Proceedings Paper:

Aftab, M.S. and Rossiter, J.A. (2022) Predictive functional control with explicit pre-conditioning for oscillatory dynamic systems. In: Proceedings of 2021 European Control Conference (ECC 2021). 2021 European Control Conference (ECC 2021), 29 Jun - 02 Jul 2021, Delft, Netherlands (Virtual Conference). IEEE, pp. 204-209. ISBN: 9781665479455.

https://doi.org/10.23919/ECC54610.2021.9655126

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

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

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

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

I. Introduction

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

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

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

One proposal [7] recommends altering the input by separating and subsequently cancelling the un-wanted dynamics to obtain convergent predictions. This method provides many-fold performance improvements while retaining the basic PFC characteristics but lacks practicality as the proposed minimum-moves shaping produces aggressive input activity. Another input shaping proposal [6] ensures relatively less aggressive control moves by allowing predictions to converge over many more samples. This method, tested on numerous simulation models and hardware application, outperforms the predecessor but relies on some rather less intuitive offline computations. Yet another proposal utilises the partial fraction decomposition of higher-order models [8] into several first-order systems to avail a simple tuning procedure. For oscillatory dynamics [9], however, such a decomposition explicitly embeds complex number algebra into computations limiting its practicality. Suggested modifications (within the same paper) guarantee real number computation but at the price of increased coding requirements.

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

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

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

II. Problem Statement

Consider an n^{th} order stable SISO system

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

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

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

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

III. Model Pre-conditioning

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

A. Simple Pole-Placement Compensator

Assume that feedback compensation of $G(z)$ with $C(z)$, as shown in Fig. 1, results in the transfer function $T(z)$ which provides smooth and monotonically convergent prediction behaviour. Then one may write

$$T(z) = \frac{C(z)G(z)}{1 + C(z)G(z)} = \frac{\beta(z)}{\alpha(z)} \tag{3}$$

After simple manipulations, this leads to

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

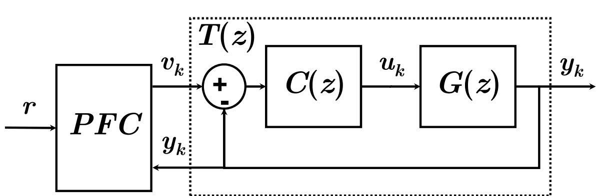

Fig. 1. PPFC structure comprising inner feedback pre-compensation and outer PFC loop.

The open-loop zeros $b(z)$ become compensator poles that could cause stability issues especially with unstable zeros. To avoid this, we set $\beta(z) = Kb(z)$, $K \neq 0$ and get

$$C(z) = K\frac{a(z)}{\alpha(z) - Kb(z)} \tag{4}$$

From (4), it is obvious that the compensator actually *cancels* the open-loop poles $a(z)$ and places new poles given by $\alpha(z)$. Would such a design based on pole-cancellation be acceptable? Let us try to understand the rationale behind pole-placement in the context of PFC.

B. Rationale behind Pole-Cancellation

At this point, readers are reminded of the main design objective, that is to obtain smooth and well-settled model predictions because conventional PFC lacks flexibility to handle oscillatory dynamics efficiently. Traditional PID and lag-lead compensation have been proven ineffective in completely eliminating oscillations especially with higher-order dynamics [15]. The obvious alternative in this case is pole-placement (4).

While it is best to avoid cancelling open-loop unstable poles, researchers report that the decision to either shift or cancel a real stable pole is merely based on design trade-off between either having the shifted pole appear as zero of the sensitivity function to output disturbance, or having the cancelled mode appear as pole of the sensitivity function to input disturbance [16]. A similar argument holds for complex conjugate pole pair, albeit cancellation in this case implies oscillatory input disturbance rejection.

Another concern is related to inexact pole-zero cancellation which is almost always inevitable due to modelling errors. For unstable poles this may lead to output divergence, but for stable open-loop poles the impact depends upon magnitude of the residues from partial fraction expansion and consequently on the design specification for satisfactory performance. Nevertheless, one should not forget that the pole-placement compensator (4) is assisted by an outer PFC loop, which is tuned for performance, robustness and disturbance rejection.

Although a clear understanding would require formal sensitivity analysis, here we demonstrate the performance of a pole-placement technique in combination with PFC, shown in Fig. 2, against a pole-shifting compensator in the presence of output disturbance (constant 0.25 amplitude starting at 25^{th} sample) and input disturbance (constant -0.25 amplitude starting at 60^{th} sample) for G_1 (see Section V-A) with deliberately introduced modelling errors. Evidently pole-placement provides better output disturbance rejection whereas input disturbance rejection is equivalent for both controllers. Moreover, while it may not always be possible to design a pole-shifting controller, the proposed pole-placement compensator always exists for any order dynamic model (1).

From now onwards, we shall focus on the attributes of pole-placement compensator and the design simplicity it brings within the PFC framework.

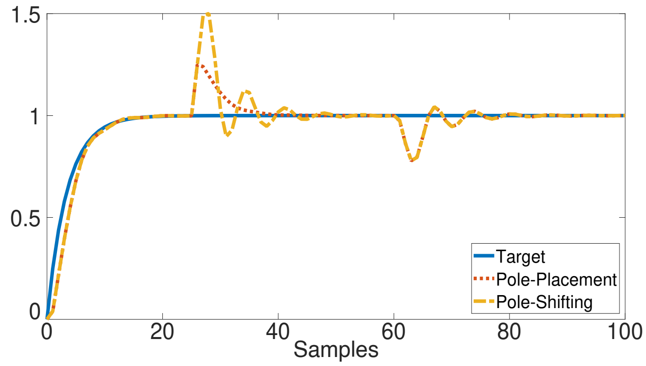

Fig. 2. PFC comparison with pole-cancellation and pole-shifting compensation for G_1 in the presence of input and output disturbances with modelling mismatches

C. Selecting Compensated Dynamics

In order to design a pole-placement compensator, appropriate selection of $\alpha(z)$ is extremely important. Ideally one would want the compensated model $T(z)$ to exhibit non-oscillatory behaviour. A good starting point then is to place the "new" poles of $T(z)$ at the projection of dominant oscillatory poles of $G(z)$ along the real axis. This would effectively filter out the unwanted oscillations without compromising convergence speed [6]. Mathematically

$$T(z) = K\frac{b(z)}{\alpha(z)}; \quad \alpha(z) = a^-(z)\alpha^+(z) \tag{5}$$

with $\alpha^+(z) = (1 - p_r z^{-1})^2$.

It is necessary for internal stability that all poles of $C(z)$ remain inside the unit circle. This can be guaranteed by keeping K within the stable range. However it is quite tedious to obtain an analytical expression for K for higher than second-order systems and therefore we recommend using a graphical tool such as root locus (see, for example [17]) on the denominator of (4) to find the stability margin of $C(z)$.

D. Second-Order Models

Before moving on to the next design stage, it is pertinent to discuss pre-conditioning of under-damped second-order systems exclusively. Assume $a(z) = 1 + a_1 z^{-1} + a_2 z^{-2}$ and $b(z) = b_1 z^{-1} + b_2 z^{-2}$ and also that the oscillatory modes are $z_a = p_r \pm jp_i$. Then according to the preceding discussion $\alpha(z) = 1 - 2p_r z^{-1} + p_r^2 z^{-2}$ is the pole polynomial of the compensated model. Theorem 1 below provides analytical expressions for the stable K range.

Theorem 1: A pre-conditioning compensator designed for second-order model using (4)-(5) is guaranteed stable if:

$$K < \begin{cases} min\left[\dfrac{(1-p_r)^2}{b_1(1-z_z)}, -\dfrac{(1+p_r)^2}{b_1(1+z_z)}\right]; & -\infty < z_z \leq -1 \\ \dfrac{(1-p_r)^2}{b_1(1-z_z)}; & -1 < z_z < p_r \\ \dfrac{1-p_r^2}{|b_1 z_z|}; & z_z > p_r \end{cases}$$

where $z_z = -b_2/b_1$ is the system zero.

Proof: The controller poles are:

$$z_C = \frac{(2p_r + Kb_1) \pm \sqrt{(2p_r + Kb_1)^2 - 4(p_r^2 - Kb_2)}}{2}$$

For convenience, we substitute $x_1 = 0.5(2p_r + Kb_1)$ and $x_2 = 0.5\sqrt{(2p_r + Kb_1)^2 - 4(p_r^2 - Kb_2)}$. Then $z_{C_1} = x_1 - x_2$ and $z_{C_2} = x_1 + x_2$.
(i) if $z_z \leq -1$ then by increasing K, z_{C_1} moves towards z_z whereas z_{C_2} goes towards $+\infty$. For stability we must ensure $z_{C_1} > -1$ and $z_{C_2} < 1$. These two conditions transform into:

$$z_{C_1} > -1 \implies K < -\frac{(1+p_r)^2}{b_1(1+z_z)}$$

$$z_{C_2} < 1 \implies K < \frac{(1-p_r)^2}{b_1(1-z_z)}$$

Depending on the actual position of p_r, one of the controller poles is relatively nearer to the stability boundary. Therefore:

$$K < min\left[\frac{(1-p_r)^2}{b_1(1-z_z)}, -\frac{(1+p_r)^2}{b_1(1+z_z)}\right]; -\infty < z_z \leq -1 \tag{6}$$

(ii) for $-1 < z_z < p_r$, z_{C_1} can never leave the unit circle. Therefore it is sufficient to check only:

$$K < \frac{(1-p_r)^2}{b_1(1-z_z)}; \quad -1 < z_z < p_r \tag{7}$$

(iii) $z_z > p_r$ results in complex conjugate controller poles. In this case, guaranteed stability $|z_C| < 1$ implies:

$$K < \frac{1-p_r^2}{|b_1 z_z|}; \quad z_z > p_r \tag{8}$$

which completes the proof. ■

Remark 1: For some systems one or more zeros might be located at p_r. To prevent inadvertent pole-zero cancellation in such cases, we suggest replacing p_r in the preceding analysis with $p_r + \epsilon$, where $\epsilon \to 0$ and $|p_r + \epsilon| < 1$.

IV. Nominal PFC Design

The design stage implements the pre-conditioned model predictions within a PFC framework, as shown in Fig. 1.

A. The PPFC Control Law

Similar to conventional PFC, the Pre-conditioned PFC (PPFC) drives the output prediction $y_{k+i|k}$ exponentially closer to the set point r with each time step. This convergence mainly depends upon the target pole ρ defined by $\rho = exp(-3T/CLTR)$ where T and $CLTR$ are sampling time and desired closed-loop settling time respectively. Mathematically, the PFC law is derived from the target:

$$y_{k+i|k} = r - (r - y_k)\rho^i; \quad i > 0 \tag{9}$$

On the other hand, eqn. (5) i.e. $\alpha(z)y(z) = Kb(z)v(z)$ provides i-step ahead prediction information as follows:

$$y_{k+i|k} = KH_i \underrightarrow{v}_k + KP_i \underleftarrow{v}_{k-1} + Q_i \underleftarrow{y}_k; \quad i > 0 \tag{10}$$

where H_i, P_i and Q_i depend upon model parameters. For an N^{th} order model:

$$\underset{\rightarrow}{v}_k = \begin{bmatrix} v_k \\ v_{k+1} \\ \vdots \\ v_{k+i-1} \end{bmatrix}; \underset{\leftarrow}{v}_{k-1} = \begin{bmatrix} v_{k-1} \\ v_{k-2} \\ \vdots \\ v_{k-N+1} \end{bmatrix}; \underset{\leftarrow}{y}_k = \begin{bmatrix} y_k \\ y_{k-1} \\ \vdots \\ y_{k-N+1} \end{bmatrix}$$

By keeping a constant future input i.e. $v_{k+i} = v_k, \forall i > 0$, the i-step ahead model prediction (10) is matched to the target (9) at one future point called the coincidence point n_y i.e. at $i = n_y$. This results in the PPFC law

$$v_k = \frac{r - (r - y_k)\rho^{n_y} - (KP_{n_y}\underset{\leftarrow}{v}_{k-1} + Q_{n_y}\underset{\leftarrow}{y}_k)}{Kh} \tag{11}$$

where $h = \sum_{j=1}^{n_y} H_{n_y}^j$ and $H_{n_y}^j$ is the j^{th} element of H_{n_y}. Naturally the main interest is in finding actual input u_k that drives the plant. It is evident from Fig. 1

$$u(z) = C(z)[v(z) - y(z)] \tag{12}$$

Lemma 1: Formulation (12) is equivalent to:

$$u(z) = K\frac{a^+(z)}{\alpha^+(z)}v(z)$$

Proof: Substituting $C(z) = Ka(z)/[\alpha(z) - Kb(z)]$ and $y(z) = [b(z)/a(z)]u(z)$ in (12), we get:

$$u(z) = K\frac{a(z)}{\alpha(z)}v(z)$$

further $\alpha(z) = a^-(z)\alpha^+(z)$ and $a(z) = a^-(z)a^+(z)$ imply:

$$u(z) = K\frac{a^+(z)}{\alpha^+(z)}v(z) \tag{13}$$

which is a simple equivalent of (12). ■

Extracting either u_k or v_k from the other simply requires vector/matrix multiplication as shown below. Hence the associated coding requirement is elementary.

$$\begin{aligned} u_k &= K\hat{a}^+\underset{\leftarrow}{v}_k - \hat{\alpha}^+\underset{\leftarrow}{u}_{k-1} \\ v_k &= \frac{\hat{\alpha}^+}{K}\underset{\leftarrow}{u}_k - \hat{a}^+\underset{\leftarrow}{v}_{k-1} \end{aligned} \tag{14}$$

where vectors $\hat{a}^+$ and $\hat{\alpha}^+$ contain coefficients of polynomials $a^+(z)$ and $\alpha^+(z)$ respectively.

Remark 2: Subtleties related to prediction-bias removal and offset-free tracking have been omitted from (11) as these do not affect the main analysis and results. Numerical examples nonetheless include relevant algebra. Readers are referred to [3] for details.

B. Tuning Procedure

Though PFC tuning has traditionally been heuristic, researchers have managed to establish some generic guidelines for simpler systems [7], [18]. Pre-compensation effectively changes the oscillatory open-loop step response of $G(z)$ to smoother and more settled behaviour, and therefore standard PFC tuning procedures can be fully utilised. In this study, the PPFC control law (11) depends upon three significant parameters: the coincidence horizon n_y, the target pole ρ and the compensator gain K. While a judicious selection of n_y and ρ is of paramount importance, surprisingly K does not affect the closed-loop performance.

Theorem 2: The plant input u_k is independent of compensator gain K.

Proof: In z-domain, eqn. (11) can be written as:

$$v(z) = \frac{(1 - \rho^{n_y})r(z) + [\rho^{n_y} - Q(z)]y(z)}{K[h + P(z)]}$$

where $P(z) = \sum_{j=1}^{N-1} P_{n_y}^j z^{-j}$ and $Q(z) = \sum_{j=0}^{N-1} Q_{n_y}^j z^{-j}$. It follows from Lemma 1:

$$u(z) = \cancel{K}\frac{a^+(z)}{\alpha^+(z)} \cdot \frac{(1 - \rho^{n_y})r(z) + [\rho^{n_y} - Q(z)]y(z)}{\cancel{K}[h + P(z)]}$$

or equivalently in the time-domain:

$$u_k = \frac{\sum_{j=0}^{2} a_j^+(1 - \rho^{n_y})r - (\tilde{P}_{n_y}\underset{\leftarrow}{u}_{k-1} + \tilde{Q}_{n_y}\underset{\leftarrow}{y}_k)}{h} \tag{15}$$

for suitable $\tilde{P}_{n_y}$ and $\tilde{Q}_{n_y}$. Hence the plant input is independent of K. Nevertheless, appropriate selection of K is still necessary to maintain internal stability. ■

The recommended tuning procedure [5] suggests choosing n_y within the range $k_L \leq n_y \leq k_U$, where k_L and k_U represent the time samples when the normalised unit-step response of $T(z)$ reaches approximately 0.4 and 0.8 respectively with significant gradient. As for the target pole, one may compare several first-order responses with differing ρ against the normalised step-response to find an intercept within the above-mentioned n_y range. See, for instance, Figs. 3 and 5.

C. Constraint Management

Knowledge of u_k from (14) can facilitate constraint handling in fairly straightforward manner. Instead of adopting computation-intensive algorithms, such as back-calculation [4], one may opt for input and output predictions for constraint assessment. The only caveat, however, is the need to recalculate corrected v_k, using (14), if input violations are detected. Furthermore, output constraints must be validated before inputs, since y_k is based on the pre-conditioned model predictions and hence depend upon v_k. One may use prediction equation (10) by selecting such value of v_k closest to the one obtained with (11) that satisfies all constraints for sufficiently large validation horizon n_c i.e.

$$y_{min} \leq KH_jv_k + KP_j\underset{\leftarrow}{v}_{k-1} + Q_j\underset{\leftarrow}{y}_k \leq y_{max} \tag{16}$$

where $j = 1, 2, \ldots, n_c$ and $n_c \gg n_y$.

Remark 3: Constraint handling with the pre-conditioned model predictions (16) is guaranteed recursively feasible as long as n_c is sufficiently large [19]. This, however, may not be true with open-loop oscillatory predictions.

V. Numerical Examples

This section demonstrates the efficacy of PPFC algorithm with two numerical examples. To better understand its advantages, a comparison of closed-loop performance is drawn against the standard PFC for two challenging processes: G_1

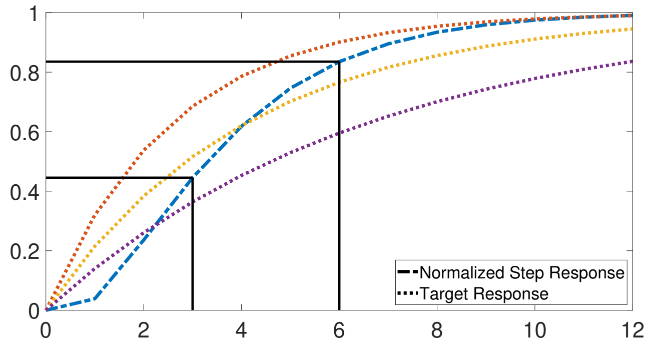

Fig. 3. Target responses with $\rho = [0.6(red), 0.75(yellow), 0.9(purple)]$ overlaying the normalised step response of T_1

is a second-order oscillatory system whereas G_2 is slightly non-minimum phase third-order under-damped system.

A. Example-1 (Second-Order Model)

Consider a second-order under-damped system (17) with $|\Delta u_k| \leq 0.45$, $-0.25 \leq u_k \leq 1.75$ and $0 \leq y_k \leq 1.05$. For fair comparison, both PPFC and PFC control laws will be tuned identically.

$$G_1 = \frac{0.1z^{-1} + 0.4z^{-2}}{1 - z^{-1} + 0.8z^{-2}} \tag{17}$$

The oscillatory modes of G_1 are $z_a = 0.5 \pm j0.742$, whereas $z_z = -4$ is the system zero. For stability the compensator gain should be $K < 0.5$ as obtained from (6). Consequently the pre-conditioned prediction model with $K = 0.25$ is:

$$T_1 = \frac{0.025z^{-1} + 0.1z^{-2}}{1 - z^{-1} + 0.25z^{-2}} \tag{18}$$

Next we find n_y and ρ. Fig. 3 shows the pre-conditioned step response overlaid with various first-order target responses and suggests $3 \leq n_y \leq 5$ as a suitable coincidence horizon window. Evidently target dynamics with $\rho = 0.6$ or $\rho = 0.9$ do not match predictions within the desirable n_y range and hence would need over-actuation or under-actuation to enforce an intercept. However, a sensible choice would be $\rho = 0.75$ which gets an exact match at $n_y = 4$.

Efficacy of the PPFC algorithm is obvious with the constrained nominal closed-loop performance shown in Fig. 4. Specifically we observe that the PPFC plant output (upper

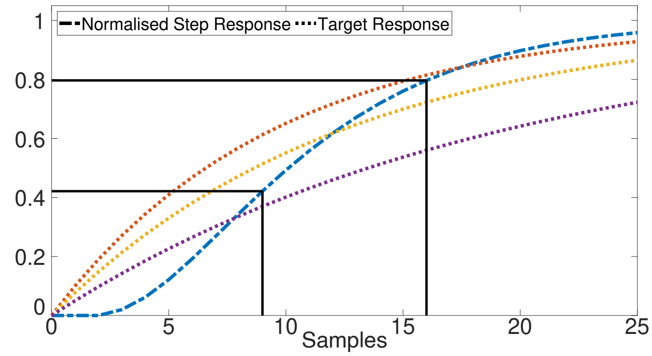

Fig. 5. Targets with $\rho = [0.9(red), 0.92(yellow), 0.95(purple)]$ overlaying the normalised step response of T_2

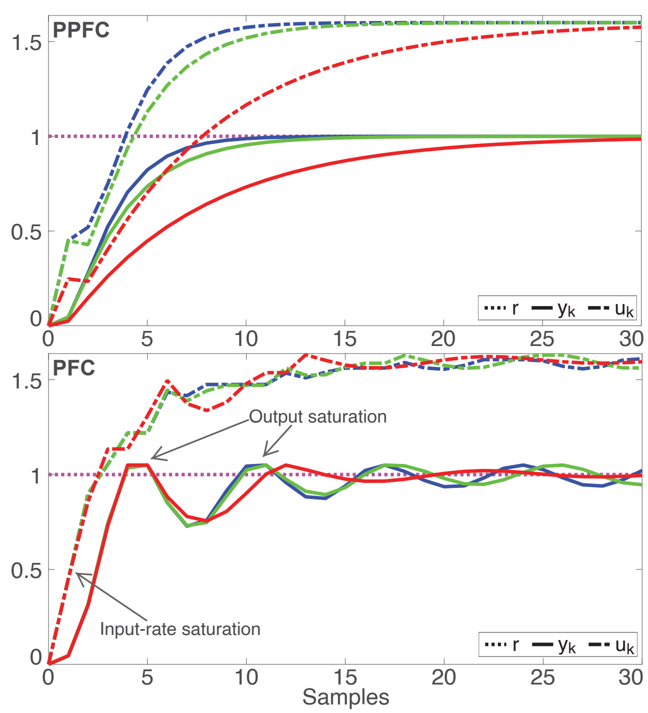

Fig. 4. Nominal constrained performance with G_1, $n_y = 4$ and $\rho = [0.6(blue), 0.75(green), 0.9(red)]$

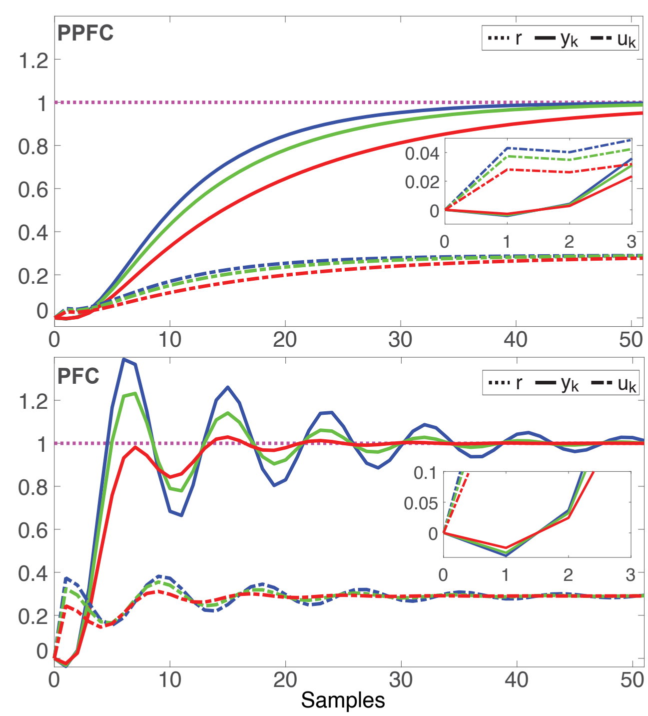

Fig. 6. Nominal unconstrained performance with G_2, $n_y = 12$ and $\rho = [0.9(blue), 0.92(green), 0.95(red)]$

figure) is smooth and oscillation-free, and strongly linked to the corresponding ρ. Additionally no constraint violation is visible, although control input for faster target dynamics is relatively aggressive which reinforces the expectation. The conventional PFC (lower figure) appears incapable within transient period. There seems no obvious link between the output and target dynamics (ρ), and inputs are generally too aggressive. As a result, input-rate and output saturation are seen within first 10 samples.

B. Example-2 (Higher-Order Model)

Consider the model G_2 with *no constraints*

$$G_2 = \frac{-0.1z^{-1} + 0.4z^{-2} + 0.2z^{-3}}{1 - 2.1z^{-2} + 1.69z^{-2} - 0.445z^{-3}} \quad (19)$$

with $a^+(z) = 1 - 1.6z^{-1} + 0.89z^{-2}$ and $a^-(z) = 1 - 0.5z^{-1}$. The oscillatory modes are located at $z_{a^+} = 0.8 \pm j0.5$. Therefore, we define $\alpha^+(z) = 1 - 1.6z^{-1} + 0.64z^{-2}$, $\alpha^-(z) = a^-(z)$ and get the compensated model:

$$T_2 = K\frac{-0.1z^{-1} + 0.4z^{-2} + 0.2z^{-3}}{1 - 2.1z^{-2} + 1.44z^{-2} - 0.32z^{-3}} \quad (20)$$

The root locus plot (not shown) suggests $K < 0.04$ for compensator stability. We choose $K = 0.02$ as its numerical value does not affect the closed-loop performance. Fig. 5 shows the normalised step response of T_2, which clearly recommends a coincidence point within $9 \leq n_y \leq 16$. An intercept between step response and target exists with $\rho = 0.92$ at $n_y = 12$. To enforce an intercept with faster targets, one would need to over-actuate the input which in practice may result in constraint violations.

Nevertheless we examine the closed-loop behaviour with $\rho = [0.9, 0.92, 0.95]$ and $n_y = 12$ in Fig. 6, which presents a rather contrasting display of performance between the PPFC and simple PFC algorithms. Key observations are: Simple PFC with constant future input fails to damp the oscillatory modes of G_2. With the proposed scheme, while v_k remains constant within the coincidence horizon, the plant input u_k is aptly parametrised to overcome the under-damping. This difference is obvious from the varied control dynamics for both controllers. With PPFC, the target pole ρ seems more effective. This is clearly evident in the amount of time the outputs take to settle for both the algorithms. To sum up, while the conventional PFC algorithm may not be suitable for oscillatory dynamics, it is possible to improve its capabilities via pre-conditioning in straightforward fashion.

VI. Conclusion

This paper has proposed a pre-conditioned PFC design methodology for under-damped dynamic systems. The proposed pre-conditioning stage, based on pole-placement, transforms the unsettled open-loop model dynamics into smoother prediction behaviour, known to work well with the standard PFC algorithm. The overall design process is fairly straightforward; one that does not complicate standard constraint handling and controller tuning, and also retains the key attributes of original PFC i.e. simplicity and intuitiveness. We have demonstrated that the PPFC algorithm in essence parametrises control action to efficiently handle oscillations, a quality not present in the conventional PFC. Moreover the main tuning parameter, the target pole, shows improved efficacy in relation to the closed-loop performance as seen in the numerical simulations.

Nevertheless, as discussed in Section III-B, the improved set-point tracking with pole-placement pre-conditioning comes at the price of rather oscillatory input disturbance rejection, but a relatively better response to output disturbance compared to an equivalent pole-shifting compensator balances out this shortcoming. A formal analysis of sensitivity functions is required to fully understand the pros and cons of pole-placement and this constitutes our future research work.

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[16] R. H. Middleton and S. F. Graebe, “Slow stable open-loop poles: to cancel or not to cancel,” *Automatica*, vol. 35, no. 5, pp. 877–886, 1999.

[17] B. C. Kuo, *Digital Control Systems*. Oxford University Press Inc, 1995.

[18] J. A. Rossiter, “A priori stability results for pfc,” *International journal of control*, vol. 90, no. 2, pp. 289–297, 2017.

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