issue. Besides, this work also presents a more formal comparison of the closed-loop response and constrained performance between PFC and LPFC with two different parameterisations: (i) one based on the inputs and (ii) another based on the input increments. There are two main findings which are worth highlighting:

1. Mapping Laguerre directly onto the inputs is more preferable than mapping onto the input increments as this enables faster transients and a better usage of the full input range.

2. However, the downside is that it needs to utilise an implied terminal constraint and, as is well known in the literature, terminal constraints can cause conflicts with other constraints and thus, at times, need to be managed carefully.

5.4.1 Parameterisation of LPFC Based on Input Increment

The choices of parameterisation via inputs and input increments are not exactly equivalent and thus would lead to different results in general (Abdullah & Rossiter, 2019b). Noting the definition of first-order Laguerre polynomial in (5.4), an almost equivalent definition could use the input increments and hence:

$$\Delta u_k = \nu_k; \quad \Delta u_{k+1|k} = a\nu_k; \quad \Delta u_{k+2|k} = a^2\nu_k; \quad \dots \quad (5.17)$$

in this case the implied input trajectory would be:

$$u_k = u_{k-1} + \nu_k; \quad u_{k+1|k} = u_{k-1} + (1 + a)\nu_k; \quad u_{k+2|k} = u_{k-1} + (1 + a + a^2)\nu_k; \quad \dots \quad (5.18)$$

Next, note the properties of the geometric sequence $1, 1 + a, 1 + a + a^2, \dots$. It is known that

$$S_a = \sum_{i=0}^{\infty} a^i = \frac{1}{1 - a} \quad \text{and} \quad \sum_{i=0}^n a^i = \frac{1 - a^{n+1}}{1 - a} = (1 - a^{n+1})S_a \quad (5.19)$$

Using this properties, the equivalent sequence of (5.18) can be represented as:

$$\begin{bmatrix} \Delta u_k \\ \Delta u_{k+1|k} \\ \Delta u_{k+2|k} \\ \vdots \end{bmatrix} = \begin{bmatrix} \nu_k \\ a\nu_k \\ a^2\nu_k \\ \vdots \end{bmatrix} \Rightarrow \begin{bmatrix} u_k \\ u_{k+1|k} \\ u_{k+2|k} \\ \vdots \end{bmatrix} = \begin{bmatrix} u_{k-1} + \nu_k \\ u_{k-1} + (1 - a^2)S_a\nu_k \\ u_{k-1} + (1 - a^3)S_a\nu_k \\ \vdots \end{bmatrix} \quad (5.20)$$

Algorithm 5.3. LPFC2: *An alternative PFC control law using a Laguerre parameterisation of the input increment trajectory is defined as follows:*

1. Define the input trajectory from (5.20).

2. Define the output prediction n_y -steps ahead using (5.8) with (5.20) and hence define

$$y_{p,k+n_y|k} = H(u_{k-1} + \underbrace{\begin{bmatrix} 1 \\ (1-a^2)S_a \\ (1-a^3)S_a \\ \vdots \end{bmatrix}}_{L_2} \nu_k) + P u_{\leftarrow k} + Q y_{m,\leftarrow k} + d_k \quad (5.21)$$

3. Substituting output predictions (5.21) into (2.2) with a suitable n_y , the PFC control law can be defined as:

$$\nu_k = \frac{(1 - \lambda^{n_y})R + \lambda^{n_y} y_{p,k} - (h_{n_y} u_{k-1} + P_{n_y} u_{\leftarrow k} + Q_{n_y} y_{m,\leftarrow k} + d_k)}{H_{n_y} L_2} \quad (5.22)$$

where the implied control input becomes:

$$u_k = u_{k-1} + \nu_k \quad (5.23)$$

Corollary 5.2. *The prediction classes for PFC given in (2.10) and with Laguerre based on input increments (5.17) suffer from a critical weakness. In both cases the asymptotic value of the predicted output is highly unlikely to be close to the desired target of R . This is because, the value of u_k satisfying the PFC law definition $r_{k+n} = y_{p,k+n}$ in general will be inconsistent with $u_k = E[u_{ss}]$.*

5.4.2 Constraint Handling Properties for Different Parameterisation of LPFC

The constraint handling procedure for LPFC2 is same with LPFC by using the inequalities vector of (5.15), the only required modifications are to set $u_{ss} = u_{k-1}$ and $L = L_2$. Nevertheless, there are some subtleties which are worth highlighting and link to feasibility.

Lemma 5.1. *For the nominal case, recursive feasibility is guaranteed with LPFC2, irrespective of the choice of target R .*

Proof. Assuming feasibility at sample $k-1$, then the choice $\Delta u_{k+i|k} = \Delta u_{k+i|k-1}$, $\forall i \geq 0$ will give rise to predictions which satisfy constraints (5.12). The choice $\nu_k = a_l \nu_{k-1}$ will enable this choice of future inputs and thus a feasible solution exists at the current sample and, clearly, this statement can be made recursively. \square

Lemma 5.2. *For the nominal case, recursive feasibility is not guaranteed with LPFC.*

Proof. The potential weakness with input prediction class (5.5) is emphasised in the first term $u_k = u_{ss,k} + \eta_k$ as this contains a value, specifically $u_{ss,k}$ which may or may not be feasible. Moreover, consideration of the implied increments shows that $\Delta u_k = u_{ss,k} - u_{ss,k-1} + \eta_k$ could be very large if there is a significant change in $u_{ss,k}$ (that is, $u_{ss,k} \neq u_{ss,k-1}$). To be more precise, the sequence of proposed inputs from the previous sample can be laid alongside the proposed sequence at the current sample:

$$\underset{\rightarrow}{u}_{k|k-1} = \begin{bmatrix} u_{ss,k-1} + a\eta_{k-1} \\ u_{ss,k-1} + a^2\eta_{k-1} \\ u_{ss,k-1} + a^3\eta_{k-1} \\ \vdots \end{bmatrix} \quad : \quad \underset{\rightarrow}{u}_{k|k} = \begin{bmatrix} u_{ss,k} + \eta_k \\ u_{ss,k} + a\eta_{k-1} \\ u_{ss,k} + a^2\eta_k \\ \vdots \end{bmatrix} \quad (5.24)$$

From these it is clear that one can only ensure $\underset{\rightarrow}{u}_{k|k-1} = \underset{\rightarrow}{u}_{k|k}$ if $u_{ss,k} = u_{ss,k-1}$ is unchanged. Without the ability to remain on the same prediction class, recursive feasibility cannot be assured. \square

Theorem 5.3. *In order to ensure recursive feasibility while using LPFC, the user must retain the option to modify $u_{ss,k}$ as required.*

Proof. It is a consequence of Lemma 5.1 whereby the option to choose $u_{ss,k} = u_{ss,k-1}$ enables the selection of $\underset{\rightarrow}{u}_{k|k-1} = \underset{\rightarrow}{u}_{k|k}$, hence guaranteeing feasibility. \square

The requirement in Theorem 5.3 is analogous to reference governor strategies (Gilbert & Tan, 1991) and is unsurprising and indeed also a well known issue within mainstream MPC. That is, large changes in the target can give rise to transient infeasibility where there is a terminal constraint as implicit with input trajectory (5.5) and this is easiest dealt with by slowing the change in target (equivalently modifying the implied terminal constraint). This work will use examples to compare such a strategy with the use of LPFC2 which is more analogous to GPC (Clarke & Mohtadi, 1989) in not having an implied terminal constraint and thus does not require this additional check.

5.4.3 Results

This paper considers several non-first order dynamics where it is known that the conventional PFC is adequate for such systems. The simulation studies will compare the prediction consistency, closed-loop performance and efficacy of constraint handling between PFC, LPFC (with input parameterisation) and LPFC2 (with input increment parameterisation). Since the tuning efficacy of LPFC has been discussed as in the previous two sections and so this section illustrates whether the proposed adoptions of LPFC2 changes any of those insights or not. This thesis only presents one of the given examples in Abdullah & Rossiter (2019b) that is process G_6 :

$$G_6 = \frac{-0.04z^{-1} + 0.1z^{-2}}{1 - 1.4z^{-1} + 0.45z^{-2}}; \quad n = 5; \quad \lambda = 0.7 \quad (5.25)$$

- Figure 5.7 shows that LPFC gives the best prediction where its response is the closest to the target trajectory followed by PFC and LPFC2. This outcome is because the flexibility in LPFC allows a large initial input to get a fast transient and then a gradual decay to the desired steady-state. By contrast, PFC tries to manage everything with the constant input and thus fails. LPFC2 has a different weakness: as the increments Δu_k all have the same sign, the required early increments to satisfy the control law inevitably lead to an asymptotic input trajectory which grows too large and ironically, also implies a less aggressive initial input move which could provide relatively slow transients compared to PFC and LPFC.

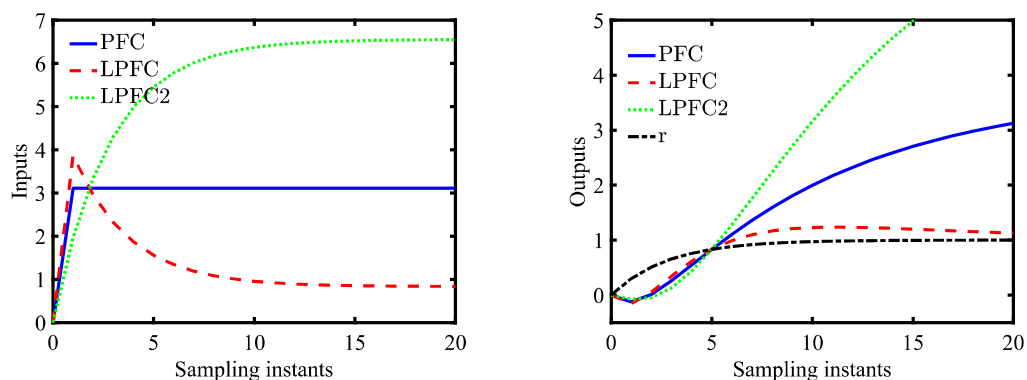


Figure 5.7: Input and output prediction of PFC, LPFC ,LPFC2 for process G_6 .

2. For the closed-loop response (see Figure 5.8), none of the algorithms is able to get close to the desired dynamic/target trajectory when $n \gg 1$. However, LPFC is marginally faster during the intermediate transients than PFC whereas LPFC2 has slow initial transients but ultimately converges to the steady-state slightly more quickly.

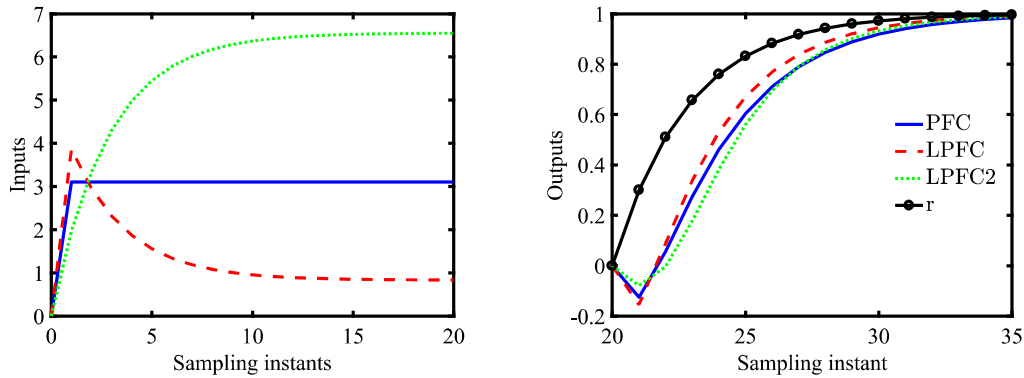


Figure 5.8: Closed-loop response of PFC, LPFC ,LPFC2 for process G_6 .

3. For constraint handling (refer Figure 5.9), as expected LPFC has by far the best performance because it exploits the input most effectively. By contrast, because PFC assumes a constant future input, the input values available become highly restricted to be close to the steady-state because otherwise, the long-range output predictions would exceed the upper output limit. LPFC2 has a slow initial transient, again because the shape of the input trajectory will only meet the upper constraints in the long term. However, in the medium term LPFC2 can exploit input values beyond u_{ss} and thus eventually converges in a timescale not dissimilar from LPFC.

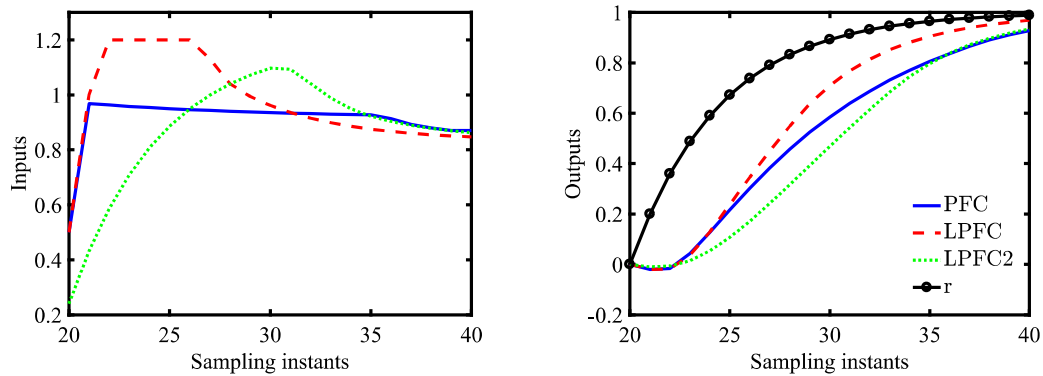


Figure 5.9: Constrained performance of of PFC, LPFC ,LPFC2 for process G_6 .

In summary, a comparison of the two alternative parameterisations indicates that in most cases, mapping Laguerre directly onto the inputs is preferable rather than the input increments, since this provides better prediction consistency and closed-loop performance. However, if necessary, a user may need to modify the implied terminal constraint which is similar to the reference governor approach in MPC to avoid recursive infeasibility.

5.5 LPFC FOR AN INTEGRATING PROCESS

Generally, when dealing with a system that has a marginally stable dynamics, the constant future input assumption of PFC will produce a divergent open-loop prediction (Richalet & O'Donovan, 2009; J. A. Rossiter & Haber, 2015; J. A. Rossiter, 2016). Consequently, it may lead to poor closed-loop performance, prediction inconsistency, and also a failure in constraint implementation. To deal with this types of plant, PFC practitioners often employ the Transparent Control as discussed before in Chapter 2, Section 2.2 (Richalet & O'Donovan, 2009). In practice, this structure is implementable within a constrained and uncertain environments. However, the controller still leads to an ill-posed decision making, which may impact its closed-loop performance (Abdullah & Rossiter, 2016) and constraints handling (Abdullah et al., 2017). Besides, the interaction between inner and outer loops also makes the tuning and constraint implementation less transparent and, within the literature, there is no clear or systematic guideline to deal with this issue.

The UKACC 2018 conference paper (Abdullah & Rossiter, 2018a) which is attached in Appendix E extends the use of LPFC for handling an integrating process. Since the underlying frame work of LPFC shapes the future input trajectory to converge to its steady state value, it may stabilise the open-loop prediction without the cascade structure. Besides, with a proper tuning of Laguerre pole a_l , a user may obtain a better closed-loop performance and prediction consistency to facilitate more reliable constraint management. In fact, the required modification to deploy this algorithm is straightforward and thus in line with the simplicity requirement of PFC.

5.5.1 LPFC Control Laws for Integrating Process

For an integrating process, since one of its poles will reside on the origin, the expected steady state input becomes zero for a constant set point. These dynamics are still compatible with the LPFC law with the only required modification being to define $u_{ss} = 0$. Hence, the future input dynamics of (5.5) becomes:

$$u(z) = \frac{\eta}{1 - a_l z^{-1}} \quad (5.26)$$

This dynamics will give input predictions that settle exponentially at zero with a speed linked to Laguerre pole a_l . Besides, the value η_k will affect the implied steady-state outputs since it has an affine dependence on the integral of the future input.

Algorithm 5.4. *For integrating process, a similar algorithm as in (5.9) is used except that u_{ss} term is removed.*

$$\eta_k = \frac{(1 - \lambda^{n_y})R + \lambda^{n_y}y_{p,k} - (P_{n_y}u_{\leftarrow k} + Q_{n_y}y_{m,k} + d_k)}{H_{n_y}L} \quad (5.27)$$

Similarly, to satisfy the implied constraints, this control law uses the similar procedure as in Algorithm 5.2 to solve for η_k . As noted in Remark 5.2, a user also may further improve the prediction consistency of LPFC with a proper tuning of Laguerre pole a_l by selecting a proper value that it less than λ . In brief, LPFC provides several benefits such as:

1. It offers a simple and systematic framework to handle an integrating process.

2. It stabilises the output prediction without a cascade structure thus the constraint handling procedure is more straight forward.
3. The Laguerre pole a_l can be utilised to control the speed of convergence to improve the prediction consistency and efficacy of constrained solution.
4. The implied structure of LPFC in conjunction with constraints means that a recursive feasibility guarantee for nominal case is provided (J. A. Rossiter et al., 2001).

5.5.2 Results

This paper considers a first order servo system G_7 (5.28) with an integrator as a plant to demonstrate the benefit of using LPFC compared to the cascade PFC (TPFC).

$$G_7 = \frac{0.0095z^{-2} + 0.0073z^{-1}}{1 - 1.45z^{-1} + 0.45z^{-2}} \quad (5.28)$$

The simulation will focus on the tuning process and the concept of well-posed decision in addition to the efficacy of constraint handling. The results demonstrates that:

1. In the unconstrained case, Figure 5.10 shows that although TPFC with different choices of gain K manages to track the trajectory set point with almost equivalent speed, the prediction consistency for both tunings are still poor. A similar scenario is observed when tuning LPFC with $a = \lambda$ where there is still noticeable inconsistency between predictions (blue dotted line) and the closed-loop behaviour (blue line). However, tuning the Laguerre pole with $a_l = 0.56$ improves the prediction consistency and the overall closed-loop performance.
2. Figure 5.11 demonstrates that when implementing the output constraint $\bar{y} = 0.8$, TPFC provides slower convergence (around 10 samples) and a more conservative constrained solution due to the prediction inconsistency. However, with a well tuned LPFC, the solution becomes more accurate (it converges around 5 samples) and less conservative in satisfying the limit.

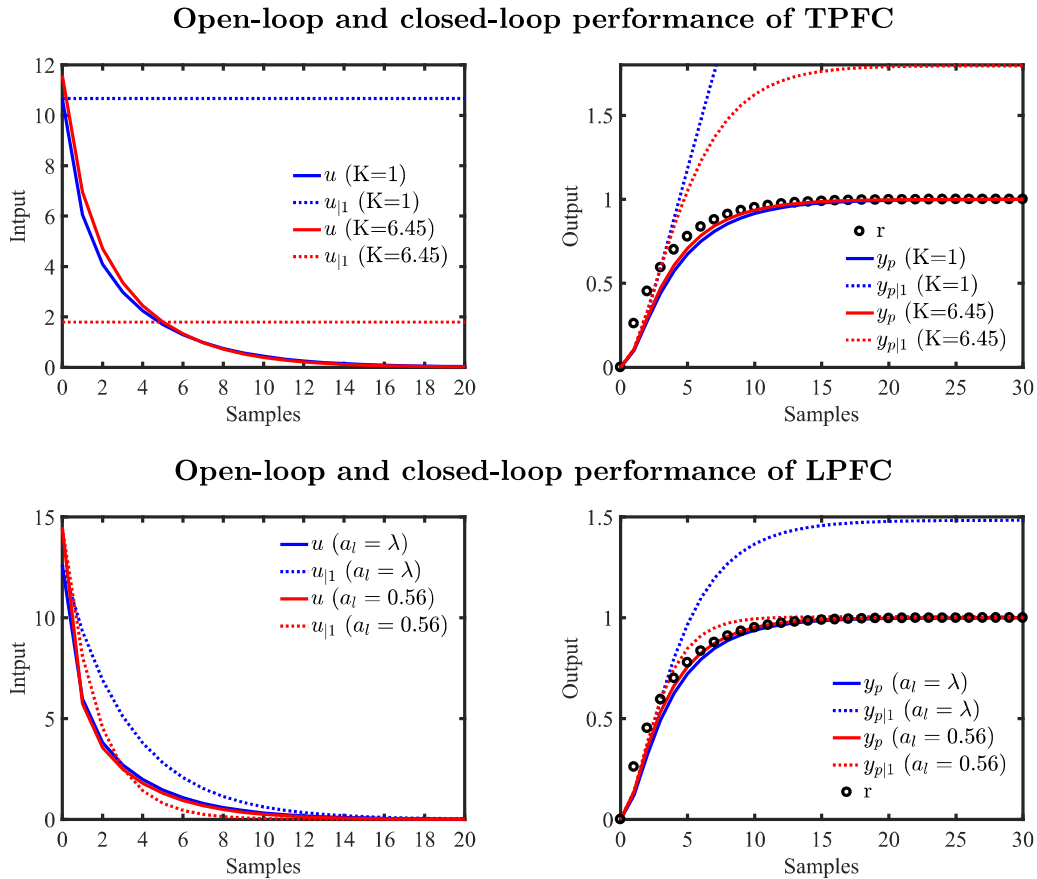


Figure 5.10: Open-loop and closed-loop performance of TPFC and LPFC with different tuning for process G_7 .

Nevertheless, there is a potential weakness with LPFC due to the use of an IM structure. A small offset error may occur if there is a plant-model mismatch or the real plant is not in fact integrating. In this case, when the input u_{ss} is set to zero, the correction in target trajectory cannot be made. The other limitation of LPFC is that its first order shaping is not able to handle an oscillatory or unstable process since its underlying dynamic is not capable enough to stabilise these undesirable poles. Hence, other alternative structures that can surpass this limitation are more desirable.

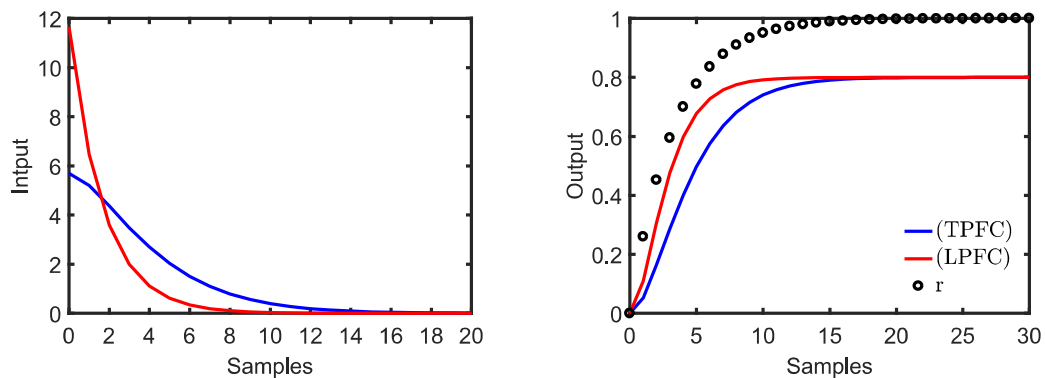


Figure 5.11: Constrained performance of TPFC and LPFC with bounded output $\bar{y} = 0.8$.

5.6 SUMMARY

This chapter has presented several contributions that are related to the development of Laguerre based PFC for improving prediction consistency, tuning efficacy and constraint handling while extending the usage to handle an integrating process and pointing out its potential weaknesses. The main highlights of this chapter are:

1. The UKACC 2016 conference paper (Abdullah & Rossiter, 2016) in Appendix B proves that LPFC can provide better prediction consistency and overall closed-loop performance compared to the traditional PFC.
2. The IFAC 2017 conference paper (Abdullah et al., 2017) in Appendix C proposes a more systematic constraint handling procedure using LPFC and demonstrates that it gives a more accurate and a less conservative constrained solution compared to the traditional multiple-regulators approach of PFC.
3. The IJC 2018 journal paper (Abdullah & Rossiter, 2019b) in Appendix D provides a performance comparison with several non-first order systems between PFC and different parameterisations of LPFC with inputs and input increments. The paper also points out that although LPFC provides the best performance compared to the other two approaches, a user needs to implement an extra terminal constraint to ensure recursive feasibility, which is common within the mainstream MPC practice.

4. The UKACC 2018 conference paper (Abdullah & Rossiter, 2018a) in Appendix E extends the use of Laguerre based PFC to handle an integrating process where the proposed controller provides better performance when compared to the traditional PFC cascade structure.

Nevertheless, these developments are only valid for simple and stable dynamics. Although LPFC is implementable with an integrating system, yet in uncertain cases such as in the presence of noise, disturbance and parameter uncertainty, it may provide poor closed-loop performance and a potential offset error. Hence, the next chapter will discuss the development of another alternative method for PFC that can specifically handle more challenging processes.

Chapter 6

POLE SHAPING PFC

It should be well noted now that the default PFC concept is more challenging to tune for a plant with an open-loop integrating, oscillatory or unstable prediction. The previous two techniques LPFC and PP-PFC can only handle one specific type of challenging dynamics process namely an integrating and oscillating, respectively. Hence, the second objective of this work is to propose a new systematic approach to PFC that can be generalised to handle these processes. This section provides a comprehensive summary of the Process 2018 journal paper (Abdullah & Rossiter, 2018d) which is attached in Appendix F, that develops an alternative PFC structures known as Pole Shaping PFC (PS-PFC). Section 6.1 begins with a brief discussion on the inspired concept of Pole Cancellation PFC developed by J. A. Rossiter (2016) and Section 6.2 presents the author's contribution in extending its concept to develop the PS-PFC. Section 6.3 presents several numerical examples from the paper and the final Section 6.4 provides the overall summary for this chapter.

6.1 POLE CANCELLATION PFC (PC-PFC)

Pole Cancellation PFC (PC-PFC) is another alternative method for handling different types of challenging dynamics systems that was originally used in MPC (J. A. Rossiter, 2018). Since the constant input assumption of typical PFC does not provide enough flexibility to control those challenging dynamics systems (J. A. Rossiter & Haber, 2015; Rawlings & Muske, 1993; Mosca & Zhang, 1992), it is crucial to first stabilise the prediction before implementing the nominal control law (J. Rossiter & Kouvaritakis, 1994). Hence, the first

step is to factorise the poles in the denominator:

$$\Delta y(z) = \frac{b(z)}{a(z)} \Delta u(z); \quad a(z) = a^-(z)a^+(z) \quad (6.1)$$

where $a^+(z)$ contains the undesirable poles. Utilising the Toeplitz/Hankel form (J. A. Rossiter, 2018), the future output predictions can be computed as:

$$[C_{a-\Delta}][C_{a^+}]y_{\rightarrow k+1} + H_A y_{\leftarrow k} = C_b \Delta u_{\rightarrow k} + H_b \Delta u_{\leftarrow k} \quad (6.2)$$

Rearranging prediction (6.2) in more compact form produces:

$$y_{\rightarrow k+1} = [C_{a-\Delta}]^{-1}[C_{a^+}]^{-1}[C_b \Delta u_{\rightarrow k} + \underbrace{H_b \Delta u_{\leftarrow k} - H_A y_{\leftarrow k}}_{\mathbf{p}}] \quad (6.3)$$

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

Lemma 6.1. *Next, the future input sequence $\Delta u_{\rightarrow k}$ is selected at each sample, such that the following equality is satisfied:*

$$[C_b \Delta u_{\rightarrow k} + \mathbf{p}] = C_{a^+} \gamma \quad (6.4)$$

where γ is a convergent sequence or a polynomial, will ensure that the corresponding output predictions in (6.3) do not contain the undesirable modes in a^+ .

Proof. This is self evident by substitution of (6.4) into (6.3) which gives:

$$y_{\rightarrow k+1} = [C_{a-\Delta}]^{-1}[C_{a^+}]^{-1}[C_{a^+}]\gamma = [C_{a-\Delta}]^{-1}\gamma \quad (6.5)$$

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

Remark 6.1. *Requirement (6.4) can be solved by a small number of simultaneous equations (J. A. Rossiter, 2018) where the minimal order solution can be represented as:*

$$\Delta u_{\rightarrow k} = P_1 \mathbf{p}; \quad \gamma = P_2 \mathbf{p} \quad (6.6)$$

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

To ensure the future manipulated control moves are convergent while adding some flexibility for modifying the output predictions, the input requirement in (6.6) can be enhanced to:

$$\Delta \underline{u}_{\rightarrow k} = P_1 \mathbf{p} + C_{a^+} \phi \quad (6.7)$$

where the vector parameter ϕ denotes the d.o.f. within the predictions.

Theorem 6.1. *Using the new shaped input (6.7) ensures that the undesirable modes do not appear in the output predictions, irrespective of the choice of ϕ . The output predictions are convergent if ϕ is finite dimensional or a convergent infinite dimensional sequence.*

Proof. Substitute input prediction (6.7) into output prediction (6.4), the predictions become:

$$\begin{aligned} \underline{y}_{\rightarrow k+1} &= [C_{a^-} \Delta]^{-1} [C_{a^+}]^{-1} [C_b \Delta \underline{u}_{\rightarrow k} + \mathbf{p}] \\ &= [C_{a^-} \Delta]^{-1} [C_{a^+}]^{-1} [C_{a^+} \gamma + C_b C_{a^+} \phi] \\ &= [C_{a^-} \Delta]^{-1} [C_b \phi + \gamma] \end{aligned} \quad (6.8)$$

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

$$\underline{y}_{\rightarrow}(z) = \frac{[1, z^{-1}, z^{-2}, \dots][\gamma + C_b \phi]}{a^-(z)} = \frac{\gamma(z) + b(z)\phi(z)}{a^-(z)} \quad (6.9)$$

It is known from Lemma 6.1 that the contribution from γ gives a convergent prediction and thus overall convergence is obvious as long as $\phi(z)$ is convergent (or an FIR). \square

Remark 6.2. *Noting the definition of \mathbf{p} in (6.3), the n_y -step ahead output prediction with prediction class (6.7,6.9) can be put in more common form as:*

$$y_{k+n_y|k} = H_{s,n_y} \phi + P_{s,n_y} \Delta_{\leftarrow} u_k + Q_{s,n_y} \underline{y}_k \quad (6.10)$$

where H_{s,n_y} , P_{s,n_y} , and Q_{s,n_y} are suitable matrices and the additional subscript ‘s’ is used to denote shaping and ϕ is taken to be FIR (equivalently a finite dimensional vector). Note however that typically for PFC ϕ is a scalar. Also, it is easy to show (J. A. Rossiter, 2002) that choosing $\phi = 0$ will automatically give the same input predictions as deployed at the previous sample which enables consistency of predictions from one sample to the next.

Algorithm 6.1. [PC-PFC] *After selecting a suitable coincidence horizon n_y and desired closed-loop pole λ . The d.o.f ϕ is computed by substituting prediction (6.10) of PC-PFC*

into equality (2.2) and thus:

$$\phi = \frac{1}{h_{n,s}} \left[(1 - \lambda^{n_y})r + \lambda^{n_y}y_k - Q_{n_y,s}y_k - P_{n_y,s}\Delta u_k \right] \quad (6.11)$$

then, the current input increment Δu_k is determined simply by inserting ϕ into the predicted input of (6.7).

Remark 6.3. *It has been shown in the work of J. A. Rossiter (2016) that PC-PFC can improve the significance of λ as the tuning parameter when handling unstable and oscillating processes. Besides, the implied modification also provides a recursive feasibility guarantee when satisfying the limits. Nevertheless, the manipulated input dynamics is not always implementable or desirable for some cases, since cancelling the specified poles in just a few control moves often lead to an over-aggressive input trajectory (J. A. Rossiter, 2016, 2002; Mosca & Zhang, 1992).*

6.2 THE CONCEPT OF POLE SHAPING PFC (PS-PFC)

Noting the benefits and weaknesses of the Pole Cancellation PFC (PC-PFC) as discussed in Remark 6.3, this work develops an alternative shaping method known as Pole Shaping PFC (PS-PFC) to improve its reliability. Cancelling the undesired pole directly will lead to several issues such as sensitivity to uncertainty or worst instability (J. A. Rossiter, 2018). This scenario may happen because of the controller is highly depending on the accuracy of the represented model. Hence, rather than cancelling the undesired poles directly, PS-PFC changes these poles to be inside the unit circle so that it can stabilise the open-loop prediction while allowing the output modes to evolve over many more samples (thus, the term "shaping" is used to name this controller). With this modification, the required control input will be less aggressive than the previous method.

6.2.1 PS-PFC Control Law

The formulation given in this section is directly linked to the previous PC-PFC control law. It is known that dead-beat pole cancellation can require aggressive inputs and the minimal order solutions to (6.4) are in effect dead-beat input predictions (Rawlings & Muske,

1993; J. A. Rossiter, 2016). Although dead-beat solutions are easy to define and thus have some advantages in terms of computation and transparency, in practice, a user may desire a less aggressive shaping that is more implementable in a real system. Alongside this, the popularity of dual mode approaches in the literature (Scokaert & Rawlings, 1998) is partially because they allow the implied input predictions to converge to the steady-state asymptotically rather than in a small finite number of steps. Thus, the proposal here is to introduce some poles say $\alpha(z)$ in the implied solutions for $\gamma(z)$, $\phi(z)$ used in (6.9) to obtain a smoother solution of (6.4).

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

Proposal 6.1. *By definition, the integrator has a pole on the unit circle, that is factor $(1 - z^{-1})$, and conversely, cancelling the pole as in (6.4) is equivalent to enforcing a pole on the origin, that is factor $(1 - 0z^{-1})$. Hence the choice of pole factor $\alpha = (1 - 0.5z^{-1})$ represents a simple half-way house trade off between these two choices.*

Proposal 6.2. *For a process with significant under-damping, the implied $\alpha(z)$ will have only real poles which are chosen to be close to the real parts of the oscillatory poles. This will reduce the undesirable oscillation in the output predictions, but not change the convergence speed, albeit the input may then be somewhat oscillatory.*

Proposal 6.3. *For open-loop unstable systems, a simple default solution simply inverts the unstable poles, that is, defining $\alpha(z)$ such that $a^+(z_i) = 0 \Rightarrow \alpha(1/z_i) = 0$.*

Remark 6.4. *A reader should also be noted that the concept of PS-PFC is different from the traditional pole placement technique (Aström & Murray, 2010) and PP-PFC which was discussed in Chapter 4. The PS-PFC concept replaces the undesirable poles by placing new poles to obtain a stable open-loop prediction rather than getting a desired closed-loop performance. For this case, the standard PFC regulator will enforce the stabilised open-loop response to follow the first order target trajectory as discussed before in Chapter 2, Section 2.1.3.*

Lemma 6.2. *The dynamics $\alpha(z)$ will be present in the predictions if the following Diophantine equation is used to solve the input/output prediction pairing.*

$$\begin{aligned} b(z)w(z) + \alpha(z)p(z) &= a^+(z)\hat{\gamma}(z); \quad p(z) = [1, z^{-1}, \dots]\mathbf{p} \\ \Rightarrow \quad \Delta_{\rightarrow}u(z) &= \frac{w(z)}{\alpha(z)}, \quad \underline{y}(z) = \frac{\hat{\gamma}(z)}{a^-(z)\Delta(z)\alpha(z)} \end{aligned} \quad (6.12)$$

Proof. First note that (6.12) is equivalent to solving:

$$[C_b C_\alpha^{-1} \mathbf{w} + \mathbf{p}] = C_{a^+} C_\alpha^{-1} \hat{\gamma} \quad (6.13)$$

and moreover eqn.(6.13) follows directly from enforcing (6.4) while assuming $\Delta_{\rightarrow k}u = C_\alpha^{-1} \mathbf{w}$. Hence, substituting this $\Delta_{\rightarrow k}u$ into (6.3) gives:

$$\begin{aligned} \underline{y}_{\rightarrow k+1} &= [C_{a^- \Delta}]^{-1} [C_{a^+}]^{-1} [C_b \Delta_{\rightarrow k}u + \mathbf{p}] \\ &= [C_{a^- \Delta}]^{-1} [C_{a^+}]^{-1} [C_b C_\alpha^{-1} \mathbf{w} + \mathbf{p}] \\ &= [C_{a^- \Delta}]^{-1} [C_{a^+}]^{-1} [C_{a^+} C_\alpha^{-1} \hat{\gamma}] = C_{a^- \Delta}^{-1} C_\alpha^{-1} \hat{\gamma} \end{aligned} \quad (6.14)$$

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

Remark 6.5. *The new requirement (6.13) can be solved similarly to (6.4) where the minimal order solution for \mathbf{w} and $\hat{\gamma}$ are:*

$$\mathbf{w} = \hat{P}_1 \mathbf{p}; \quad \hat{\gamma} = \hat{P}_2 \mathbf{p} \quad (6.15)$$

Theorem 6.2. *A convergent prediction class which embeds both the desired asymptotic poles and some degrees of freedom (d.o.f.) can be defined from:*

$$\mathbf{w} = \hat{P}_1 \mathbf{p} + C_{a^+} \phi; \quad \Delta_{\rightarrow k}u = [C_\alpha]^{-1} [\hat{P}_1 \mathbf{p} + C_{a^+} \phi] \quad (6.16)$$

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

Proof. Based on superposition, the additional component in \mathbf{w} , that is $C_{a^+} \phi$, necessarily cancels the undesirable poles and gives overall convergent output predictions. So using (6.14), then:

$$\begin{aligned} \underline{y}_{\rightarrow k+1} &= C_{a^- \Delta}^{-1} C_{a^+}^{-1} [C_b \Delta_{\rightarrow k}u + \mathbf{p}] \\ &= C_{a^- \Delta}^{-1} C_\alpha^{-1} \hat{\gamma} + C_{a^- \Delta}^{-1} C_{a^+}^{-1} C_\alpha^{-1} [C_{a^+} C_b \phi] \\ &= C_{a^- \Delta}^{-1} C_\alpha^{-1} [\hat{\gamma} + C_b \phi] \end{aligned} \quad (6.17)$$

□

Remark 6.6. By extracting the n_y^{th} row and noting the definition of \mathbf{p} in (6.3), the n_y step ahead prediction from (6.17) can be rearranged in a more general form as:

$$y_{k+n_y|k} = h_{n_y,\alpha}\phi + P_{n_y,\alpha}\Delta_{\leftarrow}u_k + Q_{n_y,\alpha}y_{\leftarrow}k \quad (6.18)$$

for suitable $h_{n_y,\alpha}, P_{n_y,\alpha}, Q_{n_y,\alpha}$ and it is noted that as is conventional for PFC, ϕ has just a single non-zero parameter in order to retain computational simplicity and have just a single d.o.f. for satisfying the control law (2.2).

Algorithm 6.2. [PS-PFC] After selecting a suitable n_y and λ , the d.o.f ϕ is computed by substituting prediction (6.18) of PS-PFC into equality (2.2) and thus:

$$\phi = \frac{1}{h_{n_y,\alpha}} \left[(1 - \lambda^{n_y})r + \lambda^{n_y}y_k - Q_{n_y,\alpha}y_{\leftarrow}k - P_{n_y,\alpha}\Delta_{\leftarrow}u_k \right] \quad (6.19)$$

then, the current input increment Δu_k is determined simply by inserting ϕ into the predicted input of (6.16).

6.2.2 Constraint Handling of PS-PFC with Recursive Feasibility

Noting the definition of future input increments in (6.16) and output predictions in (6.18), the constraints inequalities for (5.12) can be defined as:

$$\begin{aligned} L\underline{\Delta u} &\leq C_\alpha^{-1}[\hat{P}_1\mathbf{p} + C_{a^+}\phi] \leq L\overline{\Delta u}; \\ L\underline{u} &\leq C_{I/\Delta}C_\alpha^{-1}[\hat{P}_1\mathbf{p} + C_{a^+}\phi] + Lu_{k-1} \leq L\bar{u} \\ L\underline{y} &\leq H_\alpha\phi + P_\alpha\Delta_{\leftarrow}u_k + Q_\alpha y_{\leftarrow}k \leq L\bar{y} \end{aligned} \quad (6.20)$$

where $C_{I/\Delta}$ is a lower triangular matrix one ones and L is a vector of ones with appropriate dimension (typically a horizon long enough to capture the core dynamics in the predictions). It should be noted that the validation horizon n_i for the predictions used in (6.20) will in general be much longer than the coincidence horizon used in (6.19) as one needs to ensure that the implied long range predictions satisfy constraints. The inequalities can be combined for convenience as follows (although this is not necessary for on-line coding where efficient alternatives may exist):

$$C\phi \leq \mathbf{f}_k \quad (6.21)$$

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

Algorithm 6.3. [PS-PFC with constraint handling] At each sample:

1. Define the unconstrained value for ϕ from (6.19).
2. Update the vector \mathbf{f}_k of (6.21) (it is noted that C does not change).
3. Use a simple loop covering all the rows of C as follows:
 - (a) Check satisfaction of the i th constraint using: $\mathbf{e}_i^T C \phi \leq f_{k,i}$.
 - (b) If $\mathbf{e}_i^T C \phi > f_{k,i}$, then set $\phi = (f_{k,i}) / [\mathbf{e}_i^T C]$, else leave ϕ unchanged.

Theorem 6.3. In the presence of constraints, **Algorithm 6.3** is recursively feasible where the computed ϕ will not only enforce the input/output predictions to satisfy constraints at the current sample but also guarantees that one can make the same statement at the next sample.

Proof. By definition, the choice of $\phi = 0$ ensures feasibility in the nominal case because the input component $\hat{P}_1 \mathbf{p}$ is the unused part of the input prediction from the previous sample and this is known to satisfy constraints by assumption. One can ensure feasibility at start-up by beginning from a sensible state. \square

It is worth noting that using the pre-stabilised/shaped predictions is essential for this recursive feasibility result which is not available for more conventional PFC approaches for which the implied long range predictions may be divergent. Thus, **Theorem 6.3** is an important contribution of this work.

Remark 6.7. *It is noted that recursive feasibility is an essential property to implies stability (Sokaert et al., 1999). However, recursive feasibility alone is not sufficient to guarantee stability; the Lyapunov theorem needs to be respected which is only possible if the quadratic cost function is used as in MPC. Although **Algorithm 6.3** allows recursive feasibility which is a strong result, ironically the use of PC-PFC or PS-PFC does not give any a priori stability and/or performance guarantees in general which is a well understood weakness of PFC approaches (J. A. Rossiter, 2017) and a consequence of wanting a very simple and cheap control approach.*

6.3 RESULTS

Several numerical examples are presented in (Abdullah & Rossiter, 2018d) to compare the performance between PS-PFC, PC-PFC and conventional PFC ranging form:

1. Integrating process:

$$G_8 = \frac{0.1z^{-1} + 0.4z^{-2}}{(1 - 0.8z^{-1})(1 - z^{-1})} \quad (6.22)$$

2. Oscillatory process:

$$G_9 = \frac{0.85z^{-1} - 1.5z^{-2} + 0.85z^{-2}}{(1 - 0.6z^{-1})(1 - 1.6z^{-1} + 0.8z^{-2})} \quad (6.23)$$

3. Unstable process:

$$G_{10} = \frac{0.2z^{-1} - 0.26z^{-2}}{(1 - 0.9z^{-1})(1 - 1.5z^{-1})} \quad (6.24)$$

The results shows that:

1. For open-loop behaviour in Figure 6.1, PS-PFC produces the best prediction behaviour because it ensures convergent predictions with less aggressive input activity than given by PC-PFC and PFC.
2. For closed-loop performance in Figure 6.2, PS-PFC (using a default choice of α) gives the best trade-off between the rate of convergence and the aggressiveness of input activity compared to PFC and PC-PFC.

3. For constraint handling in Figure 6.3, both PS-PFC and PC-PFC satisfy constraints while retaining recursive feasibility throughout and converge safely. However, PS-PFC provides a smoother input transition and better constrained performance.

In essence, the proposed PS-PFC algorithm gives a pragmatic and simple proposal for deriving input and output prediction pairs which do not require aggressive inputs during transients compared to the more classical alternative approach of PC-PFC. Besides, using the proposed parameterisations also allows a simple proof of recursive feasibility so that the constraint handling can be performed more safely and reliably. Other than these numerical examples, this paper also has validated the practicality of PS-PFC controller by successfully implementing it in laboratory hardware, that is the Quanser SVR02 servo based unit.

6.4 SUMMARY

This chapter has presented a summary of the Process 2018 journal paper (Abdullah & Rossiter, 2018d), which introduces a novel input shaping method known as Pole Shaping PFC to handle different types of challenging dynamics system with far less aggressive input demand compared to the traditional Pole Cancellation PFC. Besides, PS-PFC also provides a more general approach to handle different types of challenging dynamics systems compared to the other two previously developed methods of PP-PFC and LPFC with the additional advantage of a guaranteed recursive feasible constrained solution.

Nevertheless, since PP-PFC utilises the Realigned Model (RM) structure, the control law may become more sensitive to measurement noise and parameter uncertainty (J. A. Rossiter, 2018). This limitation is because of its underlying algorithm becoming too dependent on the accuracy of the model representing the plant. Although, theoretically it can provide an offset-free correction in the presence of disturbance and model-mismatch, yet the RM structure is still not able to withstand high-frequency measurement noise especially with a higher-order process. Hence, the next chapter will propose the use of T-filter in PS-PFC to tackle this limitation together with a formal sensitivity analysis for different types of PFC structures to measure their robustness.

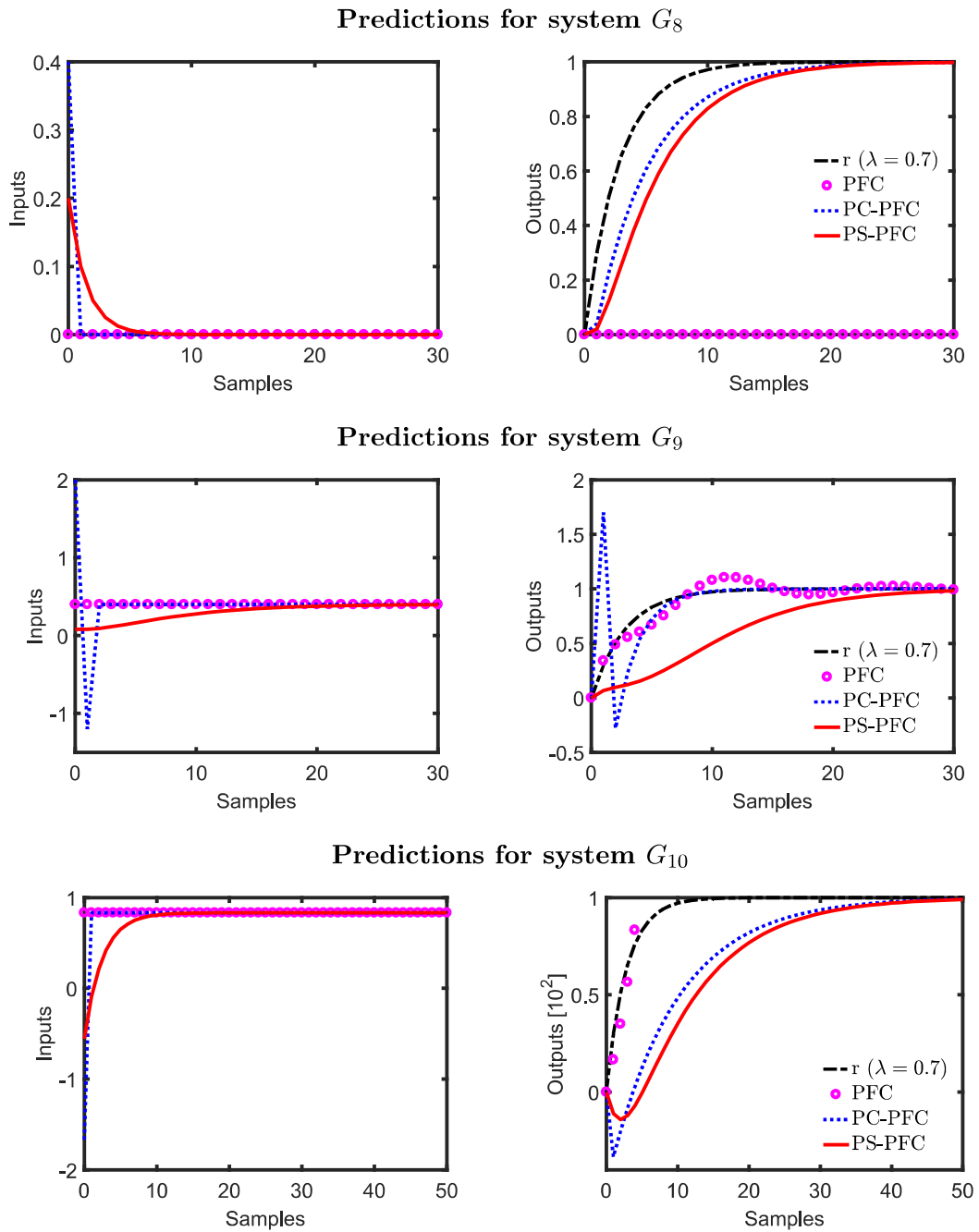


Figure 6.1: Input and output predictions with PFC, PC-PFC and PS-PFC for processes G_8 , G_9 , G_{10} .

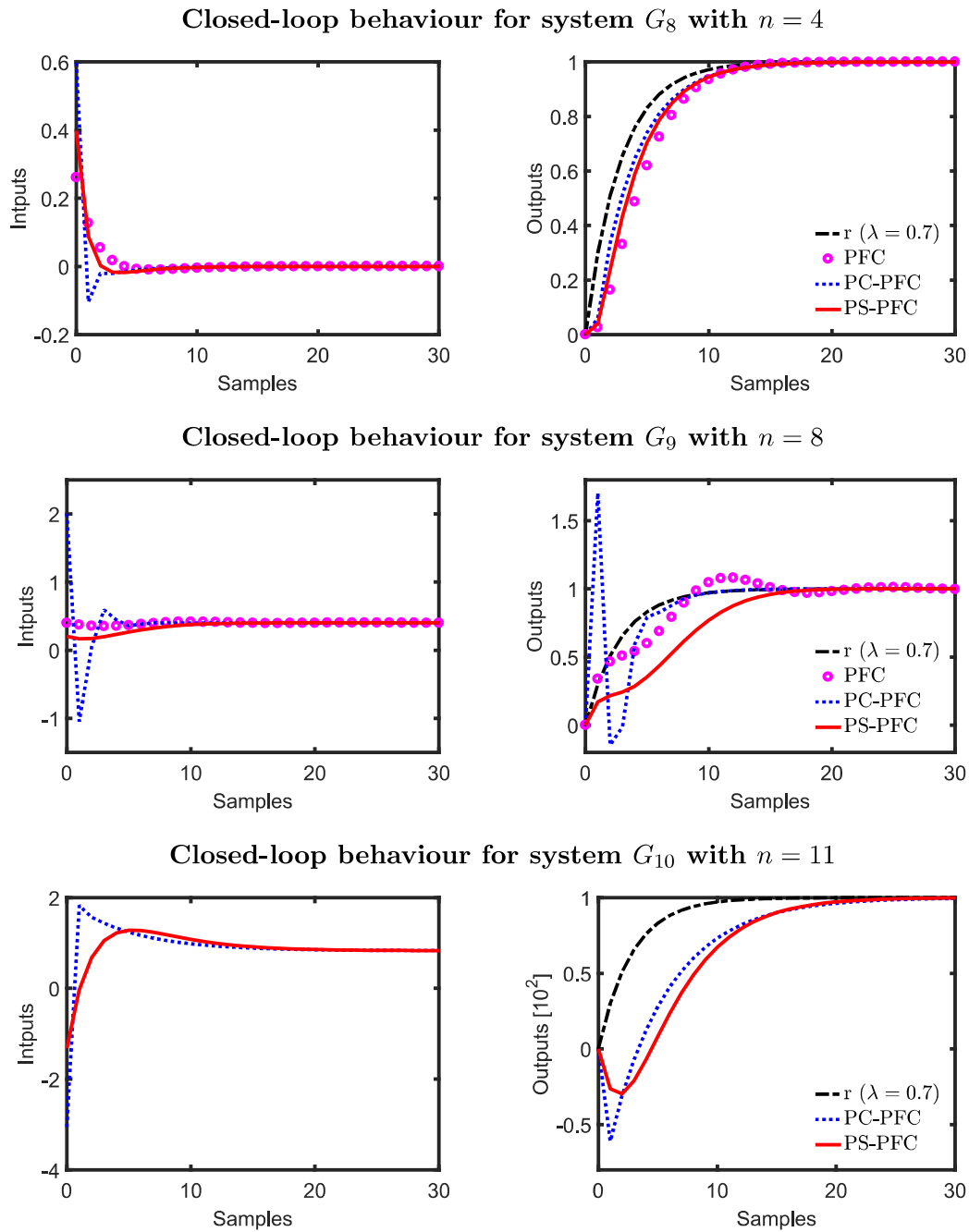
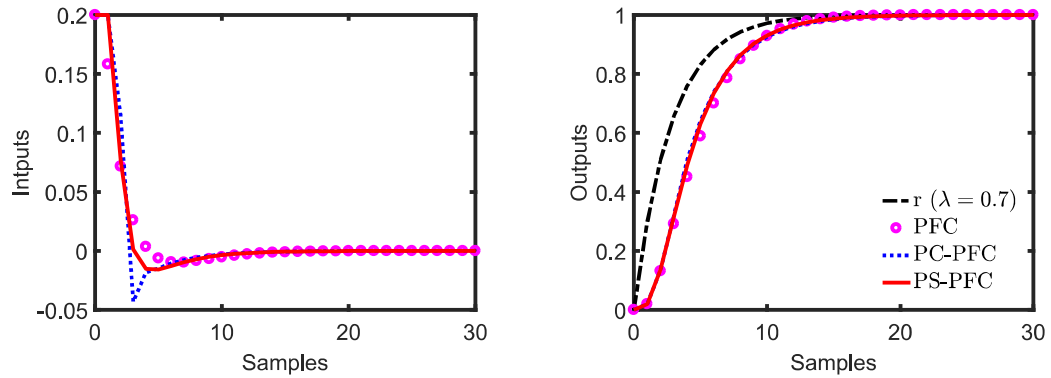


Figure 6.2: Closed-loop input and output behaviour of PC-PFC and PS-PFC for processes G_8, G_9, G_{10}

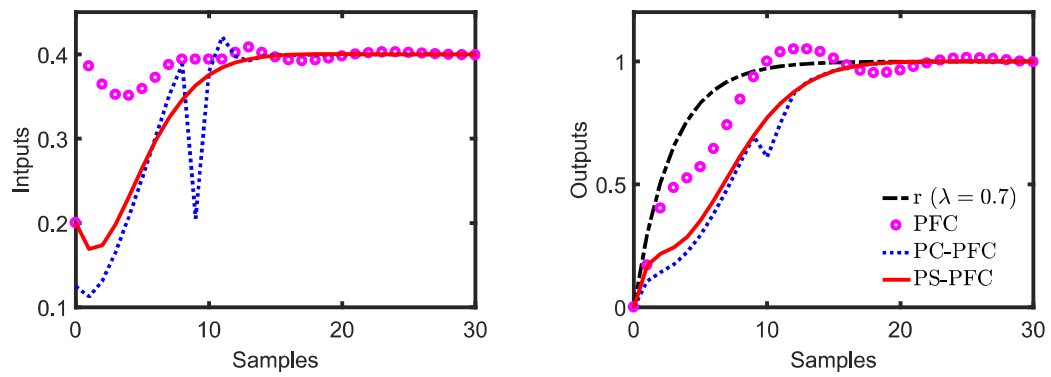
Closed-loop constrained behaviour for system G_8 with $n = 4$ and

$$-0.2 \leq u_k \leq 0.2; -0.2 \leq \Delta u_k \leq 0.2; -1.2 \leq y_k \leq 1.2$$



Closed-loop constrained behaviour for system G_9 with $n = 8$ and

$$-1 \leq u_k \leq 1; -0.2 \leq \Delta u_k \leq 0.2; -1.05 \leq y_k \leq 1.1$$



Closed-loop constrained behaviour for system G_{10} with $n = 11$ and

$$-1 \leq u_k \leq 1; -0.2 \leq \Delta u_k \leq 0.2; -0.2 \leq y_k \leq 1.2$$

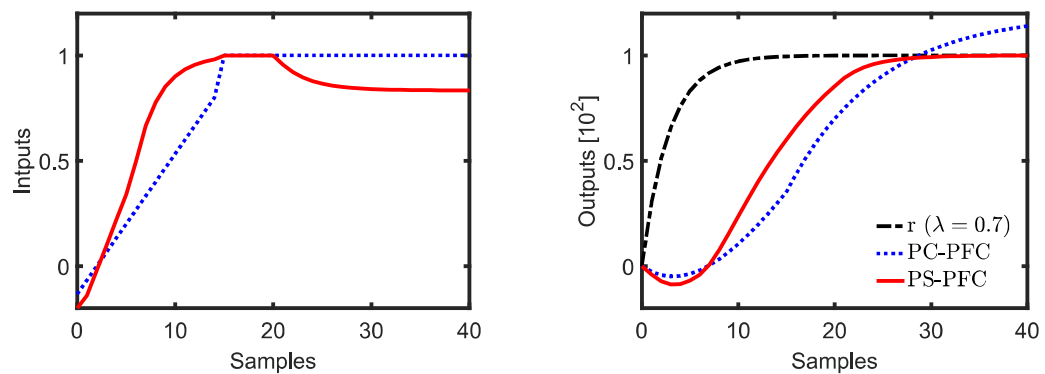


Figure 6.3: Constrained input and output behaviour of PFC, PC-PFC and PS-PFC for processes G_8, G_9, G_{10} .

Chapter 7

SENSITIVITY ANALYSIS

This chapter presents a comprehensive summary from three different papers (Abdullah & Rossiter, 2018b,c, 2019a) that focus on the sensitivity analysis for varying PFC structures. It is noted that the previous two chapters have developed the Laguerre PFC to control stable and straightforward dynamics processes while Pole Shaping PFC for challenging dynamics systems. Hence, the third objective of this work is to analyse and compare the sensitivity trade-off of these structures to measurement noise, disturbances and parameter uncertainty. Before analysing those control laws, Section 7.1 provides the initial analysis between the IM and RM structure of PFC and proposes the use of a T-filter to improve the robustness of a RM structure. Section 7.2 continues to analyse the sensitivity trade-off when using LPFC compared to traditional PFC. Since the underlying concept of PS-PFC utilising a RM structure is quite sensitive to high-frequency measurement noise, Section 7.3 proposes the use of a T-filter to handle this issue while providing its formal sensitivity analysis. The final section 7.4 provides the summary for this chapter.

7.1 SENSITIVITY ANALYSIS OF DIFFERENT PFC STRUCTURES

As discussed before in Chapter 2, PFC can utilise two main prediction structures which are the Independent Model (IM) and Realigned Model (RM). Between these two options, IM has become a standard structure for PFC to handle uncertainties (Richalet & O'Donovan, 2009). Although the implementation of IM is straightforward, its usage is only limited to a stable process, where for unstable dynamics, PFC practitioners often imply the cascade structure to retain its stability. Conversely, the use of a RM structure is more general which can be implemented to different types of dynamical processes, yet this structure may

become highly sensitive to measurement noise due to its dependency on the plant output measurements. Nevertheless, a user also should consider the use of a T-filter to surpass the limitation of a RM structure, which has become a conventional practice in the GPC framework (Yoon & Clarke, 1995; J. A. Rossiter, 2018).

Hence, the main contribution of this work is to propose the use of a T-filter within a PFC framework while analysing the sensitivity trade-off between three prediction models namely the RM, RM with T-filter and IM structures when handling measurement noise and disturbances. The associated sensitivity functions are derived and validated via both closed-loop simulation and real-time implementation. Since the sensitivity relationships are system dependent, performing this off-line analysis is beneficial to measure the robustness of a controller. The detailed formulation, derivation and discussion of this work are published in the ECC 2018 conference paper (Abdullah & Rossiter, 2018b), which is attached in Appendix G.

7.1.1 PFC with RM Structure

Without loss of generality and for clarity of presentation, this work assumes an underlying CARIMA model to represent the prediction of RM, RM with T-filter and IM structures. A similar representation as in the previous chapters is used except that now the model is in a difference form, where:

$$y_{k+n_y|k} = H\Delta_{\rightarrow}u_k + P\Delta_{\leftarrow}u_k + Qy_k \quad (7.1)$$

This structure provides an offset free correction due to the presence of an integrator, where the current and past output is measured directly from a plant. Substituting prediction (7.1) into equality (2.2) gives the control law of:

$$\Delta u_k = \frac{1}{H_{n_y,1}} \left[(1 - \lambda^{n_y})r + \lambda^{n_y}y_k - Q_{n_y}y_k - P_{n_y}\Delta_{\leftarrow}u_k \right] \quad (7.2)$$

where the constant future input assumption of PFC means $\Delta u_{k+i} = 0$ for $i > 0$, hence only the first column ($H_{n_y,1}$) of matrix H_{n_y} is used for this case. The control law can be represented in a vector form by rearranging (7.2) in terms of parameters F , N and \hat{D} with obvious definitions:

$$\Delta u_k = Fr - Ny_k - \hat{D}\Delta_{\leftarrow}u_k \quad (7.3)$$

Although the formulation in (7.3) can be implemented directly, it is easier to utilise a transfer function form for analysing its sensitivity (J. A. Rossiter, 2018). The vectors of

$$\begin{aligned} N &= [N_0, N_1, N_2, \dots, N_n] \\ \hat{D} &= [\hat{D}_0, \hat{D}_1, \hat{D}_2, \dots, \hat{D}_n] \end{aligned} \quad (7.4)$$

are defined in the z domain as:

$$\begin{aligned} N(z) &= N_0 + N_1z^{-1} + N_2z^{-2} + \dots + N_nz^{-n} \\ \hat{D}(z) &= \hat{D}_0 + \hat{D}_1z^{-1} + \hat{D}_2z^{-2} + \dots + \hat{D}_nz^{-n} \\ D(z) &= 1 + z^{-1}\hat{D}(z) \end{aligned} \quad (7.5)$$

Noting the definitions of Δu_k and y_k , the sensitivity functions can be derived in a fixed closed loop form of:

$$D(z)\Delta u_k = F(z)r - N(z)y_k \quad (7.6)$$

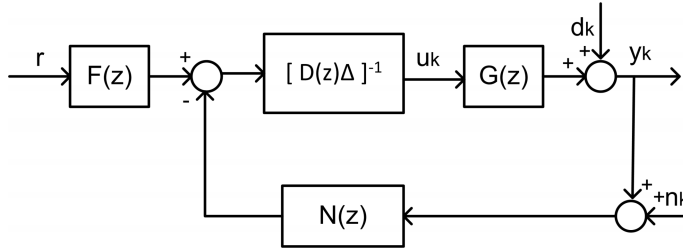


Figure 7.1: PFC with RM structure equivalent block diagram.

Figure 7.1 indicates the equivalent block diagram (unconstrained case) and adds measurement noise n and output disturbance d . From the structure, the effective control law can be simplified to $K(z) = N_c(z)[D_c(z)\Delta]^{-1}$. Assuming system $G(z) = b(z)a(z)^{-1}$, the closed-loop pole polynomial $P_c(z) = 1 + K(z)G(z)$ is represented as:

$$P_c(z) = D(z)a(z)\Delta + N(z)b(z) \quad (7.7)$$

The sensitivity of the input to noise is derived by finding the transference from u to n (refer to Figure 7.1):

$$\begin{aligned} S_{un} &= K(z)[1 + K(z)G(z)]^{-1} \\ &= N(z)P_c(z)^{-1}a(z) \end{aligned} \quad (7.8)$$

Similarly, the sensitivity of output to disturbances is obtained by solving the transference from d to y :

$$\begin{aligned} S_{yd} &= [1 + K(z)G(z)]^{-1} \\ &= a(z)P_c(z)^{-1}D(z)\Delta \end{aligned} \quad (7.9)$$

7.1.2 PFC with T-filter

Other than the IM structure, T-filter is often considered as an essential tool in the practical implementation of predictive controller that uses a transfer function model (Clarke et al., 1987; J. A. Rossiter, 2018). It acts as a low pass filter to eliminate high frequency measurement noise from a plant without affecting the nominal tracking performance of predictive control (Yoon & Clarke, 1995). The framework proposed here is a two stage design whereby PFC is first tuned for performance tracking, then the T-filter is employed to improve the sensitivity. Conceptually, the measurement output is low-pass filtered before prediction and anti-filtered after prediction to restore the predicted data back to the correct domain before deploying the nominal algorithm. The schematic structure is shown in Figure 7.2 where it reduces the impact of high frequency noise on the prediction while retaining the valuable low frequency dynamics.

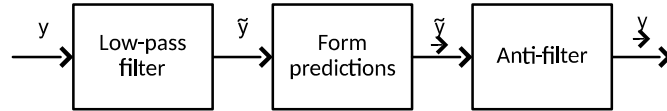


Figure 7.2: PFC prediction structure with T-filter.

The desired T-filter T^{-1} is deployed as $\tilde{y}_k = y_k T^{-1}$ or $T\tilde{y}_k = y_k$. Define the filtered predictions up to horizon n_y as follows:

$$\tilde{y}_{\rightarrow k+1} = H\Delta_{\rightarrow} \tilde{u}_k + P\Delta_{\leftarrow} \tilde{u}_k + Q\tilde{y}_k \quad (7.10)$$

The relationship between the filtered and unfiltered predicted data can be represented using Toeplitz/Hankel form (J. A. Rossiter, 2018):

$$\begin{aligned} y_{\rightarrow k+1} &= C_T \tilde{y}_{\rightarrow k+1} + H_T \tilde{y}_{\leftarrow k} \\ \Delta_{\rightarrow} u_k &= C_T \Delta_{\rightarrow} \tilde{u}_k + H_T \Delta_{\leftarrow} \tilde{u}_k \end{aligned} \quad (7.11)$$

substituting (7.11) into (7.10) gives:

$$\underbrace{C_T^{-1}[y_{\rightarrow k+1} - H_T \tilde{y}_k]}_{\tilde{y}_k} = H \underbrace{C_T^{-1}[\Delta_{\rightarrow} u_k - H_T \Delta_{\leftarrow} \tilde{u}_k]}_{\Delta_{\rightarrow} \tilde{u}_k} + P \Delta_{\leftarrow} \tilde{u}_k + Q \tilde{y}_k \quad (7.12)$$

Multiplying through by C_T and grouping common terms:

$$\tilde{y}_k = H \Delta_{\rightarrow} u_k + \tilde{P} \Delta_{\leftarrow} \tilde{u}_k + \tilde{Q} \tilde{y}_k \quad (7.13)$$

where $\tilde{P} = [C_T P - H H_T]$ and $\tilde{Q} = [H_T + C_T Q]$. The difference between (7.13) and (7.3) are the last two terms which now are based on past filtered data. Hence, applying a similar control law, a PFC with T-filter can be formulated as:

$$D_t(z) \Delta u_k = F(z) r - N_t(z) y_k \quad (7.14)$$

where $D_t(z) = \frac{D(z)}{T(z)}$ and $N_t(z) = \frac{N(z)}{T(z)}$. From the equivalent block diagram representation in Figure 7.3, the closed-loop pole polynomial, sensitivity of the input to noise and sensitivity of the output to disturbances are constructed as:

$$\begin{aligned} P_t(z) &= D_t(z) a(z) \Delta + N_t(z) b(z) \\ S_{un} &= N_t(z) P_t(z)^{-1} a(z) \\ S_{yd} &= a(z) P_t(z)^{-1} D_t \Delta \end{aligned} \quad (7.15)$$

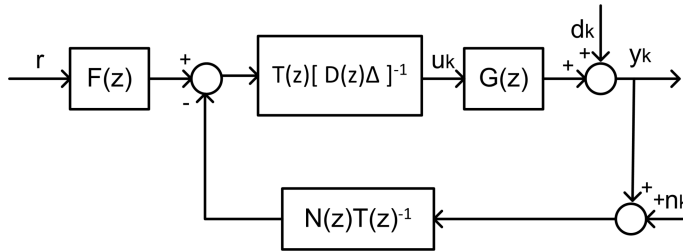


Figure 7.3: PFC with T-filter equivalent block diagram.

Remark 7.1. *It can be shown that the closed-loop poles of PFC with a T-filter $P_t(z)$ are related to the equivalent poles of PFC by $P_t(z) = P_c(z)T(z)$ and also that the inclusion of T-filter cannot affect the nominal tracking performance (J. A. Rossiter, 2018).*

7.1.3 PFC with IM structure

The prediction using an IM structure can also be represented using a CARIMA model. Similar as before, the only difference is that now the parameter is in the difference form of:

$$y_{p,k+n_y|k} = H\Delta_{\rightarrow}u_k + P\Delta_{\leftarrow}u_k + Qy_{\leftarrow m,k} + (y_{p,k} - y_{m,k}) \quad (7.16)$$

The new equality between prediction (7.16) with the target trajectory (2.2) at n_y step ahead is given as:

$$H_{n_y}\Delta_{\rightarrow}u_k + P_{n_y}\Delta_{\leftarrow}u_k + Q_{n_y}y_{\leftarrow m,k} - y_{m,k} = (1 - \lambda^{n_y})(R - y_{p,k}) \quad (7.17)$$

Since the future input increment Δu_{k+i} is assumed zero for $i > 0$ and H_{n_y} is reduced to $H_{n_y,1}$, the PFC control law is:

$$\Delta u_k = \frac{1}{H_{n_y,1}} \left[(1 - \lambda^{n_y})r - (1 - \lambda^{n_y})y_k - Q_{n_y}y_{\leftarrow m,k} + y_{m,k} - P_{n_y}\Delta_{\leftarrow}u_k \right] \quad (7.18)$$

For suitable F, N_i, M_i, \hat{D} , one can rearrange (7.18) as:

$$\Delta u_k = Fr - N_i y_{\leftarrow m,k} - M_i y_k - \hat{D}\Delta_{\leftarrow}u_k \quad (7.19)$$

Transforming (7.19) into an equivalent transfer function format, the fixed closed loop is constructed as:

$$D(z)\Delta u_k = F(z)r - N_i(z)y_{m,k} - M_i(z)y_k \quad (7.20)$$

The model output can be determined exactly from the model $y_{m,k} = b(z)a(z)^{-1}u_k$ and hence equation (7.20) can be replaced by (see Figure 7.4 for the effective loop structure):

$$\underbrace{[D(z)\Delta + N_i(z)b(z)a(z)^{-1}]}_{D_i(z)} u_k = F(z)r - M_i(z)y_k \quad (7.21)$$

The sensitivities for IM structure of Figure 7.4 are obtained analogously to the PFC with RM structure by substituting parameter $D(z)\Delta$ with $D_i(z)$, and $N(z)$ with $M_i(z)$ in equation (7.7-7.9). The closed-loop pole polynomial and sensitivities are:

$$\begin{aligned} P_i(z) &= D_i(z)a(z) + M_i(z)b(z) \\ S_{un} &= M_i(z)P_i(z)^{-1}a(z) \\ S_{yd} &= a(z)P_i(z)^{-1}D_i(z) \end{aligned} \quad (7.22)$$

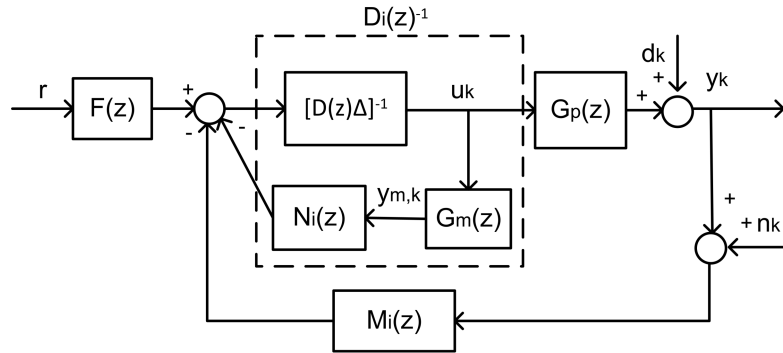


Figure 7.4: PFC with IM structure equivalent block diagram.

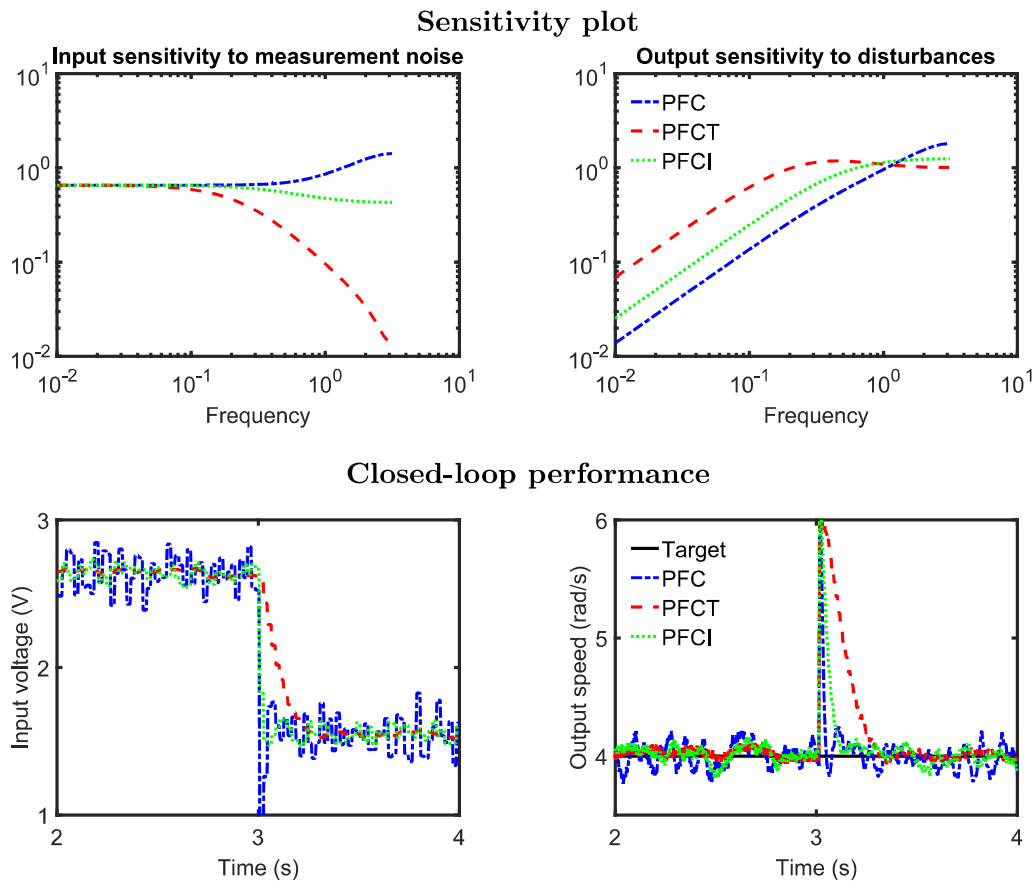
7.1.4 Results

This paper analyses the sensitivity of PFC with a RM structure (PFC), PFC with a T-filter (PFCT) and PFC with an IM structure (PFCI) for a first order real plant, that is, a Quanser SRV02 servo based unit G_{11} (Apkarian et al., 2012), and second order numerical examples which represent an over-damped process G_{12} .

$$G_{11} = \frac{0.8338}{1 - 0.455z^{-1}} \quad (7.23)$$

$$G_{12} = \frac{z^{-2} + 0.3z^{-1}}{1 - 1.2z^{-1} + 0.32z^{-2}} \quad (7.24)$$

1. For a first order Quanser servo system, all the PFC structures will use the same tuning parameters ($\lambda = 0.7$ and $n_y = 3$). Figure 7.5 shows that in the high frequency range, the first order PFCT ($T = 1 - 0.8z^{-1}$), gives the lowest sensitivity to noise followed by PFCI and PFC. However, the output of PFCT becomes more sensitive to low and mid frequency disturbances compared to PFCI and PFC. The real response validates this analysis and it can be concluded that for this case, it may be worth to have a slower disturbance rejection (which is less likely to occur) to get the best noise sensitivity with the T-filter.
2. For a second order over-damped process, Figure 7.6 demonstrates that the input of PFCT2 with second order filter ($T_2 = (1 - 0.8z^{-1})^2$) gives the lowest input sensitivity

Figure 7.5: Sensitivity analysis for process G_{11} .

to noise followed by PFCI, PFCT1 with first order filter ($T = 1 - 0.8z^{-1}$) and PFC. Nevertheless, over filtering the measurement as with PFCT2 leads to a poor output reaction to disturbances in the low or mid frequency range compared to other structures. The closed-loop response validates this analysis and for this specific case, there is a reasonable argument that the IM structure provides a good sensitivity trade off between noise and disturbances.

Although generic conclusions are not applicable in this study, it is clearly shown that the popular IM structure does not always give the best trade-off between the uncertainties as demonstrated in the first example. In some cases, using a low pass filter such as a T-filter can enable a good sensitivity trade-off between noise and disturbances. However, the sensitivity of PFC structures are system dependent and thus the best option may not be

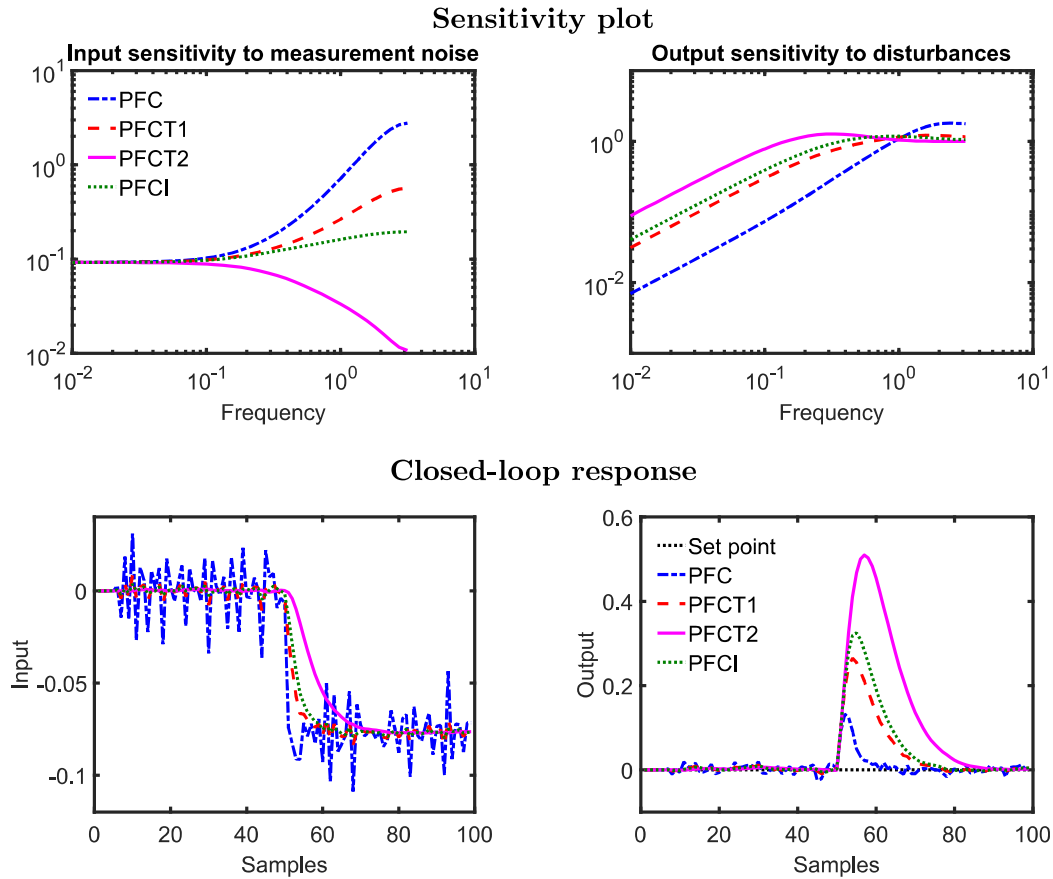


Figure 7.6: Sensitivity analysis for G_{12} .

clear a priori, as the latter example indicated a likely preference for using the IM approach. Hence, production of off-line sensitivity plots is essential to give insight into the robustness of differing PFC structures.

7.2 SENSITIVITY ANALYSIS OF LAGUERRE PFC

It is noted that Laguerre PFC can provide several benefits such as improving the prediction consistency, closed-loop performance and constraint handling. However, it is also well-known that an input shaping method, in general, can affect the loop sensitivity of a system since the algorithm becoming too dependent on the accuracy of a representative model. Hence, the UKACC 2018 conference paper (Abdullah & Rossiter, 2018c), which is attached in Appendix H presents a formal sensitivity analysis of LPFC by considering the

effect of noise, unmeasured disturbances and parameter uncertainty. The performance will be benchmarked with the nominal PFC to get some insight into the sort of sensitivity trade-off that ones should expect when using LPFC. Similar as before, the sensitivity plots from bode diagrams and closed-loop simulations are used to measure the controller robustness. The scope of this work is only focused on a simple and stable dynamics system as LPFC is only effective for this process. Since a Laguerre polynomial is better parametrised with an input rather than input increments, both PFC and LPFC take the IM structure for handling noise, disturbance and parameter uncertainty.

7.2.1 Sensitivity Functions for PFC and LPFC

Generally, both PFC (2.10) and LPFC (5.9) control laws can be converted into a fixed structure by grouping the common terms where for an IM structure:

$$D_i(z)u_k = F(z)r - M(z)y_{p,k} \quad (7.25)$$

By referring to Figure 7.7, the effective control law can be simplified to $K(z) = M(z)[D_i(z)]^{-1}$.

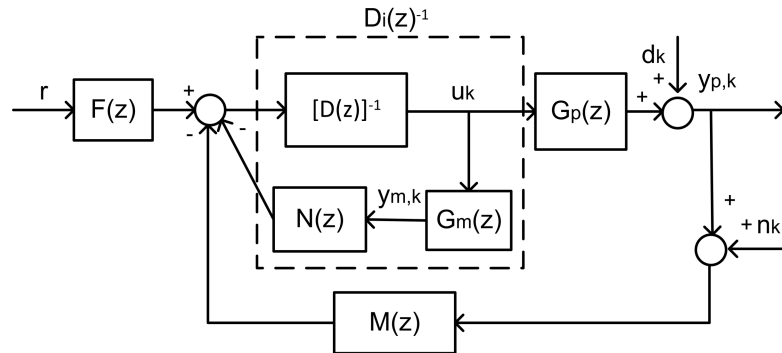


Figure 7.7: IM structure equivalent block diagram for PFC and LPFC.

Assuming system $G(z) = b(z)a(z)^{-1}$, the closed-loop pole polynomial $P_i(z) = 1 + K(z)G(z)$ is represented as:

$$P_i(z) = D_i(z)a(z) + M(z)b(z) \quad (7.26)$$

The sensitivity of the input to noise is derived by finding the transference from $n(z)$

to $u(z)$:

$$S_{un} = K(z)[1 + K(z)G(z)]^{-1} = M(z)P_i(z)^{-1}a(z) \quad (7.27)$$

Similarly, the sensitivity of output to disturbance is obtained by solving the transference from $d(z)$ to $y(z)$:

$$S_{yd} = [1 + K(z)G(z)]^{-1} = a(z)P_i(z)^{-1}D_i(z) \quad (7.28)$$

The multiplicative uncertainty is modelled as $G(z) \rightarrow (1 + \delta)G(z)$, for δ a scalar (possibly frequency dependent). Thus the closed-loop pole sensitivity to multiplicative uncertainty becomes:

$$\begin{aligned} P_c &= [1 + G(1 + \delta)K] = 0 \\ S_g &= GK[1 + K(z)G(z)]^{-1} = M(z)P_i(z)^{-1}b(z) \end{aligned} \quad (7.29)$$

From the derivation, it is clear that the main difference between the sensitivity functions for PFC and LPFC is the parameter $D_i(z), F(z), M(z)$.

7.2.2 Results

This paper considers a second order over-damped process G_{13} . For the first example, both PFC and LPFC are tuned using a faster λ compared to its slowest open-loop pole ($\lambda = 0.7, n_y = 7$). The second example demonstrates the effect of loop sensitivity when the controllers are tuned using slower pole $\lambda = 0.92$ (almost similar with the slowest open-loop pole) with $n_y = 9$. In order to analyse the sensitivity trade-off between PFC and LPFC, the Bode plots of each sensitivity functions are plotted together with their closed-loop bandwidth. The outcomes of this analysis are given as:

$$G_{13} = \frac{0.1z^{-1} + 0.4z^{-2}}{(1 - 0.5z^{-1})(1 - 0.9z^{-1})} \quad (7.30)$$

1. For the first case, Figure 7.8 shows that LPFC has a higher bandwidth compared to PFC. Since LPFC has a faster dynamics, it becomes less sensitive in rejecting low-frequency disturbance. However, higher bandwidth requires more aggressive input activity, and thus LPFC becomes more sensitive to measurement noise and modelling uncertainty compared to conventional PFC. The closed-loop simulation in Figure 7.9 validates the analysis where LPFC is slightly less robust than PFC in handling noise

and uncertainty, yet better in rejecting disturbance and tracking the target, but that observation is most likely linked to the difference in implied closed-loop poles with LPFC delivering the desired pole while PFC is not doing so.

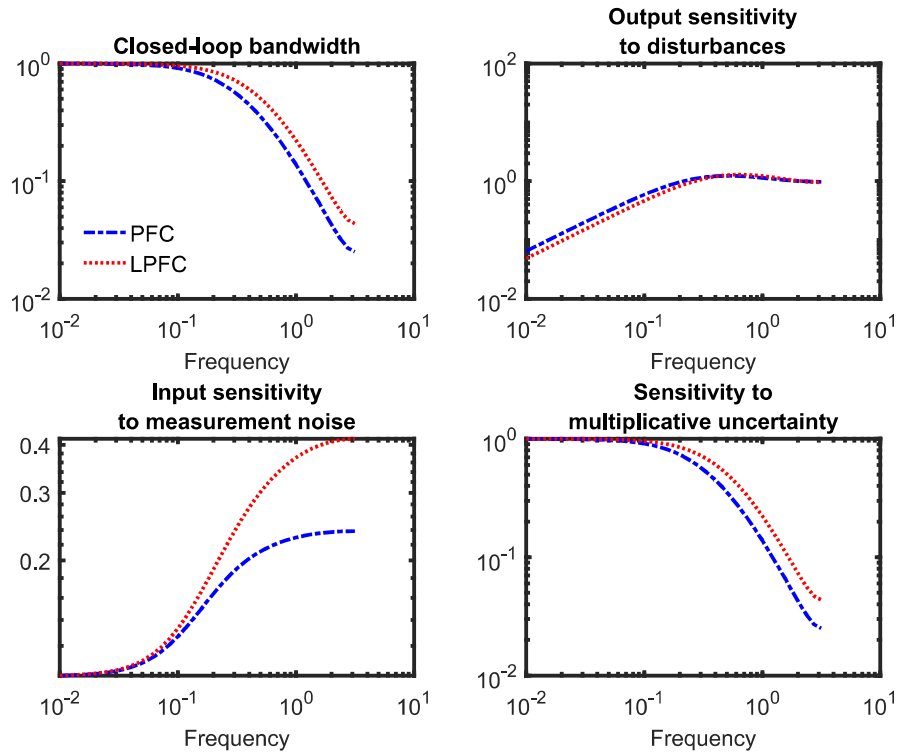


Figure 7.8: Sensitivity plot for process G_{13} with $\lambda = 0.7$ and $n = 7$.

- For the second example, when LPFC is tuned to have a similar bandwidth with PFC, Figure 7.10 demonstrates that both controllers provide similar sensitivity outcomes with respect to disturbance, noise and modelling uncertainty. Besides, the closed-loop simulation in Figure 7.11 validates the analysis.

In summary, it is clear that the controller sensitivity is related to the achieved closed-loop bandwidth. Since LPFC is better at delivering the target λ compared to PFC, in consequence, for the same λ , LPFC is usually more highly tuned and thus more sensitive to noise and modelling uncertainty. However, where the two control laws give similar

closed-loop poles (perhaps by deploying different λ), their sensitivities are similar. Therefore, LPFC is a better base on which to explore the trade-offs in the sensitivity, as there is a stronger connection between the tuning parameters and the achieved closed-loop performance (Abdullah & Rossiter, 2016) in addition to better constraint handling due to its well-posed decision and prediction consistency as discussed in (Abdullah et al., 2017).

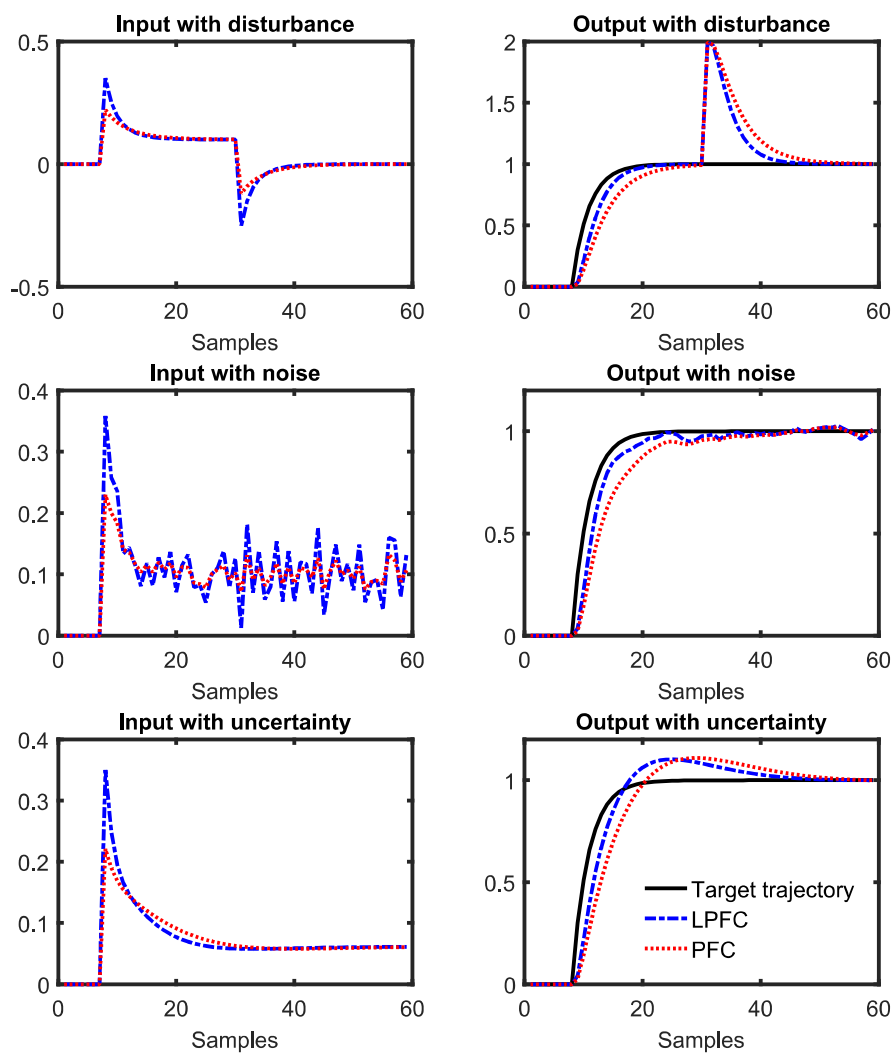


Figure 7.9: Closed-loop response of process G_{13} with $\lambda = 0.7$ and $n = 7$ in the presence of disturbance, noise, and uncertainty.

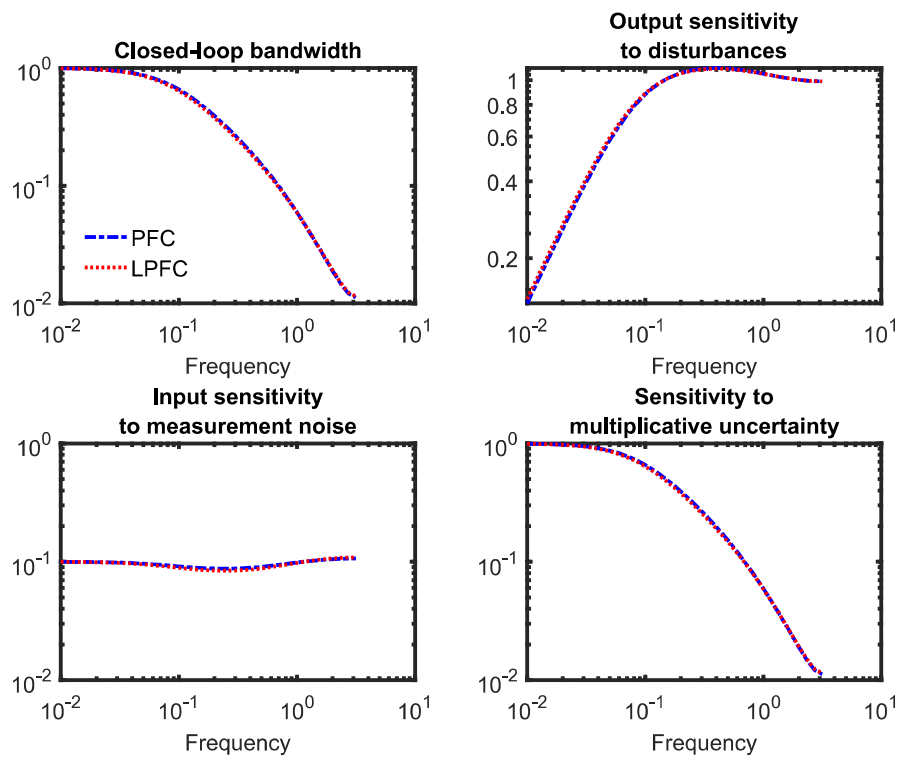


Figure 7.10: Sensitivity plot for process G_{13} with $\lambda = 0.92$ and $n = 9$.

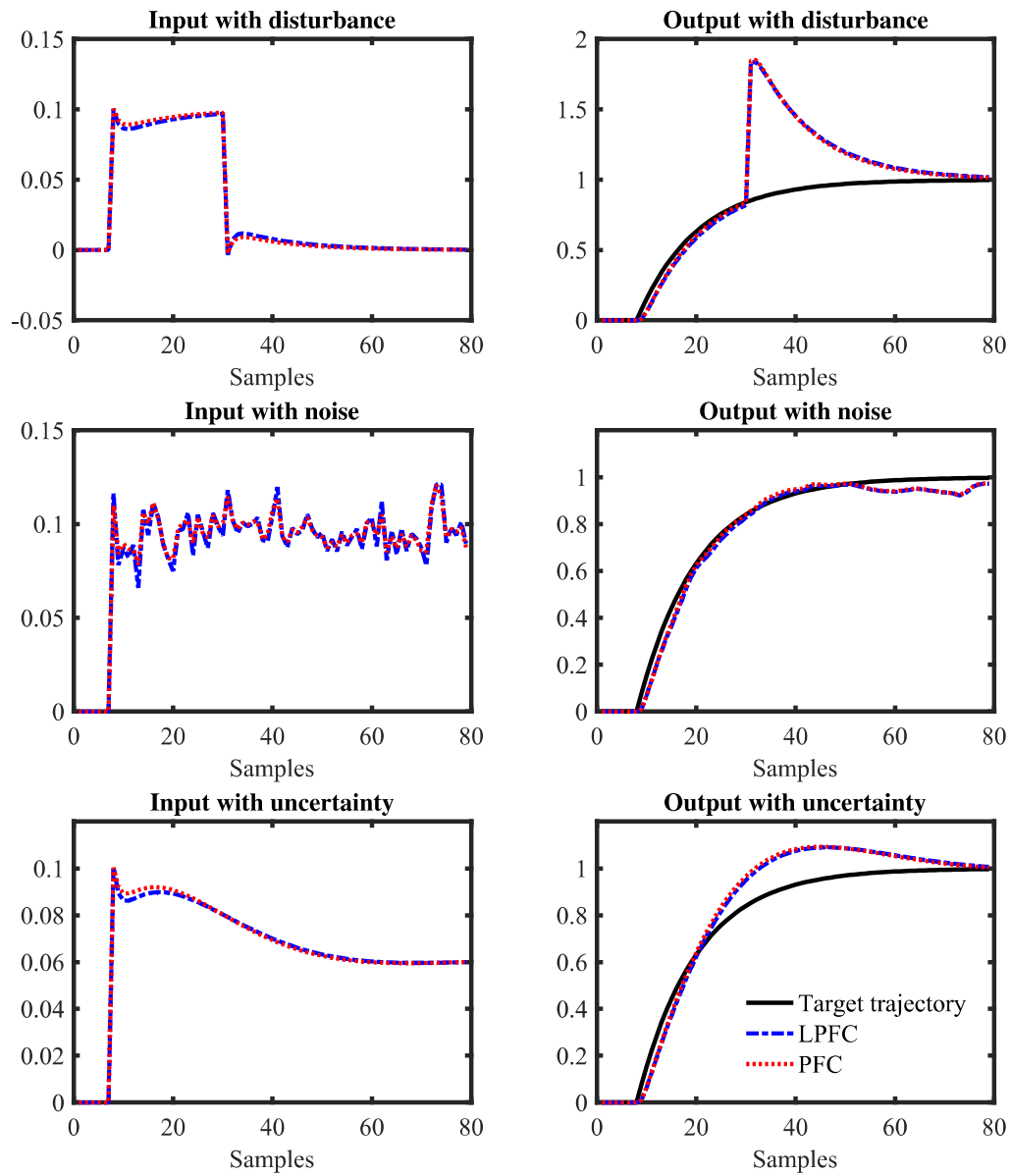


Figure 7.11: Closed-loop response of process G_{13} with $\lambda = 0.92$ and $n = 9$ in the presence of disturbance, noise and uncertainty.

7.3 SENSITIVITY ANALYSIS OF PS-PFC WITH T-FILTER

This section presents a sensitivity analysis of the Pole Shaping PFC (PS-PFC) algorithm, that was proposed in the previous chapter to handle processes with open-loop divergent or oscillatory dynamics. Notably, PS-PFC utilises the RM structure for unbiased predictions, where it is well known that this structure can naturally handle uncertainty and disturbance. Nevertheless, it may be very sensitive to a high frequency measurement noise which can result in poor control performance or at worst instability (J. A. Rossiter, 2018). In order to improve this issue, a T-filter based prediction is proposed while analysing its trade-off against the sensitivity to disturbances and parameter uncertainty. As to the authors' knowledge, no-one has looked at this in the context of shaped predictions and thus this proposal forms the main contribution of this work. The detailed formulations, discussions and results are available in the CCTA 2019 conference paper (Abdullah & Rossiter, 2019a) which is attached in Appendix I.

7.3.1 Sensitivity Functions for PS-PFC with and without T-filter

The derivation of PS-PFC with a T-filter has a direct link with the presented formulation in Chapter 6. In order to implement a T-filter with a PS-PFC control law, the matrix C_A needs to be separated from prediction (7.13). Noting the definition of H , P , Q in (7.1) and \tilde{P} , \tilde{Q} , the new representation can be formed as:

$$\mathbf{y}_{\rightarrow k+1} = [C_{a-\Delta}]^{-1}[C_{a+}]^{-1}[C_b\Delta\mathbf{u}_{\rightarrow k} + \underbrace{\tilde{H}_b\Delta\tilde{\mathbf{u}}_{\leftarrow k} - \tilde{H}_A\tilde{\mathbf{y}}_{\leftarrow k}}_{\mathbf{p}}] \quad (7.31)$$

where $\tilde{H}_b = C_T H_b - C_b H_T$ and $\tilde{H}_A = C_A H_T + C_T H_A$. Now, the control law with a T-filter follows a similar derivation of PS-PFC in Section 7.1.2, but using filtered prediction (7.31) instead of (6.3).

Remark 7.2. *If $T = 1$, the prediction of (7.31) becomes nominal as in (6.3) given that $C_T = 1$ and $H_T = 0$. Thus the derivation of sensitivity functions for PS-PFC with and without T-filter can be generalised.*

Based on receding horizon principle, only the first computed input is implemented. Since the first row of matrix C_α^{-1} and C_{a+} is always 1, the input trajectory of (6.16) and the implied control law can be simplified to:

$$\Delta u_k = [\hat{P}_1 \mathbf{p} + \phi] \quad (7.32)$$

The control law is then rearranged into compact form by substituting ϕ_k from the filtered prediction of (7.31) into (7.32):

$$\Delta u_k = \hat{P}_1 \mathbf{p} + \frac{1}{h_{n_y, \alpha}} \left[(1 - \lambda^{n_y})r + \lambda^{n_y} y_k - Q_{n_y, \alpha} \tilde{y}_k - P_{n_y, \alpha} \Delta \tilde{u}_k \right] \quad (7.33)$$

As \mathbf{p} is representing both past filtered input and output, equation (7.33) is rearranged in a more general form by noting the definition of $y_k = C_T \tilde{y}_k + H_T \tilde{y}_{k-1}$ and grouping the common term of r , \tilde{y}_k , and $\Delta \tilde{u}_k$ into:

$$\begin{aligned} \Delta u_k &= Fr - N \tilde{y}_k - \hat{D} \Delta \tilde{u}_k \\ &= Fr - NT^{-1} y_k - \hat{D} T^{-1} \Delta u_k \end{aligned} \quad (7.34)$$

Noting the definitions of $\Delta \tilde{u}_k$ and y_k , the sensitivity functions are derived based on a fixed closed-loop form of:

$$D(z)T(z)^{-1} \Delta u_k = F(z)r - N(z)T(z)^{-1} y_k \quad (7.35)$$

Figure 7.3 indicates the equivalent block diagram in the presence of measurement noise n and output disturbance d . Here, the overall control law can be simplified to $K(z) = N_c(z)[D_c(z)\Delta]^{-1}$. Assuming system $G(z) = b(z)a(z)^{-1}$, the closed-loop pole polynomial $P_c(z) = 1 + K(z)G(z)$ is represented as:

$$P_c(z) = D(z)a(z)\Delta + N(z)b(z) \quad (7.36)$$

Hence the associated sensitivity of input to noise, output to disturbance and parameter uncertainty are given as:

$$S_{un} = K(z)[1 + K(z)G(z)]^{-1} = N(z)P_c(z)^{-1}a(z) \quad (7.37)$$

$$S_{yd} = [1 + K(z)G(z)]^{-1} = a(z)P_c(z)^{-1}D(z)\Delta \quad (7.38)$$

$$S_g = GK[1 + K(z)G(z)]^{-1} = M(z)P_i(z)^{-1}b(z) \quad (7.39)$$

7.3.2 Results

This subsection presents two numerical examples to demonstrate the impact of a T-filter on the loop sensitivity in the presence of noise, disturbance and parameter uncertainty. The selected process with its tuning parameters are given as:

1. A 3rd order under-damped process with two under-damped poles and $\alpha(z) = (1 - 0.8z^{-1})(1 - 0.8^{-1})$:

$$G_{14} = \frac{0.85z^{-1} - 1.5z^{-2} + 0.85z^{-2}}{(1 - 0.6z^{-1})(1 - 1.6z^{-1} + 0.8z^{-2})}; \quad \lambda = 0.7; n_y = 4 \quad (7.40)$$

2. A 2nd order unstable system with one pole is outside the unit circle and $\alpha(z) = (1 - 0.833z^{-1})$:

$$G_{15} = \frac{0.4z^{-1} - 0.1z^{-2}}{(1 - 0.5z^{-1})(1 - 1.2z^{-1})} \quad \lambda = 0.7; n_y = 6 \quad (7.41)$$

The sensitivity plots are shown in Figures 7.12 and 7.13, where $T_1 = 1$ represents PS-PFC without a T-filter and $T_2 = 1 - 0.8z^{-1}$ is PS-PFC with a first order filter. For both cases, the overall outcome can be summarised as:

- PS-PFC with a T-filter manages to reduce the overall sensitivity in the high-frequency domain, yet becomes more sensitive in the low and mid frequency domains compared to PS-PFC without a T-filter.
- This scenario means that the process will respond better with a T-filter in the presence of high-frequency noise, but worse in rejecting low or mid frequency disturbances and parameter uncertainty.

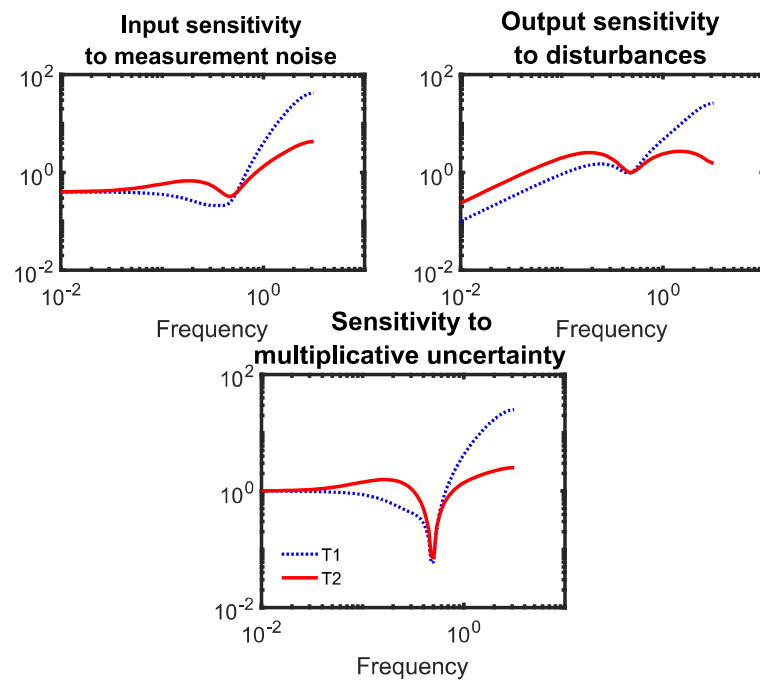
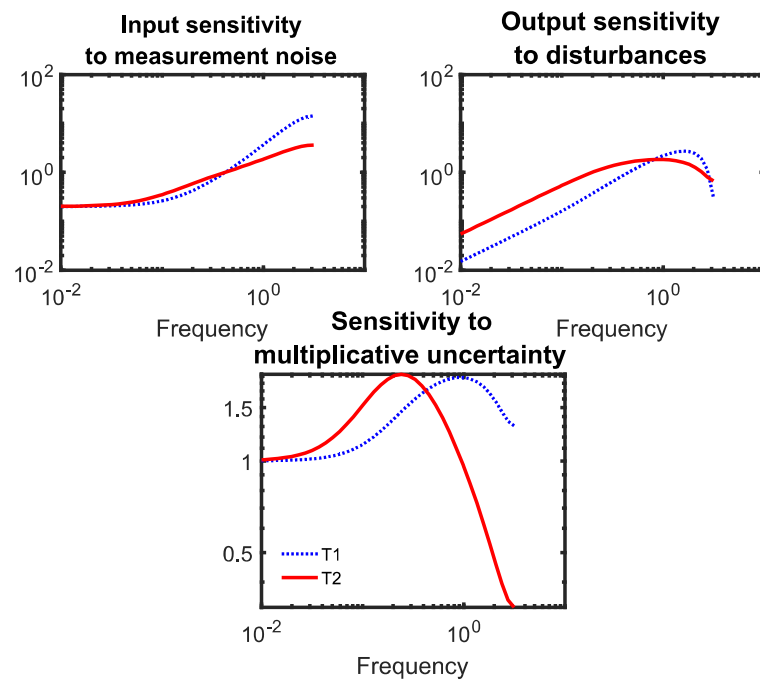
In order to validate this off-line analysis, the closed-loop responses for both processes are simulated with three different conditions. Figures 7.14 and 7.15 demonstrate that:

1. When Gaussian random white noise corrupts the output measurement of both processes, PS-PFCT rejects noise better compared to PS-PFC, thus reducing the overall input fluctuation.

2. In the presence of an output disturbance, PS-PFCT is slower in rejecting it compared to PS-PFC, yet its input activity is far less aggressive.
3. When there is a parameter uncertainty (consider plant dynamics different from the model), both controllers provide offset-free tracking, but the response with PS-PFCT is a bit slower compared to PS-PFC.

In addition to these numerical examples, the paper also has implemented the proposed control law on real laboratory hardware, specifically the Quanser servo with a flexible arm to track its angular position. This process consists of two complex poles and one integrating pole along with some plant model mismatch. The bode plot shows a similar trade-off in sensitivity to those noted in the numerical examples. However, when implemented in a real plant, only the second-order T- filter is implementable, where else the input fluctuation is unacceptable/unsafe. The controller manages to track the alternating set-point without offset error while reducing the oscillation in the flexible link. Hence, in this case, the usage of a T-filter is essential for the effective utilisation of PS-PFC.

In essence, the associated sensitivity analysis of PS-PFC for noise, disturbance and parameter uncertainty with and without the T-filter indicates that the benefits are similar to those achieved in the context of GPC. Although without generic proofs, it is still clear that use of T-filter often helps in reducing the sensitivity of an input to measurement noise with only relatively small deterioration in disturbance rejection.

Figure 7.12: Sensitivity plots for G_{14} with different T-filters.Figure 7.13: Sensitivity plots for G_{15} with different T-filters.

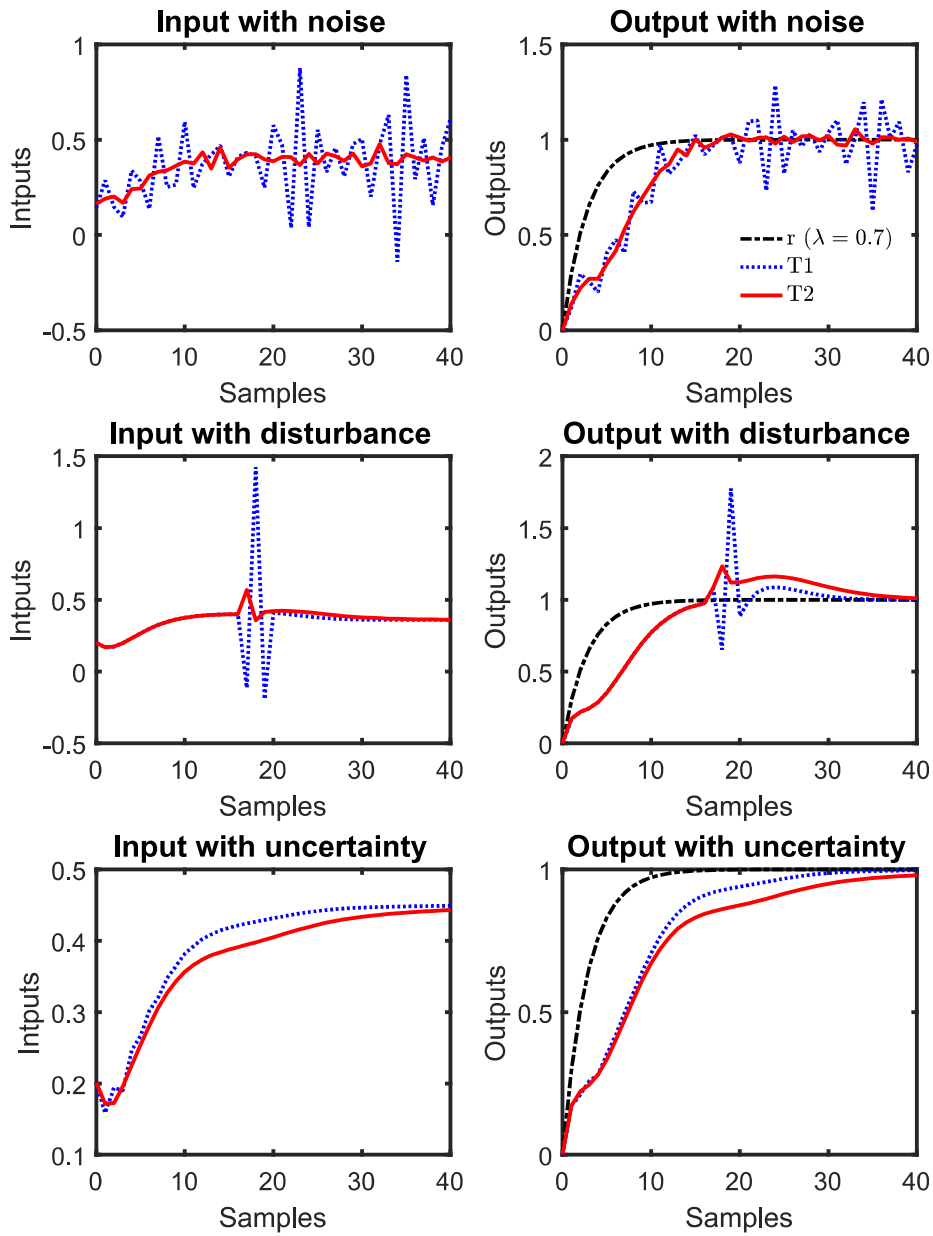


Figure 7.14: Closed-loop responses of G_{14} in the presence of noise, output disturbance and parameter uncertainty.

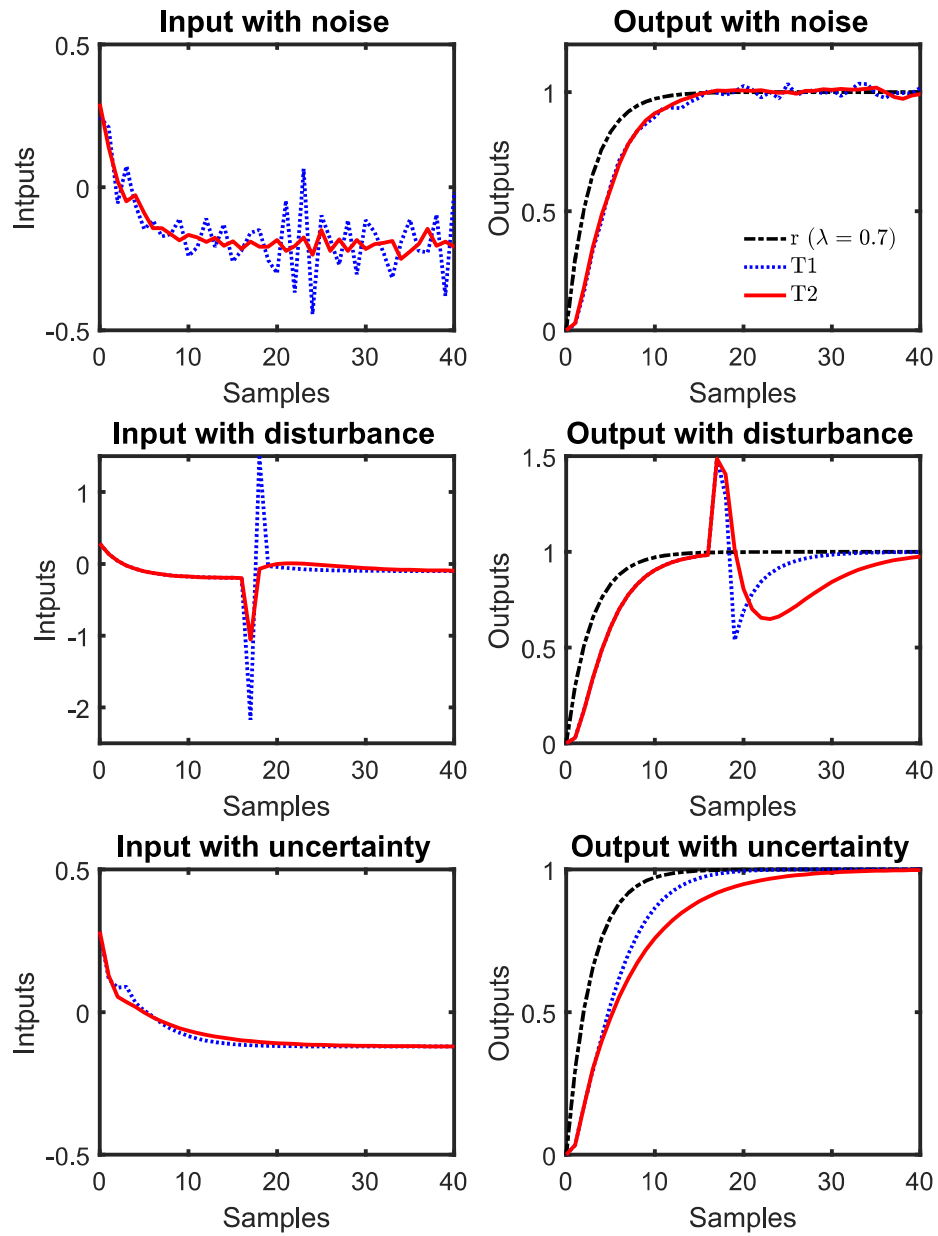


Figure 7.15: Closed-loop responses of G_{15} in the presence of noise, output disturbance and parameter uncertainty.

7.4 SUMMARY

This chapter has presented a comprehensive summary of three different works to tackle our third research objective which is to analyse the sensitivity of PFC with different structures where:

1. Abdullah & Rossiter (2018b) shows that using IM is not always beneficial especially when dealing with high-frequency noise, hence other alternative structure such as the use of T-filter with RM also should be considered.
2. Abdullah & Rossiter (2018c) presents the sensitivity trade-off that one should expect when using LPFC where a user needs to sacrifice some of its sensitivity features for getting a better closed-loop performance and other associated benefits.
3. Abdullah & Rossiter (2019a) implements a T-filter in a PS-PFC framework to improve its input sensitivity to measurement noise with only relatively small deterioration in rejecting disturbance and parameter uncertainty. However, this trade-off is essential to ensure that the control law is providing an implementable computed input.

Overall, it is apparent that the sensitivity is system dependent. Hence, in practice, an off-line sensitivity analysis is essential to gain some insight into the robustness with differing prediction structures. The findings from this work are only the initial starting point to motivate the needs for a more comprehensive robust analysis in the future for different types of PFC such as the cascade structure, pole placement and others.

Chapter 8

CONCLUSIONS AND FUTURE WORKS

This final chapter is divided into two sections whereon Section 8.1 presents the overall conclusion of this research and Section 8.2 suggests several potential future works that can be carried out to further enhance the concept of PFC.

8.1 FINAL CONCLUSIONS

From the literature survey, it is noted that PFC only received a few attention in the academic literature considering its capability to replace the use of traditional PID controller in many small-scale applications. Most of the current developments are focusing on the hybridisation of PFC with other advanced techniques rather than improving the control law itself. Thus, this research has presented several novel developments of PFC to improve some of its significant weaknesses such as the tuning efficacy, constrained performance, reliability in handling different types of challenging dynamics processes and robustness analysis to uncertain cases. Although a reliable general solution for all the problems is not available, yet this work has provided some of the best options for each different cases.

8.1.1 Stable and simple dynamics system

The implementation of Laguerre PFC specifically with an inputs parameterisation for a system with simple and stable dynamics seems promising considering it manages to provide a better prediction consistency, overall closed-loop performance and constraint handling compared to the traditional PFC approach. Besides, this work also proposes the use of a systematic and cost-effective constraint handling method as in MPC rather

than the traditional multiple-regulators scheme, where it provides more accurate and less conservative constrained solutions. Since the overall modification for the proposed control law is straightforward, the simplicity of tuning and implementation of the original concept can be retained. Nevertheless, when using LPFC, a user may need to modify or slow down the implied terminal constraints to avoid infeasibility in some cases. This procedure may need extra computation, yet it is still relatively simpler than MPC.

8.1.2 Challenging dynamics systems

For challenging dynamics system, this research has explored three alternative methods namely Laguerre PFC (LPFC), Pole Placement PFC (PP-PFC) and Pole Shaping PFC (PS-PFC) to improve the reliability of the PFC concept. It is noted that LPFC and PP-PFC are only implementable with integrating and oscillating dynamics processes, respectively. Besides, these structures have their limitations, for example, LPFC may provide a small offset error in the presence of plant model mismatch due to the setting of $u_{ss} = 0$, while PP-PFC constraint handling is only effective for a single prediction. Conversely, the PS-PFC concept can be generalised for different types of challenging dynamics systems with the proposed default shaping of α . This control law can provide stable closed-loop performance with a smooth input dynamic while guaranteeing recursive feasible constrained solutions. However, a user should also be concerned with the use of Realigned Model structures where additional filtering may be required for reducing its sensitivity to measurement noise.

8.1.3 Sensitivity Analysis

In general, the sensitivity of PFC structures is system dependent where the best option may not be apparent a priori. Hence, this research has provided an off-line sensitivity analysis for comparing the robustness of differing PFC structures. The first analysis is employed to compare the sensitivity between PFC with a RM structure, PFC with an IM structure and PFC with a T-filter. Although a general conclusion is not applicable, it is noted that a RM structure is highly sensitivity to measurement noise. Conversely, the IM structure also is not always the best option to improve this sensitivity. For some cases, using a T-filter with a RM structure may provide a better improvement and sensitivity trade-off.

The second analysis is performed to the proposed LPFC control law, which utilises the IM structure. It is shown that when using LPFC, a user need to pay a small trade-off by having a more sensitive controller to noise and uncertainty since it is highly tuned with a larger bandwidth compared to the conventional PFC. However, both controllers may typically have similar sensitivities when both are tuned to get similar closed-loop poles which would indicate a preference for LPFC in general due to easier tuning and other advantages.

The third analysis demonstrates the benefits of using a T-filter prediction with PS-PFC where it is originally designed with the RM structure. The associated sensitivity response of PS-PFC for noise, disturbance and parameter uncertainty with and without the T-filter indicates that the benefits are similar to those achieved in the context of GPC (J. A. Rossiter, 2018). The simulation and experiment results clearly show that the use of T-filter often helps in reducing the sensitivity of an input to measurement noise with only relatively small deterioration in disturbance rejection.

8.2 RECOMMENDATION FOR FUTURE WORK

In a nutshell, the development works in this thesis are not fully matured. There are still a lot of validations, comparisons, modifications and analysis that need to be done in order to fully understand the rigour behind these frameworks. For examples:

1. Notably, the proposed PS-PFC concept can provide stable closed-loop performance with satisfactory constrained solutions when handling different types of challenging dynamics processes. However, it also may face similar prediction consistency issues as other PFC frameworks if the implied validation horizon is larger than the coincidence horizon. Possible future work is to look at how one can effectively shape the pole α to eliminate this weakness to future improve its constrained performance.
2. If the previous recommendation is implementable, there is also a possibility to adopt a similar concept for handling a stable and simple dynamics system. This framework may generalise the use of PS-PFC for all types of system dynamics. Nevertheless, it needs to undergo several validations and comparisons with other alternative structures

such as LPFC and cascade structure to get an insight into its performance, tuning, coding, computation demand and robustness.

3. Another possible idea is to utilise the concept of implied closed-loop prediction as in the dual mode MPC to improve the constraint performance, which is not yet explored in PFC framework. Probably, the findings from this proposal may change or not change the insights that have been presented in this thesis. Hence, it is interesting to analyse how to implement this concept effectively while retaining the simplicity of formulation. Besides, one also should investigate whether it can provide recursively feasible constrained solutions.
4. Recent work of J. Zhang et al. (2018) has developed a more systematic cascade structure of PFC that can provide a direct relationship between the desired closed-loop pole with the internal tuning gain. Although, the current idea is only limited to a simple integrating dynamics, yet the concept may be extend for handling other challenging dynamics systems with proper modification. Hence, it is possible that this structure can provide a similar effect as the PS-PFC method. Again a detailed review and comparison with other structures are needed to justify what sort of benefits that it can provide.
5. Since the sensitivity analysis in this thesis only covers the influence of prediction structure, future work should consider the impact of changes in the parameters λ, n_y and the choice of shaping poles $\alpha(z)$ in PS-PFC where these parameters can also affect the loop sensitivity of a controller. Besides, a more formal robustness comparison with other alternative PFC methods such as cascade structures and pole placement also should be considered.
6. The use of state space prediction model with a Kalman filter in PFC framework should also be analysed in details whether it can provide better sensitivity trade-off compared to a T-filter.
7. Indeed the use of a first order target trajectory can provide a simple and transparent framework. However, future work should look at other alternative functions or

representation that may provide better closed-loop performance while retaining the simplicity of tuning and implementation as one of the advantages of PFC is to track ramp or parabolic set point.

8. In the literature, several works have extended the concept of PFC as a non-linear controller. Again a complete analysis should be done to provide more rigour for this applications.
9. Since PFC can be an excellent alternative to the classical PID controller, it is also interesting to employ this simple concept to a more complicated application where PFC can act as a slave controller rather than the main one so that a user can extract full benefits from this design.
10. A formal comparison between the proposed methods with other existing techniques such as MPC, LQR, pole-placement and others should also be considered which can provide useful insight to a user.

REFERENCES

- Abdullah, M., & Idres, M. (2014a). Constrained Model Predictive Control of Proton Exchange Membrane Fuel Cell. *Journal of Mechanical Science and Technology*, 28(9), 3855–3862.
- Abdullah, M., & Idres, M. (2014b). Fuel cell starvation control using Model Predictive technique with Laguerre and exponential weight functions. *Journal of Mechanical Science and Technology*, 28(5), 1995–2002.
- Abdullah, M., & Rossiter, J. A. (2016). Utilising Laguerre function in Predictive Functional Control to ensure prediction consistency. In *2016 11th UKACC International Conference on Control* (pp. 1–6).
- Abdullah, M., & Rossiter, J. A. (2018a). Alternative method for Predictive Functional Control to handle an integrating process. In *Press of 2018 12th UKACC International Conference on Control*.
- Abdullah, M., & Rossiter, J. A. (2018b). The effect of model structure on the noise and disturbance sensitivity of Predictive Functional Control. In *Press of 2018 European Control Conference*.
- Abdullah, M., & Rossiter, J. A. (2018c). A formal sensitivity analysis for Laguerre based Predictive Functional Control. In *Press of 2018 12th UKACC International Conference on Control*.
- Abdullah, M., & Rossiter, J. A. (2018d). Input shaping Predictive Functional Control for different types of challenging dynamics processes. *Processes*, 6(8), 118.
- Abdullah, M., & Rossiter, J. A. (2019a). Sensitivity analysis for an input shaping Predictive Functional Control for processes with challenging dynamics. In *Submission to 3rd IEEE Conference on Control Technology and Applications*.

- Abdullah, M., & Rossiter, J. A. (2019b). Using Laguerre functions to improve the tuning and performance of Predictive Functional Control. *Accepted by International Journal of Control*.
- Abdullah, M., Rossiter, J. A., & Haber, R. (2017). Development of constrained Predictive Functional Control using Laguerre function based prediction. *IFAC-PapersOnLine*, 50(1), 10705–10710.
- Abu, S., Fiani, P., & Richalet, J. (1991). Handling input and state constraints in Predictive Functional Control. In *Proceedings of the 30th IEEE Conference on Decision and Control* (pp. 985–990).
- Apkarian, J., Levis, M., & Gurocak, H. (2012). *Instructor workbook: SVR02 based unit experiment for LabVIEW users*. Quanser Inc.
- Åström, K. J., & Hägglund, T. (2001). The future of PID control. *Control Engineering Practice*, 9(11), 1163–1175.
- Astrom, K. J., Hang, C. C., & Lim, B. (1994). A new Smith predictor for controlling a process with an integrator and long dead-time. *IEEE Transactions on Automatic Control*, 39(2), 343–345.
- Aström, K. J., & Murray, R. M. (2010). *Feedback systems: an introduction for scientists and engineers*. Princeton university press.
- Azira, A., Osman, K., Samsudin, S., & Sulaiman, S. F. (2018). Predictive Functional Controller (PFC) with novel observer method for Pneumatic Positioning System. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 10(2-6), 119–124.
- Bemporad, A., & Morari, M. (1999). Robust Model Predictive Control: A survey. In *Robustness in Identification and Control* (pp. 207–226). Springer.
- Bishop, G., Welch, G., et al. (2001). An introduction to the kalman filter. *Proc of SIGGRAPH, Course*, 8(27599-3175), 59.
- Camacho, E. F., & Bordons, C. A. (2012). *Model Predictive Control in the process industry*. Springer Science & Business Media.

- Clarke, D. W., & Mohtadi, C. (1989). Properties of Generalized Predictive Control. *Automatica*, 25(6), 859–875.
- Clarke, D. W., Mohtadi, C., & Tuffs, P. (1987). Generalized Predictive Control- Part I. The basic algorithm. *Automatica*, 23(2), 137–148.
- Cutler, C. R., & Ramaker, B. L. (1980). Dynamic Matrix Control- A computer control algorithm. In *Joint Automatic Control Conference* (p. 72).
- De Doná, J. A., Goodwin, G. C., & Seron, M. M. (2000). Anti-windup and Model Predictive Control: Reflections and connections. *European Journal of Control*, 6(5), 467–477.
- Diordiev, A., Ursaru, O., Lucanu, M., & Tigaeru, L. (2003). A hybrid PID-Fuzzy controller for DC/DC converters. In *2003 International Symposium on Signals, Circuits and Systems* (Vol. 1, pp. 97–100).
- Dovžan, D., & Škrjanc, I. (2010). Predictive Functional Control based on an adaptive Fuzzy model of a hybrid semi-batch reactor. *Control Engineering Practice*, 18(8), 979–989.
- Farina, M., Ferrari, G. P., Manenti, F., & Pizzi, E. (2016). Assessment and comparison of distributed Model Predictive Control schemes: Application to a natural gas refrigeration plant. *Computers & Chemical Engineering*, 89, 192–203.
- Garcia, C. E., & Morari, M. (1982). Internal model control. a unifying review and some new results. *Industrial & Engineering Chemistry Process Design and Development*, 21(2), 308–323.
- Garcia, C. E., Prett, D. M., & Morari, M. (1989). Model Predictive Control: Theory and practice - A survey. *Automatica*, 25(3), 335–348.
- Gilbert, E. G., & Tan, K. T. (1991). Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9), 1008–1020.
- Giselsson, P., & Rantzer, A. (2010). Distributed Model Predictive Control with suboptimality and stability guarantees. In *49th IEEE Conference on Decision and Control* (pp. 7272–7277).

- Grewal, M. S. (2011). *Kalman Filtering*. Springer.
- Guo, J., Wang, G., Guo, Z., & Zhou, J. (2017). Augmented Predictive Functional Control for missile autopilot design. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 231(10), 1794–1803.
- Haber, R., Bars, R., & Schmitz, U. (2012). *Predictive control in process engineering: From the basics to the applications*. John Wiley & Sons.
- Han, P., Wang, G., & Wang, D. (2003). On the application of Predictive Functional Control in steam temperature systems of thermal power plant. In *Proceedings of 42nd IEEE Conference on Decision and Control* (Vol. 6, pp. 6559–6564).
- Hashizume, S. (2015). *Development of a Predictive Functional Control technique and practical applications to chemical processes* (Tech. Rep.). Sumitomo Chemical Co., Ltd.
- Hrovat, D., Di Cairano, S., Tseng, H. E., & Kolmanovsky, I. V. (2012). The development of Model Predictive Control in automotive industry: A survey. In *IEEE International Conference on Control Applications* (pp. 295–302).
- Jones, C. N., & Kerrigan, E. (2015). Predictive control for embedded systems. *Optimal Control Applications and Methods*, 36(5), 583–584.
- Khadir, M., & 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.
- Khan, B., & Rossiter, J. A. (2011). Triple mode MPC or Laguerre MPC: A comparison. In *Proceedings of the American Control Conference* (pp. 5189–5194).
- Kouvaritakis, B., Gossner, J., & Rossiter, J. A. (1996). A priori stability conditions for an arbitrary number of unstable poles. *Automatica*, 32(10), 1441–1446.
- Kumar, V., Nakra, B., & Mittal, A. (2011). A review on classical and Fuzzy PID controllers. *International Journal of Intelligent Control and Systems*, 16(3), 170–181.
- Kwakernaak, H., & Sivan, R. (1972). *Linear optimal control systems* (Vol. 1). Wiley-Interscience New York.

- Li, S., Liu, H., & Fu, W. (2017). An improved Predictive Functional Control method with application to PMSM systems. *International Journal of Electronics*, *104*(1), 126–142.
- Ljung, L. (1998). System Identification. In *Signal Analysis and Prediction* (pp. 163–173). Springer.
- Mayne, D. Q. (2014). Model Predictive Control: Recent developments and future promise. *Automatica*, *50*(12), 2967–2986.
- Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Sokaert, P. O. (2000). Constrained Model Predictive Control: Stability and optimality. *Automatica*, *36*(6), 789–814.
- Mayne, D. Q., Seron, M. M., & Raković, S. (2005). Robust Model Predictive Control of constrained linear systems with bounded disturbances. *Automatica*, *41*(2), 219–224.
- Mosca, E., & Zhang, J. (1992). Stable redesign of predictive control. *Automatica*, *28*(6), 1229–1233.
- Muñoz, W. A. P., Sellier, A. G., & Castro, S. G. (2018). The Predictive Functional Control and the management of constraints in GUANAY II autonomous underwater vehicle actuators. *IEEE Access*, *6*, 22353–22367.
- Pedersen, T. S., Nielsen, K. M., Hindsborg, J., Reichwald, P., Vinther, K., & Izadi-Zamanabadi, R. (2017). Predictive Functional Control of superheat in a refrigeration system using a Neural Network Model. *IFAC-PapersOnLine*, *50*(1), 43–48.
- Qin, S. J., & Badgwell, T. (2003). A survey of industrial Model Predictive Control technology. *Control Engineering Practice*, *11*(7), 733–764.
- Qin, S. J., & Badgwell, T. A. (1997). An overview of industrial Model Predictive Control technology. In *AIChE Symposium Series* (Vol. 93, pp. 232–256).
- Qin, S. J., & Badgwell, T. A. (2000). An overview of nonlinear Model Predictive Control applications. In *Nonlinear Model Predictive Control* (pp. 369–392). Springer.
- Rawlings, J. B., & Muske, K. R. (1993). The stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, *38*(10), 1512–1516.

- Richalet, J. (2007). Industrial application of Predictive Functional Control. *Nonlinear Model Predictive Control, Software and Applications*.
- Richalet, J., Estival, J., & Fiani, P. (1995). Industrial applications of Predictive Functional Control to metallurgical industries. In *Proceedings of the 4th IEEE Conference on Control Applications* (pp. 934–942).
- Richalet, J., & O’Donovan, D. (2009). *Predictive Functional Control: Principles and industrial applications*. Springer Science & Business Media.
- Richalet, J., & O’Donovan, D. (2011). Elementary Predictive Functional Control: A tutorial. In *2011 International Symposium on Advanced Control of Industrial Processes (ADCONIP)* (pp. 306–313).
- Richalet, J., Rault, A., Testud, J., & Papon, J. (1978). Model predictive heuristic control: Applications to industrial processes. *Automatica*, *14*(5), 413–428.
- Rossiter, J., Haber, R., & Zabet, K. (2016). Pole-placement Predictive Functional Control for over-damped systems with real poles. *ISA Transactions*, *61*, 229–239.
- Rossiter, J., & Kouvaritakis, B. (1994). Numerical robustness and efficiency of Generalised Predictive Control algorithms with guaranteed stability. *IEEE Proceedings-Control Theory and Applications*, *141*(3), 154–162.
- Rossiter, J., & Kouvaritakis, B. (2001). Modelling and implicit modelling for predictive control. *International Journal of Control*, *74*(11), 1085–1095.
- Rossiter, J. A. (2002). Predictive Functional Control: More than one way to prestabilise. *IFAC Proceedings Volumes*, *35*(1), 289–294.
- Rossiter, J. A. (2016). Input shaping for PFC: How and why? *Journal of Control and Decision*, *3*(2), 105–118.
- Rossiter, J. A. (2017). A priori stability results for PFC. *International Journal of Control*, *90*(2), 289–297.
- Rossiter, J. A. (2018). *A first course in predictive control*. CRC press.

- Rossiter, J. A., Gossner, J., & Kouvaritakis, B. (1997). Constrained cautious stable predictive control. *IEEE Proceedings on Control Theory and Applications*, 144(4), 313–323.
- Rossiter, J. A., & Haber, R. (2015). The effect of coincidence horizon on Predictive Functional Control. *Processes*, 3(1), 25–45.
- Rossiter, J. A., Kouvaritakis, B., & Cannon, M. (2001). Computationally efficient algorithms for constraint handling with guaranteed stability and near optimality. *International Journal of Control*, 74(17), 1678–1689.
- Rossiter, J. A., Kouvaritakis, B., & Rice, M. (1998). A numerically robust state-space approach to stable-predictive control strategies. *Automatica*, 34(1), 65–73.
- Rossiter, J. A., & Richalet, J. (2002). Handling constraints with Predictive Functional Control of unstable processes. In *Proceedings of the American Control Conference* (Vol. 6, pp. 4746–4751).
- Rossiter, J. A., & Wang, L. (2008). Exploiting Laguerre functions to improve the feasibility/performance compromise in MPC. In *47th IEEE Conference on Decision and Control* (pp. 4737–4742).
- Rossiter, J. A., Wang, L., & Valencia-Palomo, G. (2010). Efficient algorithms for trading off feasibility and performance in predictive control. *International Journal of Control*, 83(4), 789–797.
- Sarah, G., Garna, T., Bouzrara, K., Ragot, J., & Messaoud, H. (2014). Online identification of the bilinear model expansion on Laguerre orthonormal bases. *International Journal of Control*, 87(3), 441–463.
- Satoh, T., Kaneko, K., & Saito, N. (2012). Improving tracking performance of Predictive Functional Control using disturbance observer and its application to table drive systems. *International Journal of Computers Communications & Control*, 7(3), 550–564.
- Scokaert, P. O., Mayne, D. Q., & Rawlings, J. B. (1999). Suboptimal model predictive control (feasibility implies stability). *IEEE Transactions on Automatic Control*, 44(3), 648–654.

- Scokaert, P. O., & Rawlings, J. B. (1998). Constrained Linear Quadratic Regulation. *IEEE Transactions on Automatic Control*, 43(8), 1163–1169.
- Škrjanc, I. (2007). A decomposed-model Predictive Functional Control approach to air-vehicle pitch-angle control. *Journal of Intelligent and Robotic Systems*, 48(1), 115–127.
- Škrjanc, I., & Matko, D. (2000). Predictive Functional Control based on Fuzzy model for heat-exchanger pilot plant. *IEEE Transactions on Fuzzy Systems*, 8(6), 705–712.
- Škultéty, J., Miklovičová, E., & Bars, R. (2013). Predictive synchronous generator excitation control based on Laguerre model. *Journal of Electrical Engineering*, 64(3), 173–179.
- Tatjewski, P. (2007). *Advanced control of industrial processes: Structures and algorithms*. Springer Science & Business Media.
- Utkin, V. I. (1993). Sliding mode control design principles and applications to electric drives. *IEEE transactions on industrial electronics*, 40(1), 23–36.
- Valencia-Palomo, G., & Rossiter, J. (2011). Programmable Logic Controller implementation of an auto-tuned predictive control based on minimal plant information. *ISA Transactions*, 50(1), 92–100.
- Vazquez, S., Leon, J. I., Franquelo, L. G., Rodriguez, J., Young, H. A., Marquez, A., & Zanchetta, P. (2014). Model Predictive Control: A review of its applications in power electronics. *IEEE Industrial Electronics Magazine*, 8(1), 16–31.
- Visioli, A. (2006). *Practical PID control*. Springer Science & Business Media.
- Wahlberg, B. (1991). System identification using Laguerre models. *IEEE Transactions on Automatic Control*, 36(5), 551–562.
- Wang, L. (2001). Continuous time Model Predictive Control design using orthonormal functions. *International Journal of Control*, 74(16), 1588–1600.
- Wang, L. (2004). Discrete Model Predictive Controller design using Laguerre functions. *Journal of Process Control*, 14(2), 131–142.
- Wang, L. (2009). *Model Predictive Control system design and implementation using MATLAB®*. Springer Science & Business Media.

- Wang, R., Li, X., Ahmed, Q., Liu, Y., & Ma, X. (2018). Speed control of a marine engine using Predictive Functional Control based PID controller. In *2018 Annual American Control Conference (ACC)* (pp. 3908–3914).
- Xu, M., Li, S., & Cai, W. (2005). Practical receding-horizon optimization control of the air handling unit in HVAC systems. *Industrial & Engineering Chemistry Research*, *44*(8), 2848–2855.
- Yang, X. Y., Xu, D. P., Han, X. J., & Zhou, H. N. (2005). Predictive Functional Control with modified Elman neural network for reheated steam temperature. In *International Conference on Machine Learning and Cybernetics* (Vol. 8, pp. 4699–4703).
- Yang, X. Y., Yue-Gang, L., Xu, D.-p., & Zhang, X.-F. (2006). Nonlinear Predictive Functional Control based on support vector machine. In *2006 International Conference on Machine Learning and Cybernetics* (pp. 926–929).
- Yiming, S., & Bin, Y. (2012). Predictive Functional cascade temperature control for bench-scale batch reactor. In *31st Chinese Control Conference* (pp. 4079–4084).
- Yongquan, Y., Ying, H., & Bi, Z. (2003). A PID neural network controller. In *Proceedings of the International Joint Conference on Neural Networks* (Vol. 3, pp. 1933–1938).
- Yoon, T.-W., & Clarke, D. W. (1995). Observer design in receding-horizon predictive control. *International Journal of Control*, *61*(1), 171–191.
- Zabet, K., & Haber, R. (2017). Robust tuning of PFC (Predictive Functional Control) based on first-and aperiodic second-order plus time delay models. *Journal of Process Control*, *54*, 25–37.
- Zabet, K., Rossiter, J. A., Haber, R., & Abdullah, M. (2017). Pole-placement Predictive Functional Control for under-damped systems with real numbers algebra. *ISA Transactions*, *71*, 403–414.
- Zhang, J. (2017). Design of a new PID controller using Predictive Functional Control optimization for chamber pressure in a coke furnace. *ISA Transactions*, *67*, 208–214.
- Zhang, J., Rossiter, J., Xie, L., & Su, H. (2018). Predictive Functional Control for integral system. In *Process System Engineering*.

-
- Zhang, R., & Jin, Q. (2018). Design and implementation of hybrid modelling and PFC for oxygen content regulation in a coke furnace. *IEEE Transactions on Industrial Informatics*.
- Zhang, R., Li, P., Ren, Z., & Wang, S. (2009). Combining Predictive Functional Control and PID for liquid level of coking furnace. In *IEEE International Conference on Control and Automation* (pp. 314–318).
- Ziegler, J. G., & Nichols, N. B. (1942). Optimum settings for automatic controllers. *Trans. ASME*, 64(11).

Appendix A

**POLE-PLACEMENT PREDICTIVE
FUNCTIONAL CONTROL FOR
UNDER-DAMPED SYSTEMS WITH REAL
NUMBERS ALGEBRA**

K. Zabet, J. A. Rossiter, R. Haber and M. Abdullah

This paper has been published in:
ISA Transactions 2018, Vol. 71

Declaration form

**POLE-PLACEMENT PREDICTIVE FUNCTIONAL CONTROL FOR UNDER-DAMPED SYSTEMS WITH REAL
NUMBER ALGEBRA**

(ISA transaction 2018, Vol. 71)

Contributions of authors:

K. Zabet

Provided the initial idea of the overall frameworks, formulations, codes, simulations and the first draft of this paper.

R. Haber

Supervised K. Zabet, proofread the paper and become the corresponding author.

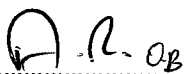
J. A. Rossiter

Supervised M. Abdullah and proofread the paper.

M. Abdullah

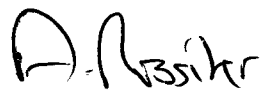
Improved the formulation for Pole-Placement PFC using real number algebra to be more effective which can further reduce the computation burden of this controller. Provided the newly improved coding and implemented it in the simulated case and real laboratory hardware. Wrote and rewrote several sections in the paper.

Signatures:



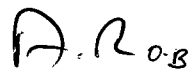
K. Zabet

(First author)



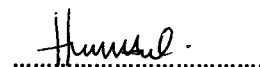
J.A. Rossiter

(Second author)



R. Haber

(Third author)



M. Abdullah

(Forth author)



Pole-placement Predictive Functional Control for under-damped systems with real numbers algebra



K. Zabet^a, J.A. Rossiter^b, R. Haber^{a,*}, M. Abdullah^b

^a Cologne Univ. of Applied Sciences, Inst. of Plant and Process Engineering, Betzdorfer Str. 2, D-50679 Koeln, Germany

^b Dept. of Automatic Control and Systems Eng., University of Sheffield, S1 3JD, UK

ARTICLE INFO

Article history:

Received 8 December 2016

Received in revised form

31 May 2017

Accepted 2 August 2017

Available online 1 September 2017

Keywords:

Predictive Functional Control

Pole-placement

Under-damped system

Real number algebra

ABSTRACT

This paper presents the new algorithm of PP-PFC (Pole-placement Predictive Functional Control) for stable, linear under-damped higher-order processes. It is shown that while conventional PFC aims to get first-order exponential behavior, this is not always straightforward with significant under-damped modes and hence a pole-placement PFC algorithm is proposed which can be tuned more precisely to achieve the desired dynamics, but exploits complex number algebra and linear combinations in order to deliver guarantees of stability and performance. Nevertheless, practical implementation is easier by avoiding complex number algebra and hence a modified formulation of the PP-PFC algorithm is also presented which utilises just real numbers while retaining the key attributes of simple algebra, coding and tuning. The potential advantages are demonstrated with numerical examples and real-time control of a laboratory plant.

© 2017 ISA. Published by Elsevier Ltd. All rights reserved.

1. Introduction

PFC (Predictive Functional Control) [1] is probably the most successful industrial implementation of MPC (Model Predictive Control) based on the numbers and breadth of applications. The main reason for this is relatively simple in that the coding requirements are similar to that for PID (Proportional-Integral-Derivative) controllers and thus the PFC strategy is a competitor with PID rather than more expensive plant wide or system wide approaches. Moreover, it has some advantages over PID in that the tuning mechanism is intuitive being based mainly on a desired time constant (equivalently settling time or convergence rate) and also it embeds a reasonable level of systematic constraint handling using relatively low computational complexity.

Nevertheless, the main weakness of conventional PFC is the same as its strength, that is the relative simplicity [2,3]. Although execution and coding are straightforward for systems with over-damped or simple dynamics, a different picture emerges with systems with less desirable open-loop dynamics [4]. Consequently, although a conventional PFC [1] can work with systems of integrators, open-loop unstable processes and non-minimum-phase characteristics, often the tuning is difficult and the implementation less simple and intuitive. Thus one purpose here is to develop

a modified PFC approach which retains the core attributes of simplicity but more specifically, retains intuitive insight during the design which means the approach is simple for technicians to deploy.

Predictive control algorithm can be calculated by properly planning the manipulated signal sequence via minimizing a cost function. The idea of pole-placement design for predictive control is not new. Pole-placement state-feedback design for optimizing continuous-time predictive control was applied in [5] and extended this algorithm for the constrained case in [6]. GPC (Generalized Predictive control) [7] has two degrees of freedom and allows a design based on pole-placement, see [8] and [9]. Investigations of the stability of PFC for first-order process models [10] were followed by a pole-placement PFC controller recommended for higher-order, over-damped processes in [11].

This paper has a focus on systems with significant under-damped dynamics in the open-loop and first considers the efficacy of a routine PFC implementation. It is demonstrated via a number of examples, that the efficacy is variable which motivates the need for an improved algorithm. Earlier literature has discussed the possibility of shaping the input predictions [4], but although often effective, that approach has the disadvantage of requiring some moderately difficult algebra/coding and there is still a need to fully understand the robustness to uncertainty of such approaches. This paper takes an alternative approach which is to explore and develop a recently proposed alternative the PP-PFC (Pole-Placement PFC) [11]. The main contribution here is to consider the extent to which this approach is suitable for handling under-damped

* Corresponding author.

E-mail addresses: j.a.rossiter@sheffield.ac (J.A. Rossiter), robert.haber@th-koeln.de (R. Haber).

systems. Moreover, as will be seen, a secondary benefit is additional flexibility in the choice of target poles to include mild under-damping; such an option is not available to conventional PFC.

A simplistic implementation of the proposed PP-PFC algorithm for underdamped systems is shown to rely on complex number algebra and this has some possible negative consequences. Firstly, the computational effort is slightly greater, although that could be considered trivial in practice. Secondly however, the requirement for complex number algebra in itself could be a problem as many low level process control units (where PFC would be applied alongside competitor approaches such as PID) do not support complex number algebra. In view of these observations, a second contribution of this paper is to propose algorithms which circumvent the complex number algebra in a relatively simple fashion, thus allowing straightforward coding, maintenance and tuning.

Section 2 will give a basic background on conventional PFC and demonstrate the potential difficulties when applying this to under-damped systems. Section 3 will introduce the pole-placement PFC approach for systems with real poles followed by Section 4 which will discuss how this approach is extended to cope with complex poles, that is under-damped systems. Section 5 will then develop an alternative formulation of PP-PFC which uses just real number algebra. Section 6 gives numerical examples and also some simulations on hardware.

2. Background of PFC

This section gives a brief review of a basic PFC algorithm and demonstrates a normal tuning procedure.

2.1. PFC concepts

The basic principle underlying PFC approaches is that the desired output dynamic is close to that of a first-order response with a specified pole λ . The hope is that if one, recursively at each sample, ensures the prediction of the system behavior is close to the desired dynamic, then the closed-loop behavior is likely to be close to that dynamic. Hence, for a desired steady-state set value of r , a typical target trajectory r^* , expressed in discrete time, takes the form¹:

$$r^*(k) = \frac{(1 - \lambda)z^{-1}}{1 - \lambda z^{-1}} r(k). \tag{1}$$

In the interest of simple computation, PFC differs from more standard MPC approaches in that it uses the prediction at just a single point, the so called coincidence horizon, here denoted by a n_y step ahead prediction. The control law is defined by forcing the system prediction to match the target dynamic of $r^*(k)$ at a point n_y steps ahead, as illustrated in Fig. 1.

In practice, the system output $y_p(k)$ is not beginning from zero, so the target trajectory is one which follows a first-order dynamic from the current point $y_p(k)$ to the correct steady-state, that is:

$$r^*(k + i) = r(k) - \lambda^i [r(k) - y_p(k)], \quad i \geq 1. \tag{2}$$

PFC is defined by forcing coincidence n_y steps ahead and thus the control law is defined from the equality:

$$y_p(k + n_y) = r(k) - \lambda^{n_y} [r(k) - y_p(k)]. \tag{3}$$

¹ In the following the case of a stepwise change in the reference signal is assumed. The same algorithm works for stepwise change in the output additive disturbance, as well.

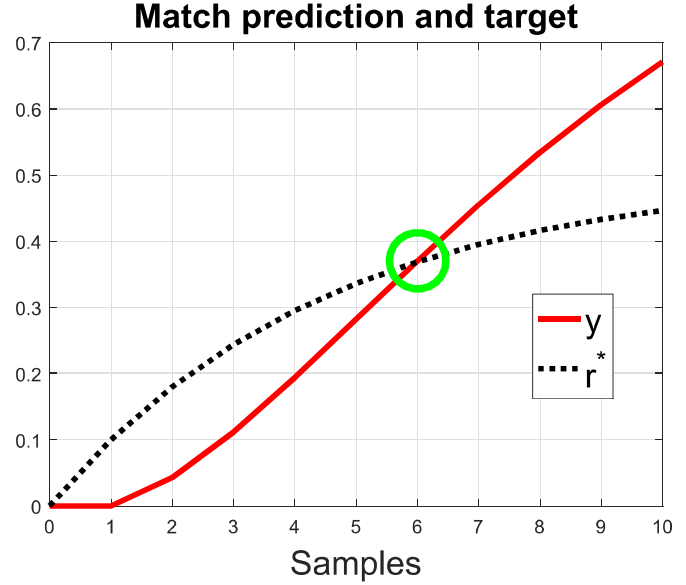


Fig. 1. Illustration of PFC target dynamic r^* and coincidence of the output prediction y_p with target dynamic $n_y = 6$ samples ahead.

Mismatch between process output y_p and model output y_m is assumed constant during the prediction horizon and hence offset-free tracking can be achieved with a minor modification to take account of this bias. The system prediction is given by the model prediction plus an estimated disturbance $d(k)$ (variants of this exist but are not central to the current paper):

$$y_p(k + n_y) = y_m(k + n_y) + d(k), \quad d(k) = y_p(k) - y_m(k). \tag{4}$$

Simplification 1. The n_y steps ahead prediction $y_p(k + n_y)$ depends upon the future choices of control actions. As PFC is premised on being as simple as possible, a typical assumption is that the future inputs remain constant, that is $(u(k + i) = u(k), \quad i \geq 1)$. This has the advantage that only one decision variable is needed so the desired selection to satisfy (3) is straightforward to code (this also applicable with non-linear processes).

Simplification 2. In order to maintain simple coding, PFC overcomes the complexity of prediction algebra by using partial fractions to express the n th-order model $G_m(z)$ as a sum of first-order models [1,2,12] and hence:

$$\left. \begin{aligned} y_m(k) &= G_m(z)u(k), \\ G_m(z) &= \sum_{i=1}^n G_i(z), \end{aligned} \right\} \Rightarrow y_m(k) = \sum_{i=1}^n G_i(z)u(k) = \sum_{i=1}^n \frac{b_i z^{-1}}{1 + a_i z^{-1}} u(k). \tag{5}$$

The effective structure of the model is illustrated in Fig. 2 where G_p represents the real (unknown) process and G_i denote the partial fraction expansion of the assumed model $G_m(z)$. In practice this means that the independent model deployed in PFC code comprises a number of first-order independent models running in parallel; clearly the coding and computation requirement for each is trivial.

The advantage of this parallel formation is that n_y steps ahead predictions can be defined explicitly and without the need for costly or cumbersome prediction algebra [13]. To be precise, the predictions for the model can be expressed as the sum of the predictions of a number of first-order models with component outputs $y_m^{(i)}$, that is:

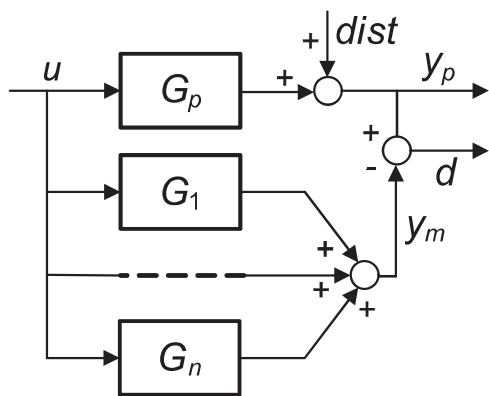


Fig. 2. Parallel model format alongside the actual process G_p .

$$y_m(k + n_y) = \sum_{i=1}^n \left[b_i \frac{1 - (-a_i)^{n_y}}{1 + a_i} u(k) + (-a_i)^{n_y} y_m^{(i)}(k) \right]. \quad (6)$$

Algorithm 1. (PFC) A simple PFC control law can now be constructed by using (3) and prediction (6) in (4). Hence, solve the following for $u(k)$:

$$(1 - \lambda^{n_y})[r(k) - y_p(k)] = \sum_{i=1}^n \left[1 - (-a_i)^{n_y} \right] \left[\frac{b_i}{1 + a_i} u(k) - y_m^{(i)}(k) \right]. \quad (7)$$

Rearrange to determine the input as:

$$u(k) = \frac{(1 - \lambda^{n_y})[r(k) - y_p(k)] + \sum_{i=1}^n \left[1 - (-a_i)^{n_y} \right] y_m^{(i)}(k)}{\sum_{i=1}^n b_i \frac{1 - (-a_i)^{n_y}}{1 + a_i}}. \quad (8)$$

The terms in this law are simple to compute.

Remark 1. This paper does not discuss issues such as ramp targets, system delays and constraints in order to avoid unnecessarily complicated presentation which would distract from the core concepts and contributions presented here. The proposals of this paper carry over to such scenarios in a straightforward fashion. The required modifications are well known in the literature and in fact imply relatively minor changes to the algebra and coding.

2.2. Efficacy of PFC tuning parameters when applied to under-damped systems

This section considers what might be a weakness of PFC which is the underlying motivation for the paper. That is, the main tuning parameter, namely the desired convergence rate λ , is often ineffective and not a good representation of the closed-loop dynamic that results. Clearly this undermines one core selling point and thus should be improved.

Some simple guidance exists for tuning PFC [1–3] but in practice, these methods are underpinned by the requirement to do a form of global search over potential parameters. For a straightforward system and constant targets there are two tuning parameters: (i) the coincidence horizon n_y and (ii) the target closed-loop pole λ . The user can use trial and error over expected reasonable values and choose the pairing that gives them closest to the desired performance. The readers can do this themselves and will find that for many systems the process works well, which is not surprising given the wide spread commercial success of PFC. Specifically, the design procedure is most effective when the process is first-order or heavily damped. However, for other processes, the procedure can be less effective [3,11].

- Fig. 3 shows the possible pole positions for different pairings of

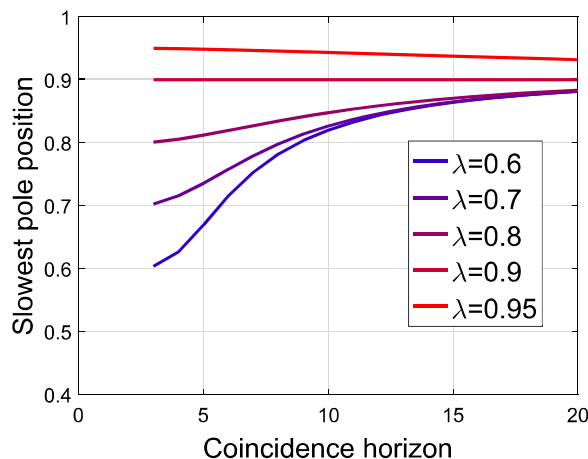


Fig. 3. Illustration of PFC closed-loop poles with different choices of λ and n_y on the over-damped example P_1 . Note $n_y < 3$ gives closed-loop instability.

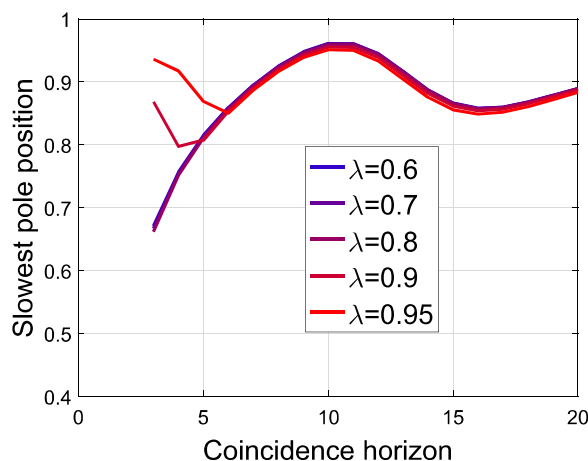


Fig. 4. Illustration of PFC closed-loop poles (absolute values as complex) with different choices of λ and n_y on the under-damped example P_2 .

tuning parameters on an over-damped system $P_1 = (-z^{-1} + 4z^{-2})/(1 - 1.4z^{-1} + 0.45z^{-2})$. It is clear that good pairings exist in that the closed-loop dynamics can be close to the target dynamic and thus a simple PFC design procedure can be effective.

- Fig. 4 shows the possible closed-loop poles with different pairings of tuning parameters for a specific under-damped system $P_2 = (0.4z^{-1} + 0.08z^{-2})/(1 - 1.6z^{-1} + 0.8z^{-2})$. (This process is equivalent to example M , given in (38)). This case is less clear but because while the link between target dynamic and desired dynamic may be achieved for small n_y , tuning is more difficult because the responses are quite sensitive to the choice of coincidence horizon. This inconsistency of result for different n_y could be worrying.
- Fig. 5 shows a different under-damped and non-minimum-phase example N (given in (39)). In this case it is not easily possible to find a good pairing of parameters. Worse still, it is clear the system is closed-loop unstable for nearly all reasonable choices and thus in this case, PFC would be a potentially unsafe approach.

3. Pole-placement PFC

The previous section has demonstrated that the nominal PFC algorithm of (8) may be ineffective for systems with difficult dynamics and more specifically, that the role of the tuning parameter

λ can be weak [3]. In view of this, some recent work [11] considered a minor modification with the aim of making the tuning more effective and thus having real physical meaning to potential users so that they can use it intuitively, as was also intended.

This section will give a quick review of the proposed modification, the so called PP-PFC approach.

3.1. PFC with a first-order model

PFC has been particularly effective in industry partially because many real systems have dynamics which are close to first-order and it is easy to show [3] that for a first-order system, the PFC tuning parameters work perfectly, as long as one uses a coincidence horizon of one. In other words, the target pole λ becomes the closed-loop pole exactly in the nominal case $y_m = y_p = y$.

- For a first-order model with $n_y = 1$, the control law (8) is given as follows:

$$\left. \begin{aligned} y(k+1) &= b_1 u(k) - a_1 y(k), \\ y(k+1) &= (1-\lambda)r(k) + \lambda y(k), \end{aligned} \right\} \Rightarrow u(k) = \frac{(1-\lambda)r(k) + (a_1 + \lambda)y(k)}{b_1}. \quad (9)$$

- Rearranging and substituting the corresponding control action back into the system dynamics gives:

$$\begin{aligned} y(k+1) &= b_1 \frac{(1-\lambda)r(k) + (a_1 + \lambda)y(k)}{b_1} - a_1 y(k) \\ &= (1-\lambda)r(k) + \lambda y(k). \end{aligned} \quad (10)$$

From which it is clear that the closed-loop behavior is represented by a first-order model with unity gain (no steady-state offset) and the desired pole λ .

3.2. Pole-placement PFC

The main motivation for PP-PFC algorithm is to exploit the efficacy of PFC for first-order systems in order to propose an equally simple process that will work on higher-order systems as it is known (Section 2.2) that tuning for higher order systems [3] is not nearly so straightforward or effective in general.

The key concept within the proposal is to treat each submodel G_i shown in Fig. 2 as if it had an independent input and then deploy a nominal PFC algorithm to compute what that input should be in order to achieve some specified dynamic, say pole ρ_1 . The next core concept is to exploit linearity and linear combinations. The algorithm takes a linear combination of all the proposed inputs to determine the desired input to the real system. By utilizing a sensible constraint (that the partial contributions of each individual inputs sum to unity), it is easy to show that the desired dynamic is then achieved in the nominal case $d(k) = y_p(k) - y_m(k) = 0$.

Algorithm 2. (PP-PFC): The PP-PFC algorithm for achieving a target closed-loop pole comprises the following steps.

1. Define targets for each individual submodel G_i based on the model steady-state gains of output $y_m^{(i)}$ (5) using the formulae:

$$r^{(i)}(k) = \frac{\gamma_i}{\sum_{j=1}^n \gamma_j} r(k); \quad \gamma_i = \frac{b_i}{1 + a_i}. \quad (11)$$

2. Identify proposed inputs for each submodel ($i = 1, \dots, n$) using the control law (9):

$$\begin{aligned} u^{(i)}(k) &= \frac{(1-\rho_1)[r^{(i)}(k) - d^{(i)}(k)] + (a_i + \rho_1)y_m^{(i)}(k)}{b_i}; \\ d^{(i)}(k) &= \frac{\gamma_i d(k)}{\sum_{j=1}^n \gamma_j}. \end{aligned} \quad (12)$$

3. Form a linear combination of these inputs to determine the process input as:

$$u(k) = \sum_{i=1}^n \beta_i u^{(i)}(k); \quad \sum_{i=1}^n \beta_i = 1. \quad (13)$$

Next we demonstrate that the desired pole ρ_1 is achieved before discussing how the remaining freedom in β_i might be used.

Lemma 1. The control law of (12), (13) ensures that the target pole ρ_1 becomes a closed-loop pole in the nominal case (thus $d(k) = 0$).

Proof. The control law (13) is presented by using z^{-1} as the shifting time operator, $z^{-1}x(k) = x(k-1)$, and using (5), (11) and (12):

$$\begin{aligned} u(k) &= \sum_{i=1}^n \beta_i \frac{(1-\rho_1)r^{(i)}(k) + (a_i + \rho_1)y_m^{(i)}(k)}{b_i} \\ &= K_c r(k) + \sum_{i=1}^n \beta_i \frac{(\rho_1 + a_i)}{b_i} \frac{b_i z^{-1}}{1 + a_i z^{-1}} u(k); \end{aligned}$$

where, $K_c = (1-\rho_1) \sum_{i=1}^n \frac{\beta_i \gamma_i}{b_i \sum_{j=1}^n \gamma_j}$. (14)

Rearranging this it is clear the characteristic polynomial of the closed-loop poles $p_c(z)$ has ρ_1 as a root:

$$\begin{aligned} \{p_c(z) = 0\} &\equiv \left\{ 1 - \sum_{i=1}^n \beta_i \frac{(\rho_1 + a_i)z^{-1}}{1 + a_i z^{-1}} = 0 \right\} \\ p_c(\rho_1) &= 1 - \sum_{i=1}^n \beta_i \frac{\rho_1 + a_i}{\rho_1 + a_i} = 1 - \sum_{i=1}^n \beta_i = 0. \end{aligned} \quad (15)$$

□

It is important that a sensible choice is made for the values of β_i as, while any choice satisfying (13) will give the desired closed-loop pole, the choice made also has an impact on the other closed-loop poles. Indeed, the remaining flexibility in the values of β_i can be used to assign the other closed-loop poles at values $\rho_i, i = 2, \dots, n$ using a partial fraction by the following definitions [11].

$$\beta_j = \frac{\prod_{i=2}^n (a_j + \rho_i)}{\prod_{i=1, i \neq j}^n (a_j - a_i)}, \quad \forall j = 1, 2, \dots, n. \quad (16)$$

Theorem 1. Using the choice of β_i in (16) results in all the poles $\rho_i, i = 2, \dots, n$ becoming closed-loop poles.

Proof. The overall implied control law is given as:

$$\begin{aligned} u(k) &= \sum_{i=1}^n \beta_i u^{(i)}(k) = (1-\rho_1) \sum_{i=1}^n \frac{\beta_i r^{(i)}(k)}{b_i} + \sum_{i=1}^n \beta_i \frac{(a_i + \rho_1)y_m^{(i)}(k)}{b_i} \\ &= \frac{1-\rho_1}{\sum_{j=1}^n \gamma_j} \sum_{i=1}^n \frac{\beta_i \gamma_i r(k)}{b_i} + \sum_{i=1}^n \beta_i \frac{(a_i + \rho_1)z^{-1}}{1 + a_i z^{-1}} u(k). \end{aligned} \quad (17)$$

Substituting in from (15) and (17):

$$(1 - \rho_1) \sum_{i=1}^n \frac{\beta_i \gamma_i}{b_i} = \prod_{i=1}^n \frac{1 - \rho_i}{1 + a_i} \Rightarrow$$

$$u(k) = \frac{r(k)}{\sum_{j=1}^n \gamma_j} \left[\prod_{i=1}^n \frac{1 - \rho_i}{1 + a_i} + \left[1 - \prod_{i=1}^n \frac{1 - \rho_i z^{-1}}{1 + a_i z^{-1}} \right] u(k) \right]. \quad (18)$$

The implied characteristic polynomial of the closed-loop poles is given as:

$$\{p_c = 0\} \equiv \left[\prod_{i=1}^n \frac{1 - \rho_i z^{-1}}{1 + a_i z^{-1}} \right] = 0. \quad (19)$$

From this it is clear that ρ_i are the closed-loop poles. □

Remark 2. The stability of PP-PFC is guaranteed in the nominal case as a natural corollary of [Theorem 1](#) whereby the positions of the poles are all known and have to be selected to be inside the unit circle.

4. Extending PP-PFC to systems with complex poles

This section forms a main contribution of this paper which is to extend PP-PFC to systems with under-damped modes. The significance of this change is because the partial fraction expansion implicit in (5) will lead to complex poles and residues, and in turn this means that the control laws of (13) imply complex inputs. In the first instance there is a need to consider whether the use of complex numbers is important or indeed whether PP-PFC is still effective and simple to design and implement.

The reader should note a core point which is that, if the PP-PFC algorithm continues to work effectively with under-damped modes, then it solves a tuning challenge for conventional PFC as tuning for [Algorithm 1](#) can be a significant challenge in the presence of oscillatory predictions. For simplicity this presentation will assume just a single pair of complex poles; this is reasonable as PFC would rarely be used on very high-order models given that low-order models usually capture the core dynamics. Moreover, notwithstanding this, the results will automatically carry over anyway.

4.1. Partial fraction expansion with complex coefficients

Consider a model $G_m(z)$ which has roots at $-a_1, -a_2, \dots, -a_n$ with a_1, a_2 a complex conjugate pair. A partial fraction expansion of $G_m(z)$ into first-order terms is:

$$G_m(z) = \frac{n(z)}{(1 + a_1 z^{-1})(1 + a_2 z^{-1}) \dots (1 + a_n z^{-1})} = \sum_{i=1}^n \frac{b_i z^{-1}}{1 + a_i z^{-1}}. \quad (20)$$

It is noted that the residues b_1, b_2, \dots, b_n will be complex conjugates.

4.2. PFC law for a process with complex coefficients

A quick review of the previous section will reveal that none of the algebra required numbers to be purely real and the algebra and pole computations should equally apply for complex numbers. The obvious consequence is that a system with complex coefficients should still be amenable to the PP-PFC control law of (13). In fact, the only requirement that needs careful checking is that the input $u(k)$ to be implemented to the real process must be real.

Lemma 2. Consider the submodel $G_i(z) = \frac{b_i z^{-1}}{1 + a_i z^{-1}}$ where both b_i, a_i are complex and find the corresponding control law using (9). The implied output dynamics must follow the desired first-order trajectory with dynamic λ .

Proof. This is already evident from (10) in Section 3. However, closer inspection reveals that the corresponding input signal $u^{(i)}$ is not real due to the presence of $r^{(i)}, b_i, a_i$ in the law [definition \(12\)](#). Nevertheless, as this is a simulation model, not a real process, that issue is not important. □

Lemma 3. Notwithstanding the fact that the implied input $u^{(i)}(k)$ is complex, nevertheless applying a control law which utilises $\beta_1 u^{(1)}(k) + \beta_2 u^{(2)}(k)$ as defined in (13) will result in a real input as long as $\beta_2 = \beta_1^*$ (means complex conjugate).

Proof. The overall implied control law associated to a pair of complex poles is given as:

$$u(k) = \beta_1 u^{(1)}(k) + \beta_1^* u^{(1)*}(k)$$

$$= (1 - \rho_1) \left[\frac{\beta_1 r^{(1)}(k)}{b_1} + \frac{\beta_1^* r^{(1)*}(k)}{b_1^*} \right] + \beta_1 \frac{(a_1 + \rho_1) y_m^{(1)}(k)}{b_1}$$

$$+ \beta_1^* \frac{(a_1^* + \rho_1^*) y_m^{(1)*}(k)}{b_1^*}. \quad (21)$$

Here all the terms are complex conjugates and hence the resulting term $u(k)$ is real. It should also be remarked that the condition that $\sum \beta_i = 1$ implies that $\beta_2 = \beta_1^*$. □

Lemma 4. Notwithstanding the fact that the implied input could be complex, nevertheless applying a control law as defined in (13) will result all the desired closed-loop poles being achieved, even when ρ_i are defined as complex numbers.

Proof. This follows automatically from [Lemma 3](#) as algebra is not affected by the use or not of complex numbers. □

Theorem 2. Notwithstanding the fact that the implied input $u^{(i)}(k)$ is complex, nevertheless applying a control law which utilises $\sum_{i=1}^n \beta_i u^{(i)}(k)$ as defined in (13) will result in a real input as long as β_i are calculated based on (16), irrespective of the choices of ρ_i .

Proof. The core difference in this proof is to allow complex choices for the poles and showing that all the desired poles are achieved while retaining a real input. The overall implied control law is given as:

$$u(k) = \sum_{i=1}^n \beta_i u^{(i)}(k) = (1 - \rho_1) \sum_{i=1}^n \frac{\beta_i r^{(i)}(k)}{b_i} + \sum_{i=1}^n \beta_i \frac{(a_i + \rho_1) y_m^{(i)}(k)}{b_i}$$

$$= \frac{1 - \rho_1}{\sum_{j=1}^n \gamma_j} \sum_{i=1}^n \frac{\beta_i \gamma_i r^{(i)}(k)}{b_i} + \sum_{i=1}^n \beta_i \frac{(a_i + \rho_1) z^{-1}}{1 + a_i z^{-1}} u(k). \quad (22)$$

Substituting from (19):

$$(1 - \rho_1) \sum_{i=1}^n \frac{\beta_i \gamma_i}{b_i} = \prod_{i=1}^n \frac{1 - \rho_i}{1 + a_i} \Rightarrow$$

$$u(k) = \frac{r(k)}{\sum_{j=1}^n \gamma_j} \prod_{i=1}^n \frac{1 - \rho_i}{1 + a_i} + \left[1 - \prod_{i=1}^n \frac{1 - \rho_i z^{-1}}{1 + a_i z^{-1}} \right] u(k). \quad (23)$$

Again it is clear that any terms appear in complex conjugate pairs. □

Remark 3. This section has proved that the desired closed-loop poles of ρ_i are achieved for any choices of desired poles and any

open-loop poles, irrespective of whether they are complex or real. In all cases, the proposed control law of (13) produces a real input. However, it is emphasised that the underlying signals implied in the independent model of Fig. 2 will be complex numbers and as this model is retained in the control law implementation, it assumes that complex number algebra is supported by the operating system.

It is worth repeating that a key benefit of PP-PFC as opposed to conventional PFC is that the user can now guarantee the behavior of the nominal closed-loop and achieve the desired dominant dynamics. This section has shown that a simplistic implementation of PP-PFC on systems with under-damped dynamics is effective.

5. Implementable PP-PFC using real numbers algebra

The main weakness of PP-PFC as presented in the previous section is the reliance on complex number algebra. However, many operating systems used to implement control do not support complex number algebra. Consequently there is a need to develop an alternative implementation which uses only real number algebra.

Two alternative implementations are developed in this section: i) handling the real and imaginary components explicitly and ii) a formulation of the algorithm avoiding complex numbers altogether. Readers should note that the case of target poles ρ_i being complex is also included as this gives the designer extra flexibility which can be useful, and this is a novel contribution to the PFC field.

5.1. Calculating real and imaginary parts separately

For complex numbers expressed in Cartesian coordinates, the real and imaginary parts can be handled with real number algebra as follows. In this method, each component of the complex numbers (real and imaginary part) is calculated separately for example, consider $x = \text{Re}\{x\} + j\text{Im}\{x\}$ and $y = \text{Re}\{y\} + j\text{Im}\{y\}$, then:

$$xy = \underbrace{[\text{Re}\{y\}\text{Re}\{x\} - \text{Im}\{y\}\text{Im}\{x\}]}_{\text{real part}} + j \underbrace{[\text{Re}\{y\}\text{Im}\{x\} + \text{Im}\{y\}\text{Re}\{x\}]}_{\text{imaginary part}} \tag{24}$$

Lemma 5. *The update equation of independent model $G_m \Rightarrow y_m^{(i)}(k) = b_i u(k-1) - a_i y_m^{(i)}(k-1)$ can be handled using the following two separate computations.*

$$\begin{aligned} \text{Re}\{y_m^{(i)}(k)\} &= \text{Re}\{-a_i\text{Re}\{y_m^{(i)}(k-1)\} - \text{Im}\{-a_i\}\text{Im}\{y_m^{(i)}(k-1)\} + \text{Re}\{b_i\}u(k-1) \\ \text{Im}\{y_m^{(i)}(k)\} &= \text{Re}\{-a_i\}\text{Im}\{y_m^{(i)}(k-1)\} + \text{Im}\{-a_i\}\text{Re}\{y_m^{(i)}(k-1)\} + \text{Im}\{b_i\}u(k-1). \end{aligned} \tag{25}$$

Lemma 6. *Only the real part of the term $\beta_i u^{(i)}(k)$ needs to be computed.*

$$\begin{aligned} \text{Re}\{\beta_i u^{(i)}(k)\} &= \text{Re}\left\{ (1 - \rho_1) \frac{\beta_i \gamma_i}{b_i \sum_{j=1}^n \gamma_j} \right\} r(k) + \text{Re}\left\{ (a_i + \rho_1) \frac{\beta_i}{b_i} \right\} \text{Re}\{y_m^{(i)}(k)\} \\ &\quad - \text{Im}\left\{ (a_i + \rho_1) \frac{\beta_i}{b_i} \right\} \text{Im}\{y_m^{(i)}(k)\}. \end{aligned} \tag{26}$$

Proof. It was established in Theorems (2) and (3) that $u(k)$ is real and therefore all the imaginary terms must cancel out and therefore need not be computed. □

Theorem 3. *Compared to PP-PFC using complex algebra, the increase in computational demand using real number algebra is inconsequential although the coding is slightly more involved.*

Proof. As is clear from (25), (26), the PP-PFC calculation of the real and imaginary parts of the actual submodels output required 10 mathematical operations on real numbers (summation and multiplication) with 4 reserved places for variables in addition to the variable $u(k)$, and the calculation of the real parts of $\beta_i u^{(i)}(k) + \beta_{i+1} u^{(i+1)}(k)$ required 5 operations on real numbers with one reserved place for the variable in addition to the $r(k)$ variable. In comparison, the PFC of (17) uses the same memory space and 11 operations on complex numbers but in truth the difference is so small that on modern computing it has small relevance. □

Remark 4. When the desired closed-loop pole ρ_1 is real (β_i are complex conjugates), the calculation of the real and imaginary parts of $y_m^{(i)*}$ can be omitted because, by inspection, these are known from $y_m^{(i)}$.

5.2. New formulation of PP-PFC algorithm using real numbers algebra

The main concept deployed next is to exploit the structure in the independent model of Fig. 2 in order to reduce the control law to an even simpler final form. Ironically, there is a partial move away from the partial fraction expansion in first-order terms to the final implementation so that the implied partial fractions are all real, although the full decomposition structure is still implicit in the control law design.

This section deploys a number of lemmata and theorems which are required to establish the final result. The reader may like to note that a key focus in many of these is to identify when terms are real or appear in complex conjugate pairs, and when they do not, so that this information can be exploited efficiently in any code. The idea is to look carefully at the computation required for each term in (22), (26).

5.2.1. Real system poles

First consider the parts of (22), (26) linked to real system poles.

Lemma 7. *The parameter β_i related to a real system pole a_i has real value if the target pole ρ_1 is real, otherwise it has complex value.*

Proof. This follows from the fact that β_i in (16) have complex values in conjugate pairs when ρ_1 is real, otherwise when ρ_1 is complex, then β_i will contain the complex ρ_1^* and thus not be in conjugate pairs. □

Lemma 8. *The parameter $(a_i + \rho_1)\beta_i$ related to a real system pole a_i has a real value irrespective of whether the target pole ρ_1 is real or complex.*

Proof. Considering (16) the parameter $(a_i + \rho_1)\beta_i$ contains the complex values in conjugate pairs.

$$(a_i + \rho_1)\beta_i = \frac{\prod_{j=1}^n (a_i + \rho_j)}{\prod_{j=1; j \neq i}^n (a_i - a_j)} \tag{27}$$

□

Theorem 4. *The real value of the proposed weighted input signal $\text{Re}\{\beta_i u^{(i)}\}$ for the submodel having real pole a_i comprises numerous*

components which can be computed off-line and stored.

$$\text{Re}\{\beta_i u^{(i)}\} = \text{Re}\left\{ \left(1 - \rho_1\right) \frac{\beta_i \gamma_i}{b_i \sum_{j=1}^n \gamma_j} \right\} r(k) + (a_i + \rho_1) \frac{\beta_i}{b_i} y_m^{(i)}(k). \tag{28}$$

Proof. This is obvious in that several of the terms above do not change.

$$K_{0,i} = \text{Re}\left\{ \left(1 - \rho_1\right) \frac{\beta_i \gamma_i}{b_i \sum_{j=1}^n \gamma_j} \right\}; \quad K_{1,i} = (a_i + \rho_1) \frac{\beta_i}{b_i};$$

$$\text{Re}\{\beta_i u^{(i)}\} = K_{0,i} r(k) + K_{1,i} y_m^{(i)}(k). \tag{29}$$

□

Remark 5. The coefficient $K_{0,i}$ is automatically real when the target pole ρ_1 is real.

5.2.2. Complex system poles

Next the paper considers the parts of (21), (26) linked to complex conjugate pairs of poles in $G(z)$.

Lemma 9. The one-step-ahead prediction models for the summed outputs of G_1, G_2 and the output of $G_{1,2} = G_1 + G_2$ must match, assuming the inputs into each are the same. This means the complex states of G_1, G_2 can be inferred from the real states of $G_{1,2}$.

Proof. This is by inspection following linearity.

$$G_{1,2} = G_1 + G_2 = \frac{b_1 z^{-1}}{1 + a_1 z^{-1}} + \frac{b_2 z^{-1}}{1 + a_2 z^{-1}}$$

$$= \frac{B_1 z^{-1} + B_2 z^{-2}}{1 + (a_1 + a_2) z^{-1} + a_1 a_2 z^{-2}} y_m^{(1,2)}(k+1) = B_1 u(k)$$

$$+ B_2 u(k-1) - (a_1 + a_2) y_m^{(1,2)}(k) - (a_1 a_2) y_m^{(1,2)}(k-1) \tag{30}$$

$$\left. \begin{aligned} y_m^{(1)}(k+1) &= b_1 u(k) - a_1 y_m^{(1)}(k) \\ y_m^{(2)}(k+1) &= b_2 u(k) - a_2 y_m^{(2)}(k) \end{aligned} \right\} \Rightarrow y_m^{(1,2)}(k+1) = y_m^{(1)}(k+1) + y_m^{(2)}(k+1). \tag{31}$$

In consequence, ignoring the dependence on the term $u(k)$ which is yet to be determined, one can write that:

$$-a_1 y_m^{(1)}(k) - a_2 y_m^{(2)}(k) = B_2 u(k-1) - (a_1 + a_2) y_m^{(1,2)}(k) - (a_1 a_2) y_m^{(1,2)}(k-1)$$

$$y_m^{(1)}(k) + y_m^{(2)}(k) = y_m^{(1,2)}(k). \tag{32}$$

Therefore, given they are conjugates, the values $y_m^{(1)}, y_m^{(2)}$ can be inferred from these simultaneous equations (noting that in both the imaginary parts are zero by definition).

$$2 \begin{bmatrix} -\text{Re}\{a_1\} & \text{Im}\{a_1\} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \text{Re}\{y_m^{(1)}(k)\} \\ \text{Im}\{y_m^{(1)}(k)\} \end{bmatrix}$$

$$= \begin{bmatrix} B_2 u(k-1) - 2\text{Re}\{a_1\} y_m^{(1,2)}(k) - a_1 a_1^* y_m^{(1,2)}(k-1) \\ y_m^{(1,2)}(k) \end{bmatrix} \Rightarrow \begin{bmatrix} \text{Re}\{y_m^{(1)}(k)\} \\ \text{Im}\{y_m^{(1)}(k)\} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & \text{Im}\{a_1\} \\ 1 & \text{Re}\{a_1\} \end{bmatrix} \begin{bmatrix} B_2 u(k-1) - 2\text{Re}\{a_1\} y_m^{(1,2)}(k) - a_1 a_1^* y_m^{(1,2)}(k-1) \\ y_m^{(1,2)}(k) \end{bmatrix}. \tag{33}$$

□

Lemma 10. The parameters β_i, β_{i+1} related to complex conjugate

poles a_i and a_{i+1} are complex conjugates if the target pole ρ_1 is real, otherwise β_i, β_{i+1} are not complex conjugates.

Proof. From (16) both β_i and β_{i+1} are complex conjugates if ρ_1 is real, otherwise if ρ_1 is complex then both β_i and β_{i+1} are not conjugate pairs. □

Lemma 11. The parameters $(a_i + \rho_1)\beta_i$ and $(a_{i+1} + \rho_1)\beta_{i+1}$ related to a complex conjugate pair of poles a_i and a_{i+1} , are complex conjugates irrespective of whether the target pole ρ_1 is real or complex.

Proof. Considering (16), all the terms appear in conjugate pairs.

$$(a_i + \rho_1)\beta_i = \frac{\prod_{j=1}^n (a_i + \rho_j)}{\prod_{j=1, j \neq i}^n (a_i - a_j)};$$

$$(a_{i+1} + \rho_1)\beta_{i+1} = \frac{\prod_{j=1}^n (a_{i+1} + \rho_j)}{\prod_{j=1, j \neq i+1}^n (a_{i+1} - a_j)}. \tag{34}$$

□

Lemma 12. The real value of the proposed weighted input signal $\text{Re}\{\beta_i u_i(k) + \beta_{i+1} u_{i+1}(k)\}$ for the two submodels having complex conjugated poles a_i and a_{i+1} comprises numerous components which can be computed off-line and stored.

$$\text{Re}\{\beta_i u^{(i)}(k) + \beta_{i+1} u^{(i+1)}(k)\} = \text{Re}\left\{ \frac{1 - \rho_1}{\sum_{j=1}^n \gamma_j} \left[\frac{\beta_i \gamma_i}{b_i} + \frac{\beta_{i+1} \gamma_{i+1}}{b_{i+1}} \right] \right\} r(k)$$

$$+ 2\text{Re}\left\{ (a_i + \rho_1) \frac{\beta_i}{b_i} \right\} \text{Re}\{y_m^{(i)}(k)\}$$

$$- 2\text{Im}\left\{ (a_i + \rho_1) \frac{\beta_i}{b_i} \right\} \text{Im}\{y_m^{(i)}(k)\}. \tag{35}$$

Proof. This is obvious in that several of the terms above do not change.

$$K_{0,i} = \text{Re}\left\{ \frac{1 - \rho_1}{\sum_{j=1}^n \gamma_j} \left[\frac{\beta_i \gamma_i}{b_i} + \frac{\beta_{i+1} \gamma_{i+1}}{b_{i+1}} \right] \right\};$$

$$K_{1,i} = 2\text{Re}\left\{ (a_i + \rho_1) \frac{\beta_i}{b_i} \right\};$$

$$K_{2,i} = -2\text{Im}\left\{ (a_i + \rho_1) \frac{\beta_i}{b_i} \right\};$$

$$\text{Re}\{\beta_i u^{(i)}(k) + \beta_{i+1} u^{(i+1)}(k)\} = K_{0,i} r(k) + K_{1,i} \text{Re}\{y_m^{(i)}(k)\} + K_{2,i} \text{Im}\{y_m^{(i)}(k)\}. \tag{36}$$

□

Remark 6. The coefficient $K_{0,i}$ is automatically real when the target pole ρ_1 is real ($\beta_i = \beta_{i+1}^*$).

Theorem 5. The proposed common input signal $\text{Re}\{\beta_i u^{(i)}(k) + \beta_{i+1} u^{(i+1)}(k)\}$ for the two submodels having complex conjugate poles can be simplified to a second-order control law which is based solely on real number algebra and using the states of the second-order model $G_{i,i+1}$.

Proof. This follows from substitution of (33) into (36).

Table 1
Computational loading for different realizations of PP-PFC.

PP-PFC with complex algebra	PP-PFC with calculating real/imag. parts	PP-PFC with real algebra
11 operations	15 operations	14 operations

$$y_m^{(i,i+1)}(k+1) = B_{1,i}u(k) + B_{2,i}u(k-1) - 2\text{Re}\{a_i\}y_m^{(i,i+1)}(k) - a_i a_i^* y_m^{(i,i+1)}(k-1);$$

$$\text{Re}\{\beta_i u^{(i)}(k) + \beta_{i+1} u^{(i+1)}(k)\} = K_0^{(i,i+1)} r(k) + K_1^{(i,i+1)} y_m^{(i,i+1)}(k) + K_2^{(i,i+1)} y_m^{(i,i+1)}(k-1) + K_3^{(i,i+1)} u(k-1); K_0^{(i,i+1)} = K_{0,i};$$

$$K_1^{(i,i+1)} = \frac{K_{1,i} \text{Im}\{a_i\} - K_{2,i} \text{Re}\{a_i\}}{2 \text{Im}\{a_i\}};$$

$$K_2^{(i,i+1)} = -\frac{K_{2,i} a_i a_i^*}{2 \text{Im}\{a_i\}};$$

$$K_3^{(i,i+1)} = \frac{K_{2,i} B_{2,i}}{2 \text{Im}\{a_i\}}. \tag{37}$$

□

5.2.3. Computational load comparisons

Only the component of the control law corresponding to pairs of complex poles needs to use the formulation of (37). The contribution of submodels with real poles can use the simpler formulation of (29). From (37), the new formulated PP-PFC calculation of the actual second-order submodels output requires 7 mathematical operations on real numbers with 4 reserved places for variables in addition to the $u(k)$ variable, and the calculation of the real parts of $\beta_i u^{(i)}(k) + \beta_{i+1} u^{(i+1)}(k)$ requires 7 operations on real numbers with one reserved place for the variable in addition to the $r(k)$ variable. A simplified comparison of the alternative approaches is given in Table 1.

6. Numerical examples

This section will give some numerical examples to compare the simulation times of the control (as an indicator to the simplicity of the control action calculation) using classical PP-PFC, PP-PFC with real and imaginary parts calculation, and the new formulated PP-PFC algorithm, for various choices of ρ on two under-damped examples M, N :

$$M = \frac{0.4z^{-1} + 0.08z^{-2}}{1 - 1.6z^{-1} + 0.8z^{-2}}; \tag{38}$$

$$N = \frac{-0.66z^{-1} + 0.08z^{-2} + 0.6z^{-3}}{1 - 2.72z^{-1} + 2.626z^{-2} - 0.8924z^{-3}}. \tag{39}$$

M has poles at $-0.8 \pm 0.4j$. The choice N matches the example used in Fig. 5 which conventional PFC could not handle and has poles at $-0.9, -0.9 \pm 0.4j$. The open-loop step responses are plotted in Figs. 6 and 7, respectively.

As it is seen from the step responses, process M is of type minimum-phase and process N of type non-minimum-phase. Process M is of second-order, has a gain of 2.4 and a damping factor $\zeta = 0.234$ which causes an overshoot in the open-loop step response of about 45%. Process N is of third-order and has a gain of 1.47. The open-loop step response shows an undershoot of about -700% and an overshoot of 374%. Both oscillating processes were selected for illustration the new control algorithm as they are difficult to control.

The average simulation time of repeated 100 simulations for each case is considered in the computational loading results. Moreover, the reader will notice the additional advantage of the

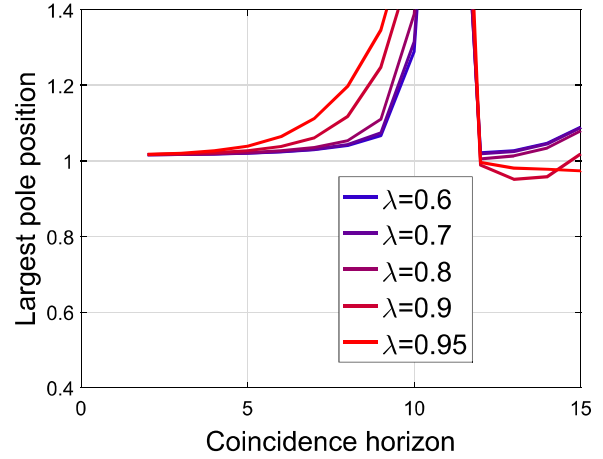


Fig. 5. Illustration of PFC closed-loop poles (closed-loop instability) with different choices of λ and $n_y=3$ on the under-damped example N .

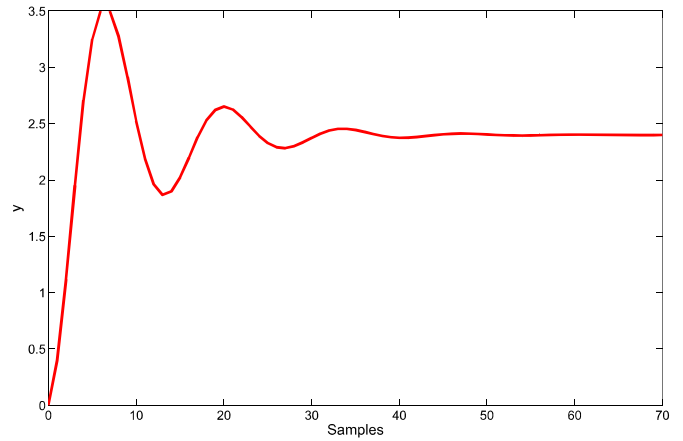


Fig. 6. Step response of system M .

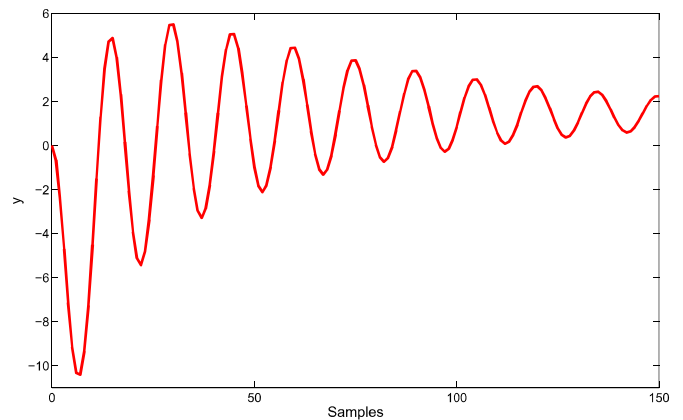


Fig. 7. Step response of system N .

proposed approach which is the ability to select a target pole as being complex which is not something that is possible in conventional PFC; such an option is reasonable in many cases where a small overshoot allows better behavior overall.

In the following simulations a stepwise change in the reference signal and the disturbance acting at the process output, as shown in Fig. 2, are applied. The algorithm can also compensate for disturbances acting at the process input, but this case is not shown here.

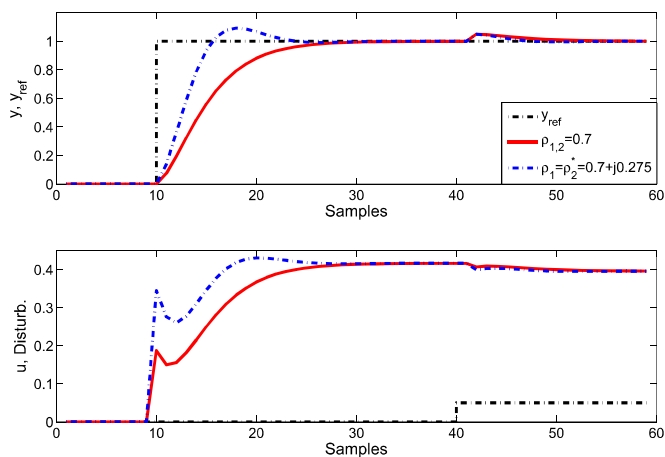


Fig. 8. Illustration of PP-PFC performance with different choices of ρ on the under-damped example M .

6.1. Example 1: PP-PFC of example M

The PP-PFC simulation of example M is given in Fig. 8 for various choices of desired closed-loop pole ρ . An output disturbance is added around the 40th sample to demonstrate the disturbance compensating ability of the approach. It is clear that the proposed algorithm has given effective control and moreover, the tuning parameter ρ has retained an intuitive link to the resulting closed-loop behavior as expected. Moreover, it is demonstrated that one can select the target pole as being complex, unlike for conventional PFC. Nevertheless, in this case a conventional PFC can also give effective control although the link to the desired λ (defined based on the dominant poles of the simulations in Fig. 8) is weaker (see Fig. 9). For interest, the reader should note that both the values of β_i have a real part of 0.5 as expected (as $\sum \beta_i = 1$), but also have a non-zero imaginary part.

The simulation times are set in Table 2. The results show that the new formulated PP-PFC have fastest control action calculations, and the PP-PFC using complex algebra have slowest control action calculations.

6.2. Example 2: PP-PFC of example N

The PP-PFC simulation of example N is given in Fig. 10. An output disturbance is added around the 70th sample. Despite the obviously very challenging dynamics of this process, the PP-PFC algorithm has given smooth control to the required target and

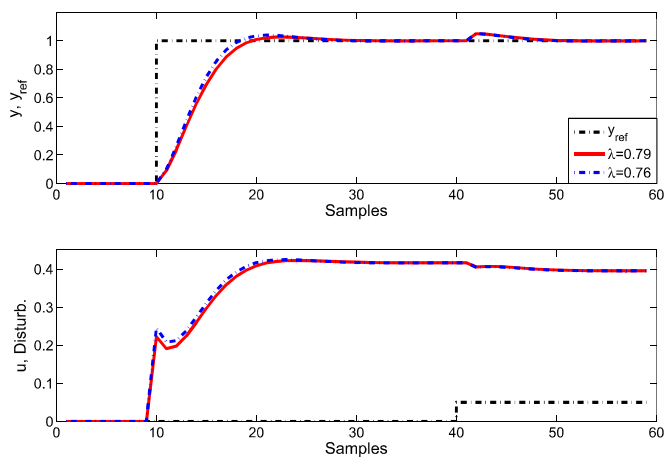


Fig. 9. Illustration of conventional PFC performance with different choices of λ on the under-damped example M .

Table 2
Relative simulation times of the different realizations of PP-PFC for example M .

ρ	PP-PFC with complex algebra	PP-PFC with calculating real/imag. parts	PP-PFC with real algebra
$\rho_{1,2} = 0.7$	100%	74%	48%
$\rho_{1,2} = 0.7 \pm j0.275$	100%	77%	48%

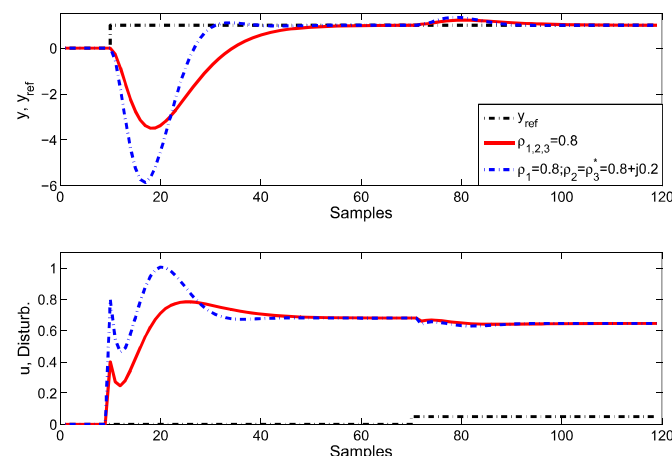


Fig. 10. Illustration of new formulated PP-PFC performance with different choices of ρ on the under-damped example N .

moreover, as desired, has maintained the intuitive link between the target dynamic ρ and the closed-loop convergences speed. Conversely, classical PFC is very sensitive to the choice of n_y and gives stable behavior only for a small range of large n_y which in effect makes the parameter λ redundant, as is clearly seen in Fig. 11; the plots are almost identical irrespective of the choice of λ and hence only relatively slow λ can be achieved.

The simulation times are set in Table 3. Also here, the results show that the proposed formulation of PP-PFC has the fastest calculations, and the PP-PFC using complex algebra has the slowest calculations.

6.3. Example 3: Constraint handling of PP-PFC for example N

For completeness, this section demonstrates that constraint handling can be embedded also in the PP-PFC algorithm in a conventional PFC manner without detriment to performance

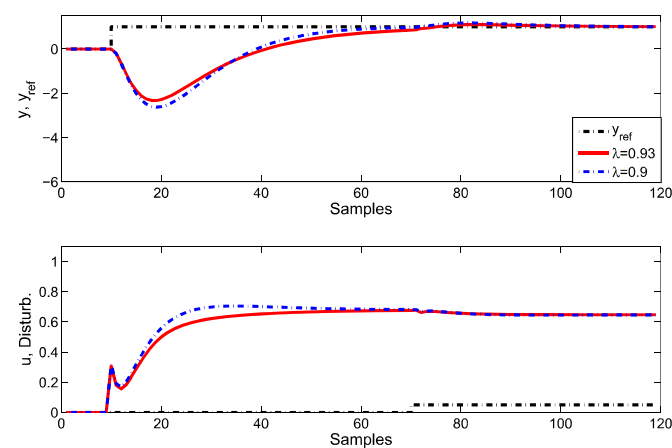


Fig. 11. Illustration of conventional PFC performance with different choices of λ on the under-damped example N .

Table 3
Relative simulation times of the different realizations of PP-PFC for example N .

ρ	PP-PFC with complex algebra	PP-PFC with calculating real/imag. parts	PP-PFC with real algebra
$\rho_{1,2,3} = 0.8$	100%	45%	42%
$\rho_1 = 0.8$	100%	41%	37%
$\rho_{2,3} = 0.8 \pm j0.2$			

beyond the inevitable loss of some performance when constraints are active. The control law is summarized as follows [11]:

1. Test whether the proposed controller output satisfies plant input absolute and rate constraints. If not, modify $u(k)$ to ensure both using saturation.
2. To ensure satisfaction of output/state constraints one must form the implied predictions over a sensible but large horizon and modify $u(k)$ as required to ensure satisfaction. This reduces to a simple *for loop* which ensures that maximum or minimum of $y_p(k+i)$ is within limits.

The constrained $u(k)$ has to be applied in the model prediction.

Fig. 12 shows the controlled and manipulated variable plots of example N with an input rate maximum limit of 0.1 per sample and an absolute maximum input limit of 0.8. As can be seen, the constraints have been handled effectively.

7. Real-time control

In this section, the proposed controller is implemented with a real laboratory hardware. This process poses its own challenges such as the measured data and the controller model may differ in value and can lead to a failure if it is not addressed properly. Other than that, the computation time of the controller need to be faster than the sampling time to avoid any delay when updating the output value. In this work, a Quanser SRV02 servo based unit powered by a Quanser VoltPAQ-X1 amplifier with a flexible joint is used as a plant. This system is operated by National Instrument ELVIS II+ multifunctional data acquisition. The plant is connected to a computer via USB connection using NI LabVIEW software as shown in Fig. 13.

The flexible joint base is mounted on the load gear of the SRV02 system. The servo angle θ together with its link will increase

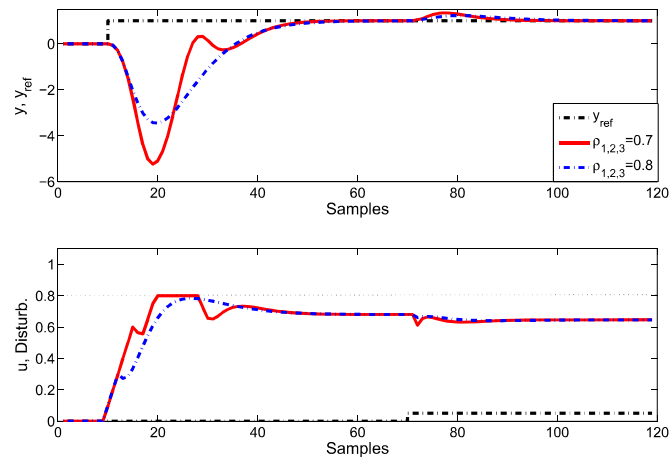


Fig. 12. Illustration of PP-PFC performance of the under-damped example N considering input constraints.

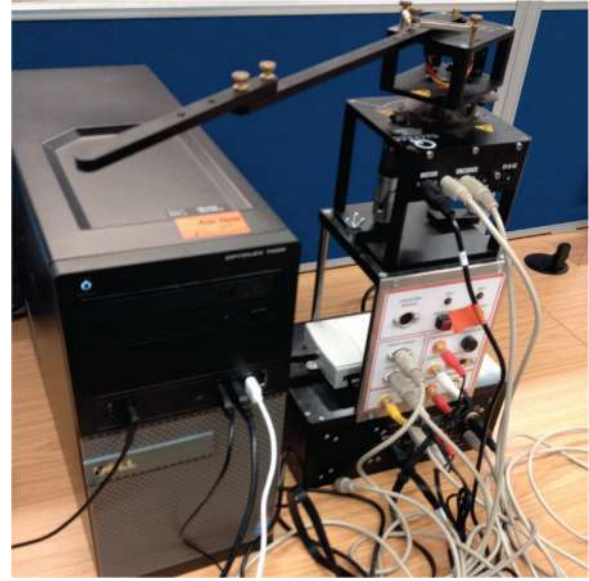


Fig. 13. The experimental plant.

positively in counter-clockwise (CCW) rotation when the supplied voltage is positive ($V_m > 0$). The same situation applied to the link deflection angle α with CCW rotation. Both $\theta(t)$ and $\alpha(t)$ are measured in radians. Fig. 14 shows a schematic of the flexible joint system [14] where the servo motor voltage V_m is acting as a control variable that generates a torque τ at the load gear to rotate the flexible joint base. On the other hand, the viscous friction coefficient of the servo B_{eq} will oppose the applied torque at the servo load gear and the friction acting on the link is denoted by the viscous damping coefficient B_l . The overall flexible joint system is assumed linear with a spring stiffness K_s .

The main objective for this task is to track the angular speed of the servo $\dot{\theta}(t)$ by manipulating the supplied voltage $V_m(t)$. The general mathematical model of the process is given as (for more details see [15]):

$$\begin{aligned} \ddot{\theta}(t) &= \frac{K_s}{J_{eq}} \alpha(t) - \frac{B_{eq}}{J_{eq}} \dot{\theta}(t) + \frac{1}{J_{eq}} \tau(t) \\ \ddot{\alpha}(t) &= -K_s \left(\frac{J_l + J_{eq}}{J_l J_{eq}} \right) \alpha(t) + \frac{B_{eq}}{J_{eq}} \dot{\theta}(t) - \frac{1}{J_{eq}} \tau(t) \\ \tau(t) &= \frac{\eta_g K_g \eta_m k_t (V_m(t) - K_g k_m \dot{\theta}(t))}{R_m} \end{aligned} \quad (40)$$

where the list and value of each corresponding SRV02 parameter used are given in Table 4. By substituting the parameter value and manipulating the algebraic equation, the control model for the plant is reduced to:

$$\begin{aligned} \ddot{\theta}(t) &= 619.05 \alpha(t) - 34.70 \dot{\theta}(t) + 61.07 V_m(t) \\ \ddot{\alpha}(t) &= -1015.62 \alpha(t) + 32.78 \dot{\theta}(t) - 61.07 V_m(t) \end{aligned} \quad (41)$$

The model in (41) is converted to a discrete-time transfer function with sampling time 0.02 s to get a direct relationship between the angular speed and input voltage as:

$$\frac{\dot{\theta}(z)}{V_m(z)} = \frac{0.8451z^{-1} - 1.556z^{-2} + 0.8457z^{-3}}{1 - 2.17z^{-1} + 1.753z^{-2} - 0.4997z^{-3}} \quad (42)$$

Fig. 15 shows the open-loop behavior of the plant and the mathematical model based on the voltage input profile. It is clear that both of the outputs exhibit under-damped behavior due to the extended joint attachment in the servo motor where there

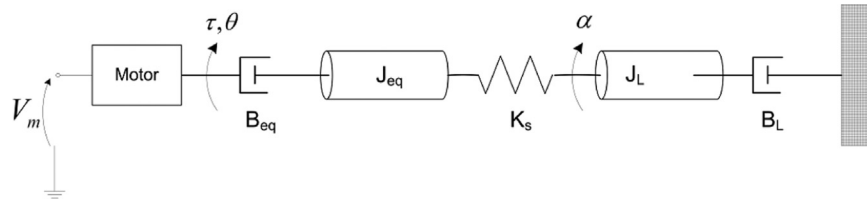


Fig. 14. Rotary flexible joint model [15].

Table 4
SRV02 servo parameters specification [14,15].

Parameters	Value
Gearbox efficiency, η_g	0.9
High-gear total gear ratio, K_g	70
Motor efficiency, η_m	0.69
Motor current-torque constant, k_t	7.68×10^{-3} Nm/A
Motor back-emf constant, k_m	7.68×10^{-3} V/(rad/s)
Motor armature resistance, R_m	2.6Ω
Spring stiffness, K_s	0.5 N/m
Viscous friction coefficient, B_{eq}	0.004 Nm/(rad/s)
Moment of inertia without external load, J_{eq}	2.08×10^{-3} kg m ²
Total moment of inertia of the arm, J_l	1.9×10^{-3} kg m ²

exist an oscillation before converging to the steady-state value. It is noted that there is a large parameter mismatch between the measured and model outputs. However, this discrepancy can be handled by the independent model structure of PFC algorithm (4).

The proposed algorithm is employed with different selections of poles according to the desired settling time. Fig. 16 demonstrates the capability of the new PP-PFC controller to track an alternating set point between -1 rad/s and 1 rad/s. The same performance as in the previous simulation is obtained. The controller managed to provide a smooth tracking to the desired target and while, retaining the intuitive link between the target dynamic ρ and the closed-loop convergence speed.

Generally, the implementation of this controller is very straightforward as it does not need any complex arithmetic compared to the traditional approach. Hence, it can be easily implemented on a low-cost hardware such as PLC (Programmable Logic Controller). In addition, the use of unit coincidence horizon simplifies both the tuning and coding processes which makes it more transparent and attractive compared to the conventional PID controller.

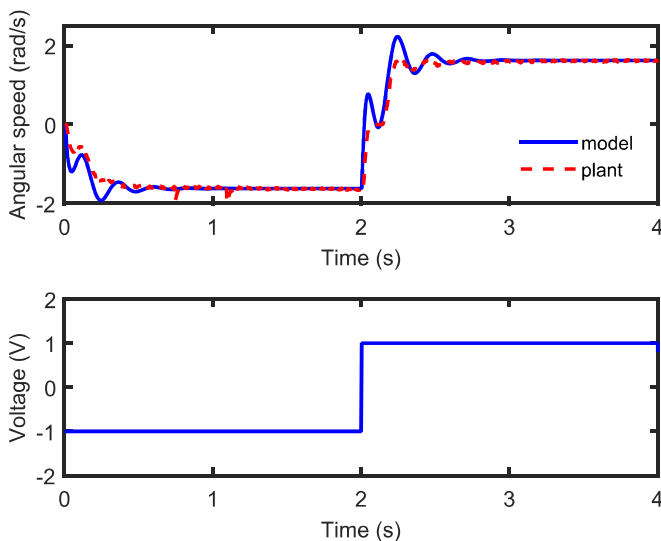


Fig. 15. The open-loop behavior of plant and model based on the supplied voltage.

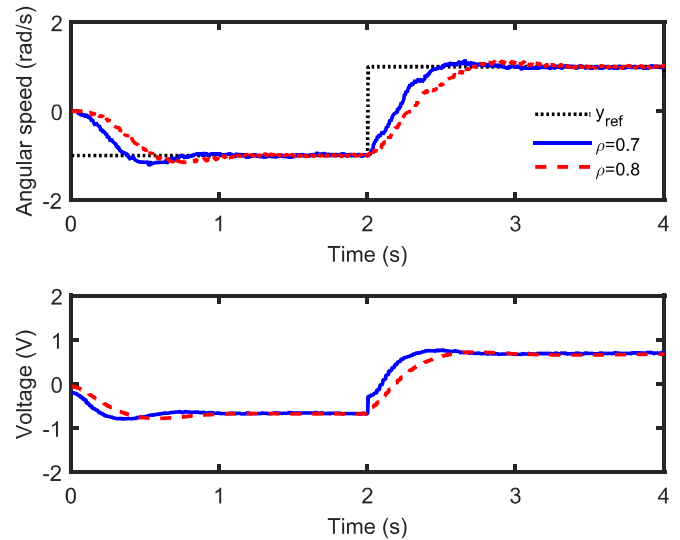


Fig. 16. Illustration of PP-PFC performance of the under-damped Quanser servo attached with flexible joint.

8. Conclusion and future work

This paper has proposed a new approach to PFC for systems with under-damped open-loop dynamics. In many cases a conventional PFC approach is difficult to tune with open-loop oscillatory dynamics and thus loses important features such as simplicity and intuition. This paper shows that by building on the partial fraction expansion commonly used in industrial PFC code to form model predictions, one can make use of the powerful results for first-order systems and apply these for high-order systems even where the partial fraction expansion gives complex residues (under-damped systems). Critically, the overall coding complexity and requirements are similar to the code of the conventional PFC but a core advantage is that the tuning options are now more straightforward than with a conventional algorithm. In fact, it is shown additionally that one is now able to select the target closed-loop pole to be complex and this is often advantageous compared to the restriction to real poles with conventional PFC.

The proposed new formulation of the PP-PFC algorithm (for systems with under-damped open-loop dynamics) reduces the calculation efforts in comparison to the conventional PP-PFC formulation because of dealing with real numbers only. A further advantage is that PP-PFC can be used in programmable logic controllers or decentralized control systems which usually do not support complex algebra; a simulation demonstration on hardware was presented.

Future work aims to look more closely at the allocation of the values β_i . There is a need to consider more carefully how these extra degrees of freedom can be utilised most effectively, while not increasing the complexity of the approach. Finally, there is also a need to compare this approach more formally with the shaping approach [4].

It is also noted that while the current approach will deal with some level of parameter uncertainty, a formal sensitivity analysis and design remains as future work.

References

- [1] Richalet J, O'Donovan D. *Predictive functional control: principles and industrial applications*. Springer Science & Business Media; 2009.
- [2] Haber R, Bars R, Schmitz U. Ch. 11: Predictive control in process engineering: from the basics to the applications, Ch. 11: Predictive functional control, Wiley-VCH (2011) 437–465.
- [3] Rossiter JA, Haber R. The effect of coincidence horizon on predictive functional control. *Processes* 2015;3(1):25–45.
- [4] Rossiter J. Input shaping for PFC: how and why?. *J. Control Decis* 2016;3(2):105–18. <http://dx.doi.org/10.1080/23307706.2015.1083408>.
- [5] Gawthrop P, Ronco E. Predictive pole-placement control with linear models. *Automatica* 2002;38:421–32.
- [6] Chen W-H, Gawthrop PJ. Constrained predictive pole-placement control with linear models. *Automatica* 2006;42:613–8.
- [7] Clarke DW, Mohtadi C, Tuffs PS. Generalized predictive control. Part I. The basic algorithm. *Automatica* 1987;23(2):137–48.
- [8] Wang W, Mao ZZ. Generalized pole-placement adaptive control algorithm and its convergence analysis. *Sel Top Model Control* 1995;2:874–8.
- [9] Fikar M, Unbehauen H, Mikles J. Design of a predictive controller based on pole-placement. In: *Proceedings of the 3rd European control conference 4* (2004) 131–135.
- [10] 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* (2005) 174–179.
- [11] Rossiter J, Haber R, Zabet K. Pole-placement Predictive Functional Control for over-damped systems with real poles. *ISA Trans* 2016;61:229–39.
- [12] 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):229–39.
- [13] Rossiter J. Notes on multi-step ahead prediction based on the principle of concatenation. In: *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 207(4) (1993) 261–263.
- [14] Quanser user manual flexible joint experiment set-up and configuration, Quanser Inc., 2012.
- [15] Apkarian J, Karam P, Levis M. *Instructor workbook set-up flexible joint experiment for LabView users*. Quanser Inc; 2012.

Appendix B

UTILISING LAGUERRE FUNCTION IN PREDICTIVE FUNCTIONAL CONTROL TO ENSURE PREDICTION CONSISTENCY

M. Abdullah and J. A. Rossiter

This paper has been published in:

Proceedings of the 11th UKACC International Conference on Control 2016

Declaration form

UTILISING LAGUERRE FUNCTION IN PREDICTIVE FUNCTIONAL CONTROL TO ENSURE PREDICTION
CONSISTENCY

(Proceeding of the 11th International Conference on Control 2016)

Contributions of authors:

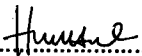
M. Abdullah

Provided the initial idea, formulations, codes, simulations and draft of this paper.

J. A. Rossiter

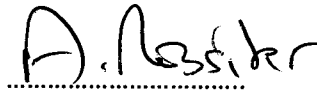
Supervised M. Abdullah and proofread the paper.

Signatures:



M. Abdullah

(First author)



J.A. Rossiter

(Second author)

Utilising Laguerre Function in Predictive Functional Control to Ensure Prediction Consistency

Muhammad Abdullah* and John Anthony Rossiter[†]

^{*†}Department of Automatic Control and System Engineering,
University of Sheffield, Mappin Street, S1 3JD, UK.

Email: MAbdullah2@sheffield.ac.uk* and j.a.rossiter@sheffield.ac.uk[†]

^{*}Department of Mechanical Engineering,

International Islamic University Malaysia, Jalan Gombak, 53100, Kuala Lumpur Malaysia.

Email: mohd_abdl@iium.edu.my

Abstract—This work proposes the use of Laguerre function in Predictive Functional Control (PFC) to produce well-posed decision making. The constant control input assumption of a classical PFC is replaced with the Laguerre polynomial and the steady state input of a system. With this slight modification, better consistency between model predictions and an actual system behaviour is achieved. In addition, the effectiveness of desired closed-loop time constant as PFC tuning parameter becomes more significant. The coding and tuning processes of the proposed approach are very straightforward and in line with the key selling points of PFC.

Keywords—Predictive Control, PFC, Laguerre Function

I. INTRODUCTION

Advancement in modern computation has triggered the usage of Model Predictive Control (MPC) in many applications such as automotive [1], chemical [2], and others. This controller offers a systematic and effective framework to handle system constraints and delays [3]. However, an MPC application is more expensive compared to other traditional controllers as it needs higher computational loads and coding complexity for prediction and optimization processes. Besides, in a real industrial application, staff often prefer to use a simpler controller such as PID rather than predictive control due to their limited understanding and experience [4].

On the other hand, Predictive Functional Control (PFC) is a simplified version of MPC [5]. Currently, this controller is widely adopted in scenarios traditionally dominated by PID due to its simplicity and ease of tuning. PFC computes the manipulated input of a system with a simplified cost function. It forces the system output to match the desired target trajectory at a specific horizon. Due to the constant future input assumption, PFC coding becomes simple and consequently, it only uses minimal computation and indeed for low order models the coding is almost trivial. The use of settling time or closed-loop time constant as a tuning parameter makes the designing process more transparent, especially to industrial users. PFC also retains similar advantages to more classical MPC algorithms such as the systematic handling of constraints and delays control problems. Taken together, these advantages explain the very wide industrial application of PFC [6].

Nevertheless, although PFC is simple and user-friendly, it is underpinned by a decision making process which at times is

ill-posed. For example, earlier work has shown that there can be a large mismatch between the open-loop predictions used for decision making and the actual closed-loop behaviour that results, especially when using small horizons [4]. Conversely, with large horizons, better prediction consistency is usually achieved, but the effectiveness of closed-loop time constant as the tuning parameter becomes less significant. This is mainly due to the constant future input assumption of PFC and so there is insufficient degrees of freedom (d.o.f.) to ensure prediction consistency when anything other than open-loop behaviour is wanted.

For systems with challenging dynamics, PFC practitioners have proposed several modifications to the default algorithm [7]. However, this leads to a more complicated PFC implementation as one needs to engage both with the selection and tuning of the required modification. One alternative way to tackle difficult dynamics is via shaping the predicted input so that it will converge smoothly to its steady state value [8]. The proposed modification is expected to give a better prediction consistency which will be useful for higher order systems and complex control problems.

Motivated by Laguerre MPC [9], [10], this paper aims to explore the potential benefits of applying Laguerre functions within a PFC framework. In MPC, this orthonormal function is utilised to enhance the horizon effect [10]. In essence a large horizon input prediction can be obtained with a small number of parameters/d.o.f.. This can reduce the implied computational burden dramatically and improve the controller performance [9], [11] where longer input horizons are needed. Moreover, the Laguerre function can capture the desired system dynamics to some extent in that its own convergence is linked to its pole which is a design parameter.

In PFC, Laguerre function is usually used as a black box model to represent a plant [12]. A high order system can be modelled with a low order Laguerre network. This will reduce the controller computation demand significantly. However, the idea of this work is to replace the PFC constant input prediction with an alternative assumption based around a Laguerre polynomial. The specific idea is to embed two dynamics within the predicted input, that is both the expected steady-state and also a deviation which converges to steady state value

with dynamics accorded to the selected Laguerre pole. This modification will give a well-posed solution, in addition to enhancing the effectiveness of close-loop time constant as a PFC tuning parameter. Section 2 provides a brief description of nominal PFC framework and formulation. Section 3 explains the modification of PFC to utilise a Laguerre function. Section 4 discusses various numerical examples. Section 5 presents the conclusion and future work.

II. NOMINAL PFC FORMULATION

This section reviews the basic framework of nominal PFC where the objective is to track a step target. In order to give focus to the key conceptual contribution of this paper, the fine details of the algebra to ensure offset correction and integral action is excluded in the formulation although obviously applied in all the numerical examples. This exclusion does not affect the validity of the analysis and tuning presented. Integral action is usually embedded in PFC using an independent model formulation; predictions are corrected using an offset term between the output of the independent model and the actual process. This technique also ensures offset-free tracking in case of small levels of uncertainty [4].

A. Target Trajectory

PFC design is based on simple human concepts. One estimates a required control action according to how fast one expects/desires the output to approach the target. More precisely, in PFC the n-step ahead system prediction $y_{k+n|k}$ is forced to move closer to the target R than the current output y_k , with a constant future input assumption $u_{k+i|k} = u_k, i = 0, \dots, n$. The target trajectory is derived based on a desired closed-loop pole of λ . The mathematical formulation of this objective is met by the predictions satisfying an equality, k samples ahead:

$$y_{k+n|k} = R - (R - y_k)\lambda^n \quad (1)$$

There are two main tuning parameters of PFC. The first is the coincidence horizon n , defined as the coincidence point at which one desires the system prediction and the target trajectory to coincide. The second is the closed loop pole λ which links to 95% of the desired settling time (often denoted as TRBF in the industry: $\lambda = e^{T/TRBF}$, T is the sampling time). An interested reader can find more detailed explanations of PFC theory and concepts in the references, e.g. [4]–[6].

B. Model Prediction and PFC Control Law

PFC utilises the n-step ahead prediction of a linear mathematical model to compute the manipulated input. In this paper and without loss of generality, a general transfer function model is used to represent the system. For such models the prediction algebra is very well known in the literature (e.g. [3]). Hence, here only the key results are provided for the sake of conciseness. For inputs u_k and outputs y_k , the n-step ahead unbiased linear model prediction is formulated as:

$$y_{k+n|k} = H_n u_k + P_n u_{k-1} + Q_n y_k \quad (2)$$

where parameters H_n, P_n, Q_n depend on the model parameters and for a model of order m :

$$\underline{u}_k = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+n-1} \end{bmatrix}; \underline{u}_{k-1} = \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \\ u_{k-m} \end{bmatrix}; \underline{y}_k = \begin{bmatrix} y_k \\ y_{k-1} \\ \vdots \\ y_{k-m} \end{bmatrix} \quad (3)$$

To compute the control input, the prediction (2) is substituted directly into (1) to obtain:

$$H_n u_k + P_n u_{k-1} + Q_n y_k = R - (R - y_k)\lambda^n \quad (4)$$

Using the constant future input prediction assumption and thus defining $h_n = \sum(H_n)$ the PFC control law reduces to:

$$\underline{u}_k = \frac{R - (R - y_k)\lambda^n - (P_n u_{k-1} + Q_n y_k)}{h_n} \quad (5)$$

Remark 1: It is convenient to define the expected value u_{ss} of constant input which will lead to no steady-state offset. For a given model and disturbance estimate the computation of this is straightforward [3].

III. LAGUERRE PFC FORMULATION

The reader is reminded of the underlying ethos behind this proposal. For a system to have other than open-loop behaviour, the input signal must include some dynamics. Therefore, in order for the input prediction to be closely matched to the desired closed-loop behaviour, then the input prediction must also have dynamics. However, the default PFC assumption is that the future input is constant, that is, it has no dynamics and therefore the default predictions can only match open-loop behaviour. The intention here is to put some dynamics into the input predictions and thus enhance the prediction capabilities of PFC enabling a matching with a greater range of desired closed-loop behaviours. This, in turn, should enable more effective and systematic tuning.

In this first subsection, the basic principle of Laguerre function and how it is used to shape the input prediction are explained briefly. The latter subsection will look at its implementation within PFC framework to generate new control law. The key focus of this formulation is to keep the coding, computational load and interpretation as simple as possible as these features are key selling points of PFC.

A. Input Parametrization using Laguerre Function

The discrete Laguerre polynomial is derived from the discretisation of continuous Laguerre function [10]. This orthonormal function is widely used in system identification due to its ability to capture the system behaviour with fewer parameters. The z-transform of discrete Laguerre polynomials are defined as:

$$L_i(z) = \sqrt{1-a^2} \frac{(z^{-1} - a)^{i-1}}{(1 - az^{-1})^i} \quad (6)$$

where a is the Laguerre pole which depends on a user selection, and i denotes the order of Laguerre polynomial. To ensure stability, the pole should be in the range of $0 < a < 1$

[10]. Although Laguerre polynomials of high order may be used in MPC [10], [11], here only a first order Laguerre polynomial is employed to shape the input prediction to ensure a smooth convergence. The first order Laguerre function, with modified scaling for simplicity, can be expressed in vector form as:

$$L_1(z) = \frac{1}{1 - az^{-1}} \equiv 1 + az^{-1} + a^2z^{-2} + \dots \quad (7)$$

Define $L_1 = [1, a, a^2, \dots, a^{n-1}]^T$. Now we are in a position to define the input prediction to be deployed in PFC.

Theorem 1: A future input parameterised as

$$u(z) = \frac{u_{ss}}{1 - z^{-1}} + \frac{\eta}{1 - az^{-1}} \quad (8)$$

will give output predictions which settle at the desired steady-state. η represents a degree of freedom.

Proof: The signal defined in (8) has the property that

$$\lim_{k \rightarrow \infty} u_k = u_{ss} \quad (9)$$

This is obvious as the definition of $L_1(z)$ shows that the components converge to zero asymptotically. By definition (see Remark 1), u_{ss} is such that:

$$\lim_{k \rightarrow \infty} u_k = u_{ss} \Rightarrow \lim_{k \rightarrow \infty} y_k = R \quad \square \quad (10)$$

As the Laguerre polynomial evolves over the horizon, it will converge to the steady state input with respect to its pole a . Noting that (8) is equivalent to $\underline{u}_k = L_1\eta + u_{ss}$, the output prediction is modified by substituting it into (2):

$$y_{k+n|k} = H_n L_1 \eta + h_n u_{ss} + P_n \underline{u}_k + Q_n y_k \quad (11)$$

B. Laguerre PFC Control Law

This section defines the PFC law using the Laguerre polynomial to shape the input predictions.

Algorithm 1: Define the output prediction n -steps ahead using (11). The PFC law is defined by substituting this prediction into (1) and solving for the d.o.f. η .

$$\eta = \frac{R - (R - y_k)\lambda^n - (P_n \underline{u}_k + Q_n y_k) - h_n u_{ss}}{H_n L_1} \quad (12)$$

Due to the receding horizon principle [10] and the definition of $L_1(z)$, the current input is defined as:

$$u_k = u_{ss} + \eta \quad (13)$$

Remark 2: It is worth noting that with this control law structure, it is straightforward to add some constraint handling capabilities, that is, modify η as necessary to ensure that the input predictions meet constraints. This method will only require a simple loop which is common practice in industrial PFC implementations [4].

IV. NUMERICAL EXAMPLES

In this section, various numerical examples (a first order system, a second order system and a third order non-minimum phase system) are presented to show the benefit of the proposed Laguerre PFC algorithm. The simulation will focus on the tuning process and the concept of well-posed decision making, that is consistency between the predictions and the actual closed-loop behaviour that results.

For each case, two figures are plotted to represent the open-loop and closed-loop input and output of the alternative controllers (1,12). The implied prediction at a given sample is denoted by subscript p (input u_p and output y_p). While for the actual closed-loop behaviour, it is expressed with subscript s (input u_s and output y_s). The target trajectory is used to represent the signal r instead of steady state target R .

The desired closed-loop pole λ and the Laguerre pole a are both set to 0.7 for all the cases. In the last subsection, the effect of varying Laguerre pole is discussed briefly. For reference, the steady state input of a system is plotted as u_{ss} . The key sign for well-posed decision making is when the prediction and behaviour are similar or relatively close to each other. This indicates that the controller only needs a minimum effort to change the system input from one sample to the next. Conversely, a drastic system input change indicates inconsistent foundation within a control law which could lead to unpredictable and undesirable effects.

A. First Order System

The first order system is:

$$G_1 = \frac{0.25z^{-1}}{1 - 0.8z^{-1}} \quad (14)$$

A suitable coincidence horizon is selected based on the coincidence point between the open loop prediction and the desired target trajectory [4]. From Fig. 1, it is clear that $n = 4$ is the suitable choice. Fig. 2 shows the performance of nominal PFC and Laguerre PFC (LPFC). The upper figure represents the output behaviour and the lower figure illustrates the input behaviour. Both of the controllers manage to track the set point R asymptotically. However, nominal PFC produces some inconsistency between the implied prediction $y_p(PFC)$ and the actual closed-loop behaviour $y_s(PFC)$. The output prediction continues to rise above the desired target due to the nominal constant input assumption. Although the actual behaviour is close to the target trajectory r , the control input $u_s(PFC)$ requires under actuation from one sample to the next as it differs from the predicted one $u_p(PFC)$. By applying a Laguerre polynomial, the consistency between output prediction $y_p(LPFC)$ and actual behaviour $y_s(LPFC)$ is improved dramatically. This shows that the controller is producing well-posed decision where the input prediction $u_p(LPFC)$ and the system input $u_s(LPFC)$ coincide with each other and converge smoothly to the system steady state input u_{ss} . In addition, with the Laguerre function, the effectiveness of λ as the tuning parameter becomes more significant compared to

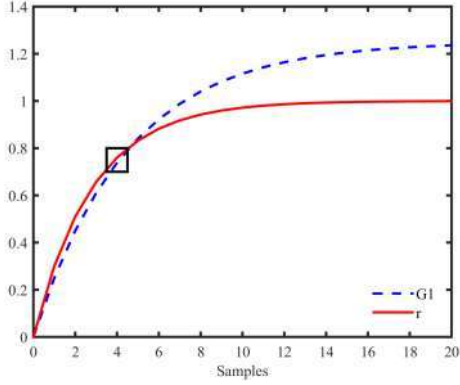


Fig. 1. 1st order step response $G1$ with target trajectory r .

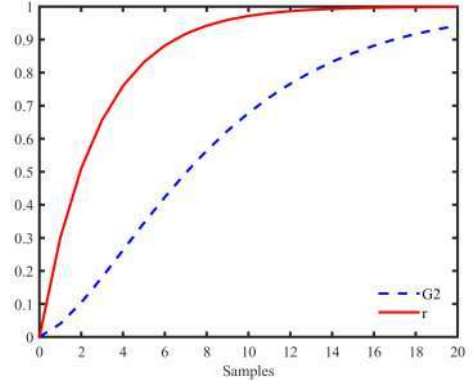


Fig. 3. 2nd order step response $G2$ with target trajectory r .

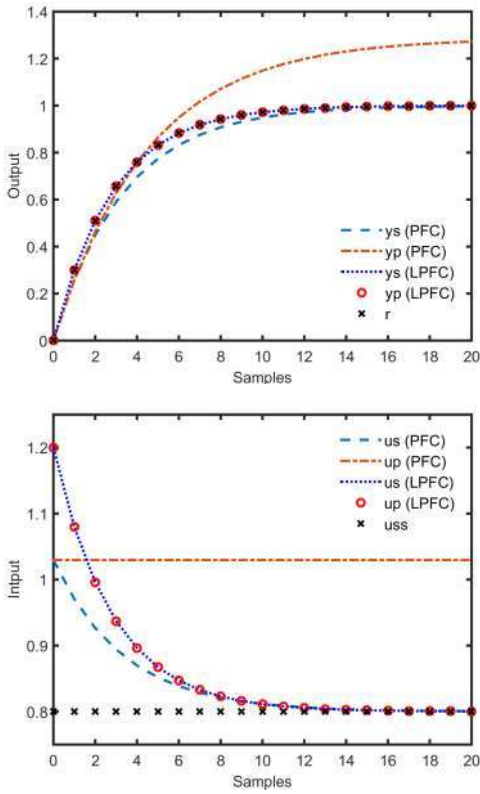


Fig. 2. 1st order system response for nominal PFC and Laguerre PFC.

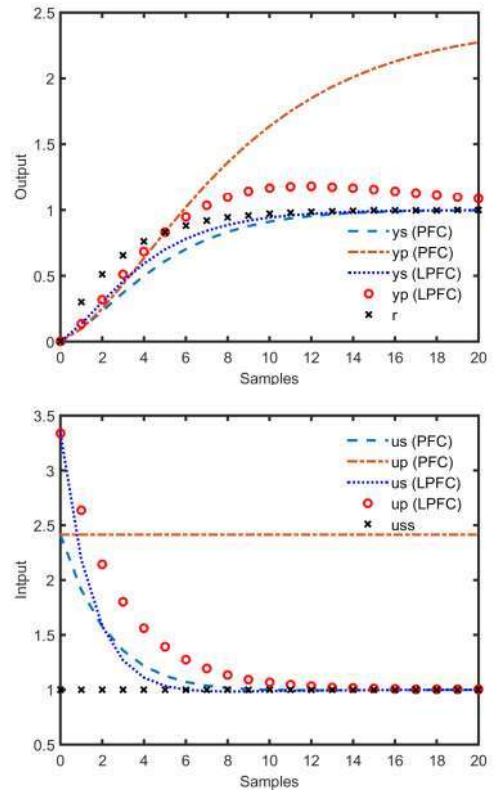


Fig. 4. 2nd order system response for nominal PFC and Laguerre PFC.

the nominal PFC. The result indicates that the system output $y_p(LPFC)$ exactly mimics the target trajectory r .

B. Second Order System

Most of the second order systems or higher have a lag in their step response [4]. Thus, it is not realistic in general to expect the system output to match a first order target trajectory in immediate transients. In this study, the selection of coincidence horizon for these particular systems is made based on the conjecture presented in [4]. A good range is defined approximately around 40% to 80% rise of the step response to its steady state value where the gradient is significant. Fig. 3

shows the step response of the 2nd order system (15) alongside the desired closed-loop behaviour ($\lambda = 0.7$). The 40% to 80% rise of the step response is approximately corresponding to the coincidence horizon of $n = 5$ and $n = 15$, respectively. Extending it to the target trajectory, the suitable horizon is selected as $n = 5$ as this gives a significant gradient.

$$G_2 = \frac{0.04z^{-1}}{1 - 1.6z^{-1} + 0.64z^{-2}} \quad (15)$$

Fig. 4 shows the performance of both PFC controllers, nominal and Laguerre PFC in tracking the set points with $n = 5$. A similar conclusion can be drawn as for the first order

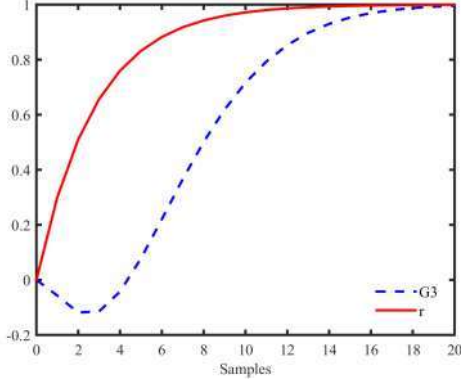


Fig. 5. 3rd order non-minimum phase system step response G_3 with target trajectory r

example. The proposed LPFC controller manages to produce a well-posed solution as can be seen from the comparison between $y_p(LPFC)$ and $y_s(LPFC)$ graphs. In addition, it also provides a closer output response $y_s(LPFC)$ with the target trajectory r compared to the nominal PFC $y_s(PFC)$. As stated earlier, a close match in early transients is not a reasonable expectation.

C. Non-minimum Phase System

For the next case, a third order non-minimum phase system is considered. The transfer function model is given as:

$$G_3 = \frac{-0.0569z^{-1} + 0.0514z^{-2} + 0.0502z^{-3}}{1 - 1.9842z^{-1} + 1.3301z^{-2} - 0.3012z^{-3}} \quad (16)$$

The step response of this system together with the desired target trajectory are plotted (see Fig. 5). A similar procedure is used as the previous second order example where the coincidence horizon is selected based on 40% to 80% rise of its step response. Thus, the horizon should be in the range of $7 \leq n_y \leq 12$. Selecting below this range will lead to a poor closed-loop performance. On the other hand, selecting above this value will put too much emphasis on the steady state value. For this case, a choice $n = 7$ was made.

Fig. 6 shows the control performance of nominal PFC and Laguerre PFC for the non-minimum phase system. As expected, neither controller is able to track the 1st order target trajectory in transients because of the non-minimum phase characteristic. However, Laguerre PFC manages to produce a well-posed decision. Without Laguerre, the output prediction $y_p(PFC)$ continues to rise beyond the target trajectory. Conversely, with Laguerre, both input prediction $u_p(LPFC)$ and output prediction $y_p(LPFC)$ converge smoothly to the desired targets.

D. Effect of Varying Laguerre Pole

An interesting observation is noted here. The convergence of the predicted input depends on the selected value of the Laguerre pole. This parameter determines how fast the predicted input converges and implicitly therefore has an impact on the

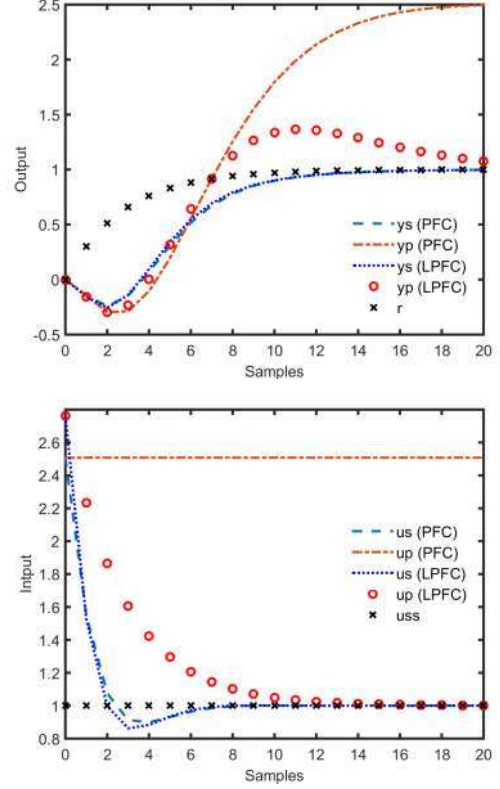


Fig. 6. 3rd order non-minimum phase system response for PFC and LPFC.

convergence rate of the output predictions. Logically, the value a should be equal to the desired closed-loop time constant λ . However, Fig. 7 illustrates the effect on closed-loop behaviour of varying the Laguerre pole a on the second order system. As one reduces the pole, the system output produces behaviour closer to the target trajectory but with a higher initial input. Conversely, if one increases the pole to a higher value, the input prediction converges slower to the steady state value with a lower initial input.

Although it is not quite as obvious as with the second order system, a similar behaviour is produced by the third order non-minimum phase system (see Fig. 8). Decreasing Laguerre pole to a smaller value will force the system input to converge faster to its steady state value. This will generate a closer output response with the target trajectory in demand of higher initial input. On the other hand, increasing the pole to a higher value, the opposite effect is gained. With this kind of ability, it will not only increase the effectiveness of λ as tuning parameter but also improve the system performance compared to the nominal PFC.

Despite these promising results, questions remain. Should the value of Laguerre pole a equal to the desired closed-loop time constant λ , or should it become the second tuning parameter to improve the system performance. Clearly, further work is required to establish detailed analysis of this issue.

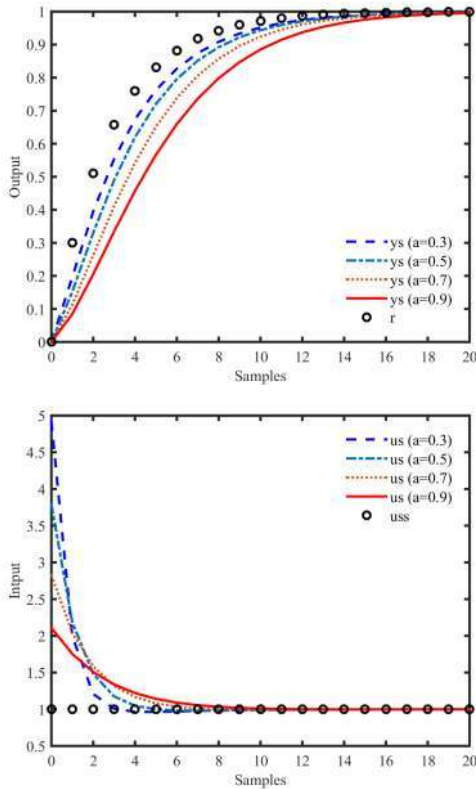


Fig. 7. Laguerre PFC for 2nd order system with different Laguerre pole a .

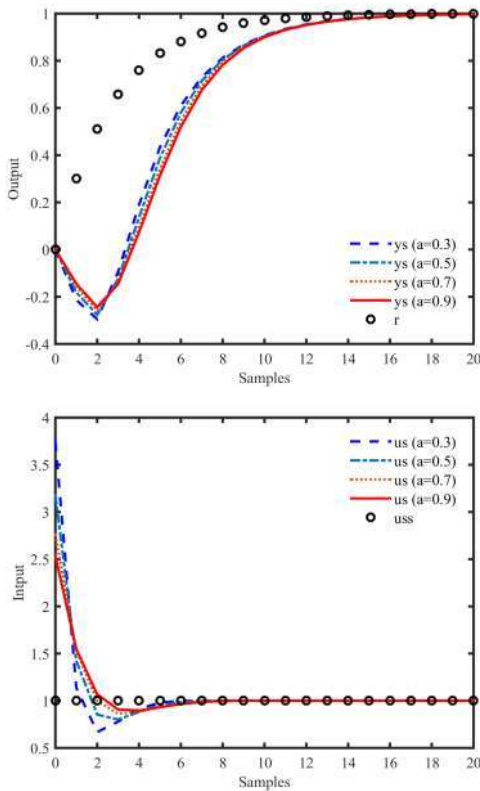


Fig. 8. Laguerre PFC for 3rd order non-minimum phase system with different Laguerre pole a .

V. CONCLUSION

This work reviews the possibility of utilizing Laguerre polynomials to shape the PFC input prediction. Instead of the constant future input assumption, Laguerre PFC forces the input prediction to converge to its steady state value throughout the horizon. This slight modification can improve the consistency of the decision making where the optimal input differs significantly from a constant. The examples have show that the proposed controller can improve the PFC performance of first order system, second order system, and third order non-minimum phase system. By improving the consistency of the underlying PFC assumptions, the proposed PFC law can also improve the effectiveness of λ as a tuning parameter, with obvious benefits to end users.

The next steps involve consideration of real industrial applications and a systematic analysis of robustness and sensitivity. The PFC controller is well known for its simplicity, it works well in many applications. However, for many scenarios a number ad-hoc constructive methods are used to improve behaviour and thus there is a need for a more systematic analysis and insight which can lead to a more consistent and flexible algorithms for such scenarios. In addition, it is also important to test the proposed method on the system with a time delay or unstable dynamics as it may sometimes degrade the closed-loop control performance in many industrial applications.

ACKNOWLEDGMENT

The first author would like to acknowledge International Islamic University Malaysia and Ministry of Higher Education Malaysia for funding this work.

REFERENCES

- [1] M. Abdullah, and M. Idres, "Constrained model predictive control of proton exchange membrane fuel cell," *J. Mech. Sci. and Technology*, vol. 28, no. 9, pp. 3855-3862, Sept. 2014.
- [2] M. A. Hasan, M. Idres, and M. Abdelrahman, "Black box nonlinear model predictive control using recurrent neural network," presented at the 2nd Int. Conf. Mechanical, Automotive and Aerospace Engineering, K.L, 2013.
- [3] J. A. Rossiter, *Model predictive control: a practical approach*, CRC Press, 2003.
- [4] J. A. Rossiter, and R. Haber, "The effect of coincidence horizon on predictive functional control," *Processes*, vol.3, no. 1, pp. 25-45, Jan. 2015.
- [5] J. Richalet, A. Rault, J. L Testud, and J. Papon, "Model predictive heuristic control: applications to industrial processes," *Automatica*, vol. 14, no. 5, pp. 413-428, 1978.
- [6] J. Richalet, and D. ODonovan, *Predictive functional control: Principles and industrial applications*, Springer Science & Business Media, 2009.
- [7] J. Richalet, G. Lavielle, and J. Mallet, *Commande Predictive*, Eyrolles, 2004.
- [8] J. A. Rossiter, "Input shaping for PFC: how and why?," *J. Control and Decision*, pp. 1-14, Sep. 2015.
- [9] J. A. Rossiter, L. Wang, G. Valencia-Palomo, "Efficient algorithms for trading off feasibility and performance in predictive control", *Int. J. Control*, vol. 83, no. 4, pp. 789-797, 2010.
- [10] L. Wang, "Discrete model predictive control design using Laguerre functions," *J. Process Control*, vol. 14, pp. 131-142, 2004.
- [11] M. Abdullah, and M. Idres, "Fuel cell starvation control using model predictive technique with Laguerre and exponential weight functions," *J. Mech. Sci. and Technology*, vol. 28, no. 5, pp. 1995-2002, May. 2014.
- [12] J. Skultety, E. Miklovicova, K. Zabet, R. Haber and R. Bars, "Predictive functional control of synchronous generator model based on Laguerre network," in *Automation and Applied Comp. Workshop*, 2012.

Appendix C

**DEVELOPMENT OF CONSTRAINED
PREDICTIVE FUNCTIONAL CONTROL
USING LAGUERRE FUNCTION BASED
PREDICTION**

M. Abdullah, J. A. Rossiter and R. Haber

This paper has been published in:
IFAC PapersOnLine, Vol. 50, No. 1

Declaration form

DEVELOPMENT OF CONSTRAINED PREDICTIVE FUNCTIONAL CONTROL USING LAGUERRE FUNCTION
BASED PREDICTION

(IFAC PaperOnLine, Vol. 50, No. 1)

Contributions of authors:

M. Abdullah

Provided the initial idea, formulations, codes, simulations and draft of this paper.

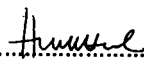
J. A. Rossiter

Supervised M. Abdullah and proofread the paper.

R. Haber

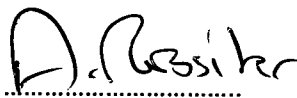
Proofread the paper and suggested some input and improvement.

Signatures:


.....

M. Abdullah

(First author)


.....

J.A. Rossiter

(Second author)


.....

R. Haber

(Third author)



Development of Constrained Predictive Functional Control using Laguerre Function Based Prediction

M. Abdullah^{*,**} J.A. Rossiter^{*} R. Haber^{***}

^{*} Department of Automatic Control and System Engineering,
University of Sheffield, Mappin Street, S1 3JD, UK.

(e-mail: MAbdullah2@sheffield.ac.uk, j.a.rossiter@sheffield.ac.uk)

^{**} Dept. of Mech. Engin., International Islamic University Malaysia.^{*}

^{***} Institute of Process Engineering and Plant Design, Cologne
University of Applied Sciences, D-50679 Köln, Germany.

Abstract: This work presents a novel constraint handling strategy for Predictive Functional Control (PFC). First, to improve prediction consistency, the constant input assumption of nominal PFC approaches is replaced with Laguerre based prediction. This substitution improves the effectiveness of using a constrained solution to prevent long-term constraint violations. Secondly, for state constraints, a simpler single regulator approach is proposed instead of switching between regulators, an approach common in the PFC literature. Simulation results verify that the proposed method manages the constraints better than the traditional approach. Moreover, despite all the modifications, the controller formulation and framework remain simple and straightforward which thus are in line with the key ethos of PFC.

© 2017, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Predictive Control, Constrained PFC, Effective Constraint Technique.

1. INTRODUCTION

Most control systems have constraints which can be identified as input constraints, rate constraints, state constraints and output constraints. If these constraints are not considered systematically in a control design, it may result in unwanted behaviour such as overshoots, long settling times, and even instability. Satisfying constraints effectively offers many attractive benefits including a higher production profit, better control performance, lower maintenance cost and safer control environment (Rossiter, 2003; Richalet and O'Donovan, 2009; Wang, 2009; Abdullah and Idres, 2014a). Clearly, these scenarios justify the need for a systematic constrained controller design.

In practice, the commonly used Proportional-Integral-Derivative (PID) controller faces difficulties in handling constraints. For example, the usage of an integrator during constraint violations can produce wind-up and/or saturation (Rossiter, 2003). Although, anti wind-up techniques can prevent this situation (Visioli, 2006), these require tuning procedures which are difficult to design and manage for different combinations of dynamics and constraints.

Conversely, Model Predictive Control (MPC) utilises a representative mathematical model to form an accurate future prediction of the system behaviour and thus satisfies constraints systematically via an optimal control approach (Rossiter, 2003; Wang, 2009). However, typical MPC strategies require a high computational demand and expensive computer hardware, thus are only suitable for

certain applications (Rossiter et al., 2010; Jones and Kerrigan, 2015). Indeed, as the number of constraints increases, the optimal constraint handling problem requires increasingly complex and demanding solvers.

Many industrial end-users are willing to trade off some loss in optimality with ease/cost of implementation. This preference has triggered the widespread acceptance of Predictive Functional Control (PFC) among industrial practitioners. PFC belongs to the family of predictive control which compute the manipulated input based on a simplified cost function. It provides some valuable properties namely intuitive tuning, simplicity in coding, low computational demand, effective handling of dead-time processes and a basic constraint handling ability (Richalet and O'Donovan, 2009). With these features, PFC has become a popular and widely used alternative to PID controllers, especially for SISO loops.

The nominal PFC utilises a constant future input assumption to reduce the computation burden and formulation complexity (Richalet and O'Donovan, 2009). This assumption can be effective in some scenarios, especially where short predictions work well enough to capture the core dynamics. However, with long predictions, the consistency with the actual closed-loop behaviour can deteriorate significantly thus invalidating any assumptions used for constraint handling (Rossiter and Haber, 2015; Abdullah and Rossiter, 2016). This break down in consistency can imply that the PFC constraint handling approach is invalid at worst and leads to poor decision making (that is, input choices) at best. Moreover, PFC practitioners commonly use an ad-hoc approach for managing state

^{*} This work is funded by International Islamic University Malaysia and Ministry of Higher Education Malaysia.

constraints, where multiple regulators that work in parallel are switched either to track the set point or satisfy the constraint depending on a supervisor decision (Richalet and O'Donovan, 2009). This structure works in most applications, but has a disadvantage in that it requires a careful tuning procedure to avoid conflicts with the internal constraints which thus counters some of the inherent benefits of a simple and transparent approach. The operation cost also may increase due to the use of multiple regulators.

This paper proposes a better constraint strategy to alleviate some of the drawbacks of the conventional approach. A Laguerre function will be utilised to improve the prediction consistency (Abdullah and Rossiter, 2016), hence instead of assuming a constant future value, the future predicted input converges to the steady state exponentially based on the desired pole. With a well-posed decision, the constrained solution will become more precise and less conservative. In addition, rather than handling state constraints with a multiple regulators scheme, a vector approach is considered to simplify the computation and tuning processes.

Section 2 provides a brief description of a traditional constrained PFC formulation. Section 3 presents the proposed Laguerre PFC scheme for constraint management. Section 4 gives a comparison between the nominal and Laguerre approaches based on two numerical examples. Finally, section 5 presents some conclusions and future work.

2. NOMINAL CONSTRAINED PFC FORMULATION

This section provides a brief review of nominal PFC including constraint handling. For simplicity of presentation of the core concepts, the main objective is to track a constant step target and moreover, the offset correction and integral action algebras are omitted, although included in the numerical examples. These simplifications do not affect the core analysis, insights and results presented. Finally, without loss of generality, the PFC formulation is constructed using a general transfer function structure.

2.1 Unconstrained PFC

The basic principle of PFC is to drive the n_y step ahead prediction of output $y_{k+n_y|k}$ nearer to the set point R than the current output y_k . The ratio is linked to a tuning parameter is the desired closed loop pole $\lambda = e^{-3T/CLTR}$, where T is the sampling time and CLTR is the desired closed loop settling time (to 95%). The basic PFC law is defined by enforcing the following equality:

$$y_{k+n_y|k} = R - (R - y_k)\lambda^{n_y} \quad (1)$$

where n_y is denoted as the coincidence horizon. There are some subtleties to ensure offset free tracking but the basic law is still (1). For a more detailed description of PFC theory and concepts, interested readers can refer to these references, e.g. (Rossiter and Haber, 2015; Richalet and O'Donovan, 2009; Haber et al., 2011).

Since the prediction algebra for general transfer functions is well known in the literature (e.g. (Rossiter, 2003)), only simplified formulations are presented here. The n_y step ahead unbiased linear prediction for inputs u_k and outputs y_k can be represented as:

$$y_{k+n_y|k} = H_{n_y} u_{\underline{k}} + P_{n_y} u_{\underline{k}} + Q_{n_y} y_{\underline{k}} \quad (2)$$

where H_{n_y} , P_{n_y} , Q_{n_y} depend on the model parameters and for systems of order m :

$$\underline{u}_k = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+n-1} \end{bmatrix}; \underline{u}_k = \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \\ u_{k-m} \end{bmatrix}; \underline{y}_k = \begin{bmatrix} y_k \\ y_{k-1} \\ \vdots \\ y_{k-m} \end{bmatrix} \quad (3)$$

The control input is solved by substituting the prediction of (2) into (1) alongside the assumption of a constant future input, namely $u_{k+i|k} = u_k, i = 0, \dots, n_y$. In consequence the parameter H_{n_y} can be simplified to $h_{n_y} = H_{n_y}[1, 1, \dots]^T$ and (1) becomes:

$$h_{n_y} u_k + P_{n_y} u_{\underline{k}} + Q_{n_y} y_{\underline{k}} = R - (R - y_k)\lambda^{n_y} \quad (4)$$

After minor rearrangement, the, PFC law reduces to:

$$u_k = \frac{R - (R - y_k)\lambda^{n_y} - (P_{n_y} u_{\underline{k}} + Q_{n_y} y_{\underline{k}})}{h_{n_y}} \quad (5)$$

2.2 Input and Input Rate Constraints

The system input is often constrained because of physical limits or indeed desired limits on temperature, pressure, voltage and others. These constraints are presented as:

$$u_{min} \leq u_k \leq u_{max} \quad (6)$$

$$\Delta u_{min} + u_{k-1} \leq u_k \leq \Delta u_{max} + u_{k-1} \quad (7)$$

where Δu_{min} and Δu_{max} are the minimum and the maximum rate, while u_{min} and u_{max} denote the minimum and maximum input. Without explicitly including these constraints in the control computation, a clipping method can be utilised (Fiani et al., 1991). When the limit in (6) or (7) is violated, the controller will treat it as an equality constraint (Wang, 2009). However, it is crucial for the model to detect possible constraint violations *a priori* (Richalet and O'Donovan, 2009). Failure to do this could introduce an overshoot in the input (and/or output) due to a mismatch between the predicted model behaviour and the actual system behaviour.

Remark 1. The input and rate constraint need only be implemented on the current input within conventional PFC because of the constant future input assumption.

2.3 State Constraints

In some applications (i.e heat treatment) an internal variable, state or output may be constrained either for an economic or safety reason. To solve this problem, the conventional PFC approach uses multiple regulators which run in parallel (see Fig. 1) (Richalet and O'Donovan, 2009; Fiani et al., 1991).

- The first regulator PFC_1 is the preferred control law and produces input $u_{1,k}$ (using (5)) to track the set point while satisfying its internal constraints. Within some validation horizon to be defined, the supervisor uses input $u_{1,k}$ to predict the future state behaviour using a prediction model such as (2). If the state predictions are within their limit, then use $u_k = u_{1,k}$.
- The second regulator PFC_2 is more conservatively tuned and tracks the state limit by manipulating input $u_{2,k}$. When the state limit is expected to be violated using PFC_1 , then use $u_k = u_{2,k}$.

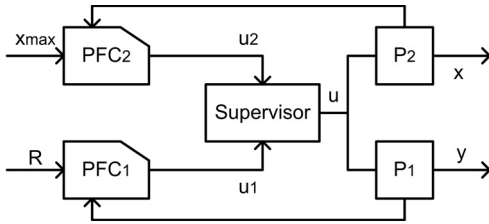


Fig. 1. Schematic of PFC considering state constraints.

- An advanced decision-making method such as fuzzy logic, look up table, or artificial neural network may be utilised for a smoother transition.

Remark 2. The second controller PFC_2 regulates the second input $u_{2,k}$ based on a state prediction equation:

$$x_{k+n_x|k} = h_{n_x} u_k + P_{n_x} u_{\leftarrow k} + Q_{n_x} x_k \quad (8)$$

where $h_{n_x}, P_{n_x}, Q_{n_x}$ denote the state model parameters. The maximum state limit x_{max} is a set as a target. With a suitable coincidence horizon n_x and the desired closed loop pole λ_x , input $u_{2,k}$ is computed as:

$$u_{2,k} = \frac{x_{max} - (x_{max} - x_k)\lambda_x^{n_x} - (P_{n_x} u_{\leftarrow k} + Q_{n_x} x_k)}{h_{n_x}} \quad (9)$$

The associated PFC is tuned, if possible, to avoid oscillations in the predictions to ensure the constraint is satisfied.

Remark 3. A suitable validation horizon for checking the predictions associated to PFC_1 should be used since the projection of $u_{1,k}$ must include the open loop time response of PFC_2 . In addition, the target pole λ_x of PFC_2 must be compatible with the need to satisfy the internal constraints of PFC_1 . Choosing a fast pole to improve the overall system response may decrease the controller robustness and introduce conflicts with the actuator limit (Richalet and O'Donovan, 2009).

3. CONSTRAINED LAGUERRE PFC FORMULATION

This section presents the formulation of PFC based on Laguerre based input predictions. By embedding exponentially decaying dynamics within the input prediction, it enables PFC to achieve a closer match to the desired closed-loop behaviour. This can improve the reliability of the constrained PFC solution. A detailed analysis and benefits of Laguerre PFC are presented in Abdullah and Rossiter (2016). Since a similar strategy to nominal PFC is adopted for input and rate constraints (Remark 1), only the state/output constraint case is presented here.

The Laguerre PFC approach requires explicit knowledge of the expected constant steady-state input u_{ss} which will lead to no steady-state offset; in fact this value is implicitly used in conventional MPC as well. For a given model and disturbance estimate, the computation of this is straightforward (Rossiter, 2003).

3.1 Unconstrained Laguerre PFC formulation

A Laguerre polynomial is often used for system identification and estimation as it can provide the ability to capture system behaviour with fewer parameters (Nurges, 1987). The z-transform of discrete Laguerre polynomials are:

$$L_j(z) = \sqrt{1-a^2} \frac{(z^{-1}-a)^{j-1}}{(1-az^{-1})^j}; \quad 0 < a < 1 \quad (10)$$

where j is the order of Laguerre function and a is the Laguerre pole which depends on a user selection. Although a high order polynomial can be used in MPC (Abdullah and Idres, 2014b; Wang, 2009), this work employs a first-order Laguerre polynomial to retain the simplicity of formulation especially when dealing with low order system. The first-order Laguerre function, with altered scaling is:

$$L_1(z) = \frac{1}{1-az^{-1}} \equiv 1 + az^{-1} + a^2z^{-2} + \dots \quad (11)$$

Define $L_1 = [1, a, a^2, \dots, a^{n-1}]^T$. Now we are in a position to define the input prediction to be deployed in PFC.

Theorem 1. A future input parametrised as

$$u(z) = \frac{u_{ss}}{1-z^{-1}} + \frac{\eta}{1-az^{-1}} \quad (12)$$

will give output predictions which settle at the desired steady-state. η represents one degree of freedom.

Proof: The signal defined in (12) has the property that

$$\lim_{k \rightarrow \infty} u_k = u_{ss} \Rightarrow \lim_{k \rightarrow \infty} y_k = R \quad \square \quad (13)$$

The implied input prediction in (12) converges to the steady state exponentially with a rate a . The associated output prediction is derived by substituting (12) into (2):

$$y_{k+n_y|k} = h_{n_y} u_{ss} + H_{n_y} L_1 \eta + P_{n_y} u_{\leftarrow k} + Q_{n_y} y_k \quad (14)$$

The following algorithm defines the PFC law using the Laguerre polynomial to shape the input predictions.

Algorithm 1. (LPFC). Define the n_y step ahead predicted output using equation (14). The PFC law is defined by substituting this prediction into (1), solving for the parameter η and then computing u_k from (12).

$$\eta = \frac{R - (R - y_k)\lambda^{n_y} - (P_{n_y} u_{\leftarrow k} + Q_{n_y} y_k) - h_{n_y} u_{ss}}{H_{n_y} L_1} \quad (15)$$

Due to the receding horizon principle (Wang, 2009) and the definition of $L_1(z)$, the current input is defined as:

$$u_k = u_{ss} + \eta \quad (16)$$

Remark 4. The value of Laguerre pole a determines the convergence speed of a system (Abdullah and Rossiter, 2016). For low order system, a reasonable choice is $a = \lambda$ where it gives a direct link to the desired target trajectory.

Remark 5. The maximum (if bigger than u_{ss}) and minimum (if smaller than u_{ss}) of the predicted future input given in (12) is the first (current) value u_k . Similarly, the maximum/minimum rate is given from $\Delta u_k = u_{ss} + \eta - u_{k-1}$. Hence, with LPFC, the maximum/minimum input rate/value (relative to expected steady-state) occur at the first sample and thus the proposed Laguerre PFC can adopt an equivalent constraint handling procedure for input constraints as standard PFC.

3.2 Efficient state and output constraint handling

To increase the efficiency of constraint strategy, an off-line prediction is utilised. The limiter computes the maximum or minimum input that is associated with all constraints being satisfied within the validation horizon. The technique has similarities to the so called ONEDOF and reference governor approaches in the literature Rossiter et al. (2001) and has the advantage of being implementable using a single simple loop at each iteration.

Lemma 2. The input constraints can be represented by a set of linear inequalities with a single variable η .

Proof: This follows directly from the observations in remark 5. The constraints can be summarised as follows:

$$\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} \eta + \begin{bmatrix} u_{ss} \\ -u_{ss} \\ u_{ss} - u_{k-1} \\ u_{k-1} - u_{ss} \end{bmatrix} \leq \begin{bmatrix} u_{max} \\ u_{min} \\ \Delta u_{max} \\ \Delta u_{min} \end{bmatrix} \quad \square \quad (17)$$

Lemma 3. State constraints can be represented by a set of linear inequalities with a single variable η .

Proof: This follows directly from computation of the state predictions as in (14) and comparison with the state limits. For example, a single state limit gives the following:

$$H_{n_x} \left\{ \begin{bmatrix} u_{ss} \\ u_{ss} \\ u_{ss} \\ \vdots \end{bmatrix} + \begin{bmatrix} 1 \\ a \\ a^2 \\ \vdots \end{bmatrix} \eta \right\} + \underbrace{P_{n_x} u_k + Q_{n_x} x_k}_{f_{n_x}(k)} \leq x_{max}$$

One can stack these inequalities over a specified horizon such that, for example:

$$\begin{bmatrix} H_1 \\ H_2 \\ \vdots \end{bmatrix} \left\{ \begin{bmatrix} u_{ss} \\ u_{ss} \\ \vdots \end{bmatrix} + \begin{bmatrix} 1 \\ a \\ \vdots \end{bmatrix} \eta \right\} + \begin{bmatrix} f_1(k) \\ f_2(k) \\ \vdots \end{bmatrix} \leq x_{max} \quad \square \quad (18)$$

Theorem 4. All the input, state and output constraints can be represented by a single vector inequality as:

$$M\eta \leq v(k) \quad (19)$$

Proof: This is a consequence of the previous two lemmata by combining all the inequalities for all the constraints. The vector M is fixed but the vector $v(k)$ varies each sample as it depends upon past system data and the estimation of the expected steady-state input u_{ss} . \square

Corollary 1. In the absence of uncertainty, the inequalities implied in (19) are always feasible, assuming feasibility at the previous sample, no changes in the target and a long enough horizon.

Proof: The structure of the input prediction (12) is such that, as long as u_{ss} does not change from one sample to the next, then one can always choose η so that the predicted input trajectory is unchanged; this is obvious from the simple exponential structure. Consequently, if there exists an η to satisfy constraints at the previous sample, there must exist a valid value at the current sample. [We shall not discuss issues linked to required horizon lengths (Gilbert and Tan, 1991) as this would take the complexity beyond reasonable expectations for PFC approaches where a lack of rigorous mathematical guarantees is accepted to allow more simplicity.] \square

Remark 6. Infeasibility can arise due to too fast or too large changes in the target (or disturbances) as this causes large changes in the value of u_{ss} . However, Laguerre PFC helps enormously in this case because the exponential structure embedded into the input prediction automatically slows down any over aggressive input responses and thus significantly increases the likelihood of feasibility being retained. In the worst case, set point changes need to be moderated (as in reference governor approaches) but such a discussion is beyond the remit of this paper.

We can now define the constraint handling algorithm.

Algorithm 2. (LPFC constrained). First ensure that the change in the steady-state value of u_{ss} is such that no absolute or rate constraints in the inputs are violated as this suggests a poorly chosen target. Hence enforce that $|u_{ss,k} - u_{ss,k-1}| < \Delta u_{max}$ and that $u_{min} \leq u_{ss} \leq u_{max}$.

Second, use the unconstrained law (15) to determine the ideal value of η and check each constraint implied in (19) using the following simple loop (subscripts denote position in a vector).

Set $\eta_{max} = \infty$, $\eta_{min} = -\infty$.

For $i=1$:end,

if $M_i\eta \not\leq v_i$ & $M_i > 0$ then define $\eta_{max} = v_i/M_i$,

if $M_i\eta \not\leq v_i$ & $M_i < 0$ then define $\eta_{min} = v_i/M_i$,

end loop.

if $\eta < \eta_{min}$, set $\eta = \eta_{min}$. if $\eta_{max} < \eta$, set $\eta = \eta_{max}$.

Note that the upper and lower limits on η to ensure feasibility update at each cycle in the loop but as all the inequalities are only ever tightened, changes lower down cannot contradict changes higher up.

3.3 Summary of benefits

This approach eliminates the careful tuning process of multiple regulators (Remark 3) since the constraint is now explicitly included in the control computation. Moreover, the algebra for computing the vectors v, M is the same as that required for computing the predictions and thus is unavoidable where constraint handling is desired and specifically, needs no input or tuning choices from the designer. This work has not investigated the implications of infeasibility due to large disturbances or set point changes any further than insisting on sensible limits to changes in u_{ss} as that is a more challenging scenario and requires a priori trade off decisions such as which constraints or requirements to sacrifice during transients.

4. NUMERICAL EXAMPLES

This section presents two numerical examples to highlight the benefit of the proposed constraint method. The first example implements output constraints while the second example operates with state constraints. For each case, two figures are plotted to represent the system input and output. The focus is to analyse and compare the constrained control performance of nominal PFC (PFC) and Laguerre PFC (LPFC). It should be noted that throughout the examples, a choice of $a = \lambda$ is used for LPFC as discussed in Remark 4.

4.1 Output Constraint Example

A first order system (20) with 0.2 input disturbance from 20s to 25s should track a constant set point ($R = 1$). For a fair comparison of PFC and LPFC, both controllers use similar tuning parameters for the desired pole ($\lambda = 0.7$) and a coincidence horizon ($n_y = 1$).

$$G_1 = \frac{0.25z^{-1}}{1 - 0.8z^{-1}} \quad (20)$$

In the unconstrained case (see Fig. 2), PFC and LPFC produce similar closed loop behaviour. The system output ($y(PFC)$ and $y(LPFC)$) exactly tracks the target

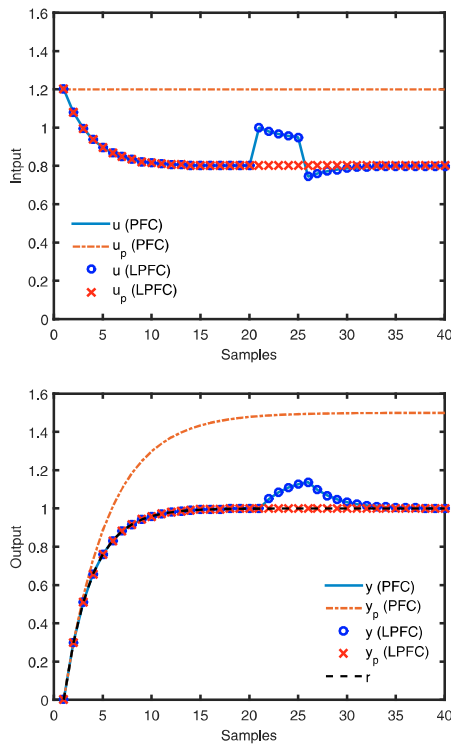


Fig. 2. Unconstrained PFC and LPFC responses.

trajectory R with settling time 8 samples and overshoots slightly in response to the disturbance. However, the initial prediction of nominal PFC $y_p(PFC)$ (displayed as computed at the first sample) is inconsistent with the actual closed-loop behaviour $y(PFC)$ because of the assumption of constant input in the prediction (i.e. $u_p(PFC)$). Nevertheless, the actual input $u(PFC)$ converges to the correct steady state value. Since LPFC embeds the exponential decay dynamics (e.g. through (12)), the input prediction $u_p(LPFC)$ matches the actual system input $u(LPFC)$ and so has better consistency between predictions and actual behaviour. This consistency is important for accurate constraint handling, to avoid conservativeness.

For the constrained case, a maximum output is set at $y_{max} = 1.05$. A validation horizon $i = 10$ is used to cover the transient period and avoid a long-term violation. However, PFC detects the output violation of $y_p(PFC)$ at the 6th sample ahead because of the ill-posed prediction (refer Fig. 2). The constraint is satisfied (Fig. 3) by the input $u(PFC)$ reducing from 1.2 to 0.9. As a result, the output $y(PFC)$ converges slower to the set point compared to $y(LPFC)$. Since LPFC produces a well-posed prediction, the output $y(LPFC)$ exactly matches the target trajectory R with a precise solution $u(LPFC)$.

4.2 State Constraint Example

Consider two processes that run in parallel. The main process P_1 and state process P_2 receive a similar manipulated input u from the regulator. For safety and economic reasons, the state is constrained at $x_{max} = 127$ with a limited input $u_{max} = 160$, and speed $\Delta u_{max} = 4$.

$$P_1 = \frac{0.0164z^{-1}}{1 - 0.9835z^{-1}}; P_2 = \frac{0.08914z^{-1} - 0.08674z^{-2}}{1 - 1.918z^{-1} + 0.92z^{-2}} \quad (21)$$

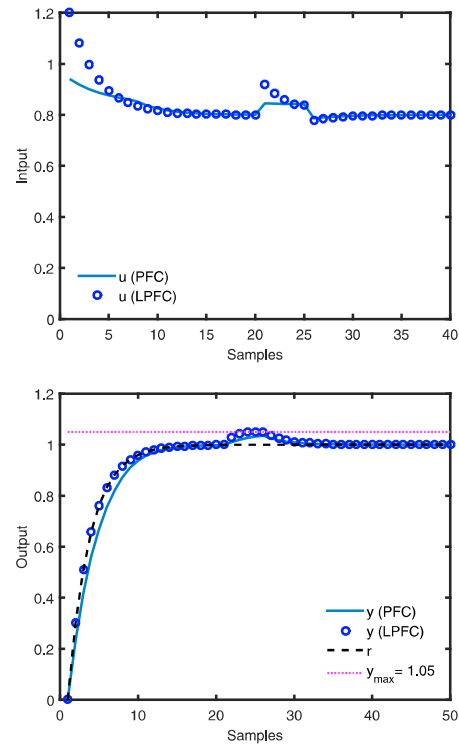


Fig. 3. Constrained PFC and LPFC responses.

For a fair comparison, Laguerre PFC (LPFC) and the first regulator of state constrained PFC (CPFC) will both use $n_y = 1$, validation horizon ($i = 68$) and pole ($\lambda = 0.975$) to track the set point ($R = 100$). Since CPFC treats the maximum state as a second target (9), the coincidence horizon ($n_x = 30$) and desired pole ($\lambda_x = 0.984$) of the second constraining regulator are selected carefully to satisfy the internal constraints (Remark 3).

Fig. 4 shows that LPFC outperforms CPFC while satisfying the state constraints. Although the state behaviours of both $x(CPFC)$ and $x(LPFC)$ are within the limits, the output settling time of $y(LPFC)$ 200 samples is almost twice as fast as $y(CPFC)$ (300+ samples) and closer to the target trajectory R . In addition, CPFC requires a careful tuning process and a higher operation cost as two regulators are used simultaneously. To respect the actuator limits, a large pole is needed to slow down the control response. Fig. 5 demonstrates the effect of poor tuning decision with a smaller pole $\lambda_x = 0.963$, where it computes a higher initial input than the maximum input $x_{max} = 160$. On the other hand, LPFC satisfies all the system constraints systematically without conflict. With Laguerre based prediction, the constrained solution becomes more precise and less conservative compared to the nominal CPFC approach.

5. CONCLUSION

This work proposes an improved constrained PFC technique to satisfy the state, output and input limits which are less conservative than the conventional PFC approach and no more onerous to code and implement. With a minimum modification, the design and formulation remain simple and straight forward. The embedding of Laguerre

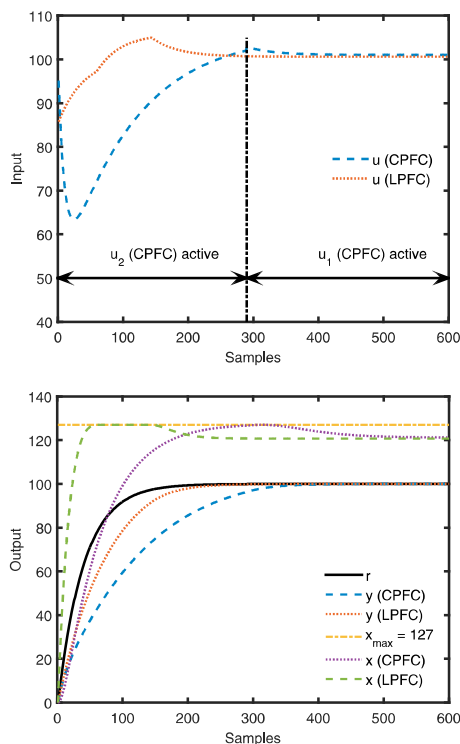


Fig. 4. Constrained CPFC and LPFC responses.

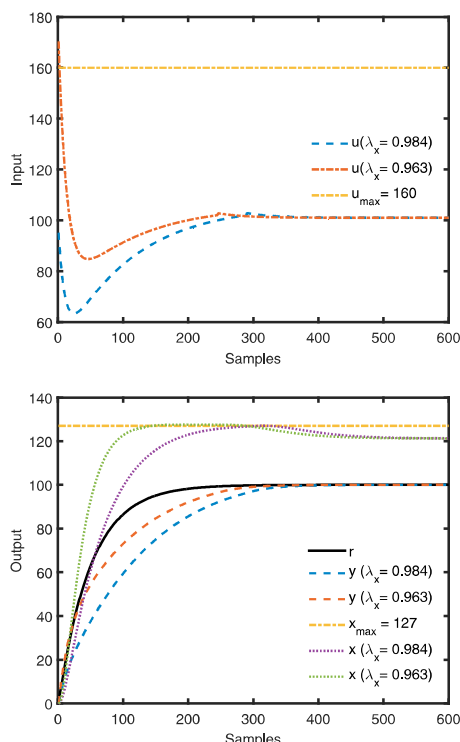


Fig. 5. CPFC responses with different poles $\lambda_x = 0.984$ and $\lambda_x = 0.963$.

input dynamics instead of constant input dynamics gives a better prediction consistency which ensures the constraint handling is more precise and less conservative. Given that the more conservative and complicated multi-regulator approach is widely adopted in many industrial applications,

we expect the proposed single constrained LPFC will offer better performance and be more cost effective. It alleviates the strict tuning requirements of the second CPFC regulator while satisfying all the constraints in a systematic fashion. As shown in the examples, the proposed method often enables faster convergence when handling the output and state constraints compared to the nominal strategy.

For future work, the robustness and sensitivity analysis of conventional PFC and LPFC will be investigated as well as the potential for more rigorous stability and feasibility guarantees, while retaining simplicity. Moreover, tests on hardware are planned. Finally, consideration will focus on whether higher order input parameterisations would be even more advantageous for higher order systems; this may involve a more complex constraint handling procedures.

REFERENCES

- Abdullah, M. and Idres, M. (2014a). Constrained model predictive control of proton exchange membrane fuel cell. *JMST*, 28(9), 3855–3862.
- Abdullah, M. and Idres, M. (2014b). Fuel cell starvation control using model predictive technique with Laguerre and exponential weight functions. *JMST*, 28(5), 1995–2002.
- Abdullah, M. and Rossiter, J.A. (2016). Utilising Laguerre function in predictive functional control to ensure prediction consistency. *UKACC*.
- Fiani, P., Richalet, J., et al. (1991). Handling input and state constraints in predictive functional control. In *CDC*, 985–990. IEEE.
- Gilbert, E.G. and Tan, K.T. (1991). Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9), 1008–1020.
- Haber, R., Bars, R., and Schmitz, U. (2011). *Predictive control in process engineering: From basics to applications, chapter 11*. Wiley-VCH, Germany.
- Jones, C.N. and Kerrigan, E. (2015). Predictive control for embedded systems. *Optimal Control Applications and Methods*, 36(5), 583–584.
- Nurges, Y. (1987). Laguerre models in approximation and identification of digital systems. *Avtomatika i Telemekhanika*, (3), 88–96.
- Richalet, J. and O'Donovan, D. (2009). *Predictive functional control: principles and industrial applications*. Springer.
- Rossiter, J.A. (2003). *Model-based predictive control: a practical approach*. CRC press.
- Rossiter, J., Kouvaritakis, B., and Cannon, M. (2001). Computationally efficient algorithms for constraint handling with guaranteed stability and near optimality. *IJC*, 74(17), 1678–1689.
- Rossiter, J., Wang, L., and Valencia-Palomo, G. (2010). Efficient algorithms for trading off feasibility and performance in predictive control. *IJC*, 83(4), 789–797.
- Rossiter, J.A. and Haber, R. (2015). The effect of coincidence horizon on predictive functional control. *Processes*, 3(1), 25–45.
- Visioli, A. (2006). *Practical PID control*. Springer.
- Wang, L. (2009). *Model predictive control system design and implementation using MATLAB®*. Springer.

Appendix D

**USING LAGUERRE FUNCTIONS TO
IMPROVE THE TUNING AND
PERFORMANCE OF PREDICTIVE
FUNCTIONAL CONTROL**

M. Abdullah and J. A. Rossiter

This paper has been accepted for publication in:
International Journal of Control 2019

Declaration form

USING LAGUERRE FUNCTIONS TO IMPROVE THE TUNING AND PERFORMANCE OF PREDICTIVE
FUNCTIONAL CONTROL

(International Journal of Control 2019)

Contributions of authors:

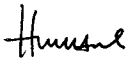
M. Abdullah

Provided the initial idea, formulations, codes, simulations and draft of this paper.

J. A. Rossiter

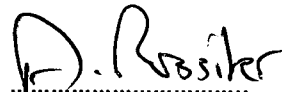
Supervised M. Abdullah and proofread the paper. Provided idea on the comparison between different parameterisations of Laguerre polynomial.

Signatures:



M. Abdullah

(First author)



J.A. Rossiter

(Second author)

ARTICLE TEMPLATE

Using Laguerre functions to improve the tuning and performance of predictive functional control

Muhammad Abdullah^{a,*} and John Anthony Rossiter^b

^{a,b} Department of Automatic Control and Systems Engineering, The University of Sheffield, S1 3JD, UK.

ARTICLE HISTORY

Compiled February 7, 2019

ABSTRACT

This paper proposes a novel modification to the predictive functional control (PFC) algorithm to facilitate significant improvements in the tuning efficacy. The core concept is the use of an alternative parameterisation of the degrees of freedom in the PFC law. Building on recent insights into the potential of Laguerre functions in traditional MPC (Rossiter et al., 2010; Wang, 2009), the paper develops an appropriate framework for PFC and then demonstrates that these functions can be exploited to allow easier and more effective tuning in PFC as well as facilitating strong constraint handling properties. The proposed design approach and the associated tuning methodology are developed and their efficacy is demonstrated with a number of numerical examples.

KEYWORDS

MPC; PFC; coincidence horizon; tuning.

1. Introduction

Model predictive control (MPC) has been very popular in the literature (Camacho & Bordons, 1999; Rossiter, 2018) for decades and very widely applied in industry (Richalet, 1993). However, the literature has given much less attention to certain approaches within the MPC portfolio, namely algorithms such as Predictive Functional Control (PFC), (Fallasohi et al., 2010; Fiani & Richalet, 1991; Haber et al., 2011; Richalet et al., 1978, 2009). This may appear somewhat surprising given the evidence that PFC is so widely used in industry (Richalet et al., 2009), however, the reasoning is simple: the tuning and general properties of PFC are difficult and weak compared to more conventional MPC algorithms (Rossiter & Haber, 2015; Rossiter, 2015; Rossiter et al., 2016) and in consequence, academic authors and reviewers are very wary since most of the academic journals are focusing on the theoretical development of a predictive controller. The same situation can be seen in most of the core textbooks of MPC as PFC concept is barely discussed, for examples Rawlings & Mayne (2009); Wang (2009) to name a few. More specifically, with the exception of a few special cases (Rossiter, 2016), PFC is not conducive to *a priori* stability guarantees and many reviewers are uncomfortable with this weakness, notwithstanding the huge successes

Corresponding author. Email: MAbdullah2@sheffield.ac.uk

in industry especially for many chemical applications (Haber et al., 2011; Richalet et al., 2009).

The prime aim of this paper is to propose a modification to PFC which improves the overall properties and thus gives the user more confidence in the resulting closed-loop behaviour and constraint handling. We do not pretend that a generic *a priori* stability proof is possible and instead emphasise that PFC should not be contrasted with more advanced MPC algorithms such as dual-mode (Rossiter et al., 1998; Rossiter, 2018; Sokaert & Rawlings, 1998) because:

- (1) PFC is a competitor with algorithms such as PID which equally have weak *a priori* properties.
- (2) PFC is orders of magnitude cheaper than dual-mode MPC and thus a comparison between the two is inappropriate. Besides the application is usually restricted to Single Input and Single Output (SISO) processes; you get what you pay for and if you want a very cheap and simple algorithm do not expect all the analysis and properties of expensive alternatives.

PFC offers properties such as prediction and systematic constraint handling not easily embedded in PID while having coding complexity that is similar to PID and tuning rules that are easy to automate and understand (Richalet et al., 2009, 2011).

In recent years, some authors have been to explore alternative parameterisations of the degrees of freedom with conventional MPC approaches. One could argue this begins with prestabilisation (e.g. Muske & Rawlings (1993)) and indeed the dual-mode approaches as proposed in the 1990s are, in effect, reparameterising the d.o.f. (Rossiter, 2018). Latterly, authors have been looking at functional approaches (Rossiter et al., 2010; Wang, 2009) whereby the future input trajectory is defined as a linear combination of a set of functions (Khan & Rossiter, 2013), where the systematic choice of function is still to a large extent an open question (Muehlebach & D'Andrea, 2017). The advantages of functions such as Laguerre over the more conventional choice of a standard basis set are that they extend the impact of the input changes over a much longer horizon and thus are more likely to be able to capture the shape of the desired closed-loop input trajectory. This simple change can lead to a reduction in the number of d.o.f. needed to manage constraints effectively and thus enables performance to be maintained with a lower computational load.

The obvious question to ask then is: to what extent can a similar parameterisation improve the properties of PFC leading to easier or more consistent tuning? Such an advance would be of significant benefit given the large number of SISO loops which involve some what challenging dynamics and constraints which make PID implementations messy and often poorly tuned whereas, by contrast, PFC may be equally cheap and also able to handle those dynamics and constraints more systematically. Of course, as with PID, the user must accept that any stability analysis is *a posteriori*.

A smaller but important contribution of this paper is also to improve the constraint handling techniques typically adopted in a conventional PFC algorithm as, in order to demonstrate the constraint handling of the proposed Laguerre approaches, it is pertinent to ensure the constraint handling is done as effectively as possible. The existing strategies popular in PFC were developed using engineering intuition and limited computing but as will be seen can be improved using insights available from more modern MPC approaches and while still incurring minimal computational loading and coding, certainly in terms of the functionality available on current cheap processors. Specifically the aim is to avoid the need for an online optimisation as that ensures the code is simple, quick and easy to maintain and validate.

This paper will propose the use of Laguerre functions for PFC and demonstrate how these can be introduced in a systematic manner. Section 2 will give standard background on PFC and briefly highlight the tuning challenges. Section 3 looks at constraint handling and proposes a systematic but simple procedure. Section 4 will introduce Laguerre functions and propose two different mechanisms for introducing these into PFC. Section 5 will give a number of numerical examples and Section 6 will give a case study on some laboratory hardware. Section 7 contains the conclusions.

2. Background on PFC

This section summarises the key assumptions, notation and principles underlying a conventional PFC algorithm and gives a brief insight into the weaknesses. As much of this is standard in the literature, detailed derivations are omitted.

2.1. Standard PFC algorithm

PFC is premised on the assumption that it is reasonable to desire the closed-loop response to follow (approximately) a first order trajectory from the current position to the desired steady-state target.

Remark 2.1. *For simplicity of exposition, as these issues only introduce more complicated algebra but do not affect the core principles, this paper will not discuss issues linked to non-zero dead-times and time-varying targets. Details are available in the references.*

Therefore, the desired output trajectory is given as:

$$r_{k+n} = R - (R - y_k)\lambda^n, \quad (1)$$

where r_{k+n} denotes the desired n-step ahead value for output y_k at sample k and λ is the desired closed-loop pole (PFC practitioners often use the desired closed-loop time constant in lieu of λ as these are equivalent) and R is the target. The unconstrained PFC law is defined by solving, for a single specified coincidence horizon n :

$$y_{k+n|k} = r_{k+n} \quad \text{with} \quad u_k = u_{k+1|k} = u_{k+2|k} = \dots, \quad (2)$$

where $y_{k+n|k}, u_{k+n|k}$ are the n-step ahead predicted values for the output and input respectively made at sample k .

In order to solve (2), the dependence of the output predictions on the assumed values $u_k = u_{k+i|k}$, $i \geq 0$ is needed. Prediction algebra is standard in the literature (e.g. Rossiter (2018)) so here we simply assume the solution can be given as:

$$y_{k+n|k} = P_n u_p + Q_n y_p + H_n \underline{u}_{\rightarrow k} + d_k; \quad u_p = \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \\ u_{k-n_b} \end{bmatrix}; \quad y_p = \begin{bmatrix} y_{k-1} \\ y_{k-2} \\ \vdots \\ y_{k-n_a} \end{bmatrix}; \quad \underline{u}_{\rightarrow k} = \begin{bmatrix} u_k \\ u_{k+1|k} \\ u_{k+2|k} \\ \vdots \end{bmatrix}, \quad (3)$$

for suitable P_n, Q_n, H_n, n_b, n_a and d_k is a term to ensure unbiased prediction (typically taken as the difference between the process measurement and an internal model output,

although these details are not central here). Note that from (2) we can write $\underline{u}_{\rightarrow k} = Lu_k$, $L = [1, 1, \dots, 1]^T$. Substituting prediction (3) into (1,2) the PFC control law can be defined as:

$$\{P_n u_p + Q_n y_p + H_n L u_k + d_k = R - (R - y_k) \lambda^n\} \Rightarrow u_k = \frac{R - (R - y_k) \lambda^n - P_n u_p - Q_n y_p - d_k}{H_n L}. \quad (4)$$

Remark 2.2. *A main selling point of PFC is the computational simplicity of control law (4). Given H_n, P_n, Q_n are needed for just a single horizon n , the computation of these can be relatively trivial and thus the overall coding requirements are elementary (Richalet et al., 2009).*

2.2. Tuning of PFC

The industrial popularity of PFC is partially down to the intuitive tuning parameters. The designer, at least in principle, chooses the desired closed loop time constant (equivalently λ). The computer can then do a quick search over different choices of coincidence horizon n , displays the associated responses and then the user can determine which value n gives the most desirable closed-loop behaviour. However, herein lies two major weaknesses (Khadir & Ringwood, 2005, 2008; Rossiter & Haber, 2015).

- (1) Often the actual closed-loop performance/dynamics are not close (Rossiter et al., 2016; Zabet et al, 2017) to the chosen pole λ which of course draws into question the value of this as a tuning parameter (main exceptions are when $n = 1$ can be chosen which is generally true only for first order plant).
- (2) An offline search over different coincidence horizons is somewhat clumsy and difficult to argue as systematic and gives no assurance that a reasonable answer will result.

Remark 2.3. *It is easy to show that PFC suffers from the prediction mismatch (Rossiter, 2018) common in open-loop MPC approaches whereby the optimised predictions may bear little resemblance to the closed-loop behaviour that results. This inconsistency can (not must) lead to poor decision making. The mismatch arises because the prediction assumption on the future input, that this remains constant, is in many cases inconsistent with the actual input trajectory that arises or indeed is required for good behaviour.*

In summary, where a process has close to first order dynamics, PFC works very well. However, as the open-loop dynamics differ more from a first order system, the usefulness of λ as a tuning parameter reduces and the selection of an appropriate coincidence horizon become less obvious. This paper seeks to propose some alternative formulations to reduce these weaknesses and, as will be noted, can be very beneficial when it comes to constraint handling.

3. Systematic constraint handling in PFC with recursive feasibility

Given PFC deploys only very simple coding to enable use on low level processors, the constraint handling is defined to be simple and thus avoids the optimisers common in more mainstream algorithms and instead uses approaches which are simpler even

than reference governer strategies (Fiani & Richalet, 1991; Gilbert & Tan, 1991).

Assume constraints, at every sample, on input and states as follows:

$$\underline{\Delta u} \leq \Delta u_k \leq \overline{\Delta u}; \quad \underline{u} \leq u_k \leq \overline{u}; \quad \underline{y} \leq y_k \leq \overline{y}, \quad (5)$$

where $\Delta u_k = u_k - u_{k-1}$ is the input increment or rate.

The simplest PFC approach deals only with input constraints and deploys a saturation approach, that is, if the proposed u_k violates (5), then move to the nearest value which does not. Readers should note that this saturation approach automatically avoids issues with integral windup and the like which can occur when using PID. Also, for systems with stable open-loop dynamics, this approach is usually safe, albeit potentially suboptimal.

The historic PFC literature (Richalet et al., 2011) deploys more involved strategies to cater for state constraints which are akin to reference governer approaches (Gilbert & Tan, 1991), though perhaps a little more cumbersome. The core principle is to deploy nested or parallel PFC loops. The outer loop supplies the target to the inner loop and is used to modify this target when there is an expectation that an unmodified target will lead to a constraint violation. Such an approach requires design and tuning of the outer loop, but also is inherently simplistic and not designed to consider a multitude of different constraints as in (5); consequently there is clear potential in modernising that approach.

First we summarise a core concept adopted as standard in the MPC literature for constraint handling and propose to use this concept in place of the historical PFC choices.

- (1) For a suitable horizon m , compute the entire set of future predictions $y_{k+i|k} = P_i u_p + Q_i y_p + H_i \underline{u}_{\rightarrow k} + d_k$, $i = 1, \dots, m$. Use the compact notation $\underline{y}_{\rightarrow k+1} = P u_p + Q y_p + H \underline{u}_{\rightarrow k} + L d_k$ to capture the output predictions in a single vector where $\underline{y}_{\rightarrow k+1} = [y_{k+1|k}, y_{k+2|k}, \dots]^T$.
- (2) Combine the output constraints, output predictions and input constraints into a single set of linear inequalities of the form:

$$C u_k \leq f_k, \quad (6)$$

$$C = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ HL \\ -HL \end{bmatrix}; \quad f = \begin{bmatrix} \overline{u} \\ -\underline{u} \\ \overline{\Delta u} \\ -\underline{\Delta u} \\ L\overline{y} \\ -Ly \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ u_{k-1} \\ -u_{k-1} \\ Pu_p + Qy_p + Ld_k \\ -Pu_p - Qy_p - Ld_k \end{bmatrix},$$

where f_k depends on past data in u_p, y_p and on the limits. The horizon for the output predictions $\underline{y}_{\rightarrow k+1}$, and thus the row dimension of H , should be long enough to capture all core dynamics!

- (3) The predictions satisfy constraints iff (6) is satisfied and thus a conventional MPC algorithm will ensure this occurs and that is consider inequalities $C u_k \leq f_k$ explicitly rather than an alternative constraint representation which may be suboptimal or approximate.

The proposed PFC constraint handling algorithm is summarised next. This uses a single simple loop to select the u_k closest to the unconstrained solution of (4) which satisfies (6).

Algorithm 3.1. *At each sample:*

- (1) Define the unconstrained value for u_k from (4).
- (2) Define the vector f_k of (6) (it is noted that C does not change).
- (3) Use a simple loop covering all the rows of C as follows:
 - (a) Check the i th constraint that is the i th row of $Cu_k \leq f_k$ using $a_i = C_i u_k - f_{k,i}$.
 - (b) If $a_i > 0$, then set $u_k = (f_{k,i})/C_i$, else leave u_k unchanged.

Theorem 3.1. *In the nominal case (when there is no change in d_k) and for stable open-loop processes, Algorithm 3.1 is guaranteed to be recursively feasible and moreover converge to a feasible value for u_k that is closest to the unconstrained choice.*

Proof. Assume feasibility at initiation and also note that for stable open-loop processes the predicted outputs are convergent for constant future inputs $u_{k+i} = u_k, \forall i > 0$. Consequently, if one has feasibility at sample $k - 1$, then the choice $u_k = u_{k-1}$ must be feasible, that is satisfy (6). Hence, as long as u_{k-1} is a possible choice (which it must be as all constraints must satisfy $C_i u_{k-1} \leq f_{k,i}$), recursive feasibility is assured and a feasible solution will lie between u_{k-1} and the unconstrained u_k . Each constraint $C_i u_k \leq f_{k,i}$ will either lower or upper bound u_k ; if $u_k < u_{k-1}$ then only the lower bounds can be active and if $u_k > u_{k-1}$ only the upper bounds. Hence, an active constraint $C_i u_k \leq f_{k,i}$ will bring u_k closer to u_{k-1} if violated by the unconstrained u_k but otherwise will have no affect. In consequence, the final u_k will be only as close to u_k as it needs to be to satisfy all the active constraints and thus, is also as close to the original unconstrained u_k as possible. \square

Remark 3.1. *Because this approach (Algorithm 3.1) deploys a very simple for-loop, coding is simple and very fast and certainly far more simple than traditional MPC approaches which often use a quadratic program but equally, more systematic and probably quicker than the ad-hoc approaches common with PID. Nevertheless, the usage is only limited for single input single output (SISO) process, as PFC is rarely used to control a multi input multi output (MIMO) system due to its limited capability (Richalet et al., 2009). Nevertheless, this offset only occurs in the implied prediction, where the closed-loop response will only used the first sample of the input. Since the manipulated input u_k value is updated at each sample time, the final output still converges to the steady state target R .*

4. PFC using Laguerre functions

The main weakness of conventional PFC was the assumption with the predictions that the future input is constant. This same weakness is present in conventional MPC algorithms such as GPC and in fact equivalent restrictions also exist in the d.o.f. within dual-mode strategies. In an effort to ameliorate these and instead propose future input trajectories which were likely to be closer to those required in closed-loop, a few authors considered Laguerre function parameterisations (Khan & Rossiter, 2013; Rossiter et al., 2010; Wang, 2009). It was shown that despite being a relatively simple change in

formulation, this helped significantly with trade-offs between the number of d.o.f. in the prediction class and the feasibility (ability to deal with constraints).

The purpose of this paper is to propose and demonstrate the potential benefits of a similar concept when applied to PFC. However we should note some fundamental differences:

- In PFC we deploy just one d.o.f. and thus can use just a single Laguerre function.
- A different trade off will be investigated, that is between tuning parameter λ and closed-loop performance achieved during both unconstrained scenarios and constraint handling scenarios

4.1. Definition of input trajectories using Laguerre functions

As we are not using the whole set of Laguerre functions and just a single one, it is easier just to state that single function. For a given pole a , the first Laguerre function is given as:

$$L_a(z) = L_a(1)[1 + az^{-1} + a^2z^{-2} + a^3z^{-3} + \dots]. \quad (7)$$

(Typically $L(1) \neq 1$ when defining multiple Laguerre functions (Rossiter et al., 2010) although for this paper this detail is optional.) An underlying assumption within this paper is that the closed-loop input will converge to the steady-state with close to first order dynamics and thus with dynamics that can be represented by $L_a(z)$ plus some constant w . That is, ideally the future input predictions are defined as:

$$u_k = w_k + \eta_k; \quad u_{k+1|k} = w_k + a\eta_k; \quad u_{k+2|k} = w_k + a^2\eta_k; \quad \dots, \quad (8)$$

where η_k is a scaling factor to be selected on-line, and w_k is a value to be defined rather than a d.o.f.. The reader will note that in effect w_k is the implied steady-state/asymptotic value for u_{k+i} within the predictions.

An almost equivalent definition could use the input increments and hence:

$$\Delta u_k = \nu_k; \quad \Delta u_{k+1|k} = a\nu_k; \quad \Delta u_{k+2|k} = a^2\nu_k; \quad \dots, \quad (9)$$

although in this case the implied input trajectory would be:

$$u_k = u_{k-1} + \nu_k; \quad u_{k+1|k} = u_{k-1} + (1+a)\nu_k; \quad u_{k+2|k} = u_{k-1} + (1+a+a^2)\nu_k; \quad \dots. \quad (10)$$

Next note the properties of the geometric sequence $1, 1+a, 1+a+a^2, \dots$. It is known that

$$S_a = \sum_{i=0}^{\infty} a^i = \frac{1}{1-a} \quad \text{and} \quad \sum_{i=0}^n a^i = \frac{1-a^{n+1}}{1-a} = (1-a^{n+1})S_a. \quad (11)$$

Lemma 4.1. *The choices of (8,9) are not exactly equivalent and thus would lead to different results in general.*

Proof. Consider the implied control increments with the choices of (8,9). First the

choice (8) gives:

$$\begin{bmatrix} u_k \\ u_{k+1|k} \\ u_{k+2|k} \\ \vdots \end{bmatrix} = \begin{bmatrix} w_k + \eta_k \\ w_k + a\eta_k \\ w_k + a^2\eta_k \\ \vdots \end{bmatrix} \Rightarrow \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \Delta u_{k+2} \\ \vdots \end{bmatrix} = \begin{bmatrix} w_k - u_{k-1} + \eta_k \\ (a-1)\eta_k \\ (a^2-a)\eta_k \\ \vdots \end{bmatrix}. \quad (12)$$

Using an equivalent notation, we can rewrite sequence (10) as follows:

$$\begin{bmatrix} \Delta u_k \\ \Delta u_{k+1|k} \\ \Delta u_{k+2|k} \\ \vdots \end{bmatrix} = \begin{bmatrix} \nu_k \\ a\nu_k \\ a^2\nu_k \\ \vdots \end{bmatrix} \Rightarrow \begin{bmatrix} u_k \\ u_{k+1|k} \\ u_{k+2|k} \\ \vdots \end{bmatrix} = \begin{bmatrix} u_{k-1} + \nu_k \\ u_{k-1} + (1-a^2)S_a\nu_k \\ u_{k-1} + (1-a^3)S_a\nu_k \\ \vdots \end{bmatrix}. \quad (13)$$

Hence it is clear that there is a significant difference in these choices of parameterisation:

- (1) Choice (8) allows the first increment Δu_k to be out of proportion to the remaining increments whereas for choice (9) this ratio is fixed. Choice (8) may be better where a disproportionately large (or small) first increment is needed.
- (2) Irrespective of the choice of w , the relative sizes of respective elements are the same for all except the first increment, so for example, taking (12,13) in turn:

$$\left\{ \frac{\Delta u_{k+i|k}}{\Delta u_{k+i+1|k}} = \frac{1}{a} \right\} \quad \text{or} \quad \left\{ \frac{\Delta u_{k+i|k}}{\Delta u_{k+i+1|k}} = \frac{a^i - a^{i-1}}{a^{i+1} - a^i} = \frac{1}{a} \right\}. \quad (14)$$

Hence the two choices would be equivalent if and only if both $w_k - u_{k-1} + \eta_k = \nu_k$ and $(a-1)\eta_k = a\nu_k$ which would require a specific choice of w_k . \square

Theorem 4.2. *The choice of parameterisation (8) has more flexibility than the choice (9) and thus, in general, is to be preferred.*

Proof. This follows immediately from the previous Lemma. Appropriate choices of η_k, ν_k make the increments identical for all bar the first, that is Δu_k . In this instance, parameterisation (8) has an additional d.o.f. w_k which can be exploited if desired and if not assigned an equivalent value to that implied in (9). \square

Proposal: As was noted above, w_k is the asymptotic value of the input within the predictions and hence an obvious choice of w_k which eliminates the need for more design decisions is to set $w_k = E[u_{ss}]$, that is the expected steady-state input value required to remove offset. This would ensure the output predictions have zero offset asymptotically.

Corollary 4.3. *The prediction classes for PFC given in (2) and with Laguerre based on input increments (9) suffer from a critical weakness. In both cases the asymptotic value of the predicted output (not to be mixed up with the closed-loop output which has no asymptotic offset assuming stability) is highly unlikely to be close to the desired target of R . This is because, the value of u_k satisfying the PFC law definition $r_{k+n} = y_{k+n}$ in general will be inconsistent with $u_k = E[u_{ss}]$.*

4.2. Unbiased prediction and steady-state estimation

The output predictions with a single control increment were given in (3). With Laguerre based predictions, that is where \underline{u}_k is taken from, for example (12), this prediction can be reformulated as:

$$y_{k+n|k} = P_n u_p + Q_n y_p + H_n \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \end{bmatrix}}_{H_{nw}} w_k + H_n \underbrace{\begin{bmatrix} 1 \\ a \\ a^2 \\ \vdots \end{bmatrix}}_{H_{na}} \eta + d_k. \quad (15)$$

The reader will note from Corollary 4.3 that w_k is defined using the expected steady-state. In simple terms, a typical PFC computation of this value is given from:

$$y_{ss} = G_{ss} u_{ss} + d_k \quad \rightarrow \quad u_{ss} = \frac{y_{ss} - d_k}{G_{ss}} = \frac{R - d_k}{G_{ss}}, \quad (16)$$

where G_{ss} is the model steady-state gain, y_{ss} is the desired output steady-state (typically R) and d_k the offset between the model output and process measurement.

Remark 4.1. *Since the value of d_k is updated at each sample, the Independent Model (IM) structure is capable to handle some level of disturbance and parameter uncertainty and give zero asymptotic offset. Although the loop and its signals are impacted by measurement noise, as measurement noise is typically assumed to be a zero mean random variable, the impact of this on offset is typically ignored in the MPC literature. Of course there is an impact on loop sensitivity which is not a topic of this paper but an interested reader may refer to the work of Abdullah & Rossiter (2018b).*

4.3. Two PFC algorithms using Laguerre function predictions

Having defined both the input and output predictions, the PFC algorithm follows the same lines as in section 2, that is, choose the d.o.f. η such that (1,2) are satisfied.

Algorithm 4.1. LPFC: *The PFC control law using a Laguerre parameterisation of the input trajectory is defined as follows:*

- (1) Define the input trajectory from (12) with $w_k = u_{ss,k}$ as defined in (16).
- (2) Define the output prediction n -steps ahead using (15).
- (3) Substituting prediction (15) into (1,2) the PFC control law can be defined as:

$$P_n u_p + Q_n y_p + H_{nw} w_k + H_{na} \eta + d_k = R - (R - y_k) \lambda^n, \quad (17)$$

$$\Rightarrow \eta_k = \frac{R - (R - y_k) \lambda^n - P_n u_p - Q_n y_p - H_{nw} w_k - d_k}{H_{na}}; \quad u_k = w_k + \eta_k.$$

For completeness, as this has some relevance with constraint handling, a second algorithm is also defined.

Algorithm 4.2. LPFC2: An alternative PFC control law using a Laguerre parameterisation of the input increment trajectory is defined as follows:

- (1) Define the input trajectory from (13).
- (2) Define the output prediction n -steps ahead using (15) with (13) and hence define

$$y_{k+n|k} = P_n u_p + Q_n y_p + H_{nw} u_{k-1} + \underbrace{H_n \begin{bmatrix} 1 \\ (1-a^2)S_a \\ (1-a^3)S_a \\ \vdots \end{bmatrix}}_{H_{nn}} \nu_k + d_k. \quad (18)$$

- (3) Substituting output predictions (18) into (1,2) the PFC control law can be defined as:

$$P_n u_p + Q_n y_p + H_{nw} u_{k-1} + H_{nn} \nu_k + d_k = R - (R - y_k) \lambda^n, \quad (19)$$

$$\Rightarrow \nu_k = \frac{R - (R - y_k) \lambda^n - P_n u_p - Q_n y_p - H_{nw} u_{k-1} - d_k}{H_{nn}}; \quad u_k = u_{k-1} + \nu_k.$$

4.4. Constraint handling

The procedure for constraint handling is analogous to that discussed in section 3 and thus is not presented in detail. The core conceptual difference is that the d.o.f is now one, as either the parameter η_k or ν_k can be selected depending on the choice of input parameterisation and this means that, in principle, the input constraints need to be checked along the entire prediction horizon. However, given the maximum magnitude increments occur at the first sample, only Δu_k one needs to be checked and similarly, the maximum/minimum of u_{k+i} has a simple dependence on η_k, ν_k, w_k so again only one value needs to be checked.

As in Algorithm 3.1, the aim is to choose the d.o.f. as close as possible to their unconstrained values, and subject to (5). Nevertheless, there are some subtleties which are worth highlighting and link to feasibility.

Lemma 4.4. For the nominal case (where there is no large change in d_k), recursive feasibility is guaranteed with input prediction class (13), irrespective of the choice of target R .

Proof. Assuming feasibility at sample $k - 1$, then the choice $\Delta u_{k+i|k} = \Delta u_{k+i|k-1}$, $\forall i \geq 0$ will give rise to predictions which satisfy constraints (5). The choice $\nu_k = a \nu_{k-1}$ will enable this choice of future inputs and thus a feasible solution exists at the current sample and, clearly, this statement can be made recursively. \square

Lemma 4.5. For the nominal case, recursive feasibility is not guaranteed with input prediction class (12).

Proof. The potential weakness with input prediction class (12) is emphasised in the first term $u_k = u_{ss,k} + \eta_k$ as this contains a value, specifically $u_{ss,k}$ which may or may not be feasible. Moreover, consideration of the implied increments shows that $\Delta u_k = u_{ss,k} - u_{k-1} + \eta_k$ could be very large if there is a significant change in $u_{ss,k}$

(that is, $u_{ss,k} \neq u_{ss,k-1}$). To be more precise, the sequence of proposed inputs from the previous sample can be laid alongside the proposed sequence at the current sample:

$$u_{\rightarrow k|k-1} = \begin{bmatrix} u_{ss,k-1} + a\eta_{k-1} \\ u_{ss,k-1} + a^2\eta_{k-1} \\ u_{ss,k-1} + a^3\eta_{k-1} \\ \vdots \end{bmatrix} : u_{\rightarrow k|k} = \begin{bmatrix} u_{ss,k} + \eta_k \\ u_{ss,k} + a\eta_{k-1} \\ u_{ss,k} + a^2\eta_k \\ \vdots \end{bmatrix}. \quad (20)$$

From these it is clear that one can only ensure $u_{\rightarrow k|k-1} = u_{\rightarrow k|k}$ if $u_{ss,k} = u_{ss,k-1}$ (equivalently w_k) is unchanged. Without the ability to remain on the same prediction class, recursive feasibility cannot be assured. \square

Theorem 4.6. *In order to ensure recursive feasibility while using prediction class (12), the user must retain the option to modify $u_{ss,k}$ as required.*

Proof. It is a consequence of Lemma 4.4 whereby the option to choose $u_{ss,k} = u_{ss,k-1}$ enables the selection of $u_{\rightarrow k|k-1} = u_{\rightarrow k|k}$, hence guaranteeing feasibility. \square

Readers may note that the requirement in Theorem 4.6 is analogous to reference governor strategies (Gilbert & Tan, 1991) and is unsurprising and indeed also a well known issue within mainstream MPC. That is, large changes in the target can give rise to transient infeasibility where there is a terminal constraint as implicit with input trajectory (12) and this is easiest dealt with by slowing the change in target (equivalently modifying the implied terminal constraint). This paper will use examples to compare such a strategy with the use of (13) which is more analogous to GPC (Clarke & Mohtadi, 1989) in not having an implied terminal constraint and thus does not require this additional check.

5. Simulation studies

The simulation studies will be sectioned into three main themes and comparisons will be made between PFC, LPFC and LPFC2.

- (1) Compare the predictions arising from the different algorithms and the extent to which these give good expectations of good performance and consistent decision making.
- (2) Consideration of tuning efficacy and performance.
- (3) Consideration of constraint handling efficacy.

Several non-first order dynamics examples will be used to emphasise a variety of challenging characteristics (for example non-minimum phase and unstable dynamics), as it is known that conventional PFC is often inadequate for such systems. These examples are:

- (1) A second order non-minimum phase system with zero at 0.4:

$$G_1 = \frac{-0.04z^{-1} + 0.1z^{-2}}{1 - 1.4z^{-1} + 0.45z^{-2}}; \quad n = 5; \quad \lambda = 0.7. \quad (21)$$

(2) A third order over-damped system with poles at 0.9241, $0.9869 \pm 0.03339j$:

$$G_2 = 10^{-3} \frac{-0.1665z^{-1} + 0.0157z^{-2} - 0.1505}{1 - 2.898z^{-1} + 2.7993z^{-2} - 0.9012z^{-3}}; \quad n = 25; \quad \lambda = 0.96. \quad (22)$$

(3) A second order unstable system with poles at 0.63 and 1.27.

$$G_3 = \frac{0.1z^{-1} - 0.2z^{-2}}{1 - 1.9z^{-1} + 0.8z^{-2}}; \quad n = 5; \quad \lambda = 0.7. \quad (23)$$

Remark 5.1. *The selection of λ depends on a user preference where the range should be in between $0 < \lambda < 1$. This parameter is directly related to the desired closed-loop time response (CLTR), where $\lambda = e^{-3T_s/CLTR}$ and T_s is the sampling time (Richalet et al., 2009). Obviously, a smaller value of λ leads to a faster convergence and vice versa. For the coincidence horizon n , this work follows the conjecture given in (Rossiter & Haber, 2015) where the point is selected in between 40% to 80% rise of the step input response to the steady-state value.*

5.1. Prediction consistency and recursive decision making

A core underlying assumption in well posed MPC algorithms is that the *optimised* prediction at the current sample is reasonable and hence, could be re-used at the next sample. However, as these examples will show that is often not the case with a conventional PFC law due to the prediction assumption that the future input is constant. Where anything other than an open-loop dynamic is required, a constant input cannot deliver the desired dynamic and thus to embed this into the predictions automatically creates a mismatch in expectations.

Figure 1 shows the optimised predictions for example G_1 at the first sample from which it is clear that although all the predictions satisfy $r_{k+n} = y_{k+n}$ ($n = 5$) as defined by (2) but elsewhere they differ significantly from the target trajectory r_k within the exception of LPFC which remains close (apart from in early transients for which, due to the non-minimum phase characteristic, tracking r_k is impossible) and also has the correct asymptotic value. The input trajectories show that the flexibility in LPFC allows a large initial input to get a fast transient and then a gradual decay to the desired steady-state. By contrast, PFC tries to manage everything with a constant input and thus fails. LPFC2 has a different weakness: as the increments Δu_k all have the same sign, the required early increments to satisfy (2) inevitably lead to an asymptotic input trajectory which grows too large and ironically, also imply a less aggressive initial input move which could imply relatively slow transients compared to PFC and LPFC.

Figure 2 shows similar behaviour for example G_2 , although in this case the disparity between the target r_k is amplified much further due to the relatively slow underlying dominant dynamics of the open-loop poles (real part 0.98), and thus open-loop predictions, compared to the desired pole ($\lambda = 0.96$).

The figure for G_3 is omitted as, being open-loop unstable, the predictions are divergent. It so happens that, in the constraint free case, effective decision making may still result from using these predictions over a short output horizon as evidenced by the numerous examples in the literature using both GPC and PFC on open-loop unstable processes (e.g. Rossiter et al. (1998)). Moreover,

Remark 5.2. *It is worth noting that, an IM structure is not suitable for open-loop*

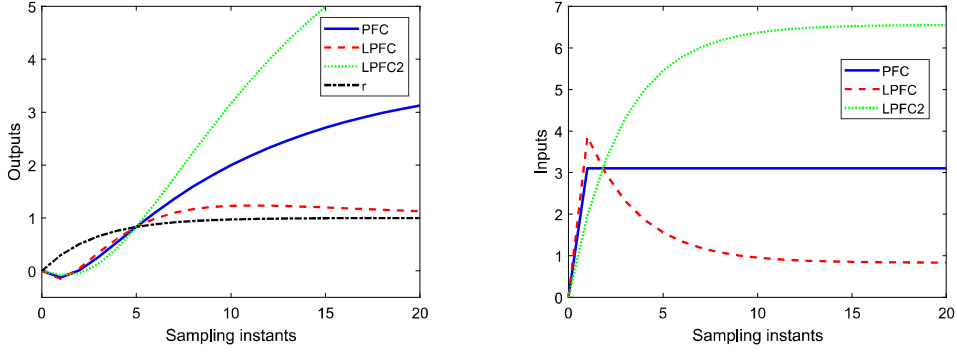


Figure 1. Output and input predictions for PFC, LPFC, LPFC2 for example G_1 .

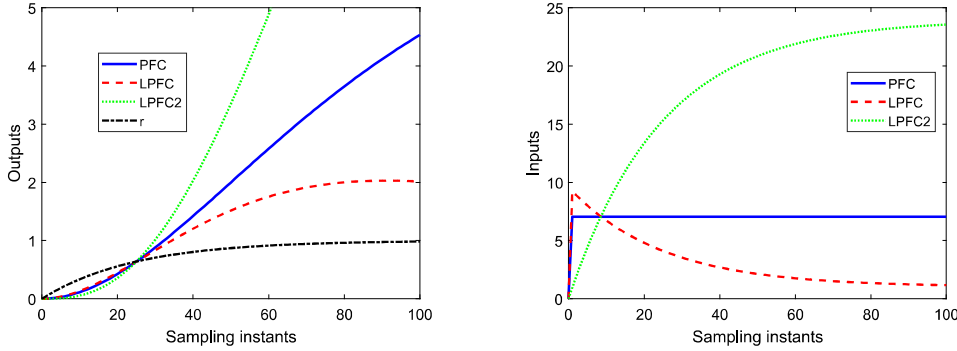


Figure 2. Output and input predictions for PFC, LPFC, LPFC2 for example G_2 .

unstable processes as would be evident should any uncertainty or noise be introduced. In such cases a classical PFC approach often deploys a cascade structure (Richalet et al., 2009), but for convenience, the results of this paper carry across directly as long as a suitable alternative prediction model is used (Clarke & Mohtadi, 1989; Rossiter, 2018). Such details are not core to this paper and thus omitted.

5.2. Tuning efficacy of PFC, LPFC and LPFC

The tuning efficacy of PFC has been discussed elsewhere (Rossiter & Haber, 2015; Rossiter et al., 2016) and so this section illustrates whether the proposed adaptations of LPFC, LPFC2 change any of those insights or not. The underlying issue is that, with the exception of cases where one can choose $n = 1$, the tuning parameter λ may have a weak correlation with the closed-loop pole that results.

Figures 3, 4, 5, 6 overlap the closed-loop behaviour with the original target for examples G_1, G_2, G_3 . From these figures three obvious conclusions are clear.

- (1) None of the algorithms is able to get close to the desired dynamic/target trajectory when $n \gg 1$.
- (2) LPFC is marginally faster during the intermediate transients than PFC whereas LPFC2 has slow initial transients but ultimately converges to the steady-state slightly more quickly.
- (3) Nevertheless, the closed-loop responses do speed up with the choice of smaller λ

(see figure 6).

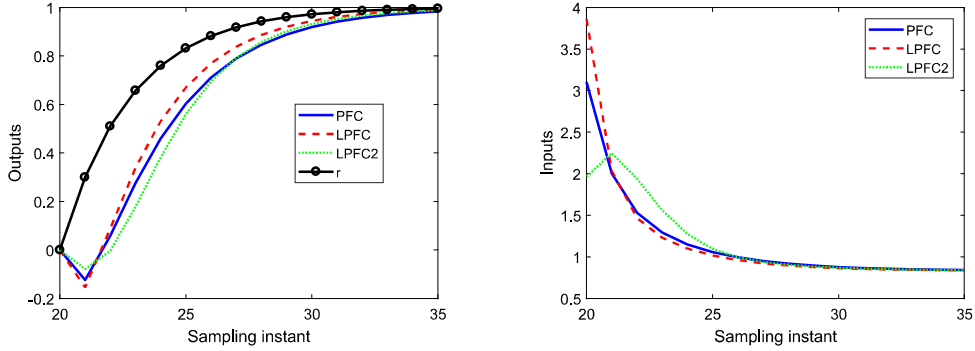


Figure 3. Closed-loop output and input behaviour for PFC, LPFC, LPFC2 for example G_1 .

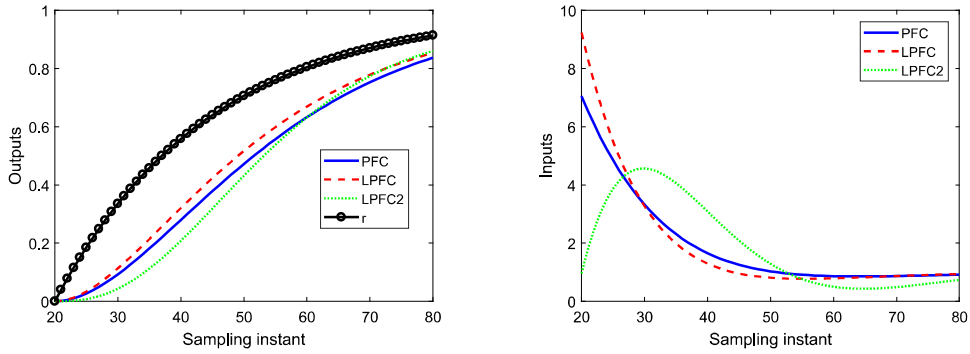


Figure 4. Closed-loop output and input behaviour for PFC, LPFC, LPFC2 for example G_2 .

5.3. Constraint handling

The reader may be puzzled as to why the improved prediction consistency illustrated in section 5.1 seems to have minimal impact on the efficacy of tuning and hence question whether the move to LPFC has any benefits? The answer to this becomes clear when one considers the constraint handling scenarios. When doing constraint handling and in particular wanting assurances of recursive feasibility (see section 4.4), consistency between predictions and the actual closed-loop behaviour is essential. The decision on how to limit u_k, η, ν_k to ensure predictions meet constraints will be ill-posed if those predictions are not representative and, of particular concern, these decisions could be unnecessarily conservative thus leading to far slower performance than necessary.

- This section will demonstrate how the inconsistency in PFC predictions can lead to extreme conservatism, whereas this is less likely to occur with LPFC.
- The section also shows a scenario where LPFC2 might be preferred to LPFC as it gives a much more straightforward mechanism for coping with large target changes.

Add constraints $\bar{u} = 1.2, \overline{\Delta u} = 0.5, \bar{y} = 1.1$ to example G_1 and limits $\bar{u} = 2, \overline{\Delta u} =$

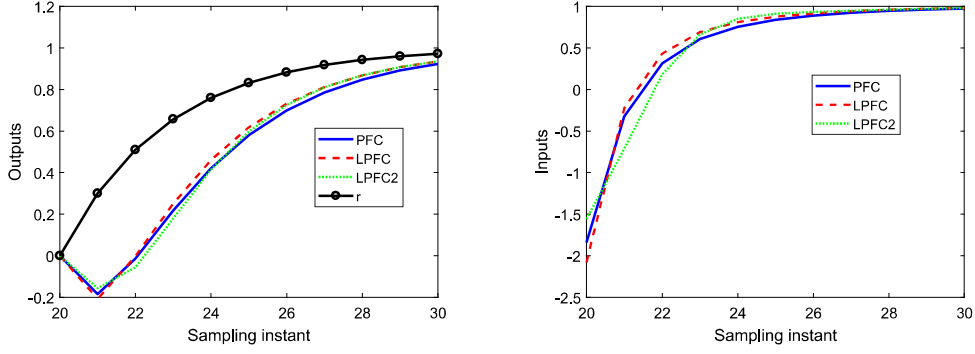


Figure 5. Closed-loop output and input behaviour for PFC, LPFC, LPFC2 for example G_3 .

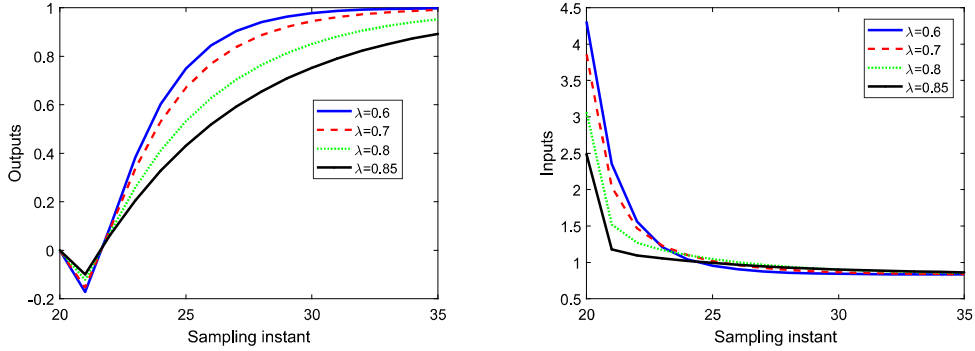


Figure 6. Closed-loop output and input behaviour for LPFC for example G_1 with various λ .

0.4, $\bar{y} = 1.1$ to example G_2 . Then the corresponding closed-loop simulations for a target of $R = 1$ are shown in Figures 7, 8. It is clear that LPFC has by far the best performance because it exploits the input most effectively. By contrast, because PFC assumes a constant future input, the input values available become highly restricted to be close to the steady-state because otherwise the long-range output predictions would exceed the upper output limit; this is obvious in the input figures where for PFC the input goes barely above its steady-state value. LPFC2 has a slow initial transient, again because the shape of input trajectory (9) will only meet the upper constraints in the long term if, for example, the first term $\Delta u_k = a\nu_k \ll \bar{u}$. However, in the medium term LPFC2 can exploit input values beyond u_{ss} and thus eventually converges in a timescale not dissimilar from LPFC.

Now consider a case where a simplistic implementation of LPFC fails, whereas PFC and LPFC2 do not. Add rate constraints to the input predictions for LPFC as given in (12).

$$\begin{bmatrix} |\Delta u_k| \\ |\Delta u_{k+1}| \\ \vdots \end{bmatrix} = \begin{bmatrix} |w_k - u_{k-1} + \eta_k| \\ |(a-1)\eta_k| \\ \vdots \end{bmatrix} \leq \begin{bmatrix} \overline{\Delta u} \\ \overline{\Delta u} \\ \vdots \end{bmatrix}. \quad (24)$$

Now, consider the case where $\overline{\Delta u} = 0.1$, $w_k = 1$, $u_{k-1} = 0$, $a = 0.8$. The first two

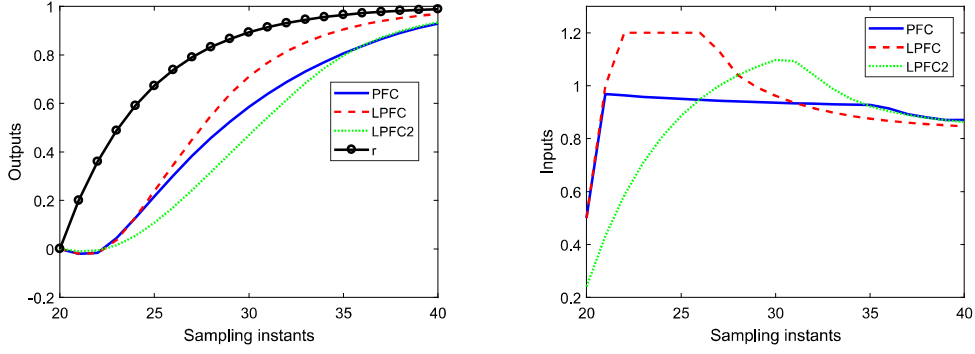


Figure 7. Closed-loop output and input behaviour for constrained LPFC for example G_1 .

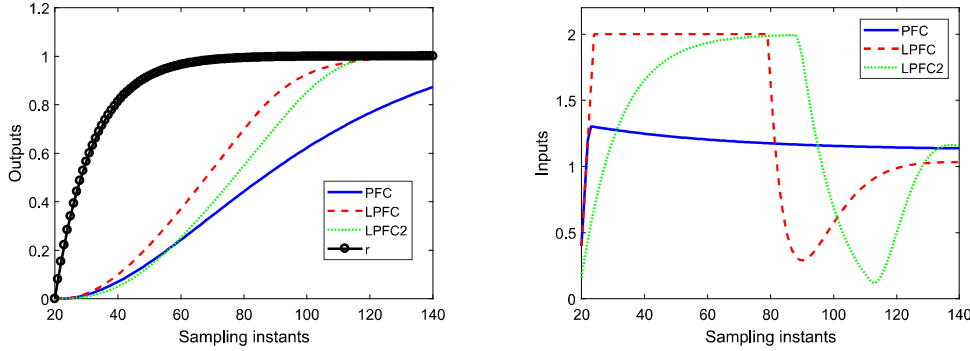


Figure 8. Closed-loop output and input behaviour for constrained LPFC for example G_2 .

inequalities reduce to:

$$\begin{bmatrix} |1 + \eta_k| \\ |-0.2\eta_k| \\ \vdots \end{bmatrix} \leq \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix} \Rightarrow \begin{matrix} -1.1 \leq \eta_k \leq -0.9 \\ |\eta| \leq 0.5 \end{matrix}. \quad (25)$$

Clearly these two conditions are inconsistent and thus LPFC is infeasible and the main factor is the input rate constraint. A simple scenario which would give rise to this inconsistency is when w_k changes significantly from one sample to the next (say due to a large change in set-point R). The requirement for the input sequence to both meet the new steady-state (while following the given decay rate) and beginning from the current u_{k-1} can easily be in conflict with rate constraints Δu . This issue is well understood in the main MPC literature and a reference governing or softening approach will be needed to avoid infeasibility.

Remark 5.3. *The reader may also be able to come up with scenarios where a large change in w_k gives rise to inconsistency between output constraints and other constraints. In general the easiest solution, which is standard in the mainstream MPC literature, is to relax the implied terminal constraint, that is slow the rate of change of w_k as much as required.*

6. Demonstrations on laboratory equipment

This section demonstrates and compares the performance of PFC and LPFC in controlling a real laboratory process that is a Quanser SRV02 servo based unit (see Figure 9). This system is powered by a Quanser VoltPAQX1 amplifier that comes with National Instrument ELVIS II+ multifunctional data acquisition device. The control algorithms are employed using National Instrument LabVIEW software in personal computer which is connected to the plant via USB wire.



Figure 9. Quanser SRV02 servo based unit.

The control objective for this case is to track the desired angular position of the servo, $\theta(t)$ by regulating the supplied voltage, $V(t)$. The mathematical model is derived based on differential equation and given as (for more detail explanation and derivation refer to Quanser (2012)):

$$J_{eq} \frac{d^2}{dt^2} \theta(t) + B_{eq} \frac{d}{dt} \theta(t) = A_m V(t), \quad (26)$$

where $J_{eq} = 0.00213 \text{ kgm}^2$ is the equivalent moment of inertia, $B_{eq} = 0.0844 \text{ Nms/rad}$ is the equivalent viscous damping parameter and $A_m = 0.129 \text{ Nm/V}$ is the actuator gain. By rearrange the model (26) and discretise it with sampling time $0.02s$, the transfer function of angular position to voltage input can be formed as:

$$G_4 = \frac{0.0095z^{-1} + 0.0073z^{-2}}{1 - 1.45z^{-1} + 0.45z^{-2}}. \quad (27)$$

Both of the controllers are tuned with $\lambda = 0.7$ and $n = 8$. Similar conclusions as in previous section can be seen from Figure 10, where LPFC is more effective in tracking the set point and converging faster (closer to the desired λ) than PFC.

Next, the following constraints are added to the process: $-6 \text{ V} < V < 6 \text{ V}$, $-3 \text{ V} < \Delta V < 3 \text{ V}$, $-0.8 \text{ rad} < \theta < 0.8 \text{ rad}$. Due to the advantage of employing better prediction consistency and well-posed decision making, LPFC manages to utilise the extra d.o.f in its future input trajectory to satisfy all the constraints better than PFC (refer Figure 11). It is notable that the constraint handling weaknesses implicit in PFC (linked to the assumed steady-state input in the predictions implying output constraint violations), means that PFC is very slow to converge to the output limit of 0.8 whereas LPFC is not affected by this weakness!

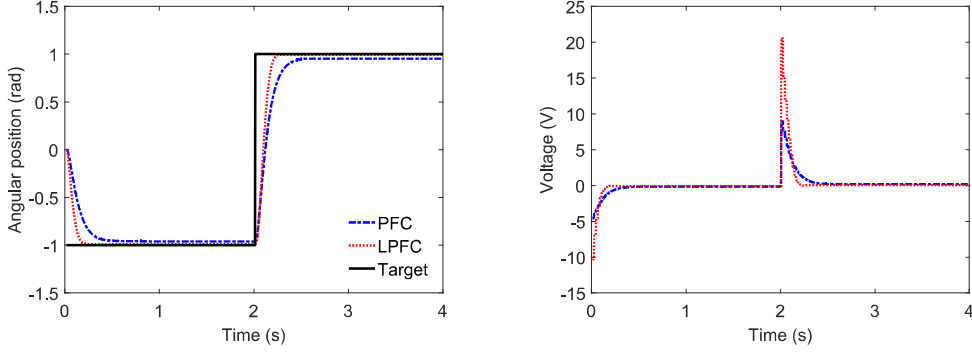


Figure 10. Performance of PFC and LPFC in tracking the Quanser SVR02 servo position.

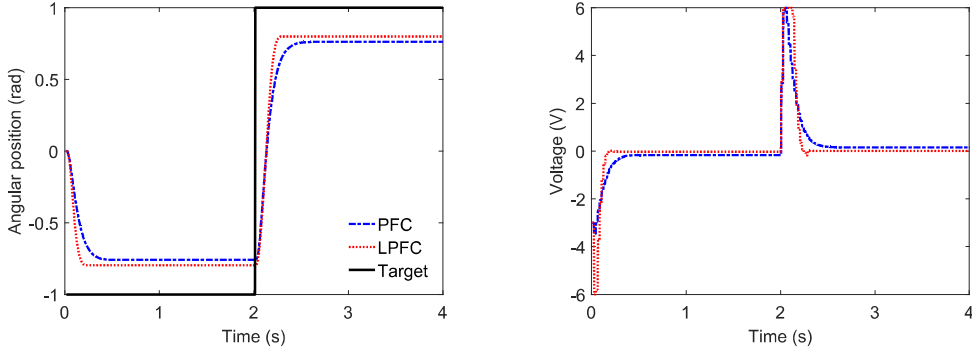


Figure 11. Constrained performance of PFC and LPFC in tracking the Quanser SVR02 servo position.

Remark 6.1. *The process used for this example has an integrator where one of its poles reside on the unit circle. Using the conventional PFC approach is not recommended since then the open loop prediction is divergent which inevitably will imply output constraint violations. The normal practice is to employ a cascade structure where the prediction is stabilised first before implementing the control law (Richalet et al., 2009). However, this cascade form of PFC is more complex to tune and has less clear cut properties so the discussion is beyond the scope of this paper. Conversely, LPFC can still be used since its future input dynamics will converge to zero assuming one uses the appropriate choice of $u_{ss} = 0$ (Abdullah & Rossiter, 2018a).*

Remark 6.2. *It is noted that there is a small offset error in Figure 10, which is far more obvious when using the conventional PFC. The main reason behind this is because the Independent Model (IM) structure used in the control law is not very effective in handling parameter uncertainty of the open-loop divergent process. This sensitivity can be improved by implementing other alternative prediction structures such as CARIMA or T-filter (Rossiter, 2018), but again the discussion of this topic is beyond the remit of this paper.*

7. Conclusions

This paper has taken the start point that Laguerre based input parameterisations have been effective for mainstream MPC algorithms and thus it is worth investigating their efficacy with PFC. We have proposed the use of Laguerre based predictions for PFC and evaluated the efficacy of this, compared to a more conventional PFC algorithm, using two different parameterisations: (i) one based on the inputs and (ii) another based on the input increments.

The most obvious conclusion is that using Laguerre functions can offer some clear benefits and more specifically, because it enables better consistency between the predictions at each sample and the resulting closed-loop behaviour, it enables more accurate constraint handling because satisfaction of constraints by the predictions is now a better representation of the actual future. By contrast, especially with regard to output constraints, a traditional PFC algorithm may have to adopt some quite conservative assumptions in order to ensure recursive feasibility and thus sacrifices performance.

The paper gives a proposal for a more formal, but computationally simple, constraint handling policy which has more rigour than traditional PFC constraint handling approaches. Then using this, a comparison of the two alternative parameterisations indicates that in most cases, mapping Laguerre directly onto the inputs is preferable to mapping onto the input increments as this enables faster transients and a better usage of the full input range. However, the one downside is the need to utilise an implied terminal constraint and, as is well known in the literature, terminal constraints can cause conflicts with other constraints and thus, at times, need to be managed carefully. Further investigations into computationally simple and efficient ways of doing this for PFC form an immediate future work.

Some other aspects which would be interesting to address next include: (i) to what extent does using a Laguerre parameterisation change, for better or worse, the underlying loop sensitivity compared to conventional PFC and (ii) is there potential to exploit other input parameterisations and if so, how can one choose these systematically and in a computationally simple manner.

References

- Abdullah, M., Rossiter, J. A. & Haber, R. (2017). Development of constrained predictive functional control using Laguerre function based prediction. In *2017 20th IFAC World Congress*, (p. 10705–10710).
- Abdullah, M. & Rossiter, J. A. (2018). Alternative method for predictive functional control to handle an integrating process. In *2018 UKACC Control Conference*.
- Abdullah, M. & Rossiter, J. A. (2018). A formal sensitivity analysis for Laguerre based predictive functional control. In *2018 UKACC Control Conference*.
- Camacho, E. F. & Bordons, C. (1999). *Model Predictive Control*. Berlin: Springer.
- Changenet, C., Charver, J. N., Gehin, D., Sicard, F. & Charmel, B. (2008). Predictive functional control of an expansion valve for controlling the evaporator superheat, *IMechE Journal of Systems and Control Engineering*, 222(6), 571–582.
- Clarke, D. W. & Mohtadi, C. (1989). Properties of generalized predictive control. *Automatica*, 25(6), 859–875.
- Fallasohi, H., Ligeret, C., & Lin-shi, X. (2010). Predictive functional control of an expansion valve for minimizing the superheat of an evaporator, *International Journal of Refrigeration*, 33, 409–418.
- Fiani, P. & Richalet, J. (1991). Handling input and state constraints in predictive functional

- control, in *Proceedings of the 30th IEEE Conference on Decision and Control*, (p. 985–990).
- Gilbert, E. & Tan, K. (1991). Linear systems with state and control constraints: The theory and application of maximal admissible sets. *IEEE Transactions Automatic Control*, 36, 1008–1020.
- Haber, R., Bars, R. & Schmitz, U. (2011). *Predictive control in process engineering: From the basics to the applications, Chapter 11: Predictive functional control*, Weinheim: Wiley-VCH.
- Haber, R., Schmitz, U. & Zabet, K. (2014). Implementation of PFC (predictive functional control) in petrochemical plant. In *2014 IFAC World Congress*, (p. 5333–5338).
- Haber, R., Rossiter, J. A. & Zabet, K. (2016). An alternative for PID control: Predictive functional control - A tutorial. In *2016 American Control Conference*, (p. 6935–6940).
- Khadir, M., & Ringwood, J. (2005). Stability issues for first order predictive functional controllers: Extension to handle higher order internal models. In *International Conference on Computer systems and Information Technology*, (p. 174–179).
- Khadir, M. & Ringwood, J. (2008). Extension of first order predictive functional controllers to handle higher order internal models. *Int. Journal of Applied Mathematics and Comp. Science*, 18(2), 229–239.
- Khan, B., & Rossiter, J.A. (2013). Alternative parameterisation within predictive control: A systematic selection. *International Journal of Control*, 86(8), 1397–1409.
- Rawlings J.B. & Mayne, D.Q. (2009). *Model predictive control: Theory and design*. Nob Hill Publication.
- Muehlebach, M. & DAndrea, R. (2017). Basis functions design for the approximation of constrained linear quadratic regulator problems encountered in model predictive control, in *2017 IEEE 56th Annual Conference on Decision and Control*, (p. 6189–6196) .
- Muske, K.R. & Rawlings, J.B. (1993), Model predictive control with linear models, *AIChE Journal*, 39(2), 262–287.
- Quanser user manual SRV02 rotary servo based unit set up and configuration. Quanser Inc, 2012.
- Richalet, J., Rault, A., Testud, J., & Papon, J. (1978). Model predictive heuristic control: Applications to industrial processes, *Automatica*, 14, 413–428.
- Richalet, J. (1993). Industrial applications of model based predictive control, *Automatica*, 29(5), 1251–1274.
- Richalet, J. & O’ Donovan, D. (2009). *Predictive functional control: Principles and industrial applications*. Springer-Verlag.
- Richalet, J. & O’ Donovan, D. (2011). Elementary predictive functional control: A tutorial. In *Int. Symposium on Advanced Control of Industrial Processes*, (p. 306–313).
- Rossiter, J.A., Wang, L., & Valencia-Palomo, G. (2010). Efficient algorithms for trading off feasibility and performance in predictive control. *International Journal of Control*, 83(4), 789–797.
- Rossiter, J.A., & Haber, R. (2015). The effect of coincidence horizon on predictive functional control, *Processes*, 3(1), 25–45.
- Rossiter, J.A. (2015). Input shaping for PFC: how and why?, *Journal of Control and Decision*, 1–14.
- Rossiter, J.A. (2016). A priori stability results for PFC. *International Journal of Control*, 90(2), 305–313.
- Rossiter, J.A., Haber, R., & Zabet, K. (2016). Pole-placement predictive functional control for over-damped systems with real poles, *ISA Transactions*, 61, 229–239.
- Rossiter, J.A., Kouvaritakis, B., & Rice, M. (1998). A numerically robust state-space approach to stable predictive control strategies. *Automatica*, 34(1), 65–73.
- Rossiter, J.A. (2018). *A first course in predictive control: 2nd edition*. CRC press.
- Scokaert, P.O. & Rawlings, J.B. (1998). Constrained linear quadratic regulation. *IEEE Transactions on Automatic Control*, 43(8), 1163–1169.
- Wang, L. (2009). *Model predictive control system design and implementation using MATLAB*. Springer Science & Business Media.
- Zabet, K., Rossiter, J.A., Haber, R., & Abdullah, M. (2017). Pole-placement predictive func-

tional control for under-damped systems with real numbers algebra, *ISA Transactions*, 71(2), 403–414.

Appendix E

**ALTERNATIVE METHOD FOR PREDICTIVE
FUNCTIONAL CONTROL TO HANDLE AN
INTEGRATING PROCESS**

M. Abdullah and J. A. Rossiter

This paper has been accepted for publication in:
The proceeding of 12th UKACC International Conference on Control 2018

Declaration form

ALTERNATIVE METHOD FOR PREDICTIVE FUNCTIONAL CONTROL TO HANDLE AN INTEGRATING
PROCESSES

(Proceeding of 12th UKACC International Conference on Control 2018)

Contributions of authors:

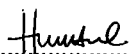
M. Abdullah

Provided the initial idea, formulations, codes, simulations and draft of this paper.

J. A. Rossiter

Supervised M. Abdullah and proofread the paper.

Signatures:



M. Abdullah

(First author)



J.A. Rossiter

(Second author)

Alternative Method for Predictive Functional Control to Handle an Integrating Process

Muhammad Abdullah* and John Anthony Rossiter[†]

^{*†}Department of Automatic Control and System Engineering,
University of Sheffield, Mappin Street, S1 3JD, UK.

Email: MAbdullah2@sheffield.ac.uk* and j.a.rossiter@sheffield.ac.uk[†]

^{*}Department of Mechanical Engineering,

International Islamic University Malaysia, Jalan Gombak, 53100, Kuala Lumpur Malaysia.

Email: mohd_abdl@iiu.edu.my*

Abstract—This work proposes an improved method for Predictive Functional Control (PFC) to handle an integrating process. Instead of assuming a constant future input, the dynamic is shaped with a first-order Laguerre polynomial so that it converges to the expected steady state value. This modification provides simpler coding and tuning compared to the conventional method in the literature. Simulation results show that the proposed controller improves the consistency of the open-loop prediction of an integrating process and thus improves closed-loop performance and constraint handling properties. The practicality of this algorithm is also validated on laboratory hardware.

Index Terms—Predictive Control, PFC, Laguerre Function, Constraints, Integrating Process, Transparent Control, Servo System.

I. INTRODUCTION

Predictive Functional Control (PFC) is a simple version of Model Predictive Control (MPC) developed in the early 1970s [1]. This algorithm only requires simple coding and low-level computation while retaining similar benefits to MPC in handling constraints and/or delays [2]. Despite its appealing characteristics, PFC receives relatively little interest in the literature [3] as it does not easily have rigorous properties such as stability assurances [4], [5] or robust feasibility [6]. However, the key selling point of PFC is the simplicity in tuning and implementation; it is a competitor to PID rather than the conventional predictive controller.

The simplistic PFC concept has several limitations, especially when dealing with an integrating process [2]. Due to their marginally stable dynamics, the constant future input assumption of PFC gives a divergent open-loop prediction [2], [7], [8]. Consequently, it may lead to poor closed-loop performance, prediction inconsistency, and also a failure in constraint implementation. Nevertheless, for low-level control applications where PID is unable to handle a constraint, PFC is still considered as an attractive option. Thus, the aim of this paper is to overcome some weaknesses while maintaining the formulation simplicity and cost effectiveness of PFC.

To deal with open-loop unstable plant, PFC practitioners often employ a cascade like structure known as transparent

control [2]. The inner loop consists of a proportional controller to stabilise the system predictions, and PFC provides the target trajectory via an outer loop. In practice, this structure often works better than PID within a constrained environment. However, the use of constant future input assumption can still lead to ill-posed decision making which impacts on both closed-loop performance [9] and constraint handling [10]. The interaction between inner and outer loops also makes the tuning and constraint implementation less transparent and, within the literature, there is no clear or systematic explanation of the approach.

Recent work has shown Laguerre PFC (LPFC) can improve closed-loop performance, prediction consistency and constraint handling for a stable system [9], [10]. This paper explores its capability to handle an integrating process. LPFC shapes the future input trajectory to converge to steady state with 1st order dynamics; this framework can stabilise the open-loop prediction of an integrating system without requiring a cascade structure, thus retaining a standard, and simpler, PFC formulation. Moreover, the Laguerre pole can be used to fine tune the closed loop performance [9], [11] and facilitate more reliable constraint management. The required modification is straightforward and thus in line with the simplicity requirement.

Section II gives some background on the nominal PFC and LPFC frameworks. Section III introduces the transparent control and LPFC law to handle an integrating process. Section IV provides a numerical example and analysis for both approaches. Section V validates the results with laboratory equipment simulations and section VI provides conclusions.

II. PFC FORMULATION FOR NOMINAL SYSTEM

This section only gives a brief review of conventional and Laguerre PFC formulations. For more detailed explanations of PFC theory and concepts can be found in these references, e.g. [2], [3], [7], [9]. To focus on the key conceptual contribution, the offset free correction is omitted from the formulation although it is applied in all the examples. Without loss of generality, all the control structures will use a general transfer function model for prediction. It is noted that the sensitivity of PFC to uncertainties constitutes future work.

This work is funded by International Islamic University Malaysia and Ministry of Higher Education Malaysia.

A. Traditional PFC

A PFC framework is based on simple human concepts and computes a required control action depending on how fast a user expects/desires the output to reach the target. There are two main components in the PFC formulation which are the desired target trajectory and system prediction. The control law is calculated by enforcing the following equality:

$$y_{k+n|k} = R - (R - y_k)\lambda^n \quad (1)$$

where $y_{k+n|k}$ is the n -step ahead system prediction at sample time k . The right hand side of (1) represents the desired trajectory of the output from y_k to the target value R with a convergence rate λ . The two main tuning parameters are:

- The coincidence horizon n defines the point where the system prediction matches the target trajectory.
- The desired closed-loop pole $\lambda = e^{-3T/CLTR}$, with T the sampling time and $CLTR$ the closed-loop time response.

Since the n -step ahead prediction algebra of a linear transfer function mathematical model is well known in the literature (e.g. [12]), here only the key results are provided. For inputs u_k and outputs y_k , the n -step ahead linear model prediction is given as:

$$y_{k+n|k} = H_n \underline{u}_k + P_n \underline{u}_k + Q_n \underline{y}_k \quad (2)$$

where parameters H_n , P_n , Q_n depend on the model parameters and for a model of order m :

$$\underline{u}_k = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+n-1} \end{bmatrix}; \underline{u}_k = \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \\ u_{k-m} \end{bmatrix}; \underline{y}_k = \begin{bmatrix} y_k \\ y_{k-1} \\ \vdots \\ y_{k-m} \end{bmatrix} \quad (3)$$

The control input is computed by substituting the prediction (2) directly into (1) to obtain:

$$H_n \underline{u}_k + P_n \underline{u}_k + Q_n \underline{y}_k = R - (R - y_k)\lambda^n \quad (4)$$

Adding a constant future input prediction assumption $u_{k+i|k} = u_k, i = 0, \dots, n$ and defining $h_n = \sum(H_n)$, the nominal PFC control law reduces to:

$$\underline{u}_k = \frac{R - (R - y_k)\lambda^n - (P_n \underline{u}_k + Q_n \underline{y}_k)}{h_n} \quad (5)$$

Remark 1: The tuning parameter λ should make the design process transparent, however the selection of coincidence horizon n affects the efficacy of λ due to the constant future input assumption [7]. With small horizons, the effectiveness of λ is more significant, but there may be poor prediction consistency with the target behaviour resulting in poor closed-loop behaviour. Conversely, larger horizons gives better prediction consistency but reducing the effectiveness of λ as a tuning parameter.

Remark 2: Prediction consistency is important for effective constraint handling thus, for some challenging dynamics, a constant input assumption may be ineffective [10].

B. Laguerre PFC (LPFC)

The LPFC approach utilises the expected constant steady-state input u_{ss} to eliminate the offset. The z -transform of discrete Laguerre polynomials are [11]:

$$L(z) = \sqrt{1-a^2} \frac{(z^{-1}-a)^{j-1}}{(1-az^{-1})^j}; \quad 0 < a < 1 \quad (6)$$

where j is the order of Laguerre function and a is the Laguerre pole which depends on a user selection between $0 < a < 1$. For simplicity of coding and concept, a first-order Laguerre polynomial is employed here, although high order polynomials may be used [11], [13], [14]. The function with altered scaling becomes an exponential decay as:

$$L(z) = \frac{1}{1-az^{-1}} \equiv 1 + az^{-1} + a^2 z^{-2} + \dots \quad (7)$$

Hence, the future input prediction is parametrised as:

$$u_{k+n} = u_{ss} + L_n \eta; \quad L_n = a^n; \quad L = [1, a, \dots, a^{n-1}] \quad (8)$$

where η is a degree of freedom. The parametrisation of (8) gives output predictions which converge to the steady-state exponentially with a rate a . The n step ahead output prediction is derived by substituting (8) into (2):

$$y_{k+n|k} = h_n u_{ss} + H_n L_n \eta_k + P_n \underline{u}_k + Q_n \underline{y}_k \quad (9)$$

Hence, the LPFC control law is defined by substituting prediction (9) into (1) and solving for parameter η as:

$$\eta_k = \frac{R - (R - y_k)\lambda^n - (P_n \underline{u}_k + Q_n \underline{y}_k) - h_n u_{ss}}{H_n L_n} \quad (10)$$

Due to the receding horizon principle [11] and the definition of $L(z)$ in (7), the current input u_k is:

$$u_k = u_{ss} + \eta_k \quad (11)$$

Remark 3: By shaping the future input dynamics, LPFC can improve prediction consistency, closed-loop performance and the efficacy of λ as a tuning parameter [9]. This improvement also provides a more accurate and less conservative solution when satisfying output/state limits [10].

III. PFC FORMULATION FOR INTEGRATING SYSTEM

An integrating process has marginally stable dynamics as one of the poles resides at the origin. This pole gives an extra challenge to the traditional PFC framework because the open-loop prediction does not converge when using the constant future input assumption. This section presents two alternative frameworks. The first subsection briefly reviews the proposed technique in the literature, so-called transparent control by [2], and the following subsection presents the proposed modification of LPFC for integrating processes. Currently, the concept is only introduced to a single integrator problem, while future work will consider a further generalisation for a plant with multiple integrators or marginally stable poles.

A. Transparent Control

Transparent control utilises two level of cascade structure (see Fig. 1). The inner loop employs a proportional gain with negative feedback to stabilise the open loop prediction, while nominal PFC controls the outer loop and eliminates any offset due to disturbance and enhances the overall dynamic performance [2].

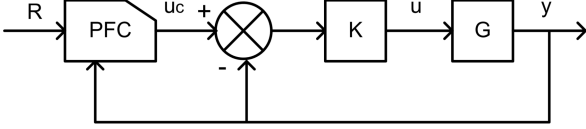


Fig. 1: Transparent PFC structure.

The inner loop with gain K will be used as a prediction model as in (12) instead of the plant model G to compute the manipulated input u_c .

$$y(s) = \frac{GK}{1 + GK} u_c(s) \quad (12)$$

The actual input u that will be send to the plant is:

$$u_k = K(u_{c,k} - y_k) \quad (13)$$

With this technique, the controlled system is able to maintain regulation during set-point changes by introducing a temporary over-compensated set-point [2]. At the same time, the outer loop will minimise the tracking error using a standard PFC formulation as discussed in section II-A.

Remark 4: Transparent PFC (TPFC) only accepts proportional gain rather than the combination with integral and/or derivative to keep the constraint implementation purely algebraic [2]. To implement input or rate constraints, a back calculation procedure is needed to transfer the information from the inner loop to the outer loop and avoiding saturation as:

$$y_k + \frac{u_{min}}{K} \leq u_{c,k} \leq y_k + \frac{u_{max}}{K} \quad (14)$$

$$y_k + \frac{\Delta u_{min} + u_{k-1}}{K} \leq u_{c,k} \leq y_k + \frac{\Delta u_{max} + u_{k-1}}{K} \quad (15)$$

Since the constraint is implemented at the current time only, there is no check that the implied predictions satisfy constraints in the future and thus recursive infeasibility may result.

Remark 5: For output or state constraints, the traditional practise utilises a multiple PFC regulators that run in parallel [2], [10]. The first regulator computes the preferred control action while the second regulator produces an input to satisfy the limit. A supervisor will choose the correct input for the plant. However, advances in computation technology mean this tedious *ad hoc* approach can be replaced with a more systematic, but simple, approach such as in [15].

B. Laguerre PFC for Integrating Process

Due to the pole on the origin, the steady state input for an integrating process is zero for a constant set point. These dynamics are still compatible with a LPFC law (section II-B) with the only required modification being to define $u_{ss} = 0$.

Theorem 1: The future input dynamics of

$$u(z) = \frac{u_{ss}}{1 - z^{-1}} + \frac{\eta_k}{1 - az^{-1}} \quad (16)$$

will give input predictions that settle exponentially at zero with a speed linked to Laguerre pole a . For an integrating process, the value η_k effects the implied steady-state outputs.

Proof: The signal defined in (16) has the property that

$$\lim_{k \rightarrow \infty} u_k = u_{ss} \Rightarrow \lim_{k \rightarrow \infty} y_k = R \quad (17)$$

When $u_{ss} = 0$, the steady-state output has affine dependence on the integral of the future input. \square

Remark 6: For simple first order system the value of a should be equal to λ [9]. However for a higher order system, selecting $a < \lambda$ will give faster convergence to steady state and thus can improve the prediction consistency.

Algorithm 1 (LPFC): For integrating process, a similar algorithm as in (10) is used except that u_{ss} term is removed.

$$\eta_k = \frac{R - (R - y_k)\lambda^n - (P_n u_k + Q_n y_k)}{H_n L_n} \quad (18)$$

Theorem 2: Using LPFC input predictions as defined in *Algorithm 1*, output, state and input constraints can be represented by a set of linear inequalities.

Proof: Output constraints can be constructed from the output predictions in (9) within the validation horizon n_i and comparison with the limits at each sample instant, e.g.:

$$y_{min} \leq H_{n_i} L_{n_i} \eta_k + P_{n_i} u_k + Q_{n_i} y_k \leq y_{max}, \forall i > 0 \quad (19)$$

The maximum/minimum input rate/value occur at the first sample, so input constraints can be formulated as:

$$u_{min} \leq \eta_k \leq u_{max} \quad (20)$$

$$\Delta u_{min} \leq \eta_k - u_{k-1} \leq u_{max} \quad (21)$$

Combining (19,20,21) it is clear that for suitable M, v_k one can represent the satisfaction of constraints by predictions using a single vector inequality of the form:

$$M \eta_k \leq v_k \quad \square \quad (22)$$

We can now define the constraint handling algorithm which is akin to methods given in [15].

Algorithm 2: [LPFC constrained] Use the unconstrained law (18) to determine the ideal value of η_k and check each constraint in (22) using a simple loop (subscripts denote position in a vector).

Set $u_{max} = \infty, u_{min} = -\infty$.

For $i=1$:end,

if $M_i \eta_k \not\leq v_i$ & $M_i > 0$ then define $u_{max} = v_i / M_i$,

if $M_i \eta_k \not\leq v_i$ & $M_i < 0$ then define $u_{min} = v_i / M_i$,

end loop.

if $\eta_k < u_{min}$, $\eta_k = u_{min}$. if $u_{max} < \eta_k$, $\eta_k = u_{max}$.

Define $u(k)$ using (16).

Note that the upper and lower limits to ensure recursive feasibility update at each cycle in the loop but as all the inequalities are only ever tightened, changes lower down cannot contradict changes higher up throughout the horizon.

Remark 7: Infeasibility can arise due to too fast or large changes in the target. However, LPFC helps in this case because the exponential structure embedded into the input prediction automatically slows down any over aggressive input responses and thus significantly increases the likelihood of feasibility being retained. In the worst case, set point changes need to be moderated as in reference governor approaches [16].

The summary benefits of this algorithm are:

- It offers a simple and systematic framework to handle an integrating process.
- It stabilises the output prediction without a cascade structure thus no back calculation process (*Remark 4*) is needed for input/rate constraints.
- The Laguerre pole a can be utilised to control the speed of convergence to improve the prediction consistency and efficacy of constrained solution.
- The implied structure of (16) in conjunction with constraints (22) means that a recursive feasibility guarantee (nominal case) is provided [15].

IV. NUMERICAL EXAMPLES AND ANALYSIS

This section presents a numerical example to demonstrate the benefit of using LPFC compared to TPFC. A first order servo system with integrator is considered as a plant where the control objective is to track the position. The discrete mathematical model with sampling time 0.02s is:

$$G = \frac{0.0095z^{-2} + 0.0073z^{-1}}{1 - 1.45z^{-1} + 0.45z^{-2}} \quad (23)$$

This simulation will focus on the tuning process and the concept of well-posed decision making that can be observed by comparing the open-loop prediction and closed-loop behaviour of the controller. In addition, the efficacy of constraint handling is also discussed in the last subsection.

A. Tuning and Performance of TPFC

The first step in implementing TPFC is to tune the proportional gain before selecting the coincidence horizon n . This gain K will determine the convergence speed and the steady state error of the inner-loop. Small gain leads to slower responses, while too large a gain causes oscillatory behaviour. The root-locus (continuous time) plot shows that the choice $K = 6.45$ is around a critical value in that higher K would give oscillatory poles (see Fig. 2).

The coincidence horizon is selected by comparing the step response with a desired first order target trajectory r with $\lambda = 0.74$ (refer to Fig. 2). Since the inner loop is second order (due to the added integrator), it is necessary [7] to choose a coincidence horizon in the range $3 \leq n \leq 8$; lower values are often preferable so here $n = 3$.

Fig. 3 shows the closed-loop and predicted (at sample $k = 0$) performance of TPFC with different values of gain K . The actual closed-loop behaviour is expressed as y (output) and u (input), while the implied predictions are denoted by signals output y_p and input u_p (corresponding to first value

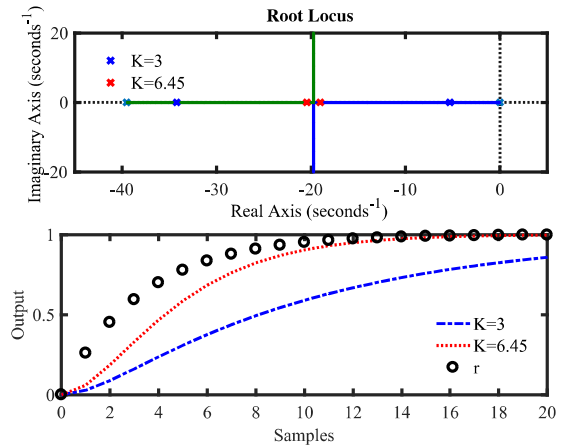


Fig. 2: Root locus and step response of $\frac{GK}{1+GK}$ with different gain K .

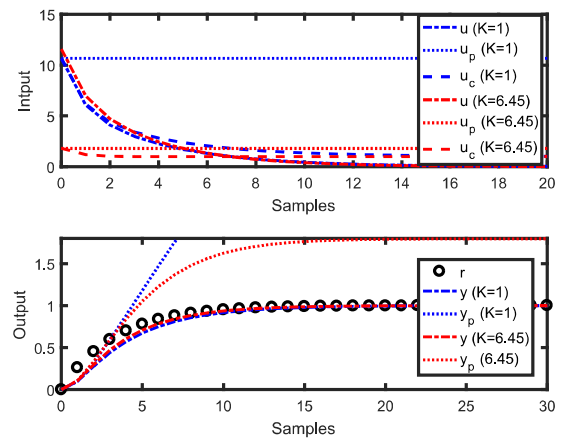


Fig. 3: Closed-loop and open-loop behaviour of G for TPFC with different K .

of input produced by PFC u_c instead of the actual input u). For both choices of K , in the unconstrained case, the closed-loop outputs y track the trajectory set point r and with almost equivalent speed. However, with $K = 1$, the implied prediction y_p has a very slow convergence in response to the constant input dynamics u_p and is inconsistent with the closed-loop behaviour y that results. With a higher gain value ($K = 6.45$), the consistency is improved but still poor. This inconsistency is likely to lead to severely flawed decision making should constraint handling be required.

B. Improvement on Prediction Consistency with LPFC.

For LPFC, due to the presence of an integrator, the coincidence horizon is selected based on the the impulse response (see Fig. 4) using the guidance of [7]; this suggests a value in the region of $n = 3$.

Ideally, for a first order system, the value of Laguerre pole a should be equal to the desired closed loop pole λ [10].

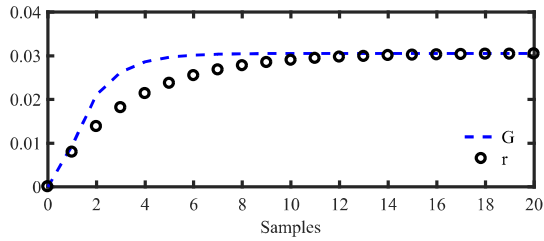


Fig. 4: Impulse response of G with target trajectory r

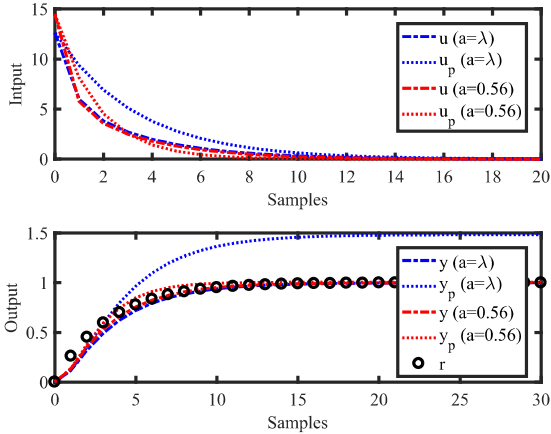


Fig. 5: Closed-loop and open-loop behaviour of G for LPFC with varying Laguerre pole a .

However, for a higher order system, this value needs to be further tuned as it has an impact on the convergence rate of the output prediction when tracking the first order target trajectory (*Theorem 1*). Fig. 5 shows that with a choice of $a = \lambda$, while the controller still tracks the set point well, there is still noticeable inconsistency between predictions y_p and the closed-loop behaviour y . However, reducing the pole to $a = 0.55$ improves the prediction consistency and the overall closed-loop performance is still good (*Remark 6*).

C. Improvement in Constrained Performance with LPFC

One of the key selling points of PFC is the computationally simple (low cost) constraint handling ability. When the system input is bounded to $u_{max} = 8$ (see Fig. 6), both of the controllers manage to track the set point and satisfy the given limit although LPFC gives a slightly better closed-loop performance due to the well posed decision. Moreover, the LPFC formulation is more straightforward to implement and does not require back calculation methods (*Remark 4*).

Fig. 7 shows the system response of both controllers when the output is limited to $y_{max} = 0.8$. The validation horizon is selected at $n_i = 10$ to cover most of the transient period and prevent constraint violation at the early stage. In this case the closed-loop response of TPFC is slower and more conservative in satisfying the limit due to the prediction inconsistency demonstrated in Fig 3. Conversely, LPFC which is based on more consistent predictions (see Fig 5) converges much faster

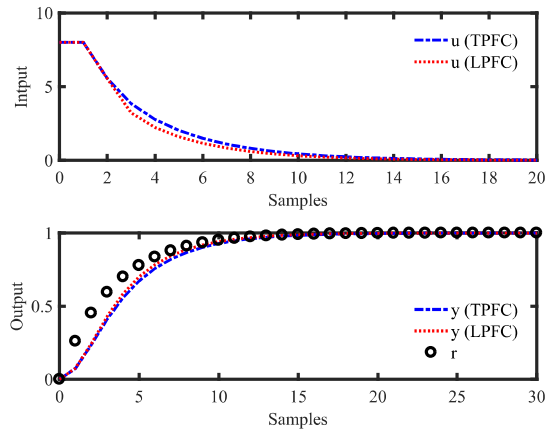


Fig. 6: Closed-loop response of G for LPFC and TPFC with bounded input ($u_{max} = 8$).

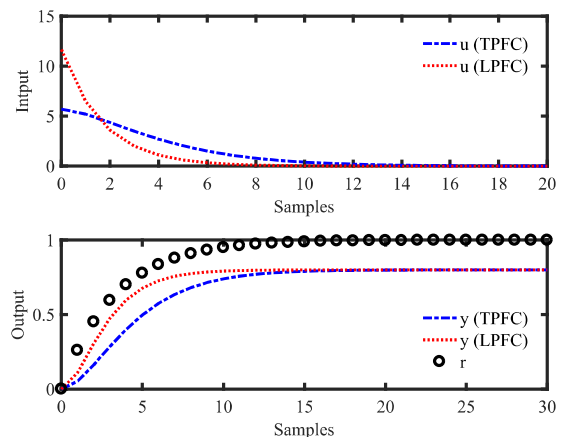


Fig. 7: Closed-loop response of G for LPFC and TPFC with bounded output ($y_{max} = 0.8$).

compared to TPFC. Clearly the constrained solution of LPFC is more accurate and less conservative.

V. IMPLEMENTATION ON REAL HARDWARE

To validate the practicality of LPFC, the algorithm is tested on a Quanser SRV02 servo based unit powered by a Quanser VoltPAQ-X1 amplifier (see Fig. 8). This system is operated by National Instrument ELVIS II+ multifunctional data acquisition. The plant is connected to a computer via a USB connection using NI LabVIEW software. The control objective is to track the servo position $\theta(t)$ by manipulating the supplied voltage $u(t)$. The mathematical model of this system is given as (for more details, refer to [17] user manual):

$$0.0254\ddot{\theta}(t) = 1.53u(t) - \dot{\theta}(t) \quad (24)$$

where $\ddot{\theta}(t)$ and $\dot{\theta}(t)$ are both servo angular acceleration and speed, respectively. Converting the continuous model in (24) to discrete form with sampling time 0.02 s, the transfer function of angular speed to voltage input becomes G as in (23).



Fig. 8: Quanser SRV02 servo based unit.

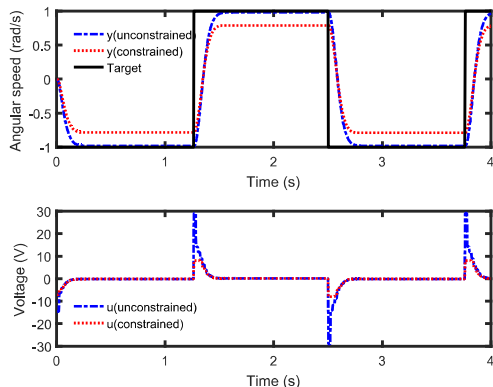


Fig. 9: Unconstrained and constrained performances of LPFC in tracking the Quanser SRV02 servo position.

The algorithm is employed with similar tuning parameters as in the previous numerical example ($\lambda = 0.74$, $n = 3$ and $a = 0.55$). Fig. 9 demonstrates the unconstrained and constrained performances of LPFC to track an alternating set point between -1 rad/s to 1 rad/s. For the unconstrained case, a similar performance to the simulation studies are obtained. The controller manages to provide a smooth tracking to the desired target while retaining the intuitive link between the target dynamic λ and the closed-loop convergences speed (CLTR = $0.2s$). In the constrained case, the implied input limits ($-8v \leq u_k \leq 8v$) and output limits ($-0.8 \leq y_k \leq 0.8$) are satisfied without any conflict by employing the systematic constraint method (*Algorithm 2*).

VI. CONCLUSIONS

This work proposes an alternative Laguerre PFC approach to control an integrating process. Since the traditional PFC formulation for integrating processes is unable to give a stable open loop prediction, the *transparent control* approach is often used. Although this cascade structure can stabilise the plant using a proportional controller, the decision making process may still be poorly posed, and notably can lead to a highly conservative solution in the presence of constraints. Conversely, by shaping input predictions using a Laguerre

polynomial, the nominal PFC method can be employed without a cascade structure. Besides, the improved prediction consistency of LPFC enables the constrained solution to become more accurate and less conservative, thus improving performance. This paper has also demonstrated the efficacy of the proposed LPFC algorithm on laboratory hardware with active constraints. Critically, the proposed algorithm is very simple to code and implement which in line with the core markets for PFC approaches.

Nevertheless, there is a potential weakness with LPFC especially when the independent model structure is used. A small offset error may occur if there is a model mismatch or the real plant is not in fact integrating. Future work aims to look more closely at this issue while providing a formal sensitivity analysis and systematic design of LPFC in handling uncertainty. Another important consideration is to analyse the alternative shaping methods which may be better tailored to deal with higher order and/or challenging dynamical.

ACKNOWLEDGMENT

The first author would like to acknowledge International Islamic University Malaysia and Ministry of Higher Education Malaysia for funding this work.

REFERENCES

- [1] J. Richalet, A. Rault, J.L. Testud and J. Papon, Model predictive heuristic control: applications to industrial processes, *Automatica*, 14(5), 413-428, 1978.
- [2] J. Richalet, and D. O'Donovan, *Predictive Functional Control: principles and industrial applications*. Springer-Verlag, 2009.
- [3] R. Haber, R. Bars, and U. Schmitz, *Predictive control in process engineering: From basics to applications, chapter 11*, Wiley-VCH, Germany, 2012.
- [4] J. Rossiter, A priori stability results for PFC, *IJC*, 90(2), 305-313, 2016.
- [5] P.O. Scokaert and J.B. Rawlings, Constrained linear quadratic regulation, *IEEE Trans. on Automatic Control*, 43(8), 1163-1169, 1998.
- [6] J. Mayne, M.M. Seron, and S. Rakovic, Robust Model Predictive Control of constrained linear systems with bounded disturbances, *Automatica*, 41(2), 219-224, 2005.
- [7] J. A. Rossiter, and R. Haber, "The effect of coincidence horizon on predictive functional control," *Processes*, 3, 1, pp. 25-45, 2015.
- [8] J. A. Rossiter, "Input shaping for PFC: how and why?," *J. Control and Decision*, pp. 1-14, Sep. 2015.
- [9] M. Abdullah and J. A. Rossiter, "Utilising Laguerre function in Predictive Functional Control to ensure prediction consistency," 11th Int. Conf. on Control, Belfast, UK, 2016.
- [10] M. Abdullah, J. A. Rossiter and R. Haber, "Development of constrained Predictive Functional Control using Laguerre function based prediction," IFAC World Congress, 2017.
- [11] L. Wang, *Model predictive control system design and implementation using MATLAB®*, Springer, 2009.
- [12] J. A. Rossiter, *Model predictive control: a practical approach*, CRC Press, 2003.
- [13] M. Abdullah, and M. Idres, Constrained Model Predictive Control of Proton Exchange Membrane Fuel Cell, *JMST*, 28(9), 3855-3862, 2014.
- [14] M. Abdullah, and M. Idres, Fuel cell starvation control using Model Predictive Technique with Laguerre and exponential weight functions, *JMST*, 28(5), 1995-2002, 2014.
- [15] J. Rossiter, B. Kouvaritakis, and M. Cannon, Computationally efficient algorithms for constraint handling with guaranteed stability and near optimality, *IJC*, 74(17), 1678-1689, 2001.
- [16] E. Gilbert and I. Kolmanovsky, Discrete-time reference governors and the non-linear control of systems with state and control constraint, *Int. J. Robust and Non-linear Control*, 5, 487-504, 1995.
- [17] *Quanser user manual SRV02 rotary servo based unit set up and configuration*. Quanser Inc, 2012.

Appendix F

**INPUT SHAPING PREDICTIVE FUNCTIONAL
CONTROL FOR DIFFERENT TYPES OF
CHALLENGING DYNAMICS PROCESSES**

M. Abdullah and J. A. Rossiter

This paper has been published in:

Processes 2018, Vol. 6

Declaration form

**INPUT SHAPING PREDICTIVE FUNCTIONAL CONTROL FOR DIFFERENT TYPES OF CHALLENGING
DYNAMICS PROCESSES**

(Processes 2018, Vol. 6)

Contributions of authors:

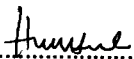
M. Abdullah

Provided the initial idea, formulations, codes, simulations and draft of this paper.

J. A. Rossiter

Supervised M. Abdullah and proofread the paper.

Signatures:



M. Abdullah
(First author)



J.A. Rossiter
(Second author)

Article

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

Muhammad Abdullah ^{1,2} and John Anthony Rossiter ^{1,*} 

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

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

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

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



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

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

1. Introduction

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

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

PFC practitioners often deploy a type of cascade structure to handle a challenging dynamics process, where an inner loop is used to improve the dynamics for an outer loop to control [14,15]. This modification enables a user to retain an independent model (IM) structure that handles uncertainties, while retaining a similar tuning concept as the conventional approach. Nevertheless, the inner PFC can only accept a proportional-type controller to avoid any overcomplication when

handling constraints [6]. This restriction makes it difficult to have a systematic selection of gain, rather than an ad hoc approach. Moreover, a back-calculation or anti-wind-up technique is required to avoid any saturation while satisfying the constraints [6], which also can be conservative and thus affect performance.

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

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

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

2. Conventional PFC

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

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

where $b(z) = b_1z^{-1} + \dots$, $a(z) = 1 + a_1z^{-1} + \dots$, y_k, u_k are the outputs and inputs, respectively, at sample k , and ζ_k is an unknown zero mean random variable used to capture uncertainty.

2.1. Control Law

PFC design is based on the assumption that a closed-loop response should follow a first-order dynamic from the current state y_k to the desired target r [20]. In practice, one aims to achieve this

by ensuring a matching using the open-loop prediction, but only at a single point. In other words, the predicted output trajectory is chosen to satisfy the following equality:

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

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

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

$$\underline{y}_{\rightarrow k+1|k} = H \Delta \underline{u}_k + P \Delta \underline{u}_k + Q \underline{y}_k \quad (3)$$

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

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

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

$$H_n \mathbf{e}_1 \Delta u_k + P_n \Delta \underline{u}_{\leftarrow k} + Q_n \underline{y}_{\leftarrow k} = (1 - \lambda^n)r + \lambda^n y_k \quad (5)$$

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

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

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

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

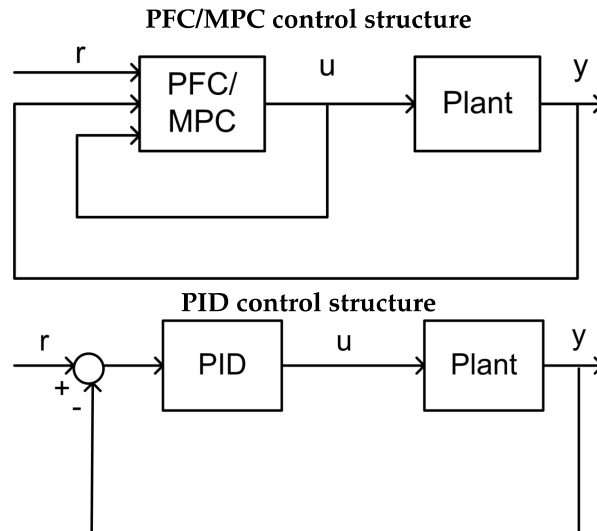


Figure 1. Comparison of flow diagrams for predictive functional control (PFC), model predictive control (MPC), and proportional integral derivative (PID).

2.2. Constraint Handling

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

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

The user needs to:

1. Find a suitable horizon m [23] over which to compute the entire set of future output predictions in a single vector: $\underline{y}_{\rightarrow k+1|k} = H\mathbf{e}_1\Delta u_k + P\underline{\Delta u}_k + Q\underline{y}_k$. The horizon for output predictions $\underline{y}_{\rightarrow k+1|k}$, and thus the row dimension of H , should be long enough to capture all core dynamics!
2. Combine the input constraints, rate constraints, and output predictions into a single set of linear inequalities:

$$C\Delta u_k \leq \mathbf{f}_k \quad (8)$$

$$C = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ H\mathbf{e}_1 \\ -H\mathbf{e}_1 \end{bmatrix}; \quad \mathbf{f}_k = \begin{bmatrix} \overline{u} - u_{k-1} \\ -\underline{u} + u_{k-1} \\ \overline{\Delta u} \\ -\underline{\Delta u} \\ L\overline{y} - P\underline{\Delta u}_k - Q\underline{y}_k \\ -L\underline{y} + P\underline{\Delta u}_k + Q\underline{y}_k \end{bmatrix}; \quad L = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

where \mathbf{f}_k depends on past data in $\underline{\Delta u}_k, \underline{y}_k$ and on the limits.

3. The input/output predictions will satisfy constraints if inequalities (8) are satisfied, and thus the PFC algorithm should consider these explicitly.

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

Algorithm 1. Simple PFC algorithm with systematic constraint handling

At each sample:

1. Define the unconstrained value for Δu_k from (6).
2. Define the vector \mathbf{f}_k of (8) (it is noted that C does not change).
3. Use a simple loop covering all the rows of C as follows:
 - (a) Check the i th constraint that is the i th row $\mathbf{e}_i^T C \Delta u_k \leq \mathbf{f}_{k,i}$ of $C \Delta u_k \leq \mathbf{f}_k$.
 - (b) If $\mathbf{e}_i^T C \Delta u_k > \mathbf{f}_{k,i}$, then set $\Delta u_k = (f_{k,i}) / [\mathbf{e}_i^T C]$, else leave Δu_k unchanged.

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

3. Input Shaping PFC

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

3.1. Pre-Stabilisation of Predictions via Pole Cancellation

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

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

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

$$[C_{a-\Delta}][C_{a^+}] \mathbf{y}_{\rightarrow k+1} + H_A \mathbf{y}_{\leftarrow k} = C_b \Delta \mathbf{u}_{\rightarrow k} + H_b \Delta \mathbf{u}_{\leftarrow k} \quad (10)$$

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

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

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

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

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

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

$$[C_b \Delta u_{\rightarrow k} + \mathbf{p}] = C_{a^+} \gamma \quad (12)$$

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

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

$$y_{\rightarrow k+1} = [C_{a^- \Delta}]^{-1} [C_{a^+}]^{-1} [C_{a^+}] \gamma = [C_{a^- \Delta}]^{-1} \gamma \quad (13)$$

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

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

$$\Delta u_{\rightarrow k} = P_1 \mathbf{p}; \quad \gamma = P_2 \mathbf{p} \quad (14)$$

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

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

$$\Delta u_{\rightarrow k} = P_1 \mathbf{p} + C_{a^+} \phi \quad (15)$$

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

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

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

$$\begin{aligned} y_{\rightarrow k+1} &= [C_{a^- \Delta}]^{-1} [C_{a^+}]^{-1} [C_b \Delta u_{\rightarrow k} + \mathbf{p}] \\ &= [C_{a^- \Delta}]^{-1} [C_{a^+}]^{-1} [C_{a^+} \gamma + C_b C_{a^+} \phi] \\ &= [C_{a^- \Delta}]^{-1} [C_b \phi + \gamma] \end{aligned} \quad (16)$$

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

$$y_{\rightarrow}(z) = \frac{[1, z^{-1}, z^{-2}, \dots][\gamma + C_b \phi]}{a^-(z)} = \frac{\gamma(z) + b(z)\phi(z)}{a^-(z)} \quad (17)$$

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

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

$$y_{k+n|k} = H_{s,n} \phi + P_{s,n} \Delta u_k + Q_{s,n} y_k \quad (18)$$

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

3.2. Pre-Stabilisation via Pole Shaping

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

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

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

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

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

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

$$\begin{aligned} b(z)w(z) + \alpha(z)p(z) &= a^+(z)\hat{\gamma}(z); \quad p(z) = [1, z^{-1}, \dots]\mathbf{p} \\ \Rightarrow \Delta_{\rightarrow} u(z) &= \frac{w(z)}{\alpha(z)}, \quad y_{\rightarrow}(z) = \frac{\hat{\gamma}(z)}{a^-(z)\Delta(z)\alpha(z)} \end{aligned} \quad (19)$$

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

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

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

$$\begin{aligned} y_{\rightarrow k+1} &= [C_{a-\Delta}]^{-1} [C_{a^+}]^{-1} [C_b \Delta_{\rightarrow k} u + \mathbf{p}] \\ &= [C_{a-\Delta}]^{-1} [C_{a^+}]^{-1} [C_b C_{\alpha}^{-1} \mathbf{w} + \mathbf{p}] \\ &= [C_{a-\Delta}]^{-1} [C_{a^+}]^{-1} [C_{a^+} C_{\alpha}^{-1} \hat{\gamma}] = C_{a-\Delta}^{-1} C_{\alpha}^{-1} \hat{\gamma} \end{aligned} \quad (21)$$

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

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

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

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

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

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

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

$$\begin{aligned} y_{\rightarrow k+1} &= C_{a^+}^{-1} C_{a^+}^{-1} [C_b \Delta u_{\rightarrow k} + \mathbf{p}] \\ &= C_{a^+}^{-1} C_\alpha^{-1} \hat{\gamma} + C_{a^+}^{-1} C_{a^+}^{-1} C_\alpha^{-1} [C_{a^+} C_b \phi] \\ &= C_{a^+}^{-1} C_\alpha^{-1} [\hat{\gamma} + C_b \phi] \end{aligned} \quad (24)$$

□

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

$$y_{k+n|k} = h_{n,\alpha} \phi + P_{n,\alpha} \Delta u_k + Q_{n,\alpha} y_k \quad (25)$$

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

3.3. Proposed Shaping PFC Control Laws

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

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

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

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

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

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

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

3.4. Constraint Handling Approaches with Recursive Feasibility

A core advantage of MPC, in general, is that the optimised predictions can be restricted to ones which satisfy constraints; the d.o.f. within the predictions are used to ensure constraint satisfaction. For PC-PFC and PS-PFC, the d.o.f. in the predictions is the variable ϕ . This section gives a brief

overview of how the constraint inequalities ensuring (7) depend upon ϕ . We will use PS-PFC and assume that the reader can easily find the equivalent matrices for PC-PFC (for which, in effect, $\alpha = 1$).

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

$$\begin{aligned} L\Delta\mathbf{u} &\leq C_\alpha^{-1}[\hat{P}_1\mathbf{p} + C_{a^+}\phi] \leq L\bar{\Delta}\mathbf{u}; \\ L\mathbf{u} &\leq C_{I/\Delta}C_\alpha^{-1}[\hat{P}_1\mathbf{p} + C_{a^+}\phi] + Lu_{k-1} \leq L\bar{\mathbf{u}}; \\ L\mathbf{y} &\leq H_\alpha\phi + P_\alpha\Delta\mathbf{u}_k + Q_\alpha\mathbf{y}_k \leq L\bar{\mathbf{y}}. \end{aligned} \quad (28)$$

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

$$C\phi \leq \mathbf{f}_k \quad (29)$$

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

Algorithm 2. [PS-PFC with constraint handling]

At each sample:

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

Theorem 3. In the presence of constraints, Algorithm 2 is recursively feasible where the computed ϕ will not only enforce the input/output predictions to satisfy constraints at the current sample, but will also guarantee that one can make the same statement at the next sample.

Proof. By definition, the choice of $\phi = 0$ ensures feasibility in the nominal case because the input component $\hat{P}_1\mathbf{p}$ is the unused part of the input prediction from the previous sample, and this is known to satisfy constraints by assumption. One can ensure feasibility at start-up by beginning from a sensible state. \square

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

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

4. Numerical Examples

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

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

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

4.1. Description of Case Studies

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

4.1.1. Case Study 1: Boiler Level Control

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

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

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

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

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

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

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

A fluidised bed reactor is used to produce a variety of multiphase chemical reactions that are highly exothermic and can be considered as unstable. The reactor bed temperature needs to be

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

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

4.2. The Impact of Input Shaping on Predictions and Feasibility

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

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

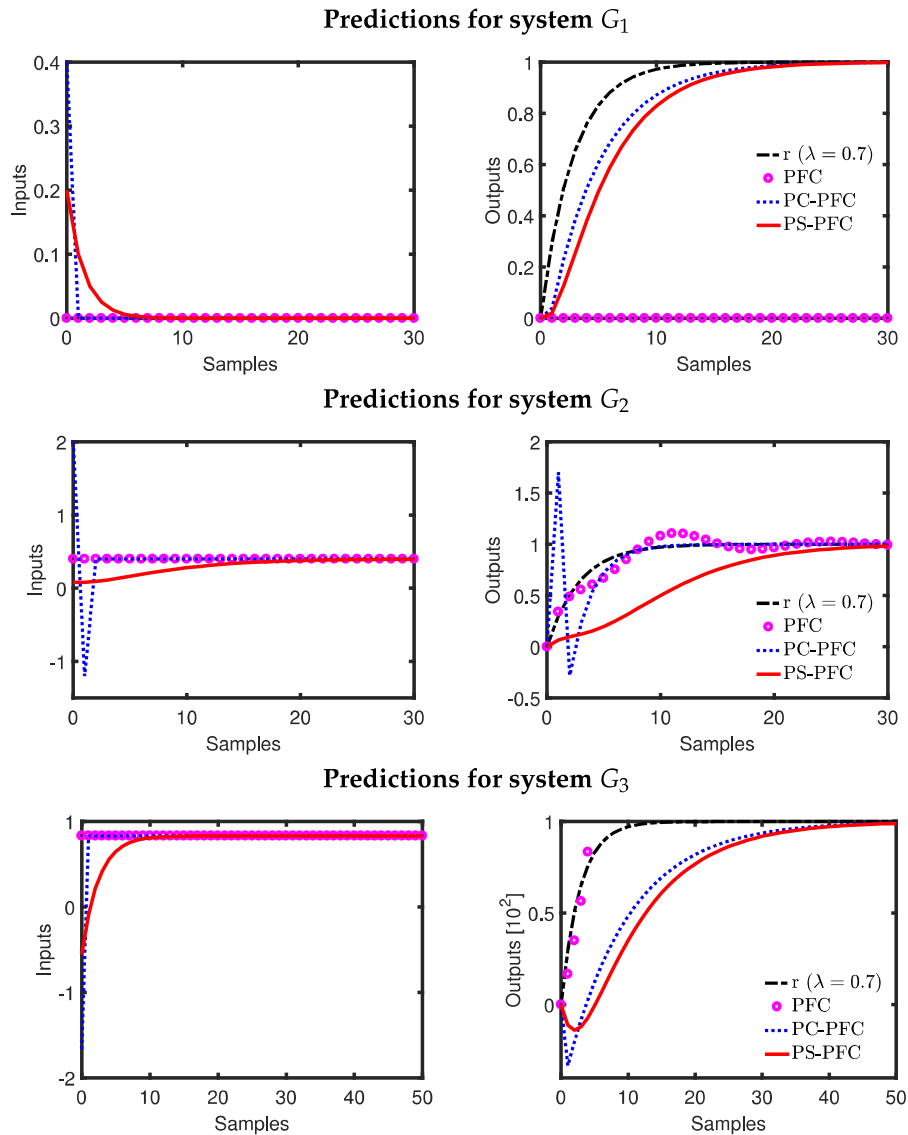


Figure 2. Input and output predictions with PFC, PC-PFC, and PS-PFC for processes G_1 , G_2 , G_3 .

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

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

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

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

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

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

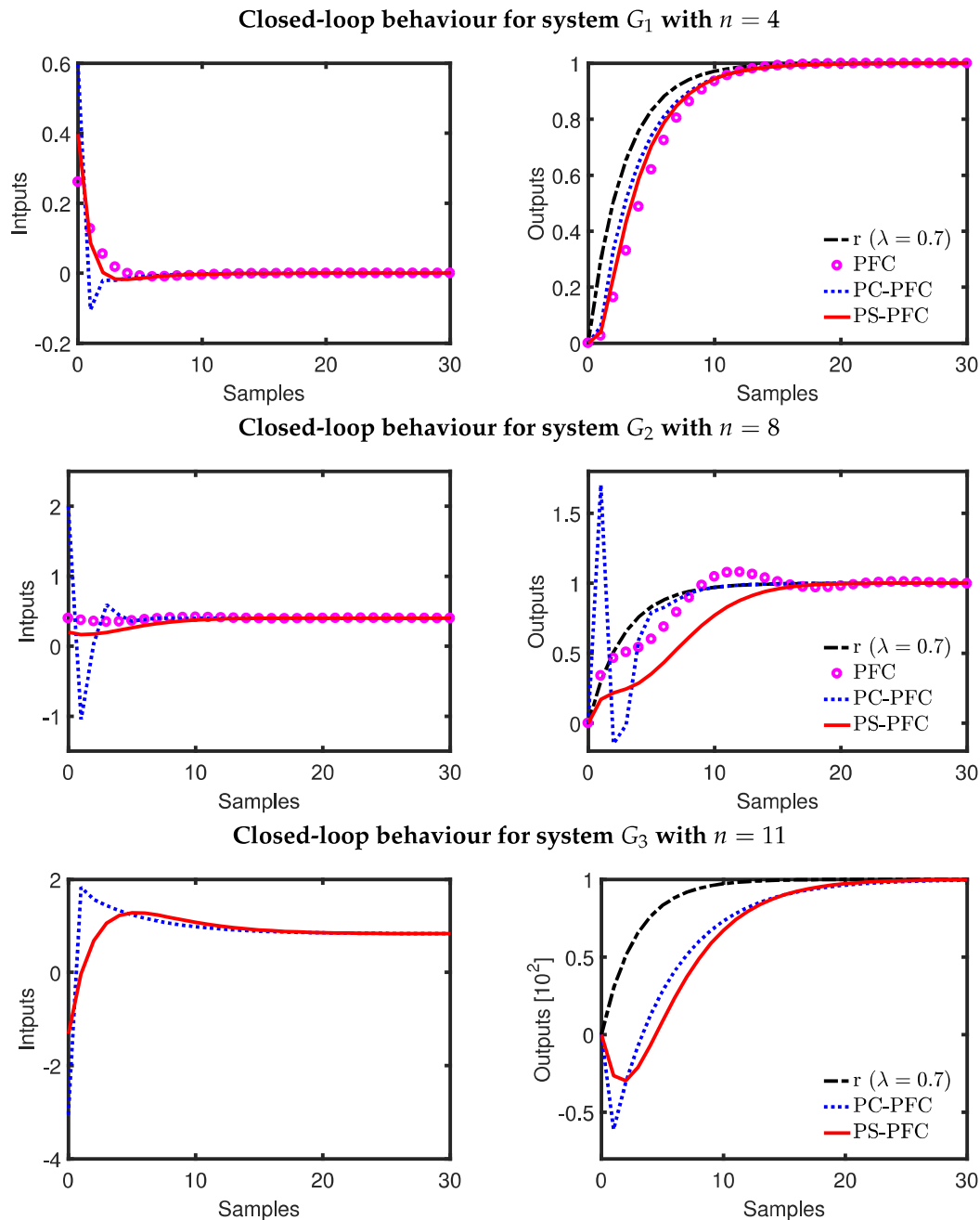


Figure 3. Closed-loop input and output behaviour of PC-PFC and PS-PFC for processes G_1, G_2, G_3 .

4.4. Constraint Handling

As noted in Section 4.2, for many dynamics a conventional PFC approach is infeasible or highly conservative because the output predictions inevitably violate constraints beyond a given horizon. Thus, conventional PFC can only be implemented by using short constraint horizons and thus with the loss of any recursive feasibility assurance; if it does work this is luck not good design and thus

should be avoided. For the case studies here, conventional PFC could only be used safely with output constraints for G_2 , although in that case we would expect some conservatism due to the oscillations in the output predictions.

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

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

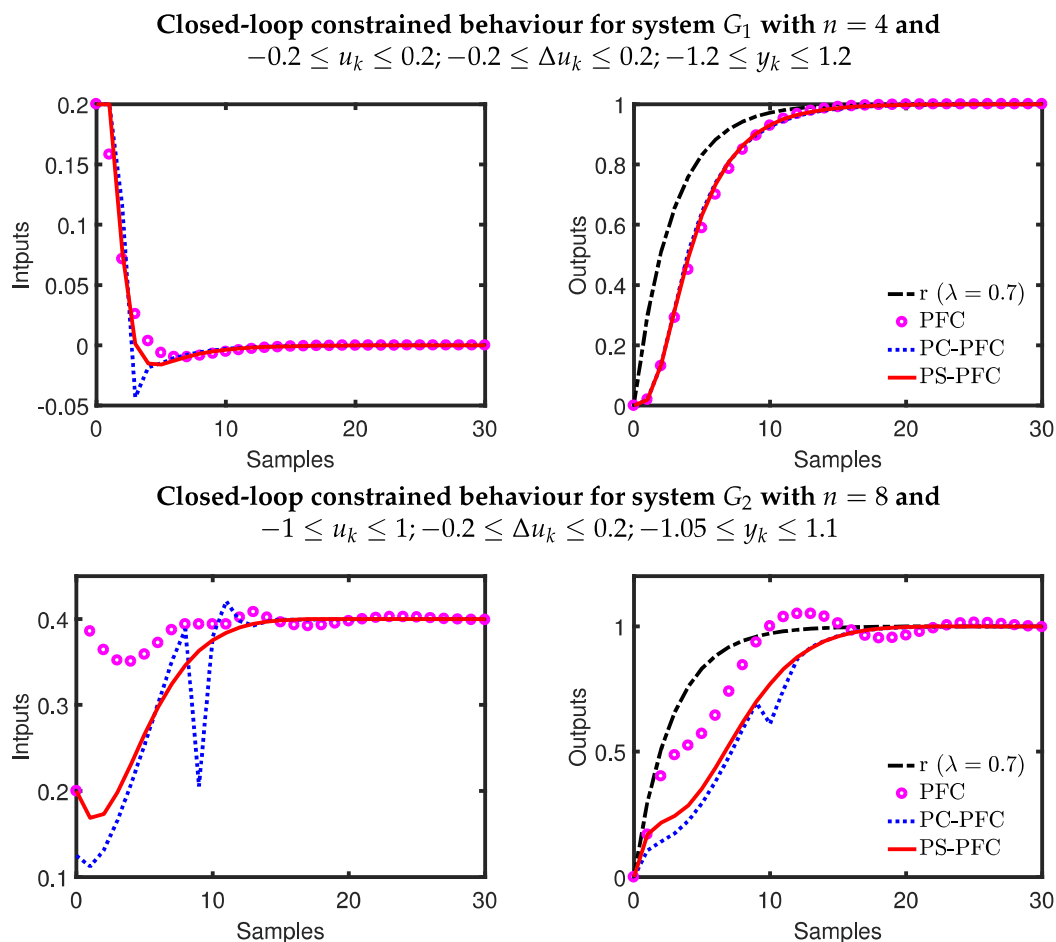


Figure 4. Cont.

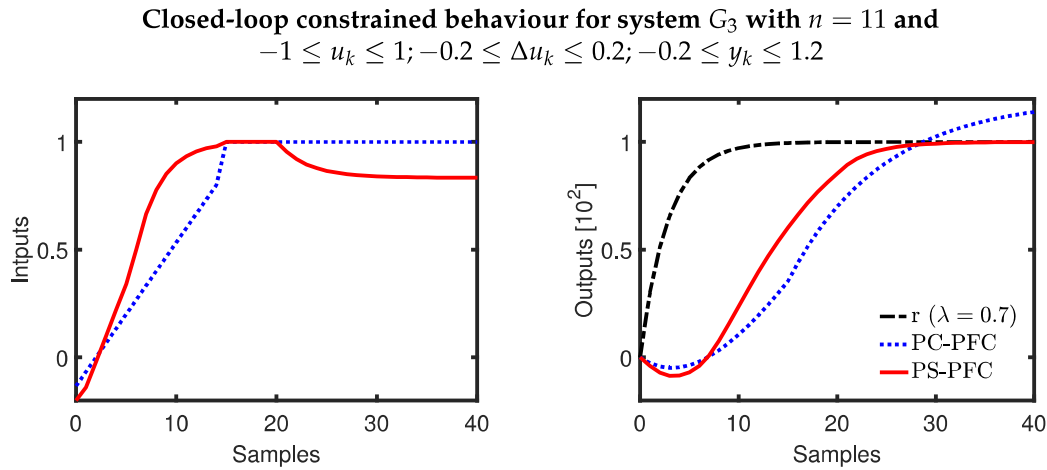


Figure 4. Constrained input and output behaviour of PFC, Pole Cancellation PFC (PC-PFC), and Pole Shaping PFC (PS-PFC) for processes G_1, G_2, G_3 .

4.5. Application of PS-PFC on Laboratory Hardware

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

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



Figure 5. Quanser SRV02 servo based unit.

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

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

The plant is set to have a CLTR of $0.2s$ (which is equivalent to $\lambda = 0.89$). Using a similar procedure to that in the previous section, the coincidence horizon is selected at $n = 4$. Figure 6 demonstrates the unconstrained and constrained performances of PS-PFC in tracking the servo position. In the unconstrained case, the controller manages to track the target position with the desired convergence speed. As for the constrained case, all the implied input limits ($-8 \leq u_k \leq 8$),

rate limits ($-3 \leq \Delta u_k \leq 3$), and output limits ($-0.8 \leq y_k \leq 0.8$) are satisfied systematically without any conflict.

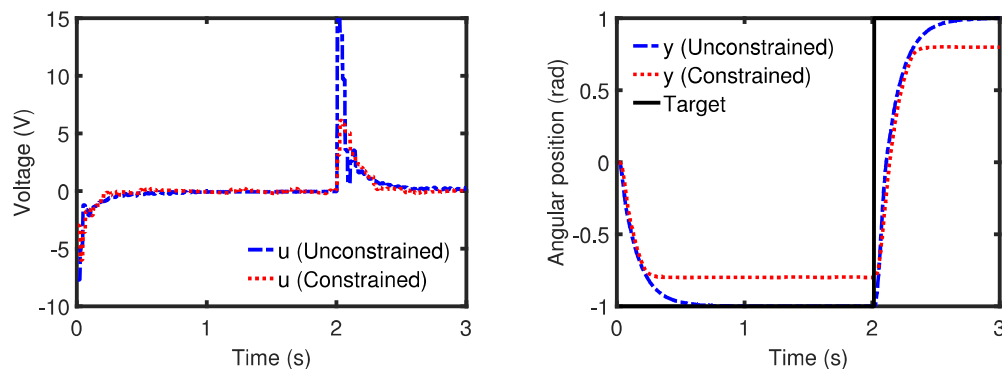


Figure 6. Unconstrained and constrained performance of PS-PFC for process G_s .

5. Conclusions

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

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

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

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

Funding: This research received no external funding.

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

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

References

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

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



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Appendix G

**THE EFFECT OF MODEL STRUCTURE ON
THE NOISE AND DISTURBANCE
SENSITIVITY OF PREDICTIVE FUNCTIONAL
CONTROL**

M. Abdullah and J. A. Rossiter

This paper has been accepted for publication in:
The proceeding of 2018 European Control Conference

Declaration form

THE EFFECT OF MODEL STRUCTURE ON THE NOISE DISTURBANCE SENSITIVITY OF PREDICTIVE
FUNCTIONAL CONTROL

(Proceeding of 2018 European Control Conference)

Contributions of authors:


M. Abdullah

Provided the initial idea, formulations, codes, simulations and draft of this paper.

J. A. Rossiter

Supervised M. Abdullah and proofread the paper.

Signatures:

.....

M. Abdullah
(First author)

.....

J.A. Rossiter
(Second author)

The effect of model structure on the noise and disturbance sensitivity of Predictive Functional Control

Muhammad Abdullah¹ and John Anthony Rossiter²

Abstract—An Independent Model (IM) structure has become a standard form used in Predictive Functional Control (PFC) for handling uncertainty. Nevertheless, despite its popularity and efficacy, there is a lack of systematic analysis or academic rigour in the literature to justify this preference. This paper seeks to fill this gap by analysing the effectiveness of different prediction models, specifically the IM structure and T-filter, for handling noise and disturbances. The observations are validated via both closed-loop simulation and real-time implementation and show that the sensitivity relationships are system dependent, which in turn emphasises the importance of performing this analysis to ensure a robust PFC implementation.

Keywords—Predictive Control, PFC, Sensitivity Analysis, Independent model, T-filter, Noise, Disturbance.

I. INTRODUCTION

Predictive Functional Control (PFC) is a variant of Model Predictive Control (MPC) that optimises a cost function solely based on a single degree of freedom (d.o.f) [1], [2]. With this simplification, PFC only requires a minimal computation and indeed, for low order models, the coding is almost trivial. In addition, PFC inherits some benefits of MPC such as systematic handling of constraints and/or systems with delays [3]. Because of its transparent tuning procedure, the controller is widely used in many industrial applications and has become a prime competitor to Proportional Integral Derivative (PID) regulators [2]–[4].

Despite its attractive attributes, the simplistic PFC concept is often unable to provide a consistent prediction [5], accurate constrained solutions [6] and effective handling of systems with challenging dynamics [7]. Several works have modified the traditional PFC framework to tackle these weaknesses either via cascade structures [2], pole-placement [7], [8] or input shaping [6], [9]–[11]. However, the derivation of these methods excludes explicit consideration of uncertainty and no attempt was made to discuss or analyse systematically the robustness of PFC.

Generally, PFC has received comparatively little attention in the academic literature because of its weaknesses in providing rigorous properties such as stability assurances [12], [13] or robust feasibility [14]. Critically however, embedding a formal robust design into the PFC formulation conflicts with the requirement for simplicity of coding and implementation [15]; a key selling point of PFC is its

simplicity. The normal option is to derive the nominal PFC controller using methods expected to give a robust design [16], [17], such as the use of a T-filter [18] or an Independent Model (IM) [2]. Since the unconstrained PFC framework provides a fixed control law, loop sensitivity can be computed and analysed to assess the controller robustness.

A conventional PFC approach often employs the IM structure to handle uncertainty [2], [3]. However, this paper argues that it is not always the best option to improve sensitivity in general. A user should also consider other alternatives such as the T-filter which may enable better trade-offs between noise and disturbance sensitivity [16], [18]. This paper compares the robustness of these two structures and their sensitivity functions are derived and benchmarked with a nominal PFC based on a CARIMA model. The analysis may help a user to get some insight into how to improve the controller robustness via selecting a suitable PFC structure rather than requiring a more complicated robust design [14].

This paper consists of five main sections. Section II discusses the basic formulation and derivations of sensitivity functions for three different PFC structures: CARIMA model, T-filter and IM structure. Section III presents the analysis on a real-time example. Section IV analyses two numerical examples with a higher order dynamics and section V provides the concluding remarks.

II. PFC STRUCTURES AND SENSITIVITY FUNCTIONS

This section presents a brief formulation of three different PFC structures associated to different prediction model assumptions together with the derivation of the associated sensitivity functions. Without loss of generality, this paper assumes an underlying CARIMA model (since state space and Finite Impulse Response (FIR) models can equally be represented with an IM). Here, only a brief background on PFC is presented; more detailed derivations, theory and concepts are available in references [2]–[5].

A. PFC with a CARIMA model

The PFC framework is designed based on human intuition where one computes a required control action depending on how fast one desires the output to reach the set point. The first order target trajectory is utilised to define the desired future output by enforcing the equality [5]:

$$y_{k+n_y|k} = (1 - \lambda^{n_y})r + \lambda^{n_y}y_k \quad (1)$$

where $y_{k+n_y|k}$ is the n_y -step ahead system prediction at sample time k , the desired closed-loop pole λ controls the convergence rate from output y_k to steady-state target r , and

^{1,2} M. Abdullah and J.A. Rossiter is with Department of Automatic Control and System Engineering, The University of Sheffield, Mappin Street, S1 3JD, UK. MAbdullah2@sheffield.ac.uk, j.a.rossiter@sheffield.ac.uk@sheffield.ac.uk

¹M. Abdullah is also with the Department of Mechanical Engineering, International Islamic University Malaysia, Jalan Gombak, 53100, Kuala Lumpur Malaysia. mohd.abdl@iiium.edu.my

the coincidence horizon n_y (a tuning parameter) is when the system prediction is forced to match the target trajectory exactly [2]. Since the n_y -step ahead prediction algebra for a CARIMA model is well known in the literature (e.g. [16]), only the final form is given here. For input increments Δu_k and outputs y_k , the n_y -step ahead linear prediction model is:

$$y_{k+n_y|k} = H\Delta_{\rightarrow}u_k + P\Delta_{\leftarrow}u_k + Qy_k \quad (2)$$

where parameters H , P , Q depend on the model parameters and for a model of order m :

$$\Delta_{\rightarrow}u_k = \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+n-1} \end{bmatrix}; \Delta_{\leftarrow}u_k = \begin{bmatrix} \Delta u_{k-1} \\ \Delta u_{k-2} \\ \vdots \\ \Delta u_{k-m} \end{bmatrix}; y_k = \begin{bmatrix} y_k \\ y_{k-1} \\ \vdots \\ y_{k-m} \end{bmatrix} \quad (3)$$

Substituting prediction (2) into equality (1) gives:

$$H\Delta_{\rightarrow}u_k + P\Delta_{\leftarrow}u_k + Qy_k = (1 - \lambda^{n_y})r + \lambda^{n_y}y_k \quad (4)$$

The constant future input assumption [2], [3] of PFC means $\Delta u_{k+i} = 0$ for $i > 0$, hence only the first column (H_1) of matrix H is used to construct the control law, thus:

$$\Delta u_k = \frac{1}{H_1} \left[(1 - \lambda^{n_y})r + \lambda^{n_y}y_k - Qy_k - P\Delta_{\leftarrow}u_k \right] \quad (5)$$

The control law can be represented in a vector form by rearranging (5) in terms of parameters F , N and \hat{D} with obvious definitions:

$$\Delta u_k = Fr - Ny_k - \hat{D}\Delta_{\leftarrow}u_k \quad (6)$$

Although the formulation in (6) can be implemented directly, it is easier to utilise a transfer function form for analysing its sensitivity [16]. The vectors of

$$N = [N_0, N_1, N_2, \dots, N_n] \\ \hat{D} = [\hat{D}_0, \hat{D}_1, \hat{D}_2, \dots, \hat{D}_n] \quad (7)$$

are defined in the z domain as:

$$N(z) = N_0 + N_1z^{-1} + N_2z^{-2} + \dots + N_nz^{-n} \\ \hat{D}(z) = \hat{D}_0 + \hat{D}_1z^{-1} + \hat{D}_2z^{-2} + \dots + \hat{D}_nz^{-n} \\ D(z) = 1 + z^{-1}\hat{D}(z) \quad (8)$$

Noting the definitions of $\Delta_{\leftarrow}u_k$ and y_k in (3), the sensitivity functions are derived based on a fixed closed loop form:

$$D(z)\Delta u_k = F(z)r - N(z)y_k \quad (9)$$

Fig. 1 indicates the equivalent block diagram and adds measurement noise n and output disturbance d . From the structure, the effective control law can be simplified to $K(z) = N_c(z)[D_c(z)\Delta]^{-1}$. Assuming system $G(z) = B(z)A(z)^{-1}$, the closed-loop pole polynomial $P_c(z) = 1 + K(z)G(z)$ is represented as:

$$P_c(z) = D(z)A(z)\Delta + N(z)B(z) \quad (10)$$

The sensitivity of input to noise is derived by finding the transference from u to n (refer to Fig. 1):

$$S_{un} = K(z)[1 + K(z)G(z)]^{-1} \\ = N(z)P_c(z)^{-1}A(z) \quad (11)$$

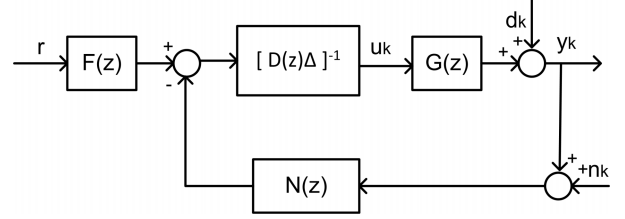


Fig. 1: PFC equivalent block diagram representation.

Similarly, the sensitivity of output to disturbance is obtained by solving the transference from y to d :

$$S_{yd} = [1 + K(z)G(z)]^{-1} = A(z)P_c(z)^{-1}D(z)\Delta \quad (12)$$

Remark 1: This work only considers the sensitivity to input noise and output disturbances. Other analysis such as parameter uncertainty S_G , output noise S_{yn} and input disturbance S_{ud} are similar but excluded to save space.

B. PFC with T-Filter (PFCT)

The T-filter acts as a low pass filter to eliminate high frequency measurement noise without affecting the nominal tracking performance [18] of predictive control, although in the literature a T-filter has yet to be applied to PFC. The framework proposed here is a two stage design whereby PFC is first tuned for performance tracking, then the T-filter is employed to improve the sensitivity. Conceptually, the measurement output is low-pass filtered before prediction and anti-filtered after prediction to restore the predicted data back to the correct domain before deploying the nominal algorithm. The procedure is illustrated in Fig. 2 and reduces the impact of high frequency noise on the prediction while retaining the valuable low frequency dynamics [16].

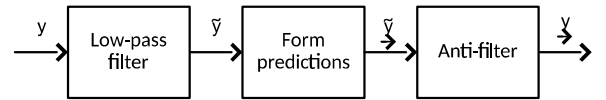


Fig. 2: PFCT prediction structure with T-filter.

The desired T-filter T^{-1} is deployed as $\tilde{y}_k = y_k T^{-1}$ or $T\tilde{y}_k = y_k$. Define the filtered predictions up to horizon n_y as follows:

$$\tilde{y}_{\rightarrow k+1} = H\Delta_{\rightarrow}\tilde{u}_k + P\Delta_{\leftarrow}\tilde{u}_k + Q\tilde{y}_k \quad (13)$$

The relationship between the filtered and unfiltered predicted data can be represented using Toeplitz/Hankel form (refer to [16] for more details):

$$y_{\rightarrow k+1} = C_T\tilde{y}_{\rightarrow k+1} + H_T\tilde{y}_k \\ \Delta_{\rightarrow}u_k = C_T\Delta_{\rightarrow}\tilde{u}_k + H_T\Delta_{\leftarrow}\tilde{u}_k \quad (14)$$

where for $T(z) = T_0 + T_1z^{-1} + \dots + T_nz^{-n}$:

$$C_T = \begin{bmatrix} T_0 & 0 & 0 & \dots \\ T_1 & T_0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \\ T_n & T_{n-1} & T_{n-2} & \dots \end{bmatrix}, H_T = \begin{bmatrix} T_1 & T_2 & \dots & T_n \\ T_2 & T_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots \\ 0 & 0 & \dots & 0 \end{bmatrix} \quad (15)$$

substituting (14) into (13) gives:

$$\underbrace{C_T^{-1}[y_{k+1} - H_T \tilde{y}_k]}_{\tilde{y}_k} = H \underbrace{C_T^{-1}[\Delta u_k - H_T \Delta \tilde{u}_k]}_{\Delta \tilde{u}_k} + P \Delta \tilde{u}_k + Q \tilde{y}_k \quad (16)$$

Multiplying through by C_T and grouping common terms:

$$y_k = H \Delta u_k + \tilde{P} \Delta \tilde{u}_k + \tilde{Q} \tilde{y}_k \quad (17)$$

where $\tilde{P} = [C_T P - H H_T]$ and $\tilde{Q} = [H_T + C_T Q]$. The difference between (17) and (2) are the last two terms which now are based on past filtered data. Hence, applying a similar control law and derivation to eqns.(4-9), a PFCT fixed control law can be formulated as:

$$D_t(z) \Delta u_k = F(z)r - N_t(z)y_k \quad (18)$$

where $D_t(z) = \frac{D(z)}{T(z)}$ and $N_t(z) = \frac{N(z)}{T(z)}$ are represented in the block diagram of Fig. 3. The closed-loop pole polynomial, sensitivity of the input to noise and sensitivity of the output to disturbances are:

$$\begin{aligned} P_t(z) &= D_t(z)A(z)\Delta + N_t(z)B(z) \\ S_{un} &= N_t(z)P_t(z)^{-1}A(z) \\ S_{yd} &= A(z)P_t(z)^{-1}D_t\Delta \end{aligned} \quad (19)$$

Remark 2: It can be shown that the closed-loop poles of PFCT $P_t(z)$ are related to the equivalent poles of PFC by $P_t(z) = P_c(z)T(z)$ and also that the inclusion of T-filter cannot affect the nominal tracking performance [16].

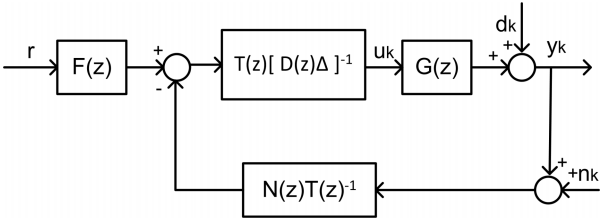


Fig. 3: PFCT control loop.

C. PFC with an Independent Model (PFCI)

As discussed before, the IM structure is often used in conventional PFC [2], [3] as the creators believe it gives better sensitivity properties in general. The implementation is equivalent to using a step response model (ignoring truncation errors [16]). Define y_m to be the model output and y the process output, then the prediction of future output in (2) is defined based on y_m and augmented with a correction term $D_k = y_k - y_{m,k}$ as follows:

$$y_{k+n_y|k} = H \Delta u_k + P \Delta u_k + Q y_{m,k} + D_k \quad (20)$$

Equating prediction (20) with the target trajectory (1) gives:

$$H \Delta u_k + P \Delta u_k + Q y_{m,k} - y_{m,k} = (1 - \lambda^{n_y})(r - y_k) \quad (21)$$

Since the future input increment Δu_{k+i} is assumed zero for $i > 0$ and H is reduced to H_1 , the PFC control law is:

$$\Delta u_k = \frac{1}{H_1} \left[(1 - \lambda^{n_y})r - (1 - \lambda^{n_y})y_k - Q y_{m,k} + y_{m,k} - P \Delta u_k \right] \quad (22)$$

For suitable F, N_i, M_i, \hat{D} , one can rearrange (22) as:

$$\Delta u_k = F r - N_i y_{m,k} - M_i y_k - \hat{D} \Delta u_k \quad (23)$$

Transforming (23) into an equivalent transfer function format, the PFCI fixed closed loop is constructed as:

$$D(z) \Delta u_k = F(z)r - N_i(z)y_{m,k} - M_i(z)y_k \quad (24)$$

The model output can be determined exactly from the model $y_{m,k} = B(z)A(z)^{-1}u_k$ and hence equation (24) can be replaced by (see Fig. 4 for the effective loop structure):

$$\underbrace{[D(z)\Delta + N_i(z)B(z)A(z)^{-1}]}_{D_i(z)} u_k = F(z)r - M_i(z)y_k \quad (25)$$

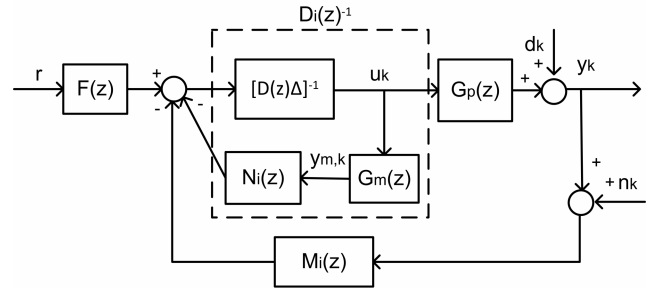


Fig. 4: PFCI control loop.

The sensitivities for IM structure of Fig. 4 are obtained analogously to CARIMA PFC by substituting parameter $D(z)\Delta$ with $D_i(z)$, and $N(z)$ with $M_i(z)$ in equation (10-12). The closed-loop pole polynomial and sensitivities are:

$$\begin{aligned} P_i(z) &= D_i(z)A(z) + M_i(z)B(z) \\ S_{un} &= M_i(z)P_i(z)^{-1}A(z) \\ S_{yd} &= A(z)P_i(z)^{-1}D_i(z) \end{aligned} \quad (26)$$

D. Summary of Control Laws

Table I summarises the sensitivity functions for PFC (Fig. 1), PFCT (Fig. 3) and PFCI (Fig. 4). The key observation is that while, the derivation and structure of all the sensitivity functions are almost same, their parameters are different and hence, different sensitivity response should be expected.

TABLE I: Sensitivity to noise and disturbance.

Algorithm	Input sensitivity to noise	Output sensitivity to disturbance
PFC	$N(z)P_c(z)^{-1}A(z)$	$A(z)P_c(z)^{-1}D(z)\Delta$
PFCT	$N_t(z)P_t(z)^{-1}A(z)$	$A(z)P_t(z)^{-1}D_t(z)\Delta$
PFCI	$M_i(z)P_i(z)^{-1}A(z)$	$A(z)P_i(z)^{-1}D_i(z)$

III. REAL TIME SYSTEM EXAMPLE

This section analyses the sensitivity of a PFC controlled Quanser SRV02 servo based unit [19] system against noise and disturbance. The servo is powered by a Quanser VoltPAQ-X1 amplifier that comes with National Instrument ELVIS II+ multifunctional data acquisition device. The controller is run by National Instrument LabVIEW software via



Fig. 5: Quanser SRV02 servo based unit.

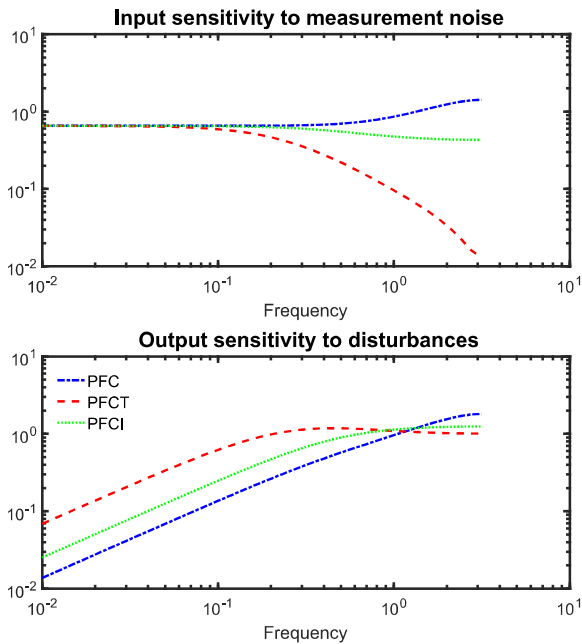


Fig. 6: Sensitivity plot for G_1 with different PFC structures.

USB connection (Fig. 5). The objective is to track the desired servo angular speed, $\dot{\theta}(t)$ by regulating the supplied voltage, $V(t)$. The mathematical model is given as [19]:

$$0.0254\ddot{\theta}(t) = 1.53V(t) - \dot{\theta}(t) \quad (27)$$

where $\ddot{\theta}(t)$ is the servo angular acceleration. Converting the model (27) to discrete form with sampling time $0.02s$, the transfer function of angular speed to voltage input is:

$$G_1 = \frac{0.8338}{1 - 0.455z^{-1}} \quad (28)$$

For a fair comparison, all PFC structures will use the same tuning parameters ($\lambda = 0.7$ and $n_y = 3$). The sensitivity functions for different loop structures: PFC, PFCT and PFCI are illustrated via Bode plots (see Fig. 6). A summary of observations is given as:

- In the high frequency range, the first order PFCT, $T = 1 - 0.8z^{-1}$ (red dashed line) gives the lowest sensitivity

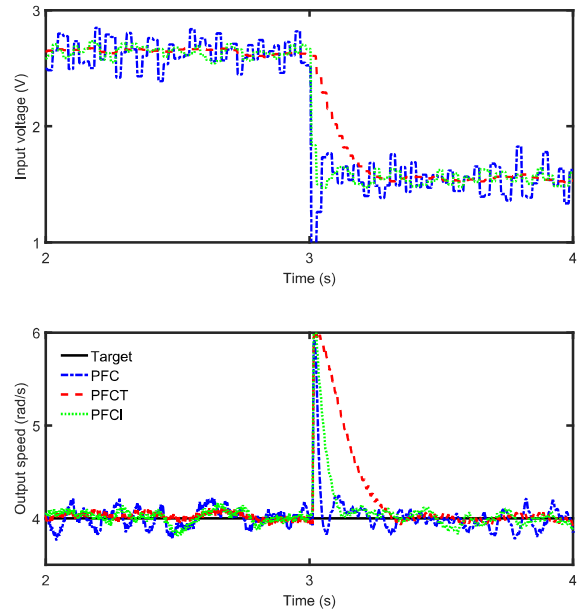


Fig. 7: Closed-loop performance of Quanser servo with different PFC structures.

to noise and disturbance followed by PFCI (green dotted line) and PFC (blue dashed-dotted line).

- The output of PFCT is more sensitive to low and mid frequency disturbances compared to PFCI and PFC.

This observation is then validated by comparing their closed-loop performance on the hardware (see Fig. 7). In this case, the desired angular speed is set at 4 rad/s and the output step disturbance ($d = 2$) entered the system at $3s$. The output measurement is corrupted by Gaussian white noise with variance of 2 . The results show:

- PFCT reduces noise transmission to the input compared to PFCI and PFC.
- PFCT rejects the output disturbance $0.2s$ slower compared to PFCI and PFC.

In summary, without filter or altered structure, the PFC input is fluctuating between $2.5V$ to $3V$. This situation may lead to a fatigue failure especially for a highly sensitive application. However, improving the sensitivity in one frequency range may make it worst at the other range and hence in practice, a trade-off to get the best overall performance is required. In this example, it may be worth to have a slower disturbance rejection (which is less likely to occur) to get the best noise sensitivity with the T-filter.

IV. ANALYSIS FOR HIGHER ORDER SYSTEMS

This section discusses the sensitivity analysis of second order systems with over and under-damped dynamics. The analysis is then validated with their closed-loop performance using Matlab simulations.

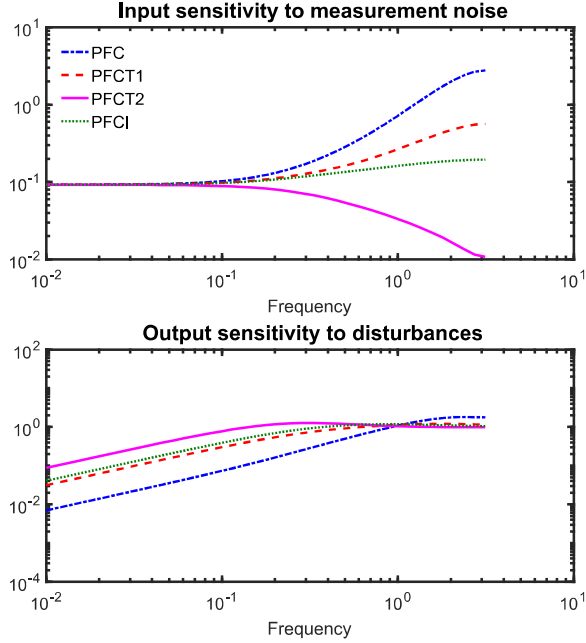


Fig. 8: Sensitivity plot for G_2 with different PFC structures.

A. Over Damped Second Order System

An over damped second order system (29) is considered here. The set point is zero and a step output disturbance ($d = 0.1$) occurs at the 50th sample. The measurement is corrupted by Gaussian random white noise. All PFCs are tuned with $\lambda = 0.7$ and $n_y = 3$.

$$G_2 = \frac{z^{-2} + 0.3z^{-1}}{1 - 1.2z^{-1} + 0.32z^{-2}} \quad (29)$$

The Bode plots in Fig. 8 show:

- The input of PFCT2, $T_2 = (1 - 0.8z^{-1})^2$ (pink line) gives the lowest input sensitivity to noise followed by PFCI (green dotted line), PFCT1, $T_1 = 1 - 0.8z^{-1}$ (red dashed line) and PFC (blue dash-dotted line).
- However, over filtering the measurement as with PFCT2 leads to a poor output reaction to disturbances in the low or mid frequency range compared to other structures.

The closed-loop simulation in Fig. 9 reflects the sensitivity analysis whereby:

- PFCT2 rejects most of the noise in input but in fact the variance with PFCI is still small.
- In the presence of the output disturbance, PFCT2 converges 7 samples slower with the highest overshoot ($y_{max} = 0.5$) compared to PFCI ($y_{max} = 0.3$) and PFCT1 ($y_{max} = 0.26$).

Although, a user can manually tune the T-filter, in this example there is a reasonable argument that the IM structure provides a good sensitivity trade off between noise and disturbances.

B. Second Order Under-damped System

A PFC controlled second order under-damped system (30) again has a zero set point and a disturbance ($d = 0.1$) at

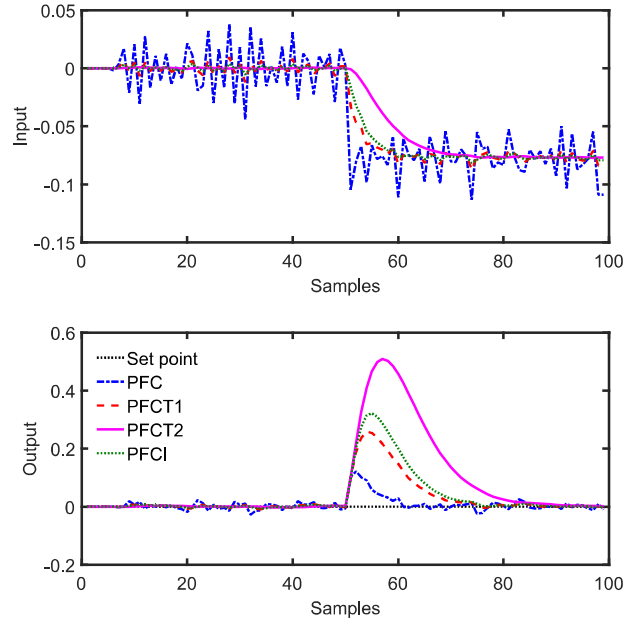


Fig. 9: Closed-loop response of G_2 with corrupted measurement noise and disturbance.

50th sample and measurement noise. The tuning parameter ($\lambda = 0.8$ and $n_y = 4$) is selected based on the conjecture presented in [5], [10].

$$G_3 = \frac{0.0565z^{-2} + 0.0495z^{-1}}{1 - 1.5643z^{-1} + 0.6703z^{-2}} \quad (30)$$

A similar pattern to the previous example is observed in the Bode diagrams of sensitivity (see Fig. 10):

- PFCT1 gives a small improvement in rejecting high frequency noise, but less than PFCI, while having almost similar disturbance sensitivity in the low frequency range compared to PFCI.
- Over filtering the measurement noise with PFCT2 leads to a more sensitive output to low frequency disturbances.

The closed-loop simulations in Fig. 11 validate the analysis whereby:

- PFCI rejects more noise compared to PFCT1 and almost the same as PFCT2.
- In the presence of the output disturbance, PFCI overshoots more than PFCT1 and less than PFCT2 but converges faster than both.

In this case, it is clearly shown that PFCI has better sensitivity trade-off between noise and disturbances, thus no filter or observer would be recommended.

V. CONCLUSIONS

This work provides a sensitivity analysis to uncertainty for different PFC structures. Although generic conclusions are not applicable, it is clearly shown that the popular IM structure does not always give the best robustness to

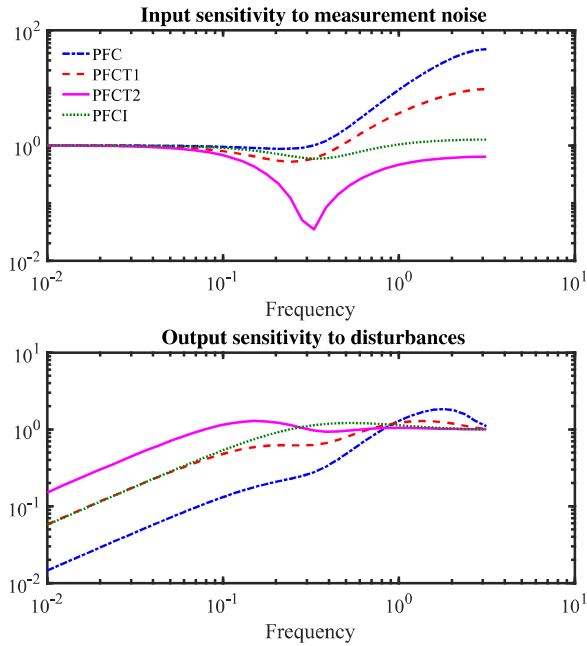


Fig. 10: Sensitivity plot for G_3 with different PFC structures.

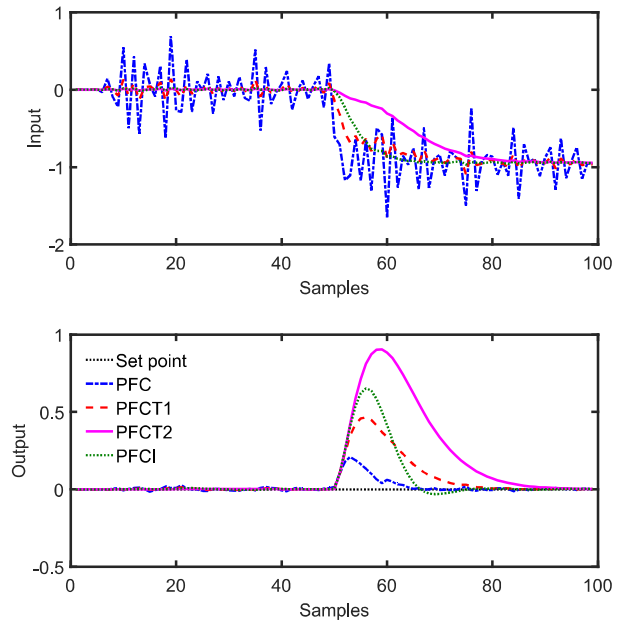


Fig. 11: Closed-loop response for system G_3 with with corrupted measurement noise and disturbance.

uncertainty especially for a simple first order system. In some cases, using a low pass filter such as a T-filter can provide a good sensitivity trade-off between noise and disturbances as shown in the hardware example of section III. However, the sensitivity of PFC structures are system dependent and thus the best option may not be clear a priori as the latter two examples indicated a likely preference for using the IM approach. Hence, production of off-line sensitivity plots is essential to give insight into the robustness of differing PFC structures and indeed, this should be extended to consider a wider range of sensitivity such as parameter uncertainty.

It is also noted that this paper did not consider the impact of changes in the parameters λ, n and one might argue that this should also be investigated. Moreover, where PFC is challenging to tune [9] and/or needs structural changes, further alternative structures may be beneficial and should be included in any offline analysis.

ACKNOWLEDGMENT

The first author would like to acknowledge International Islamic University Malaysia and Ministry of Higher Education Malaysia for funding this work.

REFERENCES

- [1] J. Richalet, A. Rault, J.L. Testud and J. Papon, Model predictive heuristic control: applications to industrial processes, *Automatica*, 14(5), 413-428, 1978.
- [2] J. Richalet, and D. O'Donovan, *Predictive Functional Control: principles and industrial applications*. Springer-Verlag, 2009.
- [3] J. Richalet, and D. O'Donovan, "Elementary Predictive Functional Control: a tutorial," Int. Symposium on Advanced Control of Industrial Processes, 2011, pp. 306-313.
- [4] R. Haber, J.A. Rossiter and K. Zabet, "An alternative for PID control: Predictive Functional Control - a tutorial," American Control Conference (ACC), 2016, pp. 6935-6940.
- [5] J. A. Rossiter, and R. Haber, "The effect of coincidence horizon on predictive functional control," *Processes*, 3, 1, pp. 25-45, 2015.
- [6] M. Abdullah, J. A. Rossiter and R. Haber, "Development of constrained predictive functional control using Laguerre function based prediction," IFAC World Congress, 2017.
- [7] 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.
- [8] 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, part 2, pp. 403-414, 2017.
- [9] J. A. Rossiter, "Input shaping for PFC: how and why?," *J. Control and Decision*, pp. 1-14, Sep. 2015.
- [10] M. Abdullah and J. A. Rossiter, "Utilising Laguerre function in predictive functional control to ensure prediction consistency," 11th Int. Conf. on Control, Belfast, UK, 2016.
- [11] M. Abdullah and J. A. Rossiter, "Alternative method for Predictive Functional Control to handle an integrating process", under review for *Advances in PID*, 2018.
- [12] M. Khadir and J. Ringwood, "Stability issues for first order predictive functional controllers: extension to handle higher order internal models," Int. Conf. on Computer Systems and Information Technology, 2005.
- [13] J. A. Rossiter, "A priori stability results for PFC, *International journal of control*, vol. 90, no. 2, pp. 305-313, 2016.
- [14] D. Q. Mayne, M. M. Seron, and S. Rakovic, "Robust model predictive control of constrained linear systems with bounded disturbances, *Automatica*, vol. 41, no. 2, pp. 219-224, 2005.
- [15] M. Khadir and J. Ringwood, "Extension of first order predictive functional controllers to handle higher order internal models, *Int. Journal of Applied Mathematics and Comp. Science*, vol. 18, no. 2, pp. 229-239, 2008.
- [16] J. A. Rossiter, *Model predictive control: a practical approach*, CRC Press, 2003.
- [17] 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.
- [18] T-W. Yoon and D.W. Clarke, Observer design in receding horizon predictive control, *International Journal of Control*, 61, 171-191, 1995.
- [19] *Quanser user manual SRV02 rotary servo based unit set up and configuration*. Quanser Inc, 2012.

Appendix H

**A FORMAL SENSITIVITY ANALYSIS FOR
LAGUERRE BASED PREDICTIVE
FUNCTIONAL CONTROL**

M. Abdullah and J. A. Rossiter

This paper has been accepted for publication in:
The proceeding of 12th UKACC International Conference on Control 2018

Declaration form

A FORMAL SENSITIVITY ANALYSIS FOR LAGUERRE BASED PREDICTIVE FUNCTIONAL CONTROL

(Proceeding of 12th UKACC International Conference on Control 2018)

Contributions of authors:

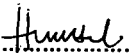
M. Abdullah

Provided the initial idea, formulations, codes, simulations and draft of this paper.

J. A. Rossiter

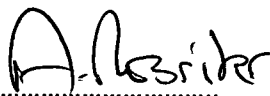
Supervised M. Abdullah and proofread the paper.

Signatures:

..........

M. Abdullah

(First author)

..........

J.A. Rossiter

(Second author)

A Formal Sensitivity Analysis for Laguerre Based Predictive Functional Control

Muhammad Abdullah* and John Anthony Rossiter[†]

^{*†}Department of Automatic Control and System Engineering,
University of Sheffield, Mappin Street, S1 3JD, UK.

Email: MAbdullah2@sheffield.ac.uk* and j.a.rossiter@sheffield.ac.uk[†]

^{*}Department of Mechanical Engineering,

International Islamic University Malaysia, Jalan Gombak, 53100, Kuala Lumpur Malaysia.

Email: mohd_abdl@iiu.edu.my*

Abstract—A Laguerre Predictive Functional Control (LPFC) is a simple input shaping method, which can improve the prediction consistency and closed-loop performance of the conventional approach (PFC). However, it is well-known that an input shaping method, in general, will affect the loop sensitivity of a system. Hence, this paper presents a formal sensitivity analysis of LPFC by considering the effect of noise, unmeasured disturbance and parameter uncertainty. Sensitivity plots from bode diagrams and closed-loop simulation are used to illustrate the controller robustness and indicate that although LPFC often provides a better closed-loop tracking response and disturbance rejection, this may involve some trade-off with the sensitivity to noise and parameter uncertainty. Finally, to validate the practicality of the results, the sensitivity of the LPFC control law is illustrated on real-time laboratory hardware.

Index Terms—Predictive Control, PFC, Sensitivity Analysis, Laguerre function, Parameter Uncertainty, Noise, Disturbance

I. INTRODUCTION

Model Predictive Control (MPC) is an optimal controller that employs a control action based on a future output prediction. Typically, MPC utilises a finite horizon prediction in the optimisation process and can explicitly take into account different types of constraints in a system [1]. Nevertheless, the implementation of this controller is often more expensive and requires higher computational effort and time compared to its competitors [2]. Hence for low-end applications, it is wiser to consider a simpler controller such as Proportional Integral Derivative (PID) or Predictive Functional Control (PFC).

Developed in 1973, PFC is known as a simplified version of MPC that minimises the output error at a single point instead of over a whole trajectory [3], [4]. With this simplification, PFC only needs simple coding and minimal computation. Although in general, the computed input is not optimal, it still retains some of the core benefits of an MPC approach such as systematic handling of constraints and/or systems with delays [4]. Besides, the use of a target first-order Closed-loop Time Response (CLTR) as one of its tuning parameters, makes the design process more transparent. Currently, this controller is widely used in many industrial applications and has become a prime competitor with PID regulators [4]–[6].

This work is funded by International Islamic University Malaysia and Ministry of Higher Education Malaysia.

Despite its attractive attributes, the simple PFC concept is often unable to provide a consistent prediction [7], accurate constrained solutions [8] and effective handling of systems with challenging dynamics [9], [10]. Several works have modified the traditional PFC framework to tackle these weaknesses either via cascade structures [4], [11], pole-placement [9], [12] or input shaping [8], [10]. However, the derivation of these methods often excludes explicit consideration of uncertainty, and only a few works have systematically discussed or analysed the robustness of PFC [13], [14]. Hence, the main objective of this work is to tackle this issue on one of its alternative structures known as Laguerre PFC (LPFC).

LPFC is defined by shaping the future predicted input trajectory with a first-order Laguerre polynomial [15], [16]. Instead of the constant input assumption of PFC, the future dynamics are now forced to converge gradually to the steady-state value. This modification can improve the prediction consistency and the significance of CLTR as a tuning parameter [16]. Furthermore, due to the well-posed decision making, satisfying constraints within a larger validation horizon becomes more accurate and less conservative [8]. However, this algorithm, as in common in MPC, is utilising the model parameters to estimate the steady state input while improving the loop performance and hence, it is worth investigating its sensitivity concerning noise, disturbances and parameter uncertainty.

Since the general unconstrained PFC framework provides a fixed control law, loop sensitivity can be computed and analysed to assess the controller robustness [3]. The performance of LPFC will be benchmarked against a nominal PFC structure to get some insight into the sort of sensitivity trade-off that ones should expect. The reader is reminded again that the scope of this work is only focused on simple and stable dynamic system; further development of LPFC to deal with challenging or unstable systems constitutes future work and in general is non-simple with a PFC approach.

This paper consists of five main sections. Section II discusses the basic formulation and derivations of sensitivity functions for PFC and LPFC. Section III presents some numerical examples. Section IV illustrates the findings are consistent with those on real-time laboratory hardware and section V gives the conclusions.

II. PFC STRUCTURES AND SENSITIVITY FUNCTIONS

This section presents a brief formulation for both PFC and LPFC together with the derivation of their sensitivity functions. More detailed derivations, theory and concepts are available in these references [3], [4], [6], [7]. Without loss of generality, this work utilises an autoregressive with exogenous terms (ARX) model with an independent model (IM) structure.

A. Conventional PFC

1) *Target trajectory*: PFC is designed to follow a closed-loop behaviour of the first order system with a delay τ (or h samples) and a time constant T_r [7]. The z-transform of the target trajectory, $r(z)$ with steady-state R is:

$$r(z) = \frac{z^{-h}(1-\lambda)}{1-\lambda z^{-1}}R \quad (1)$$

The representation of target pole, λ in (1) is equivalent to the desired closed-loop time response (CLTR) which is normally used by industrial practitioners [4]. The conversion can be presented by $T_r = CLTR/3$, where $\lambda = e^{-\frac{T}{T_r}}$ with T the sampling period.

2) *Coincidence point and degree of freedom*: The control objective of PFC is to force the system open-loop prediction, y_p to exactly match the predicted target trajectory of (1) at a selected coincidence point n samples into the future [4]. Consequently, the control law is formulated to enforce the equality:

$$y_{p,k+n|k} = (1-\lambda^n)R + \lambda^n y_{p,k} \quad (2)$$

where $y_{p,k+n|k}$ is the n -step ahead system prediction at sample time k and $y_{p,k}$ is the current process output measurement.

3) *Independent model*: The independent model (IM) structure is often used in conventional PFC [4], [5] as this is known to provide good sensitivity properties in general, yet it is only applicable to open-loop stable systems. The implementation is equivalent to using a step response model (ignoring truncation errors [1]). Both the model G_m and process G_p run in parallel using the same input u_k (see Fig.1). The error ($d_k = y_{p,k} - y_{m,k}$) between process output y_p and model output y_m is utilised to handle noise, disturbance and parameter uncertainty. Using the unbiased model prediction, the equality (2) is altered to:

$$\begin{aligned} (1-\lambda^n)R + \lambda^n y_{p,k} &= y_{m,k+n|k} + d_k \\ (R - y_{p,k})(1-\lambda^n) &= y_{m,k+n|k} - y_{m,k} \end{aligned} \quad (3)$$

4) *Control law*: The n -step ahead prediction algebra for an ARX model is well known in the literature, which can be represented using Toeplitz/Hankel form (e.g. [1]), hence only the final form is given here. For input u_k and model outputs $y_{m,k}$, the n -step ahead linear prediction model is:

$$y_{m,k+n|k} = H\underline{u}_k + P\underline{u}_k + Qy_{m,k} \quad (4)$$

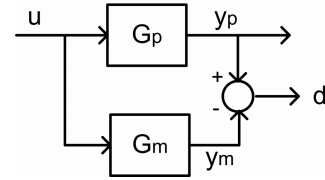


Fig. 1: The independent model structure.

where parameters H , P , Q depend on the model parameters and for a model of order m :

$$\underline{u}_k = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+n-1} \end{bmatrix}; \quad \underline{u}_k = \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \\ u_{k-m} \end{bmatrix}; \quad y_{m,k} = \begin{bmatrix} y_{m,k} \\ y_{m,k-1} \\ \vdots \\ y_{m,k-m} \end{bmatrix} \quad (5)$$

Substituting prediction (4) into equality (3) gives:

$$H\underline{u}_k + P\underline{u}_k + Qy_{m,k} - y_{m,k} = (R - y_{p,k})(1 - \lambda^n) \quad (6)$$

The constant future input assumption of PFC [3], [4] means that $u_{k+i|n} = u_k$ for $i > 0$, hence defining $h = \sum(H)$, the control law reduces to:

$$u_k = \frac{1}{h} \left[(1-\lambda^n)R - (1-\lambda^n)y_{p,k} - Qy_{m,k} + y_{m,k} - P\underline{u}_k \right] \quad (7)$$

The control law can be represented in a vector form by rearranging (7) in terms of parameters F_p , N_p , M_p and \hat{D}_p with obvious definitions:

$$u_k = F_p R - N_p y_{m,k} - M_p y_{p,k} - \hat{D}_p \Delta \underline{u}_k \quad (8)$$

Remark 1: Conventional PFC can work well with low order and simple dynamical systems, especially when the coincidence point is selected properly [7]. However, with the restricted degree of freedom (d.o.f) in its future input dynamics, an inconsistency between open-loop and closed-loop predictions will occur [7], [16]. Since the current decision making could then be ill-posed, the accuracy of a constrained solution might also be affected, especially when the validation horizon is selected far beyond the coincidence point [8].

B. Laguerre based PFC (LPFC)

1) *Future input dynamics*: The main difference between LPFC and PFC is that the future predicted input dynamics are shaped via a first-order Laguerre polynomial (in effect, a simple exponential decay function with pole a) so that it will converge to the expected steady state input u_{ss} [15], [16]. Thus, instead of the constant dynamics assumption of PFC, the future input is modified to

$$\underline{u}_k = u_{ss} + L\eta \quad (9)$$

where L is the vector ($L = [1, a, a^2, \dots, a^{n-1}]^T$) and η is a degree of freedom. For a general transfer function $G_m(z) = B(z)A(z)^{-1}$, the value u_{ss} is estimated as:

$$u_{ss} = G_m(z)^{-1}(R - d_k) \quad (10)$$

The inclusion of error term d_k in (10) is to ensure an unbiased estimation.

Remark 2: For a first-order system, a should be equal to λ to ensure consistent dynamics with the target trajectory [16]. Although for higher-order systems, the value of a can be tuned for faster convergence [15], this work will only use $a = \lambda$ to keep the sensitivity analysis transparent.

2) *LPFC control law:* The output prediction of (4) is modified with the new input dynamics of (9) to give:

$$y_{m,k+n|k} = H(u_{ss} + L\eta) + Pu_{\leftarrow k} + Qy_{\leftarrow k} \quad (11)$$

The equality of (6) now becomes:

$$HL\eta + hu_{ss} + Pu_{\leftarrow k} + Qy_{\leftarrow k} - y_{m,k} = (r - y_{p,k})(1 - \lambda^n) \quad (12)$$

and the control law is computed by solving for η as:

$$\eta = \frac{1}{HL} \left[(1 - \lambda^n)r - (1 - \lambda^n)y_{p,k} - hu_{ss} - Qy_{\leftarrow k} + y_{m,k} - Pu_{\leftarrow k} \right] \quad (13)$$

Due to the receding horizon principle [3] and the definition of $L(z)$, the current input is defined as:

$$u_k = u_{ss} + \eta \quad (14)$$

Noting the structure of u_{ss} in (10) and η in (13), the manipulated input u_k in (14) can be altered into vector form simply by rearranging the algebra and grouping the common terms into parameters F_l , N_l , M_l and \hat{D}_l so that:

$$u_k = F_l r - N_l y_{\leftarrow k} - M_l y_{p,k} - \hat{D}_l \Delta u_k \quad (15)$$

Remark 3: It has been shown in [16] that LPFC law of (15) manages to improve the prediction consistency and the efficacy of λ as tuning parameter compared to the conventional PFC law of (8). In addition, the constrained solution becomes more accurate and less conservative [8].

C. General Sensitivity function for IM structure

From the previous subsections, it is clear that both PFC and LPFC can be represented by a fixed control law as in (8) and (15). These are used in the derivation of sensitivity functions presented next to analyse their respective robustness [1].

First consider a generic formulation of the control law within an IM structure:

$$u_k = Fr - Ny_{\leftarrow k} - My_{p,k} - \hat{D}\Delta u_k \quad (16)$$

This can be represented in a transfer function form, where the vectors of

$$\begin{aligned} N &= [N_0, N_1, N_2, \dots, N_n] \\ \hat{D} &= [\hat{D}_0, \hat{D}_1, \hat{D}_2, \dots, \hat{D}_n] \end{aligned} \quad (17)$$

are defined in the z domain as:

$$\begin{aligned} N(z) &= N_0 + N_1 z^{-1} + N_2 z^{-2} + \dots + N_n z^{-n} \\ \hat{D}(z) &= \hat{D}_0 + \hat{D}_1 z^{-1} + \hat{D}_2 z^{-2} + \dots + \hat{D}_n z^{-n} \\ D(z) &= 1 + z^{-1} \hat{D}(z) \end{aligned} \quad (18)$$

Noting the definitions of $u_{\leftarrow k}$ and $y_{\leftarrow k}$ in (5), the sensitivity functions are derived based on a closed-loop form of:

$$D(z)u_k = F(z)r - N(z)y_{m,k} - M(z)y_{p,k} \quad (19)$$

alongside the model/plant equations (e.g. $y_{m,k} = B(z)A(z)^{-1}u_k$) and hence equation (19) can be replaced by:

$$\underbrace{[D(z) + N(z)B(z)A(z)^{-1}]}_{D_i(z)} u_k = F(z)r - M(z)y_{p,k} \quad (20)$$

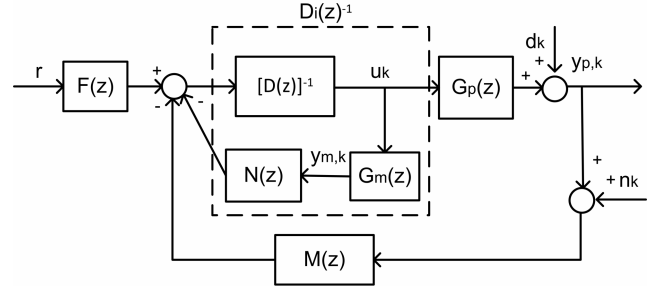


Fig. 2: PFC control loop.

Fig. 2 indicates the equivalent block diagram with the addition of measurement noise n_k and output disturbance d_k . From the structure, the effective control law can be simplified to $K(z) = M(z)[D_i(z)\Delta]^{-1}$. Assuming system $G(z) = B(z)A(z)^{-1}$, the closed-loop pole polynomial $P_i(z) = 1 + K(z)G(z)$ is represented as:

$$P_i(z) = D_i(z)A(z) + M(z)B(z) \quad (21)$$

The sensitivity of the input to noise is derived by finding the transference from $n(z)$ to $u(z)$ (refer to Fig. 2):

$$S_{un} = K(z)[1 + K(z)G(z)]^{-1} = M(z)P_i(z)^{-1}A(z) \quad (22)$$

Similarly, the sensitivity of output to disturbance is obtained by solving the transference from $d(z)$ to $y(z)$:

$$S_{yd} = [1 + K(z)G(z)]^{-1} = A(z)P_i(z)^{-1}D_i(z) \quad (23)$$

Finally, the multiplicative uncertainty is modelled as $G(z) \rightarrow (1 + \delta)G(z)$, for δ a scalar (possibly frequency dependent). Thus the closed-loop pole sensitivity to multiplicative uncertainty becomes:

$$\begin{aligned} P_c &= [1 + G(1 + \delta)K] = 0 \\ S_g &= GK[1 + K(z)G(z)]^{-1} = M(z)P_i(z)^{-1}B(z) \end{aligned} \quad (24)$$

D. Summary of Control Laws

Table I summarises some of the sensitivity functions for PFC and LPFC. It is noted that the structures of all the sensitivity functions are same, but obviously with different parameters and hence, different sensitivity responses should be expected.

TABLE I: Sensitivity functions for PFC and LPFC.

Algorithm	PFC	LPFC
S_{un}	$M_p(z)P_{i,p}(z)^{-1}A(z)$	$M_l(z)P_{i,l}(z)^{-1}A(z)$
S_{yd}	$A(z)P_{i,p}(z)^{-1}D_{i,p}(z)$	$A(z)P_{i,l}(z)^{-1}D_{i,l}(z)$
S_g	$M_p(z)P_{i,p}^{-1}B(z)$	$M_l(z)P_{i,l}^{-1}B(z)$

The polynomials $M(z)$, $D(z)$, $P_i(z)$ used a subscript p for PFC, while for LPFC the subscript is l .

III. NUMERICAL EXAMPLES

This section presents the sensitivity analysis of unconstrained second order over-damped process (25) as constraint handling would imply non-linear control. In fact, if the loop structure has low sensitivity in the nominal case, it is likely to carry over for the constrained case. For the first example, both PFC and LPFC are tuned using a faster λ compared to the slowest open-loop pole. The second example demonstrates the effect of loop sensitivity when the controllers are tuned to have almost similar closed-loop poles. The outcome of this analysis is then validated with the closed-loop simulation using Matlab.

$$G_1 = \frac{0.1z^{-1} + 0.4z^{-2}}{(1 - 0.5z^{-1})(1 - 0.9z^{-1})} \quad (25)$$

A. First example

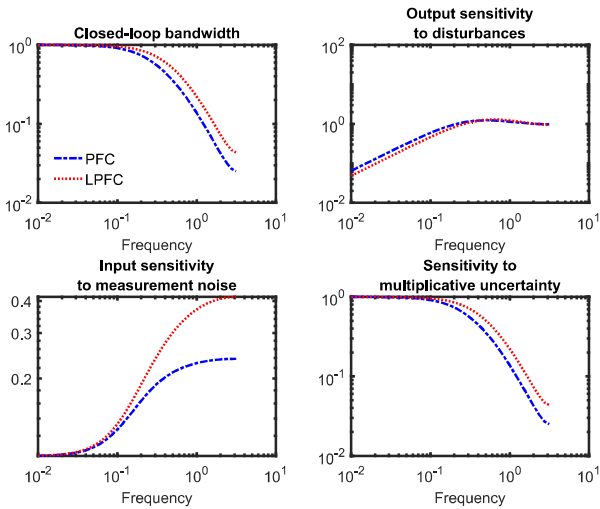


Fig. 3: Sensitivity plot for process G_1 with $\lambda = 0.7$ and $n = 7$.

In this example, the system (25) is considered to track a unit set point. The desired pole is set to $\lambda = 0.7$, while the coincidence point is tuned at $n = 7$ using conjecture presented in [7], that is corresponding to 40% to 80% rise of the step response to the steady-state value.

To analyse the trade-off between performance and robustness of PFC and LPFC, the Bode plots of each sensitivity function are plotted together with their closed-loop bandwidth (see Fig. 3). It can be observed that:

- for this particular selection of tuning parameters, LPFC (red dotted line) has a higher bandwidth compared to PFC (blue dashed line). Since LPFC has a faster dynamics, it becomes less sensitive in rejecting low-frequency disturbance..
- However, higher bandwidth requires more aggressive input activity, and thus LPFC becomes more sensitive to measurement noise and modelling uncertainty compared to conventional PFC.

One could argue that PFC has failed to deliver the desired bandwidth and if LPFC were to be tuned to give an equivalent

lower bandwidth, in all likelihood, the sensitivities would be similar.

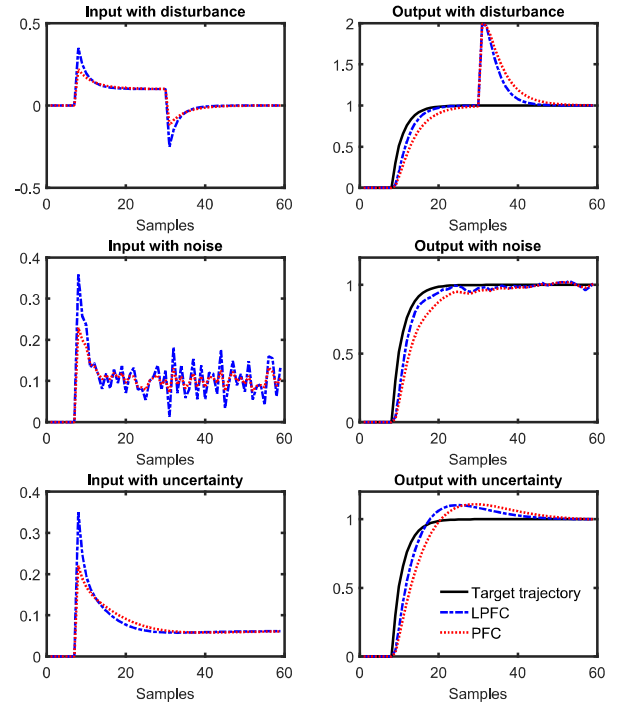


Fig. 4: Closed-loop response of process G_1 with $\lambda = 0.7$ and $n = 7$ in the presence of disturbance, noise, and uncertainty.

To validate this analysis, a closed-loop control (see Fig. 4) is simulated with three different conditions:

- 1) A step output disturbance ($d = 1$) is added to the 30th sample.
- 2) The output measurement is corrupted by Gaussian random white noise with variance of 0.1.
- 3) System $G_{1,m}$ (26) is used to predict the future dynamics instead of G_1 to demonstrate the effect of uncertainty.

$$G_{1,m} = \frac{0.12z^{-1} + 0.37z^{-2}}{1 - 1.37z^{-1} + 0.4z^{-2}} \quad (26)$$

The simulation outcomes reflect the previous sensitivity analysis whereby:

- LPFC converges approximately 2 samples faster in tracking the target and rejecting the output disturbance with almost similar overshoot ($y_{max} = 2$) compared to PFC.
- On the other hands, LPFC reacts more to the noise in the input compared to conventional PFC.
- For parameter uncertainty, both controllers manage to converge towards the steady-state value but with apparent differences in their closed-loop response.

In this example, it is clear that LPFC is slightly less robust than PFC in handling noise and uncertainty, yet better in rejecting disturbance and tracking the target, but that observation is most likely linked to the difference in implied closed-loop

poles with LPFC delivering the desired pole and PFC not doing so.

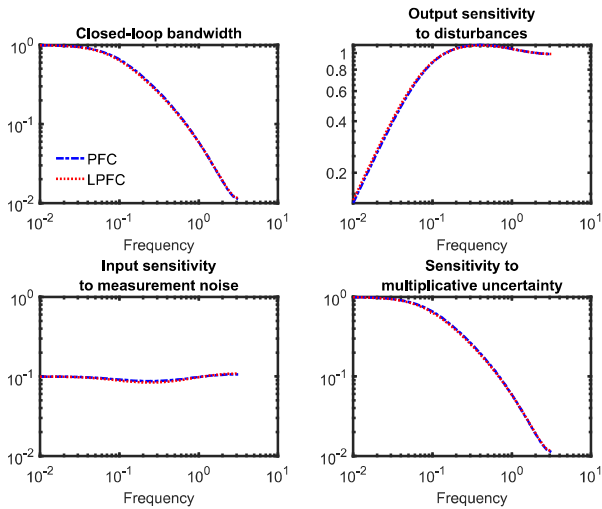


Fig. 5: Sensitivity plot for process G_1 with $\lambda = 0.92$ and $n = 9$.

B. Second example

The next example looks at the effect of sensitivity when the process (25) is tuned using slower $\lambda = 0.92$ (almost similar with the slowest open-loop pole). Based on the same procedure [7], the coincidence point $n = 9$ is selected to track a unit set point. It can be observed that (see Fig. 5):

- With the selected tuning parameters, LPFC and PFC have almost a similar bandwidth.
- As a consequence, both controllers are giving a close sensitivity outcome with respect to disturbance, noise and modelling uncertainty.

Again to validate the sensitivity analysis, the closed-loop simulation is run to track a unity set point for three different cases (similar as previous example). The outcomes in Fig. 6 demonstrates that:

- PFC and LPFC converge at the same rate and very close to the target trajectory while rejecting the disturbance with overshoot approximately around $y_{max} = 1.8$.
- Similar observation can be seen with the presence of noise and modelling uncertainty where both controllers performance are almost same.

C. Summary

In summary, for the two cases given, the controller sensitivity is related to the achieved closed-loop bandwidth. LPFC is better at delivering the target λ whereas PFC often gives a slower response than desired when large n is required. In consequence, for the same λ , LPFC is usually more highly tuned and thus more sensitive to noise and modelling uncertainty. However, where the two control laws give similar closed-loop poles (perhaps by deploying different λ), their

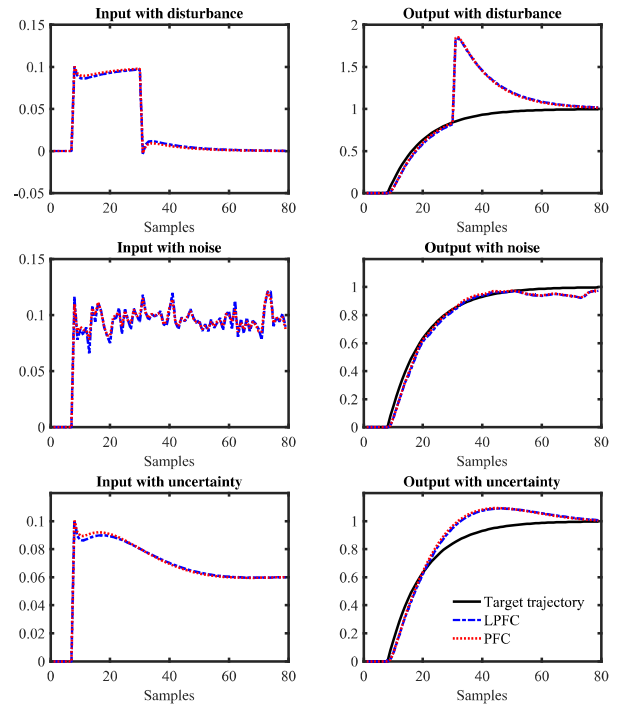


Fig. 6: Closed-loop response of process G_1 with $\lambda = 0.92$ and $n = 9$ in the presence of disturbance, noise and uncertainty.



Fig. 7: Quanser SRV02 servo based unit.

sensitivities are similar. Therefore, LPFC is a better base on which to explore the trade-offs in the sensitivity, as there is a stronger connection between the tuning parameters and the achieved closed-loop performance [16] in addition to a better constraint handling due to its well-posed decision and prediction consistency as discussed in [8].

IV. REAL TIME SYSTEM IMPLEMENTATION

This section demonstrates the practicality of LPFC to control a real system, that is a Quanser SRV02 servo based unit [17]. The servo is powered by a Quanser VoltPAQ-X1 amplifier that comes with National Instrument ELVIS II+ multifunctional data acquisition device. The controller

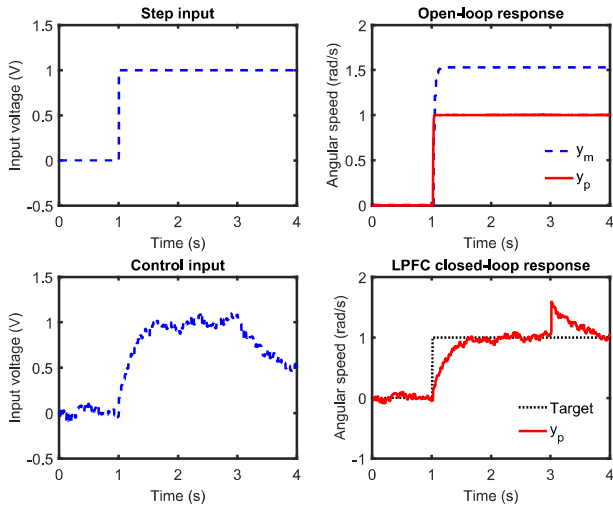


Fig. 8: Step response and LPFC closed-loop behaviour for process G_2 .

is run by National Instrument LabVIEW software via USB connection (see Fig. 7). The objective is to track the desired servo angular speed, $\dot{\theta}(t)$ by regulating the supplied voltage, $V(t)$. The mathematical model is given as [17]:

$$0.0254\ddot{\theta}(t) = 1.53V(t) - \dot{\theta}(t) \quad (27)$$

where $\ddot{\theta}(t)$ is the servo angular acceleration. Converting the model (27) to discrete form with sampling time $0.02s$, the transfer function of angular speed to voltage input becomes:

$$G_2 = \frac{0.8338}{1 - 0.455z^{-1}} \quad (28)$$

The upper Fig. 6 shows the modelling uncertainty between the process y_p and model y_m subjected to a step input u . To track the angular speed at 1 rad/s, LPFC is tuned with $n = 1$ (often a sensible choice for a first-order system [7]) with desired CLTR at $0.5s$ (equivalent to $\lambda = 0.89$). It is noted that at $3s$, there is a step output disturbance ($d = 2$) entering the system while the measurement is corrupted by Gaussian white noise with variance of 0.5 . The closed-loop response (see lower Fig. 8) shows that:

- LPFC manages to reduce some noise transmission to the input with approximate 0.2 variance from 0.5 , while rejecting the output disturbance.
- Although there is modelling uncertainty, the selected CLTR is still achieved at $0.5s$ with minimum offset error.

V. CONCLUSIONS

This work provides a formal sensitivity analysis of LPFC in the presence of noise, disturbance and modelling uncertainty. The performance is then compared with the conventional PFC control law. Indeed it is clear that when using LPFC, a user need to pay a small trade-off by having a more sensitive controller to noise and uncertainty since it is highly tuned with a larger bandwidth than conventional PFC. However, both

controllers may typically have similar sensitivities if giving similar closed-loop poles which would indicate a preference for LPFC in general due to easier tuning and other advantages as discussed in *Remark 3*.

Future work will consider the analysis of different PFC structures that deal with more challenging dynamics and unstable systems as PFC is currently has a number of ad-hoc constructive methods to improve its closed-loop behaviour. In addition, a core issue that also needs to be considered is the impact of modelling assumptions on sensitivity. This paper assumes an IM model of Fig. 1, so it would be interesting to consider how sensitivity might change with alternative prediction models such as T-filter [14].

ACKNOWLEDGMENT

The first author would like to acknowledge International Islamic University Malaysia and Ministry of Higher Education Malaysia for funding this work.

REFERENCES

- [1] J. A. Rossiter, *Model predictive control: a practical approach*, CRC Press, 2003.
- [2] L. G. Bleris and M. V. Kothare, "Real-Time implementation of Model Predictive Control," American Control Conference, Portland, USA, 2015.
- [3] J. Richalet, A. Rault, J.L. Testud and J. Papon, Model predictive heuristic control: applications to industrial processes, *Automatica*, 14(5), 413-428, 1978.
- [4] J. Richalet, and D. O'Donovan, *Predictive Functional Control: principles and industrial applications*. Springer-Verlag, 2009.
- [5] J. Richalet, and D. O'Donovan, "Elementary Predictive Functional Control: a tutorial," Int. Symposium on Advanced Control of Industrial Processes, 2011, pp. 306-313.
- [6] R. Haber, J.A. Rossiter and K. Zabet, "An Alternative for PID control: Predictive Functional Control - A Tutorial," American Control Conference (ACC), 2016, pp. 6935-6940
- [7] J. A. Rossiter, and R. Haber, "The effect of coincidence horizon on predictive functional control," *Processes*, 3, 1, pp. 25-45, 2015.
- [8] M. Abdullah, J. A. Rossiter and R. Haber, "Development of constrained predictive functional control using laguerre function based prediction," IFAC World Congress, 2017.
- [9] 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.
- [10] J. A. Rossiter, "Input shaping for PFC: how and why?," *J. Control and Decision*, pp. 1-14, Sep. 2015.
- [11] M. Khadir and J. Ringwood, "Extension of first order predictive functional controllers to handle higher order internal models," *Int. Journal of Applied Mathematics and Comp. Science*, vol. 18, no. 2, pp. 229-239, 2008.
- [12] 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*, In Press., 2017.
- [13] 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.
- [14] M. Abdullah and J. A. Rossiter, "The effect of model structure on the noise and disturbance sensitivity of Predictive Functional Control", under review for European Control Conference, 2018.
- [15] M. Abdullah and J. A. Rossiter, "Alternative Method for Predictive Functional Control to Handle an Integrating Process", under review for Advances in PID, 2018.
- [16] M. Abdullah and J. A. Rossiter, "Utilising Laguerre function in predictive functional control to ensure prediction consistency," 11th Int. Conf. on Control, Belfast, UK, 2016.
- [17] *Quanser user manual SRV02 rotary servo based unit set up and configuration*. Quanser Inc, 2012.

Appendix I

**SENSITIVITY ANALYSIS FOR AN INPUT
SHAPING PREDICTIVE FUNCTIONAL
CONTROL FOR PROCESSES WITH
CHALLENGING DYNAMICS**

M. Abdullah and J. A. Rossiter

This paper has been submitted to:

The 3rd IEEE Conference on Control Technology and Applications, 2019

Declaration form

SENSITIVITY ANALYSIS FOR AN INPUT SHAPING PREDICTIVE FUNCTIONAL CONTROL FOR
PROCESSES WITH CHALLENGING DYNAMICS

(The 3rd IEEE Conference on Control Technology and Applications, 2019)

Contributions of authors:

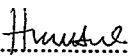
M. Abdullah

Provided the initial idea, formulations, codes, simulations and draft of this paper.

J. A. Rossiter

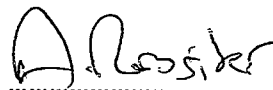
Supervised M. Abdullah and proofread the paper.

Signatures:


.....

M. Abdullah

(First author)


.....

J.A. Rossiter

(Second author)

Sensitivity Analysis of an Input Shaping Predictive Functional Control for Processes with Challenging Dynamics

Muhammad Abdullah^{1,2} and John Anthony Rossiter²

Abstract—This paper presents a formal sensitivity analysis of a Pole Shaping Predictive Functional Control (PS-PFC) algorithm, recently proposed to handle processes with open-loop divergent or oscillatory dynamics. Since the PS-PFC utilises a CARIMA based model, the control law may be sensitive to high-frequency noise. Hence, in this paper, the use of a T-filter is proposed to improve the noise sensitivity. Nevertheless, there is a trade-off in the controller response to a low-frequency disturbance and parameter uncertainty so here, both sensitivity plots and closed-loop simulations are used to analyse the controller robustness. The practicality of PS-PFC control law is then validated in real-time laboratory hardware.

Keywords—Predictive Control, PFC, Sensitivity Analysis, Input Shaping, T-filter, Noise, Disturbance, Uncertainty.

I. INTRODUCTION

Predictive Functional Control (PFC) has been widely used in many simple industrial applications due to its transparent tuning procedure [1]–[3]. Unlike its prime competitor, a Proportional-Integral-Derivative (PID) controller often needs a well-trained user to tune the gains and implement a rule-based constraint handling effectively. Since PFC adapts a similar concept to the more advanced Model Predictive Control (MPC), it inherits some of the benefits such as systematic handling of constraints and delays [2], [4]. Indeed, although the performance of PFC is often comparable to advanced MPC in the SISO case, it cannot be fairly compared with those because the optimisation is based on a single degree of freedom (d.o.f) and thus requires minimum computation, simple coding and is much faster and cheaper.

Despite the limited attention received in the academic literature, PFC has undergone several developments to tackle some of its weaknesses in providing *a priori* guarantees of stability [5], [6], consistent prediction [7] and accurate constrained solutions [8]. Besides, there are a few attempts to discuss and analyse the robustness of PFC with different prediction structures [9], [10] systematically. Nevertheless, the scope of discussion for PFC has been largely limited to stable and straightforward dynamical processes, as more challenging applications such as those with open-loop divergent or oscillatory behaviours often need a tailored technique to handle the undesirable poles effectively.

There are several frameworks to improve the effectiveness of PFC in handling challenging dynamics: cascade structures [2], pole-placement [11] or input shaping [12], where each

method has its pros and cons. The cascade structure pre-stabilises a process while retaining an Independent Model (IM) structure which can be useful for robustness to high-frequency uncertainty (standard IM structures cannot handle open-loop unstable plant and so are not discussed in this paper.). However, the tuning of the associated stabilisation gain is not always systematic or obvious. Methods such as pole-placement PFC (PP-PFC) can handle oscillatory dynamics well but are not applicable to open-loop unstable processes. Besides, PP-PFC also faces difficulty in satisfying long-range constraints as its coincidence points need to be the first sample. Direct cancellation of the unwanted poles within the predictions is another attractive method that can generalise the pre-stabilisation for different challenging dynamics, but such an approach often requires over aggressive input activity that is not entirely practical for a real process.

A recent proposal is Pole Shaping PFC (PS-PFC); here, the undesirable pole is shaped instead of cancelled to ensure smooth dynamics [13]. Although each different dynamic requires a specific shaping, the underlying framework and formulation are the same. Notably, PS-PFC has the advantage that the constrained solution is guaranteed to be recursively feasible. However, the impact on the robustness of input shaping methods is unstudied and forms the main contribution of this paper. PS-PFC can handle uncertainty and disturbance, nevertheless, the structure may be sensitive to a high-frequency measurement noise resulting in poor control performance or at worst instability [15]. Hence, the second contribution is to propose the use of T-filter to improve the input sensitivity to noise while analysing its trade-offs against the sensitivity to disturbances and parameter uncertainty as this structure is rarely used within the PFC framework [9].

Section II presents a brief formulation of nominal PFC, Pole Shaping PFC, T-filter prediction and derivation of the associated sensitivity functions. Section III analyses the loop sensitivity and performance of several numerical examples with different challenging dynamics. Section IV discusses the practicality of the proposed algorithm with real laboratory hardware and section V provides the conclusion.

II. PFC STRUCTURES AND SENSITIVITY FUNCTIONS

This section presents a brief formulation of nominal PFC control law and the concept of PS-PFC. The 2nd subsection develops the inclusion of a T-filter into the prediction structure for PS-PFC and the derivation of the associated sensitivity functions. Without loss of generality, this paper assumes an underlying CARIMA model although different prediction structures may be used for a stable process.

¹Dept. of Mech. Engineering, International Islamic Univ. Malaysia, Jalan Gombak, 53100, Kuala Lumpur Malaysia. mohd_abdl@iium.edu.my

² Dept. of ACSE, University of Sheffield, S1 3JD, UK. MAbdulah2@sheffield.ac.uk, j.a.rossiter@sheffield.ac.uk

A. Nominal PFC control law

This subsection presents a brief formulation of PFC; more detailed derivations, theory and concepts are available in references [1]–[4]. The original concept of PFC is to track a first order target trajectory by enforcing the equality [14]:

$$y_{k+n_y|k} = (1 - \lambda^{n_y})r + \lambda^{n_y}y_k \quad (1)$$

where $y_{k+n_y|k}$ is the n_y -step ahead system prediction at sample time k , λ is the desired closed-loop pole (convergence rate of output y_k to target r), and tuning parameter n_y is a coincidence horizon where the system prediction and the target trajectory are forced to match. The prediction for a CARIMA model utilises notation [15] defined for a general polynomial $f(z) = f_0 + f_1z^{-1} + \dots + f_nz^{-n}$ as follows:

$$C_f = \begin{bmatrix} f_0 & 0 & 0 & \cdots \\ f_1 & f_0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ f_n & f_{n-1} & f_{n-2} & \cdots \end{bmatrix}, H_f = \begin{bmatrix} f_1 & f_2 & \cdots & f_n \\ f_2 & f_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad (2)$$

Given the model below and defining $A(z) = \Delta a(z)$:

$$\Delta y(z) = \frac{b(z)}{a(z)}\Delta u(z) \Rightarrow y(z) = \frac{b(z)}{A(z)}\Delta u(z) \quad (3)$$

the simplified prediction structure is constructed as:

$$y_{k+n_y|k} = \underbrace{C_A^{-1}C_b}_{H}\Delta \underline{u}_k + \underbrace{C_A^{-1}H_b}_{P}\Delta \underline{u}_k + \underbrace{C_A^{-1}H_A}_{Q}\underline{y}_k \quad (4)$$

where parameters H, P, Q depend on model parameters and:

$$\Delta \underline{u}_k = \begin{bmatrix} \Delta u_k \\ \vdots \\ \Delta u_{k+n-1} \end{bmatrix}; \underline{u}_k = \begin{bmatrix} \Delta u_{k-1} \\ \vdots \\ \Delta u_{k-m} \end{bmatrix}; \underline{y}_k = \begin{bmatrix} y_k \\ \vdots \\ y_{k-m} \end{bmatrix} \quad (5)$$

Substituting the n_y th row of prediction (4) into (1) gives:

$$H_{n_y}\Delta \underline{u}_k + P_{n_y}\Delta \underline{u}_k + Q_{n_y}\underline{y}_k = (1 - \lambda^{n_y})r + \lambda^{n_y}y_k \quad (6)$$

As PFC predictions assume a constant future input [2], where $\Delta u_{k+i} = 0$ for $i > 0$, only the first column (h_{n_y}) of matrix H_{n_y} is used to construct the final control law:

$$h_{n_y}\Delta u_k = (1 - \lambda^{n_y})r + \lambda^{n_y}y_k - Q_{n_y}\underline{y}_k - P_{n_y}\Delta \underline{u}_k \quad (7)$$

Remark 1: A typical PFC can perform well with any 1st order system or high order stable and over-damped dynamics if the tuning parameters λ and n_y are selected carefully [14]. However, PFC becomes difficult to tune for open-loop unstable or oscillating process and needs a more nuanced modification to refine these undesirable dynamics [2], [14].

B. Pole Shaping PFC (PS-PFC)

The first step of pre-stabilisation is to separate the open-loop poles in the denominator of model (3) as:

$$a(z) = a^-(z)a^+(z) \quad (8)$$

where $a^+(z)$ is the undesired poles (unstable or under-damped poles) and $a^-(z)$ is the stable poles. Hence, the future output prediction of (4) can be represented as:

$$\underline{y}_{\rightarrow k+1} = [C_{a-\Delta}]^{-1}[C_{a^+}]^{-1}[C_b\Delta \underline{u}_{\rightarrow k} + \underbrace{H_b\Delta \underline{u}_{\leftarrow k} - H_A\underline{y}_{\leftarrow k}}_p] \quad (9)$$

Lemma 1: Prediction in (9) is convergent if its input is shaped to satisfy the condition of (10) at each sample

$$[C_bC_\alpha^{-1}w + p] = C_{a^+}C_\alpha^{-1}\gamma \Rightarrow \Delta \underline{u}_{\rightarrow}(z) = \frac{w(z)}{\alpha(z)} \quad (10)$$

where γ is a convergent sequence or a polynomial that ensure the corresponding prediction in (9) eliminates the undesirable poles a^+ while replacing it with the shaped poles α .

Proof: Substituting condition of (10) in prediction (9):

$$\begin{aligned} \underline{y}_{\rightarrow k+1} &= [C_{a-\Delta}]^{-1}[C_{a^+}]^{-1}[C_b\Delta \underline{u}_{\rightarrow k} + p] \\ &= [C_{a-\Delta}]^{-1}[C_{a^+}]^{-1}[C_bC_\alpha^{-1}w + p] \\ &= [C_{a-\Delta}]^{-1}[C_{a^+}]^{-1}[C_{a^+}C_\alpha^{-1}\gamma] = C_{a-\Delta}^{-1}C_\alpha^{-1}\gamma \end{aligned} \quad (11)$$

It is evident that both the desired and shaped poles are in the predictions for both the input and output. \square

Remark 2: The new requirement (10) can be solved by small number of simultaneous equation [15], where the minimal order solution for w and γ for suitable P_1, P_2 are:

$$w = P_1p; \quad \gamma = P_2p \quad (12)$$

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

A proposal to select a suitable shaping pole $\alpha(z)$ is made based on the link to the target closed-loop behaviour and/or system knowledge:

- 1) For integrating system, a pole factor $\alpha = (1 - 0.5z^{-1})$ is used as it represents a simple half-way house trade off between integrating factor $(1 - z^{-1})$ and cancelling the pole on the origin $(1 - 0z^{-1})$.
- 2) For a process with significant under-damping, the implied $\alpha(z)$ has only real poles which are chosen to be close to the real parts of the oscillatory poles. This selection reduces the undesirable oscillation in the output predictions, but does not change the convergence rate, albeit the input may be somewhat oscillatory.
- 3) For open-loop unstable systems, a simple default solution simply inverts the unstable poles, that is, defining $\alpha(z)$ such that $a^+(z_i) = 0 \Rightarrow \alpha(1/z_i) = 0$.

Theorem 1: With a choice of α , a convergent prediction class which embeds both the desired asymptotic poles and some degrees of freedom (d.o.f.) can be defined from:

$$w = P_1p + C_{a^+}\phi; \quad \Delta \underline{u}_{\rightarrow k} = [C_\alpha]^{-1}[P_1p + C_{a^+}\phi] \quad (13)$$

where convergent Infinite or Finite Impulse Response (IIR or FIR) ϕ constitutes the d.o.f.

Proof: Based on superposition. The additional component in w , that is $C_{a^+}\phi$, necessarily cancels the undesirable poles and replace it with the shaped pole α which gives overall convergent output predictions. So using (11), then:

$$\begin{aligned} \underline{y}_{\rightarrow k+1} &= C_{a-\Delta}^{-1}C_{a^+}^{-1}[C_b\Delta \underline{u}_{\rightarrow k} + p] \\ &= C_{a-\Delta}^{-1}C_\alpha^{-1}\gamma + C_{a-\Delta}^{-1}C_{a^+}^{-1}C_\alpha^{-1}[C_{a^+}C_b\phi] \\ &= C_{a-\Delta}^{-1}C_\alpha^{-1}[\gamma + C_b\phi] \end{aligned} \quad (14) \quad \square$$

Remark 3: By extracting the n_y th row and noting the definition of p in (9), the n_y step ahead prediction from (14) can be rearranged in a more general form as:

$$y_{k+n_y|k} = h_{n_y,\alpha}\phi + P_{n_y,\alpha}\Delta_{\leftarrow}u_k + Q_{n_y,\alpha}\tilde{y}_k \quad (15)$$

for suitable $h_{n_y,\alpha}, P_{n_y,\alpha}, Q_{n_y,\alpha}$ and it is noted that as is conventional for PFC, ϕ has just a single non-zero parameter in order to retain computational simplicity and also a single d.o.f. for satisfying the control law (1).

Algorithm 1: The d.o.f ϕ is computed by substituting prediction (15) into equality (1) and thus:

$$\phi = \frac{1}{h_{n_y,\alpha}} \left[(1 - \lambda^{n_y})r + \lambda^{n_y}y_k - Q_{n_y,\alpha}\tilde{y}_k - P_{n_y,\alpha}\Delta_{\leftarrow}u_k \right] \quad (16)$$

then, the current input increment Δu_k is determined simply by inserting ϕ into the predicted input of (13).

C. Shaped predictions with a T-filter

One of the reasons why PFC only received little attention in the academic literature is due to its inability in providing robust feasibility due to its simplicity requirement. Thus, a simple available option is to utilise a robust prediction structure wherein previous work [9]; a T-filter has been used on predictions (4) to improve the input sensitivity to noise. However, to the authors' knowledge no-one has looked at this in the context of shaped predictions and hence this subsection provides the required novel developments.

The T-filter serves as a low pass filter to reject high-frequency measurement noise without affecting the nominal tracking performance [16]. Theoretically, the measurement output is low-pass filtered before prediction and anti-filtered after prediction to restore the predicted data back to the correct domain before deploying the nominal algorithm. The desired T-filter T^{-1} is deployed as $\tilde{y}_k = y_k T^{-1}$ or $T\tilde{y}_k = y_k$. Define the filtered predictions as follows:

$$\tilde{y}_{k+1} = H\Delta_{\rightarrow}\tilde{u}_k + P\Delta_{\leftarrow}\tilde{u}_k + Q\tilde{y}_k \quad (17)$$

The relationship between the filtered and unfiltered predicted data can be represented using Toeplitz/Hankel form (2):

$$\tilde{y}_{k+1} = C_T\tilde{y}_{k+1} + H_T\tilde{y}_k; \quad \Delta_{\rightarrow}u_k = C_T\Delta_{\rightarrow}\tilde{u}_k + H_T\Delta_{\leftarrow}\tilde{u}_k \quad (18)$$

substituting (18) into (17) gives:

$$\underbrace{C_T^{-1}[\tilde{y}_{k+1} - H_T\tilde{y}_k]}_{\tilde{y}_k} = H \underbrace{C_T^{-1}[\Delta_{\rightarrow}u_k - H_T\Delta_{\leftarrow}\tilde{u}_k]}_{\Delta_{\leftarrow}\tilde{u}_k} + P\Delta_{\leftarrow}\tilde{u}_k + Q\tilde{y}_k \quad (19)$$

Multiplying through by C_T and grouping common terms:

$$\tilde{y}_{k+1} = H\Delta_{\rightarrow}u_k + \tilde{P}\Delta_{\leftarrow}\tilde{u}_k + \tilde{Q}\tilde{y}_k \quad (20)$$

where $\tilde{P} = [C_T P - H H_T]$ and $\tilde{Q} = [H_T + C_T Q]$. The difference between (20) and (4) are the last two terms which now are based on past filtered data.

Remark 4: When implementing T-filter to the PS-PFC control law, the matrix C_A need to be separated form

prediction (20). Noting the definition of H, P, Q in (4) and \tilde{P}, \tilde{Q} , the new representation can be formed as:

$$\tilde{y}_{k+1} = [C_{a-\Delta}]^{-1}[C_{a+}]^{-1}[C_b\Delta_{\rightarrow}u_k + \underbrace{\tilde{H}_b\Delta_{\leftarrow}\tilde{u}_k - \tilde{H}_A\tilde{y}_k}_p] \quad (21)$$

where $\tilde{H}_b = C_T H_b - C_b H_T$ and $\tilde{H}_A = C_A H_T + C_T H_A$. Now, the control law with a T-filter follows a similar derivation from (10) to (16) but using filtered prediction (21) instead of (9).

Remark 5: If $T = 1$, the prediction of (21) becomes nominal as in (9) given that $C_T = 1$ and $H_T = 0$. Thus the derivation of sensitivity functions for PS-PFC with and without T-filter can be generalised.

D. Sensitivity functions

An off-line sensitivity analysis is used to measure the controller robustness. Improving sensitivity in a specific frequency range may make it worse in another. Hence, this analysis can help to understand what sort of trade-off that one should expect when implementing a T-filter especially in the context of shaped predictions. This subsection will derive the associated sensitivity functions that are needed.

Based on the receding horizon principle, only the first computed input (e.g. (7)) is implemented. Since the first row of matrix C_{α}^{-1} and C_{a+} is always 1, the input trajectory of (13) and thus implied control law can be simplified to:

$$\Delta u_k = [P_1 p + \phi] \quad (22)$$

The control law is then rearranged into compact form by substituting ϕ_k from the filtered prediction of (21) into (22):

$$\Delta u_k = P_1 p + \frac{1}{h_{n_y,\alpha}} \left[(1 - \lambda^{n_y})r + \lambda^{n_y}y_k - Q_{n_y,\alpha}\tilde{y}_k - P_{n_y,\alpha}\Delta_{\leftarrow}\tilde{u}_k \right] \quad (23)$$

As p is representing both past filtered input and output, equation (23) is rearranged in a more general form by noting the definition of $y_k = C_T\tilde{y}_k + H_T\tilde{y}_{k-1}$ and grouping the common term of $r, \tilde{y}_k,$ and $\Delta_{\leftarrow}\tilde{u}_k$ into:

$$\begin{aligned} \Delta u_k &= F r - N\tilde{y}_k - \hat{D}\Delta_{\leftarrow}\tilde{u}_k \\ &= F r - N T^{-1}y_k - \hat{D}T^{-1}\Delta_{\leftarrow}u_k \end{aligned} \quad (24)$$

The above equation can be converted into transfer function form where the vector of $N = [N_0, N_1, N_2, \dots, N_n]$ and $\hat{D} = [\hat{D}_0, \hat{D}_1, \hat{D}_2, \dots, \hat{D}_n]$ are defined in the z domain as:

$$\begin{aligned} N(z) &= N_0 + N_1 z^{-1} + N_2 z^{-2} + \dots + N_n z^{-n} \\ \hat{D}(z) &= \hat{D}_0 + \hat{D}_1 z^{-1} + \hat{D}_2 z^{-2} + \dots + \hat{D}_n z^{-n} \\ D(z) &= 1 + z^{-1}\hat{D}(z) \end{aligned} \quad (25)$$

With the definitions of Δu_k and y_k in (5), the sensitivity functions are derived based on a fixed closed-loop form of:

$$D(z)T(z)^{-1}\Delta u_k = F(z)r - N(z)T(z)^{-1}y_k \quad (26)$$

Fig. 1 indicates the equivalent block diagram in the presence of measurement noise n and output disturbance d . The overall control law can be simplified to $K(z) = N(z)[D(z)\Delta]^{-1}$ and assuming system $G(z) = b(z)a(z)^{-1}$,

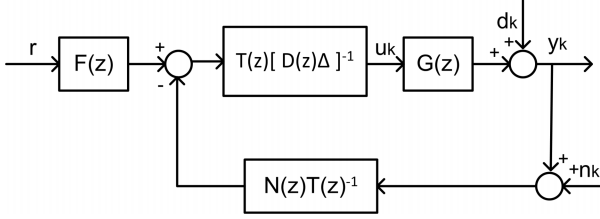


Fig. 1: PFC equivalent block diagram representation.

the closed-loop pole polynomial $P_c(z) = 1 + K(z)G(z)$ is represented as:

$$P_c(z) = D(z)a(z)\Delta + N(z)b(z) \quad (27)$$

The sensitivity of input to noise is derived via the transference from u to n (refer to Fig. 1):

$$S_{un} = K(z)[1 + K(z)G(z)]^{-1} = N(z)P_c(z)^{-1}a(z) \quad (28)$$

Similarly, the sensitivity of output to disturbance is obtained by solving the transference from y to d :

$$S_{yd} = [1 + K(z)G(z)]^{-1} = a(z)P_c(z)^{-1}D(z)\Delta \quad (29)$$

Then the multiplicative uncertainty can be modelled as $G(z) \rightarrow (1 + \delta)G(z)$, where δ is a scalar (possibly frequency dependent). Thus the closed-loop pole sensitivity to multiplicative uncertainty becomes:

$$\begin{aligned} P_c &= [1 + G(1 + \delta)K] = 0 \\ S_g &= GK[1 + K(z)G(z)]^{-1} = N(z)P_c(z)^{-1}b(z) \end{aligned} \quad (30)$$

III. NUMERICAL EXAMPLES

This section presents numerical examples and sensitivity analysis of PS-PFC with and without a T-filter for different types of challenging dynamics. It is of interest to determine whether the insights from GPC [16] carry over to the case with shaped predictions. These examples demonstrate the impact of the T-filter on the loop sensitivity and the comparison of their closed-loop performance in the presence of noise, disturbances and parameter uncertainty. Two arbitrary processes with different dynamics are considered:

- 1) A 3rd order under-damped process with two under-damped poles and $\alpha(z) = (1 - 0.8z^{-1})(1 - 0.8^{-1})$:

$$G_1 = \frac{0.85z^{-1} - 1.5z^{-2} + 0.85z^{-2}}{(1 - 0.6z^{-1})(1 - 1.6z^{-1} + 0.8z^{-2})} \quad (31)$$

- 2) A 2nd order unstable system with one pole is outside the unit circle and $\alpha(z) = (1 - 0.833z^{-1})$:

$$G_2 = \frac{0.4z^{-1} - 0.1z^{-2}}{(1 - 0.5z^{-1})(1 - 1.2z^{-1})} \quad (32)$$

A. Sensitivity analysis

Processes G_1 and G_2 are tuned with $n_y = 4$ and $n_y = 6$, respectively to track a unit set point with desired pole $\lambda = 0.7$. The coincidence horizon is selected according to the conjecture in [14], which is based on 40% to 80% rise of the step input to its steady-state value. The bode plots of each sensitivity function are plotted in Fig. 2 and 3 to

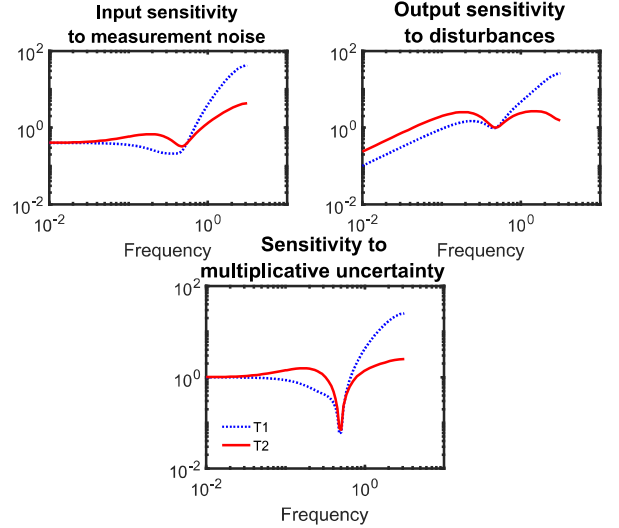


Fig. 2: Sensitivity plots for G_1 with different T-filters.

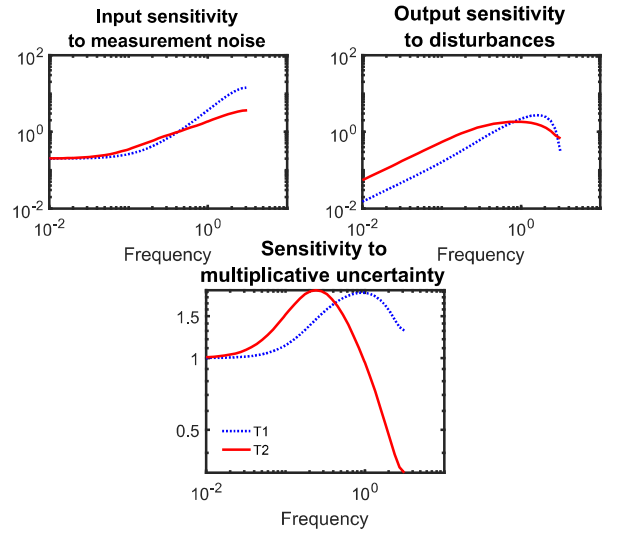


Fig. 3: Sensitivity plots for G_2 with different T-filters.

highlight the impact of a T-filter on the control loop, where $T_1 = 1$ has no filtering and $T_2 = 1 - 0.8z^{-1}$ is first order. For both cases, the overall outcome can be summarised as:

- PS-PFC with T_2 (red line) manages to reduce the overall sensitivity in the high-frequency domain, yet becomes more sensitive in low and mid frequency domain compared to T_1 (blue dotted line).
- This scenario means that the process will respond better with a T-filter in the presence of high-frequency noise, but worse in rejecting low or mid frequency disturbances and parameter uncertainty.

B. Closed-loop responses

Closed-loop responses for both processes are simulated with three different conditions (see Fig. 4 and 5) to validate the previous sensitivity analysis. The outcomes show:

- 1) When Gaussian random white noise corrupts the output measurement of both processes with the variance of

0.01, PS-PFC with T_2 rejects more noise compared to T_1 by reducing the input fluctuation approximately from variance 0.6 to 0.1 for G_1 and 0.3 to 0.1 for G_2 .

- 2) In the presence of output disturbance in G_1 ($d = 0.1$) and G_2 ($d = 0.5$) at 15th sample, responses with T_2 are slower in rejecting the disturbance than with T_1 , yet its input activity is far less aggressive.
- 3) When there is a parameter uncertainty (consider plant dynamics as $G_{1,p}$ (33) and $G_{2,p}$ (34)), both T_1 and T_2 provide offset-free tracking, but the response with T_2 is a bit slower compared to T_1 .

$$G_{1,p} = \frac{0.86z^{-1} - 1.52z^{-2} + 0.83z^{-2}}{(1 - 0.55z^{-1})(1 - 1.65z^{-1} + 0.82z^{-2})} \quad (33)$$

$$G_{2,p} = \frac{0.12z^{-1} - 0.38z^{-2}}{(1 - 0.4z^{-1})(1 - 1.1z^{-1})} \quad (34)$$

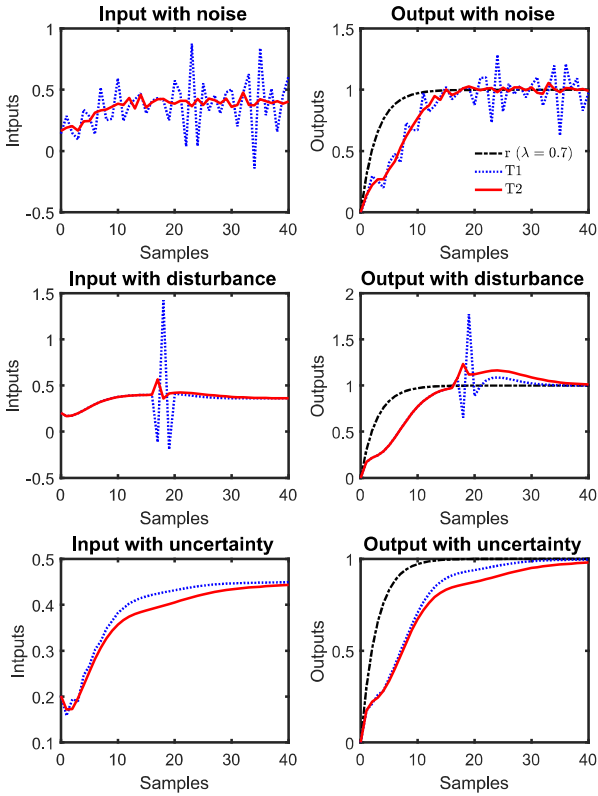


Fig. 4: Closed-loop responses of G_1 in the presence of noise, output disturbance and parameter uncertainty.

C. Summary

In practice, there is a trade-off as improving the sensitivity in one range it will become worse in another range. In both examples, the T-filter enables this trade-off to be performed as it clearly reduces input sensitivity to measurement noise thus reducing fatigue which is important for real applications. The corresponding reduction in performance when handling

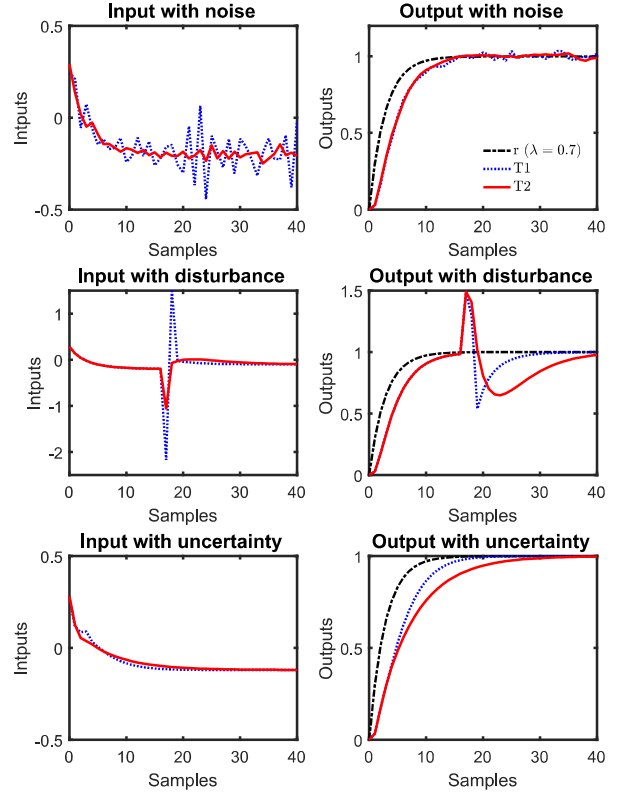


Fig. 5: Closed-loop responses of G_2 in the presence of noise, output disturbance and parameter uncertainty.

output disturbances and parameter uncertainty is less important as the integral action will deal with these uncertainties in the long term regardless. Nevertheless, although the results are similar in tone to [16], the reader is reminded that sensitivity is system dependent and hence, it is recommended to do the off-line sensitivity analysis for each case.

IV. REAL TIME SYSTEM EXAMPLE

In this section, the proposed controller is implemented in laboratory hardware. This process poses some challenges such as under-damping and the measured data and the controller model may differ in value and can lead to a failure if not addressed properly. A Quanser SRV02 servo based unit powered by a Quanser VoltPAQ-X1 amplifier with a flexible joint is used as a plant. This system is operated by National Instrument ELVIS II multifunctional data acquisition Fig. 6.

The control objective for this task is to track the angular position of the link $\theta(t)$ (measured in radians) by manipulating the supplied voltage $V(t)$. The angle θ increases positively in counter-clockwise (CCW) rotation with a positive the supplied voltage, and vice versa. A discrete-time transfer function with sampling time 0.02s is used to represent the system (for more details of the first principle model see [17]):

$$\frac{\theta(z)}{V_m(z)} = \frac{0.009625z^{-1} - 0.01011z^{-2} - 0.004445z^{-3} + 0.007632z^{-4}}{1 - 3.17z^{-1} + 3.92z^{-2} - 2.25z^{-3} + 0.5z^{-5}} \quad (35)$$

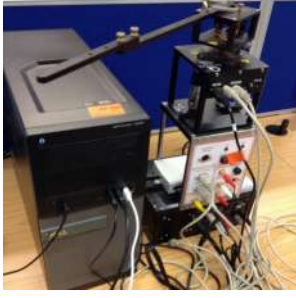


Fig. 6: Quanser SRV02 servo based unit.

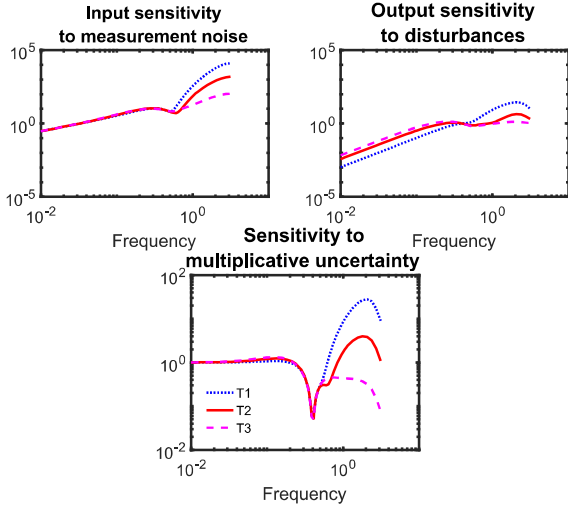


Fig. 7: Sensitivity plot for the servo kit with different PFC structures.

with two under-damped poles and one integrating pole. The control law is tuned to track an alternating set point between -1 rad/s and 1 rad/s with $\lambda = 0.7$, $n = 9$ and $\alpha(z) = (1 - 0.5z^{-1})(1 - 0.7477z^{-1})(1 - 0.7477z^{-1})$. Fig. 7 shows the sensitivity comparison with T-filter values $T_1 = 1$, $T_2 = 1 - 0.8z^{-1}$ and $T_3 = 1 - 1.6z^{-1} + 1.64z^{-2}$.

The trade-offs in sensitivity are similar to those noted in the previous section. Fig. 8 shows the real response where it is of interest that the simulation is acceptable only with T_3 as with T_1 and T_2 the input fluctuation is unacceptable/unsafe. The controller manages to track the alternating set-point without offset error while reducing the oscillation in the flexible link. Hence, in this case, the usage of a T-filter is essential for the effective utilisation of PS-PFC.

V. CONCLUSIONS

This paper proposes and demonstrates the adoption of a T-filter with the PS-PFC algorithm and thus provides a mechanism for improving sensitivity for systems where a classical PFC may not be implementable, such as for under-damping and unstable modes. The associated sensitivity analysis of PS-PFC for noise, disturbance and parameter uncertainty with and without the T-filter indicates that the benefits are similar to those achieved in the context of GPC. Hence, although generic proofs are not applicable, it is clearly shown

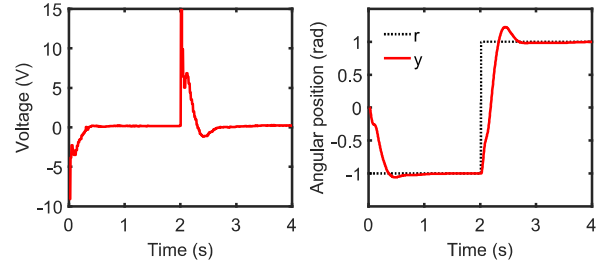


Fig. 8: Closed-loop performance of Quanser servo flexible joint with a second order T-filter.

that use of T-filter often helps in reducing the sensitivity of an input to measurement noise with only relative small deterioration in disturbance rejection. In practice, an off-line sensitivity analysis is essential to gain some insight into the robustness with differing prediction structures. Future work will look at the impact of changes in the parameters λ , n and also the choice of shaping poles $\alpha(z)$. A robustness comparison with alternative PFC methods such as cascade structures and pole placement also should be considered.

REFERENCES

- [1] J. Richalet, A. Rault, J.L. Testud and J. Papon, Model predictive heuristic control: applications to industrial processes, *Automatica*, 14(5), 413-428, 1978.
- [2] J. Richalet, and D. O'Donovan, *Predictive Functional Control: principles and industrial applications*. Springer-Verlag, 2009.
- [3] R. Haber, J.A. Rossiter and K. Zabet, "An alternative for PID control: Predictive Functional Control - a tutorial," American Control Conference (ACC), 2016, pp. 6935-6940
- [4] J. Richalet, and D. O'Donovan, "Elementary Predictive Functional Control: a tutorial," Int. Symposium on Advanced Control of Industrial Processes, 2011, pp. 306-313.
- [5] M. Khadir and J. Ringwood, "Stability issues for first order predictive functional controllers: extension to handle higher order internal models," Int. Conf. on Computer Systems and Information Tech., 2005.
- [6] J. A. Rossiter, "A priori stability results for PFC," *International journal of control*, vol. 90, no. 2, pp. 305-313, 2016.
- [7] M. Abdullah and J. A. Rossiter, "Utilising Laguerre function in predictive functional control to ensure prediction consistency," 11th Int. Conf. on Control, Belfast, UK, 2016.
- [8] M. Abdullah, J. A. Rossiter and R. Haber, "Development of constrained predictive functional control using Laguerre function based prediction," IFAC World Congress, 2017.
- [9] M. Abdullah and J. A. Rossiter, "The effect of model structure on the noise and disturbance sensitivity of Predictive Functional Control", European Control Conference, 2018.
- [10] 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.
- [11] 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, part 2, pp. 403-414, 2017.
- [12] J. A. Rossiter, "Input shaping for PFC: how and why?," *J. Control and Decision*, pp. 1-14, Sep. 2015.
- [13] M. Abdullah and J. A. Rossiter, "Input shaping predictive functional control for different types of challenging dynamics processes," Submitted to *Processes*, 2018.
- [14] J. A. Rossiter, and R. Haber, "The effect of coincidence horizon on predictive functional control," *Processes*, 3, 1, pp. 25-45, 2015.
- [15] J. A. Rossiter, *A first course in predictive control: 2nd edition*, CRC Press, 2018.
- [16] T-W. Yoon and D.W. Clarke, Observer design in receding horizon predictive control, *International Journal of Control*, 61, 171-191, 1995.
- [17] *Quanser user manual flexible joint experiment set-up and configuration*. Quanser Inc, 2012.