Article:
Zabet, K., Rossiter, J.A., Haber, R. et al. (2017) Pole-placement Predictive Functional Control for under-damped systems with real numbers algebra. ISA Transactions, 71 (Part 2). pp. 403-414. ISSN: 0019-0578

https://doi.org/10.1016/j.isatra.2017.08.002

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] *K. Zabet and R. Haber are with Cologne Univ. of Applied Sciences, Inst. of Plant and Process Engineering, D-50679 Koeln, Betzdorfer Str. 2, Germany. email: robert.haber@th-koeln.de*

[b] *J.A. Rossiter and M. Abdullah are with Dept. of Automatic Control and Systems Eng., University of Sheffield, S1 3JD, UK. email: j.a.rossiter@sheffield.ac.*

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.

Keywords: predictive functional control, pole-placement, under-damped system, real number algebra

1. Introduction

PFC (Predictive Functional Control) [1] is probably the most successful industrial implementation of model predictive control (MPC) 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 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 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 under-damped 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 form[1]:

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

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.

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:

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

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):*

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

Rearrange to determine the input as:

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

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, 2, 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].

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

- Figure 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.

- Figure 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:

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

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

$$y(k+1) = b_1 \frac{(1-\lambda) r(k) + (a_1 + \lambda) y(k)}{b_1} - a_1 y(k) = (1-\lambda) r(k) + \lambda y(k). \tag{10}$$

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 sub-model G_i based on the model steady-state gains of output $y_m^{(i)}$ (5) using the formulae:*

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

2. *Identify proposed inputs for each sub-model ($i = 1, ..., n$) using the control law (9):*

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

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

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

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

185 **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):

$$\begin{aligned} u(k) &= \sum_{i=1}^{n} \beta_i \frac{(1-\rho_1) r^{(i)}(k) + (a_i + \rho_1) y_m^{(i)}(k)}{b_i} \\ &= K_c r(k) + \sum_{i=1}^{n} \beta_i \frac{(\rho_1 + a_i)}{b_i} \frac{b_i z^{-1}}{1 + a_i z^{-1}} u(k); \end{aligned} \tag{14}$$

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

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

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

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 $\rho_i, i = 2, ..., n$ using a partial fraction by the following definitions [11].

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

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

Proof: The overall implied control law is given as:

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

Substituting in from (15) and (17):

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

$$\Downarrow \tag{18}$$

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

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

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

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 $-a_1$, $-a_2$,..., $-a_n$ with a_1, a_2 a complex conjugate pair. A partial fraction expansion of $G_m(z)$ into first-order terms is:

$$G_m(z) = \frac{n(z)}{(1+a_1z^{-1})(1+a_2z^{-1})...(1+a_nz^{-1})} = \sum_{i=1}^{n} \frac{b_iz^{-1}}{1+a_iz^{-1}}. \qquad (20)$$

It is noted that the residues b_1, b_2, ... ,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* $G_i(z) = \frac{b_iz^{-1}}{1+a_iz^{-1}}$ *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* $\beta_1u^{(1)}(k)+\beta_2u^{(2)}(k)$ *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:

$$u(k) = \beta_1 u^{(1)}(k) + \beta_1^* u^{(1)*}(k)$$
$$= (1-\rho_1)\left[\frac{\beta_1 r^{(1)}(k)}{b_1} + \frac{\beta_1^* r^{(1)*}(k)}{b_1^*}\right] + \beta_1 \frac{(a_1+\rho_1) y_m^{(1)}(k)}{b_1} + \beta_1^* \frac{(a_1^*+\rho_1) y_m^{(1)*}(k)}{b_1^*}. \tag{21}$$

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:

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

Substituting from (19):

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

$$\Downarrow \tag{23}$$

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

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 $x = \mathrm{Re}\{x\} + j\,\mathrm{Im}\{x\}$ and $y = \mathrm{Re}\{y\} + j\,\mathrm{Im}\{y\}$, then:

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

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.*

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

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

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

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 sub-models output required 10 mathematical operations on real numbers (summation and multiplication) with 4 reserved places for variables in addition to the $u(k)$ variable, and the calculation of the real parts of $\beta_i u^{(i)}(k) + \beta_{i+1} u^{(i+1)}(k)$ 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.

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

Theorem 4. *The real value of the proposed weighted input signal* $\mathrm{Re}\{\beta_i u^{(i)}\}$ *for the sub-model having real pole a_i comprises numerous components which can be computed off-line and stored.*

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

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

$$\begin{aligned} &K_{0,i} = \mathrm{Re}\left\{(1-\rho_1)\frac{\beta_i\gamma_i}{b_i\sum_{j=1}^{n}\gamma_j}\right\}; \quad K_{1,i} = (a_i+\rho_1)\frac{\beta_i}{b_i}; \\ &\mathrm{Re}\{\beta_i u^{(i)}\} = K_{0,i}r(k) + K_{1,i}y_m^{(i)}(k). \end{aligned} \quad \square \tag{29}$$

Remark 5. *The coefficent $K_{0,i}$ 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:

$$\begin{aligned} -a_1 y_m^{(1)}(k) - a_2 y_m^{(2)}(k) &= B_2 u(k-1) - (a_1 + a_2) y_m^{(1,2)}(k) - (a_1 a_2) y_m^{(1,2)}(k-1) \\ y_m^{(1)}(k) + y_m^{(2)}(k) &= y_m^{(1,2)}(k). \end{aligned} \tag{32}$$

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).

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

□

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_i 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.

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

□

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

$$\begin{aligned}\mathrm{Re}\{\beta_i u^{(i)}(k)+\beta_{i+1}u^{(i+1)}(k)\}&=\mathrm{Re}\left\{\frac{1-\rho_1}{\sum_{j=1}^{n}\gamma_j}\left[\frac{\beta_i\gamma_i}{b_i}+\frac{\beta_{i+1}\gamma_{i+1}}{b_{i+1}}\right]\right\}r(k)\\&+2\,\mathrm{Re}\left\{(a_i+\rho_1)\frac{\beta_i}{b_i}\right\}\mathrm{Re}\{y_m^{(i)}(k)\}-2\,\mathrm{Im}\left\{(a_i+\rho_1)\frac{\beta_i}{b_i}\right\}\mathrm{Im}\{y_m^{(i)}(k)\}.\end{aligned} \tag{35}$$

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

$$\begin{aligned}&K_{0,i}=\mathrm{Re}\left\{\frac{1-\rho_1}{\sum_{j=1}^{n}\gamma_j}\left[\frac{\beta_i\gamma_i}{b_i}+\frac{\beta_{i+1}\gamma_{i+1}}{b_{i+1}}\right]\right\};\\&K_{1,i}=2\,\mathrm{Re}\left\{(a_i+\rho_1)\frac{\beta_i}{b_i}\right\};\quad K_{2,i}=-2\,\mathrm{Im}\left\{(a_i+\rho_1)\frac{\beta_i}{b_i}\right\};\\&\mathrm{Re}\{\beta_i u^{(i)}(k)+\beta_{i+1}u^{(i+1)}(k)\}=K_{0,i}r(k)+K_{1,i}\,\mathrm{Re}\{y_m^{(i)}(k)\}+K_{2,i}\,\mathrm{Im}\{y_m^{(i)}(k)\}.\end{aligned} \tag{36}$$

□

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* $\mathrm{Re}\{\beta_i u^{(i)}(k)+\beta_{i+1}u^{(i+1)}(k)\}$ *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* $G_{i,i+1}$*.*

Proof: This follows from substitution of (33) into (36).

$$
\begin{aligned}
&y_m^{(i,i+1)}(k+1) = B_{1,i}u(k) + B_{2,i}u(k-1) - 2\,\mathrm{Re}\{a_i\}y_m^{(i,i+1)}(k) - a_i a_i^* y_m^{(i,i+1)}(k-1);\\
&\mathrm{Re}\{\beta_i u^{(i)}(k) + \beta_{i+1}u^{(i+1)}(k)\} = K_0^{(i,i+1)}r(k) + K_1^{(i,i+1)}y_m^{(i,i+1)}(k)\\
&\qquad\qquad + K_2^{(i,i+1)}y_m^{(i,i+1)}(k-1) + K_3^{(i,i+1)}u(k-1);\\
&K_0^{(i,i+1)} = K_{0,i}; \qquad K_1^{(i,i+1)} = \frac{K_{1,i}\,\mathrm{Im}\{a_i\} - K_{2,i}\,\mathrm{Re}\{a_i\}}{2\,\mathrm{Im}\{a_i\}};\\
&K_2^{(i,i+1)} = -\frac{K_{2,i}a_i a_i^*}{2\,\mathrm{Im}\{a_i\}}; \qquad K_3^{(i,i+1)} = \frac{K_{2,i}B_{2,i}}{2\,\mathrm{Im}\{a_i\}}.
\end{aligned}
\tag{37}
$$

□

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 sub-models with real poles can use the simpler formulation of (29). From (37), the new formulated PP-PFC calculation of the actual second-order sub-models 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 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 figure 2, are applied. The algorithm can also compensate for disturbances acting at the process input, but this case is not shown here.

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 fomulated 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 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 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.

Figure 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 it 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 posses 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 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]):

$$\begin{aligned}
\ddot{\theta}(t) &= \frac{K_s}{J_{eq}}\ \alpha(t) - \frac{B_{eq}}{J_{eq}}\ \ddot{\theta}(t) + \frac{1}{J_{eq}}\ \tau(t) \\
\ddot{\alpha}(t) &= -K_s\left(\frac{J_l + J_{eq}}{J_l J_{eq}}\right)\ \alpha(t) + \frac{B_{eq}}{J_{eq}}\ \dot{\theta}(t) - \frac{1}{J_{eq}}\ \tau(t) \\
\tau(t) &= \frac{\eta_g K_g \eta_m k_t \big(V_m(t) - K_g k_m \dot{\theta}(t)\big)}{R_m}
\end{aligned} \tag{40}$$

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:

$$\begin{aligned}
\ddot{\theta}(t) &= 619.05\ \alpha(t) - 34.70\ \dot{\theta}(t) + 61.07\ V_m(t) \\
\ddot{\alpha}(t) &= -1015.62\ \alpha(t) + 32.78\ \dot{\theta}(t) - 61.07\ V_m(t)
\end{aligned} \tag{41}$$

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:

$$\frac{\dot{\theta}(z)}{V_m(z)} = \frac{0.8451z^{-1} - 1.556z^{-2} + 0.8457z^{-3}}{1 - 2.17z^{-1} + 1.753z^{-2} - 0.4997z^{-3}} \tag{42}$$

Figure 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 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. Figure 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.

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] J. Richalet, D. O'Donovan, Predictive Functional Control: principles and industrial applications, Springer Science & Business Media, 2009.

[2] R. Haber, R. Bars, U. Schmitz, Predictive control in process engineering: from the basics to the applications, Wiley-VCH, 2011, Ch. 11: Predictive Functional Control, pp. 437–465.

[3] J. A. Rossiter, R. Haber, The effect of coincidence horizon on predictive functional control, Processes 3 (1) (2015) 25–45.

[4] J. Rossiter, Input shaping for PFC: how and why?, Journal of control and decision 3 (2) (2016) 105–118. doi:10.1080/23307706.2015.1083408.

[5] P. Gawthrop, E. Ronco, Predictive pole-placement control with linear models, Automatica 38 (2002) 421–432.

[6] W.-H. Chen, P. J. Gawthrop, Constrained predictive pole-placement control with linear models, Automatica 42 (2006) 613–618.

[7] D. W. Clarke, C. Mohtadi, P. S. Tuffs, Generalized predictive control. Part I. The basic algorithm, Automatica 23 (2) (1987) 137–148.

[8] W. Wang, Z. Z. Mao, Generalized pole-placement adaptive control algorithm and its convergent analysis, Selected Topics in Modelling and Control, 2 (1995) 874–878.

[9] M. Fikar, H. Unbehauen, J. Mikles, Design of a predictive controller based on pole-placement, 3rd European Control Conference 4 (2004) 131–135.

[10] M. Khadir, J. Ringwood, Stability issues for first order predictive functional controllers: extension to handle higher order internal models, International Conference on Computer Systems and Information Technologye (2005) 174–179.

[11] J. Rossiter, R. Haber, K. Zabet, Pole-placement Predictive Functional Control for over-damped systems with real poles, ISA transactions 61 (2016) 229–239.

[12] M. Khadir, J. Ringwood, Extension of first order predictive functional controllers to handle higher order internal models, International Journal of Applied Mathematics and Computer Science 18 (2) (2008) 229–239.

[13] J. Rossiter, Notes on multi-step ahead prediction based on the principle of concatenation, 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] J. Apkarian, P. Karam, M. Levis, Instructor workbook flexible joint experiment for LabView users, Quanser Inc, 2012.

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

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%

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%
$\rho_1 = 0.8$ $\rho_{2,3} = 0.8 \pm j0.2$	100%	41%	37%

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	2.6 Ω
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

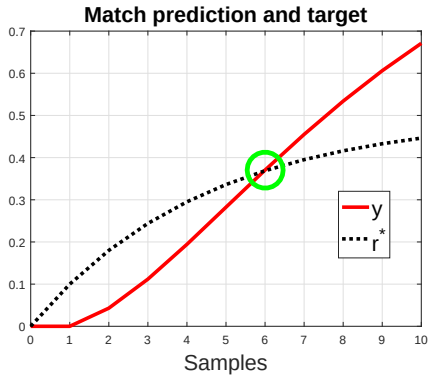

Figure 1: Illustration of PFC target dynamic r^* and coincidence of the output prediction y_p with target dynamic $n_y = 6$ samples ahead.

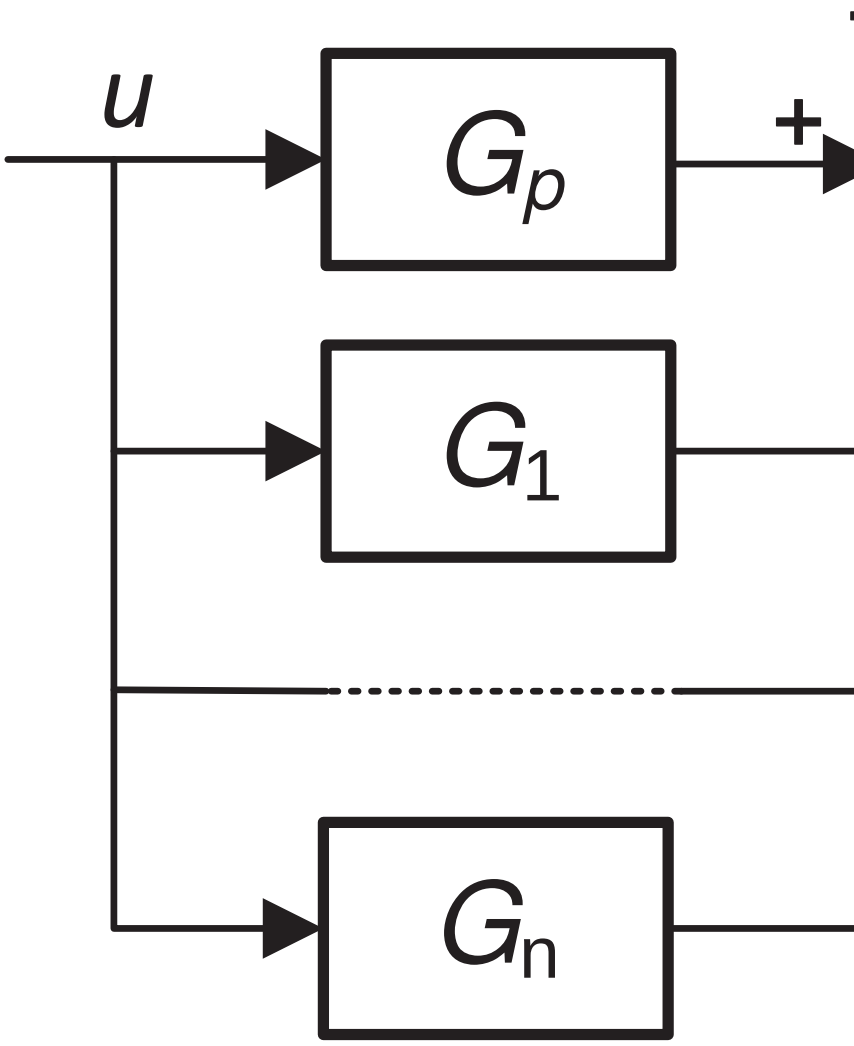

Figure 2: Parallel model format alongside the actual process G_p.

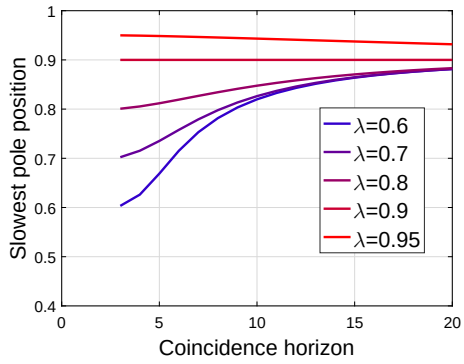

Figure 3: Illustration of PFC closed-loop poles with different choices of λ and n_y on the over-damped example P_1. Note $n_y < 3$ gives closed-loop instability.

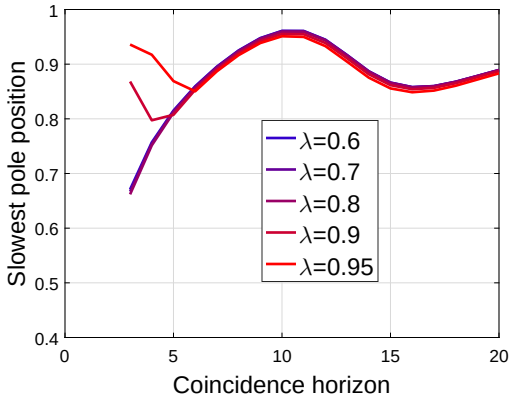

Figure 4: Illustration of PFC closed-loop poles (absolute values as complex) with different choices of λ and n_y on the under-damped example P_2.

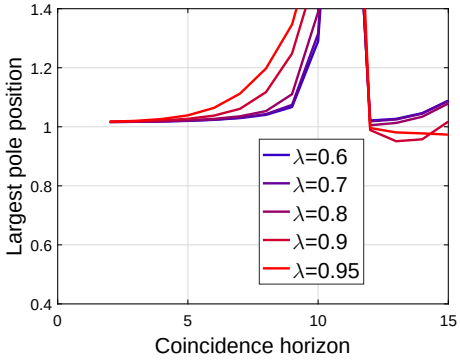

Figure 5: Illustration of PFC closed-loop poles (closed-loop instability) with different choices of λ and $n_y = 3$ on the under-damped example N.

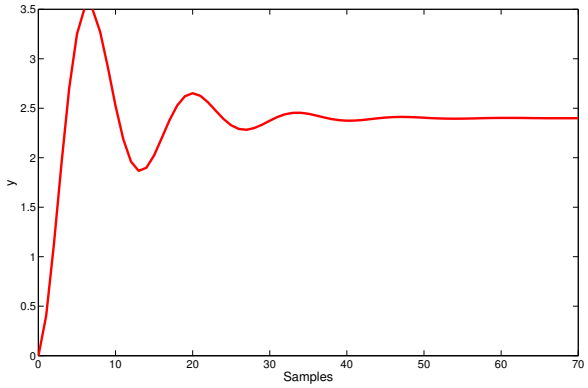

Figure 6: Step response of system M.

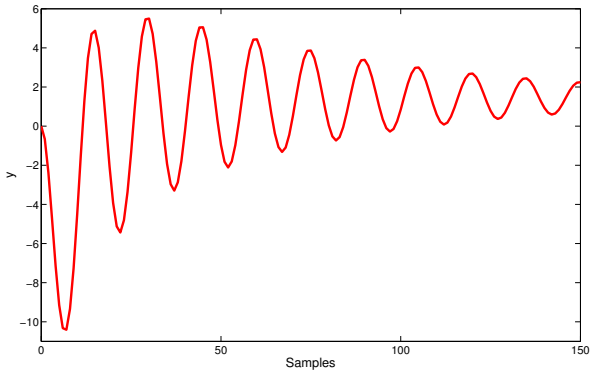

Figure 7: Step response of system N.

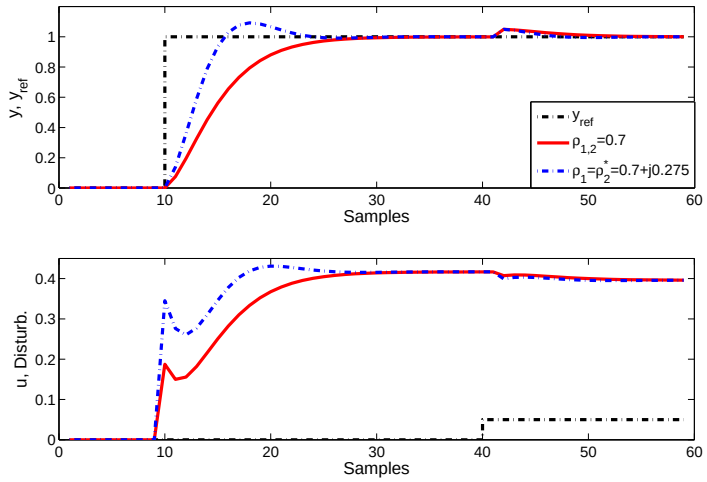

Figure 8: Illustration of PP-PFC performance with different choices of ρ on the under-damped example M.

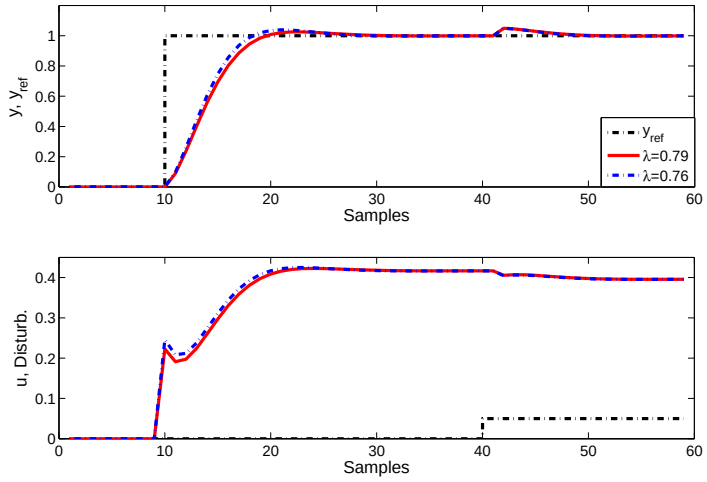

Figure 9: Illustration of conventional PFC performance with different choices of λ on the under-damped example M.

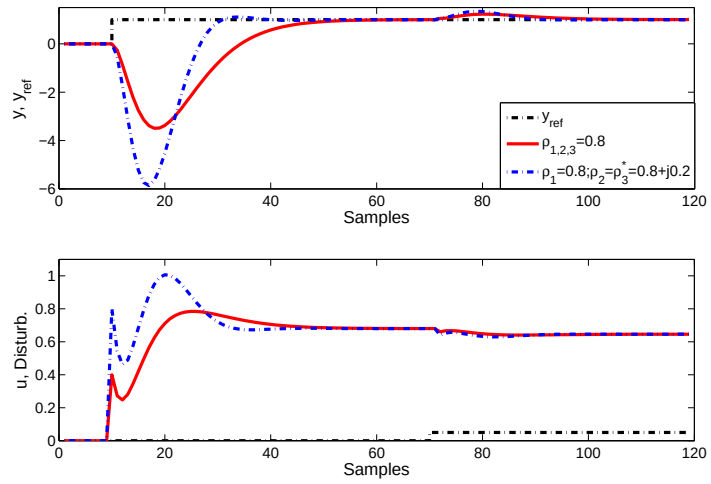

Figure 10: Illustration of new formulated PP-PFC performance with different choices of ρ on the under-damped example N.

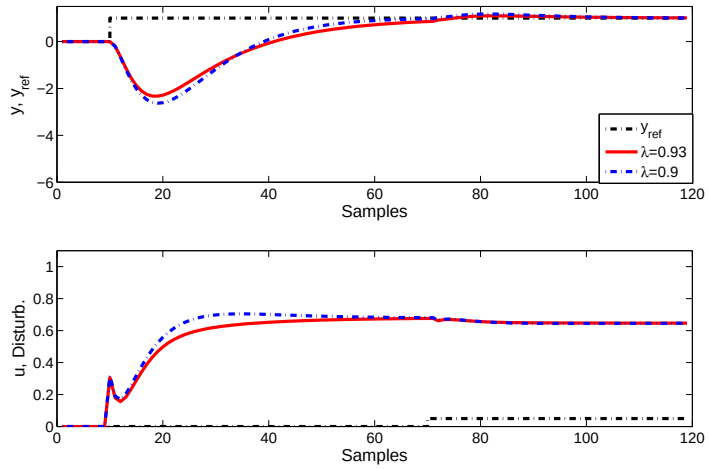

Figure 11: Illustration of conventional PFC performance with different choices of λ on the under-damped example N.

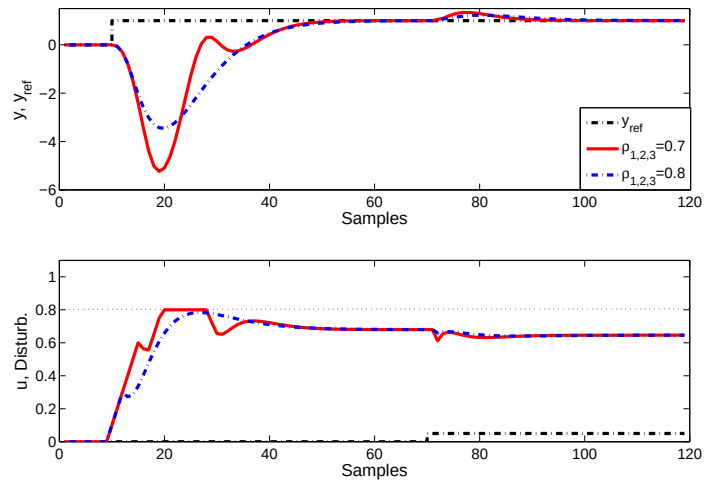

Figure 12: Illustration of PP-PFC performance of the under-damped example N considering input constraints.

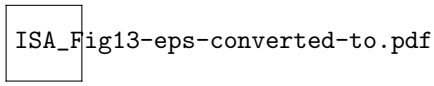

Figure 13: The experimental plant.

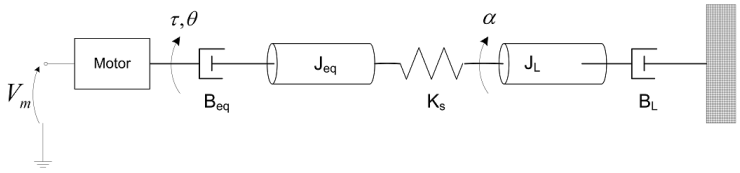

Figure 14: Rotary flexible joint model [15].

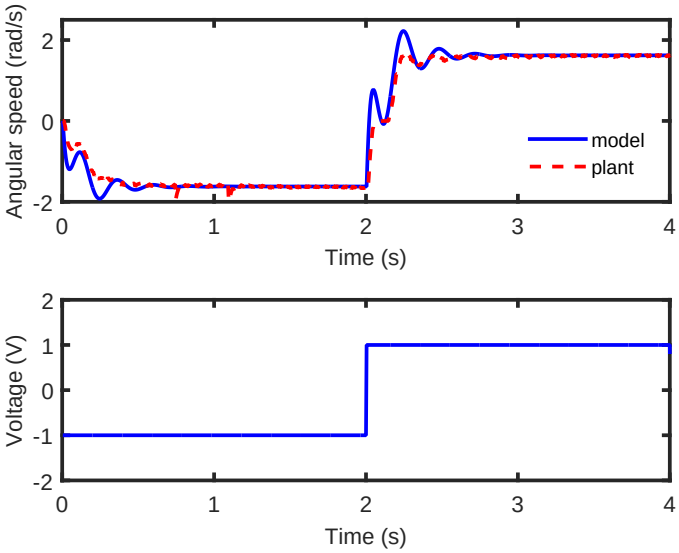

Figure 15: The open-loop behavior of plant and model based on the supplied voltage.

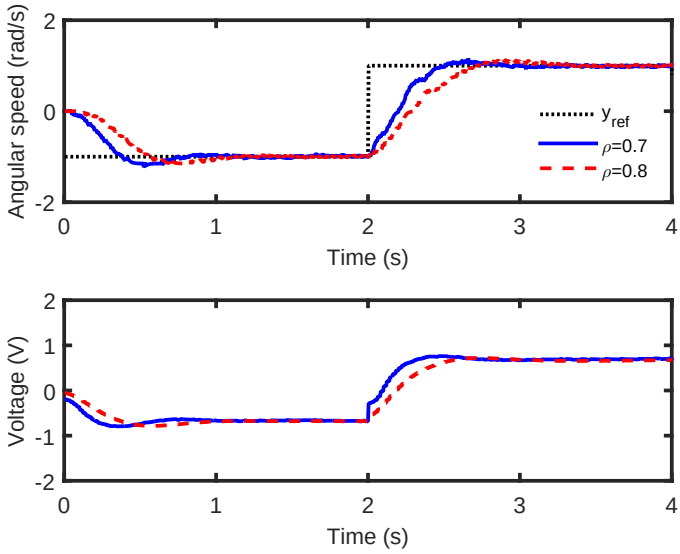

Figure 16: Illustration of PP-PFC performance of the under-damped Quanser servo attached with flexible joint.