Proceedings Paper:

Rossiter, J.A. orcid.org/0000-0002-1336-0633, Aftab, M.S. and Panoutsos, G. (2022) Exploiting Laguerre polynomials and steady-state estimates to facilitate tuning of PFC. In: 20th European Control Conference (ECC) Proceedings. ECC22 - 20th European Control Conference, 12-15 Jul 2022, London, UK. European Control Association (EUCA) , pp. 1641-1646. ISBN 978-3-9071-4407-7

https://doi.org/10.23919/ECC55457.2022.9838428

Exploiting Laguerre polynomials and steady-state estimates to facilitate tuning of PFC

John Anthony Rossiter[1], Muhammad Saleheen Aftab[1]

***Abstract*— Predictive Functional Control (PFC), a simplified and low-cost MPC algorithm, has gained considerable attention for industrial process control in the last two decades. Although with PFC, controller tuning is relatively simple and more meaningful than a PID controller, its efficacy turns poorer for larger prediction horizons–a necessity for over-damped ad non-minimum phase dynamics. This paper proposes a conceptually novel tuning mechanism based on a single choice which is: how much faster or slower than open-loop would you like the closed-loop to converge? Simulations demonstrate that this is a cheap and simple way of effective tuning, by suitably over or under actuating the open-loop control action.**

I. Introduction

The popularity of model predictive control (MPC) is taken for granted these days but most of the focus in the literature is on the more expensive products which require reliable quadratic programming (QP) optimisers for high dimensional optimisations, or indeed even more challenging non-linear optimisations [1], [2]. There is relative little attention given to the other end of the market, that is relatively low cost more akin to PID. There are still many applications where a cheap single-input-single-output (SISO) control law is required, but PID is not as effective as one would like.

A secondary issue which also has gained relatively little interest in the literature is the one of MPC tuning. While it is accepted that the input and output horizons do affect the ultimate tuning, these are not usually considered tuning knobs in themselves as the default position is to take the horizons to be as large as the computing available allows [3], [4]. Consequently the main tuning parameters are the weights in the performance index, but the relationship between the weights and properties such as bandwidth and settling time is not analytical, which means tuning could be considered as much an art form as systematic, or perhaps something amenable to an offline tuning optimisation such as with genetic algorithms [4].

There is one notable exception to the above observations and that is predictive functional control (PFC) [5], [6]. This algorithm is built on some sensible concepts that would appeal to practitioners and thus has found widespread acceptance in industry. Nevertheless, recent literature [7]–[9] has emphasised the theoretical weaknesses in the basic algorithm and thus has sought to produce modifications which retain the appeal of the underlying concepts, but give more rigour and confidence in the final control law. A simple summary of some of the core conclusions of this work is:

1) The use of a constant future input in the predictions in conjunction with a single coincidence point can lead to significant inconsistencies affecting both reliable constraint handling and behaviour [7], [12].
2) The definition of the coincidence point makes inconsistent use of target/disturbance information [17] which often results in additional lag in the responses and thus the tuning is not as intuitive as desired.
3) For systems with undesirable open-loop dynamics, some form of pre-conditionning of the predictions is essential to ensure the PFC implementation is reliable [13], [14].

This paper is focussed more on the first two points above; the proposals made could be combined with the 3rd point fairly easily but we want a simple focus as befits a short conference paper.

Specifically, this paper explores the role of the input parameterisation within PFC. Recent work, building on insights from the mainstream MPC community [15], has encouraged the use of input prediction parameterisations which converge to the steady-state asymptotically rather than instantly [16]. It has been shown that these improve constraint handling significantly, and also tuning [11]. Nevertheless, one core facet has not yet been explored in the literature and that is the role of *pseudo-open-loop* control, that is one whereby we seek to achieve open-loop dynamics but within a closed-loop including integral action. The advantage of such an approach is that the input is automatically fairly passive which in many scenarios is an advantage.

A second a more significant contribution of this paper is to propose a different flavour of tuning direction to the conventional algorithm, that is, rather than using the desired settling time as the main tuning parameter, instead using something we will call SPEED-UP. In simple terms this means, how much faster than open-loop do we want the closed-loop system to converge. SPEED-UP is a nice tuning factor because it also has a clear relationship with input activity. For example, a SPEED-UP of 2 suggests that the input will over-actuate by roughly double during transients.

Section II will given a brief introduction to classical PFC and some alternative input parameterisations, including the open-loop dynamics option. Section III will introduce the proposed new PFC approach based on SPEED-UP and then section IV will give some simulation comparisons and illustrations.

[1] Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, S1 3JD, UK (e-mail:j.a.rossiter@sheffield.ac.uk, msaftab1@sheffield.ac.uk)

II. Background on PFC

This section gives an overview of PFC and some simple alternative input parameterisations. This is used as the foundation for the proposal of the following section.

A. System definition

For convenience hereafter, and without loss of generality, take the following *nominal* transfer function model:

$$a(z)y_k = b(z)u_k + d_k \tag{1}$$

so output y_k, input u_k and d_k a disturbance estimate to cater for uncertainty. We assume that true process is similar, for example:

$$a_p(z)y_{p,k} = b_p(z)u_k; \quad d_k = y_{p,k} - y_k \tag{2}$$

Note it is assumed that the input to the process and model are the same. The model used could equally be in state space form and this assumption makes little difference to the control law derivations.

B. System prediction

Prediction is well known [4] so details are omitted here suffice to say one can determine an n-step ahead output predictions as follows, for suitable H, P, Q.

$$\underset{\rightarrow k}{y} = H\underset{\rightarrow k}{u} + Q\underset{\leftarrow k}{y} + P\underset{\leftarrow k}{u} + Ld_k \tag{3}$$

where

$$\underset{\rightarrow}{u_k} = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+n-1} \end{bmatrix}; \underset{\leftarrow}{u_k} = \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \\ u_{k-m} \end{bmatrix};$$

$$\underset{\leftarrow}{y_k} = \begin{bmatrix} y_k \\ y_{k-1} \\ \vdots \\ y_{k-m} \end{bmatrix}; \ \underset{\rightarrow k+1|k}{y} = \begin{bmatrix} y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+n} \end{bmatrix}$$

and L is a vector of ones.

C. Conventional PFC control law

PFC is based on the premise of matching the output prediction to a first order response with a given time constant. Hence, define a target trajectory as:

$$r_{k+i} = (1-\lambda^i)R + \lambda^i y_{p,k}, \quad i = 1, 2, \cdots \tag{4}$$

Note, we ignore details linked to non-zero dead-time examples for simplicity of notation; these are available in many of the references.

The PFC law is defined by forcing the prediction of (3) to match the desired trajectory (4) at a specified point n-steps ahead, assuming that the future input is constant, that is, $u_k = u_{k+i}, \forall i > 0$. Hence the PFC law is defined from:

$$e_n^T[HLu_k + Q\underset{\leftarrow k}{y} + P\underset{\leftarrow k}{u} + Ld_k] = (1-\lambda^n)R + \lambda^n y_{p,k} \tag{5}$$

where e_n is the nth standard basis vector. It is straightforward to determine u_k from (5).

D. Laguerre PFC

It was noted recently [12], [16] that the restriction of the future input to a constant did not match the expected shape of the closed-loop input and thus embedded an inconsistency between predictions and closed-loop, which in turn meant that the tuning was inevitably inconsistent. A simple improvement was to parameterise the future input using a first order Laguerre function, in essence an exponential decay so that:

$$\underset{\rightarrow k}{u} = \underbrace{\begin{bmatrix} 1 \\ \rho \\ \vdots \\ \rho^{n-1} \end{bmatrix}}_{H_\rho} \eta + \begin{bmatrix} u_{ss} \\ u_{ss} \\ \vdots \\ u_{ss} \end{bmatrix}; \tag{6}$$

where ρ is a decay factor to be chosen, η is a degree of freedom (d.o.f.) and u_{ss} is the expected steady-state so that:

$$\{u_{k+i} = u_{ss}, \ \forall i \geq 0\} \quad \Rightarrow \quad \lim_{i\to\infty} E[y_{k+i}] = R \tag{7}$$

For model (1) we can determine that, in steady-state:

$$a(1)y_{ss} = b(1)u_{ss} + d_k \quad \Rightarrow \quad E[u_{ss}] = \frac{a(1)R - d_k}{b(1)} \tag{8}$$

It is straightforward to combine the updated input prediction of (6) with predictions (3) and trajectory (4) to define the modified PFC control law as:

$$e_n^T[H(H_\rho\eta + Lu_{ss}) + Q\underset{\leftarrow k}{y} + P\underset{\leftarrow k}{u} + Ld_k] = (1-\lambda^n)R + \lambda^n y_{p,k} \tag{9}$$

Hence we solve (9) for η and substitute into (6) to determine u_k.

E. Open-loop dynamics PFC (OL)

A final simple alternative is where one is happy with the open-loop dynamics and the feedback is simply to ensure offset free tracking. Such a control law can be achieved with the simple rule:

$$u_k = E[u_{ss}] \tag{10}$$

where $E[u_{ss}]$ is indicated in (8).

Remark 1: It so happens that one can achieve an open-loop dynamics PFC using control law (9) with $\rho = 0$. This observation will prove useful in the following.

Remark 2: It should be emphasised that the open-loop method avoids use of (4) altogether. This is actually a critical part of the proposal in this paper as this means we avoid the inconsistencies highlighted in [10], [17] whereby the target information is used differently in consequent samples, leading to unexpected lag in the closed-loop behaviour.

F. Constraints

It is possible to incorporate constraint handling into PFC in a systematic and computationally simple way, and while retaining feasibility, as demonstrated in several recent papers [11], [12].

$$\begin{array}{ccc} \underline{u} & \leq \mathbf{u}_k \leq & \overline{u} \\ \underline{\Delta u} & \leq \Delta\mathbf{u}_k \leq & \Delta\overline{u} \\ \underline{y} & \leq \mathbf{y}_k \leq & \overline{y} \end{array} \tag{11}$$

and $\Delta \mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_{k-1}$.

However, as these details are not central to the contribution of this paper they are excluded for clarity and brevity.

III. Proposed PFC control law based on SPEED-UP

The key factor here is transparency of tuning. It is assumed that the operator can view the open-loop speed of response and indeed achieve this with the PFC law given in (10), or indeed equivalently (9) with $\rho = 0$. Hence it is transparent and easy for them to define a closed-loop response as being say, twice as fast, and obviously therefore having input activity twice as big.

A. Increasing speed of target trajectory compared to open-loop benchmark

In order to achieve some faster response, then we need the error convergence of the predicted behaviour of (3) to be appropriately faster for consistency. Hence, one core concept is to choose an appropriate coincidence point that will cause the suitably faster behaviour/convergence.

Begin with a benchmark behaviour that would be achieved with the open-loop method of (10), so that the predictions take the form:

$$y_{k+n|k} = e_n^T[HLu_{ss} + Q\underset{\leftarrow k}{y} + P\underset{\leftarrow k}{u} + Ld_k] \quad (12)$$

The associated n-step ahead prediction error is given as:

$$e_{k+n} = R - e_n^T[HLu_{ss} + Q\underset{\leftarrow k}{y} + P\underset{\leftarrow k}{u} + Ld_k] \quad (13)$$

Next, chose a coincidence point which has faster convergence, so implicitly the associated error is smaller by a factor of β, where β is a factor to be determined.

Lemma 1: For $\beta > 1$, a *relative* PFC control law can be defined as follows. Choose η such that:

$$\begin{aligned} R - e_n^T[HLu_{ss} + Q\underset{\leftarrow k}{y} + P\underset{\leftarrow k}{u} + Ld_k] = \\ \beta[R - e_n^T[HLu_{ss} + HH_\rho\eta + Q\underset{\leftarrow k}{y} + P\underset{\leftarrow k}{u} + Ld_k] \end{aligned} \quad (14)$$

$$\eta = \frac{(\beta-1)}{e_n^T HH_\rho \beta}[R - e_n^T[HLu_{ss} + Q\underset{\leftarrow k}{y} + P\underset{\leftarrow k}{u} + Ld_k] \quad (15)$$

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

Proof: It is clear from equation (15) that the coincidence point for the predictions with the Laguerre addition, has an associated error which is β times smaller than the error using predictions based on the open-loop approach.

The core point here is that control law (15) gives us a mechanism for achieving faster behaviour using a PFC equivalent statement; this control law is analogous to (5) with the critical exception that now there is now need for the tuning parameter λ. This difference is fundamental to the contribution of the paper as tuning is now based on relative statements rather than absolute ones.

B. Determining a precise PFC law with faster responses

First we establish a common sense observation for faster closed-loop responses, that is, a faster response requires a more aggressive input action.

Lemma 2: In simple terms, for zero initial conditions and a change in the target, a necessary condition for the response to be θ times faster is if the initial input u_k for a step change in the target to be θ times bigger. This lemma is given without proof as self evident.

Next, we look at the impact of requiring a smaller asymptotic error (as in (15)) on the initial input magnitude. The argument is that, from linearity, comparing the input activity with zero initial conditions and zero disturbance is a likely indicator of the resulting closed-loop poles and this simplifies the next stage of the analysis.

The initial input, for a change in target R and coincident point (13) and zero initial conditions, using the open-loop tuning is given in (10). (In this case the input is a constant throughout the predictions.)

Lemma 3: The initial input, for a change in target R and control law (15) is given as follows:

$$u_k = u_{ss} + \eta = \frac{R}{g(1)} + \frac{(\beta-1)}{e_n^T HH_\rho\beta}[R - e_n^T[HLu_{ss}] \quad (16)$$

This also follows directly from (15).

For convenience hereafter define the following:

$$h_\rho = e_n^T HH_\rho; \quad h = e_n^T H; \quad E[u_{ss}] = \frac{R}{g(1)} \quad (17)$$

Hence (16) can be simplified to:

$$u_k = \frac{R}{g(1)} + \frac{(\beta-1)}{h_\rho\beta}[R - h\frac{R}{g(1)})] \quad (18)$$

Theorem 1: The initial input from (18) is θ times faster than (10) if β is chosen as follows:

$$\beta = \frac{h - g(1)}{(\theta-1)h_\rho - g(1) + h} \quad (19)$$

.

Proof: Placing the two inputs (10), (18) side by side and removing the common factor R, we have:

$$\frac{\theta}{g(1)} = \left[\frac{1}{g(1)} + \frac{(\beta-1)}{h_\rho\beta}[1 - h\frac{1}{g(1)}]\right] \quad (20)$$

Create a common denominator and match the numerators, hence:

$$h_\rho\beta\theta = h_\rho\beta + (\beta-1)[g(1) - h] \quad (21)$$

Finally, solving for β gives the result in (19).

Remark 3: The derivation of the value for β was done with the nominal case and zero initial conditions for simplicity. However, as the final control law is in (15), the implied poles will be retained for the closed-loop and moreover, there will be robustness to uncertainty and offset free tracking.

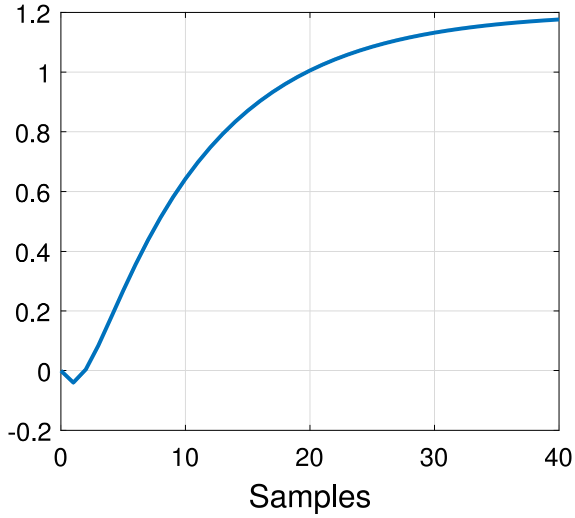

Fig. 1. Open-loop step response for system (22).

C. Summary of proposed algorithm

This section summarises the core conceptual steps and algebra needed to implement the algorithm. it is noticed that the computations are equivalent to a conventional PFC approach and thus neither more nor less complicated to code and implement. The core difference is the approach to tuning where here one adopts relative statements (faster or slower) rather than specifying desired poles/time constants precisely.

1) Verify that the open-loop behaviour is broadly acceptable so can be used as a valid benchmark.
2) Determine the desired speed-up factor θ, that is how much faster than open-loop behaviour do you want the closed-loop to be?
3) Solve for the parameter β using equation (19).
4) Determine the PFC law using equation (15).

Remark 4: Constraint handling can be handled in a conventional PFC manner using a simple *for loop* as discussed in the references. It is reiterated that recursive feasibility is automatic in the nominal case, although of course guarantees in the presence of uncertainty require computational complexity, expense and approaches which exceed the remit of PFC.

IV. Numerical comparisons

This section will demonstrate the efficacy of the proposed PFC approach as an alternative way to tune closed-loop behaviour. It needs to be re-emphasised that the method is based on the assumption that the open-loop dynamics are essentially satisfactory so this method may not be appropriate for systems with significant under-damping or open-loop instability.

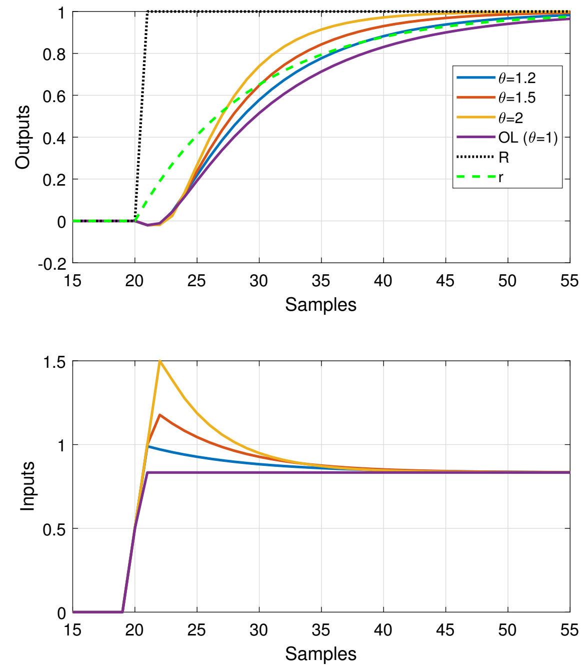

Fig. 2. Closed-loop responses for system (22) with various θ.

A. Example 1

Take the 2nd order, slightly over-damped system, with a non-minimum phase zero:

$$y(z) = \frac{-0.04z^{-1} + 0.1z^{-2}}{1 - 1.4z^{-1} + 0.45z^{-2}} \tag{22}$$

It should be remarked that the presence of the non-minimum phase zero makes a conventional PFC difficult to tune effectively and very difficult to achieve faster than open-loop behaviour!

The coincidence horizon is taken to be 15 in lieu of the slow pole at 0.9. The open-loop response is given in figure 1. The closed-loop responses for different choices of θ are shown in figure 2. It is clear that the required speed up has been achieved accurately and thus the proposed tuning parameter of θ is intuitive and easy to use.

Remark 5: The tuning parameter θ can also be used to achieve performance slower than open-loop, for example where there is a particular desire for the input to be slowly varying. This is illustrated in figure 3.

B. Example 2

Take a 3rd order, system, again with a non-minimum phase zero:

$$y(z) = \frac{0.1z^{-1} - 0.4z^{-2}}{1 - 1.85z^{-1} + 1.035z^{-2} - 0.171z^{-3}} \tag{23}$$

It should be remarked that the presence of the non-minimum phase zero makes a conventional PFC difficult to tune effectively and very difficult to achieve faster than open-loop behaviour!

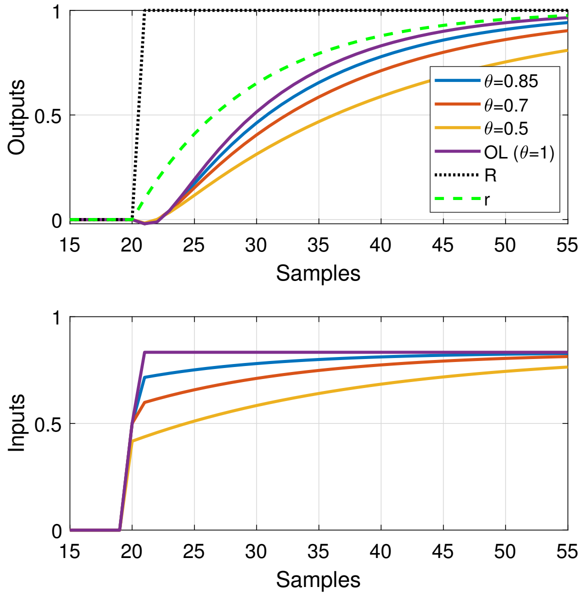

Fig. 3. Closed-loop responses for system (22) with θ chosen to slow behaviour down.

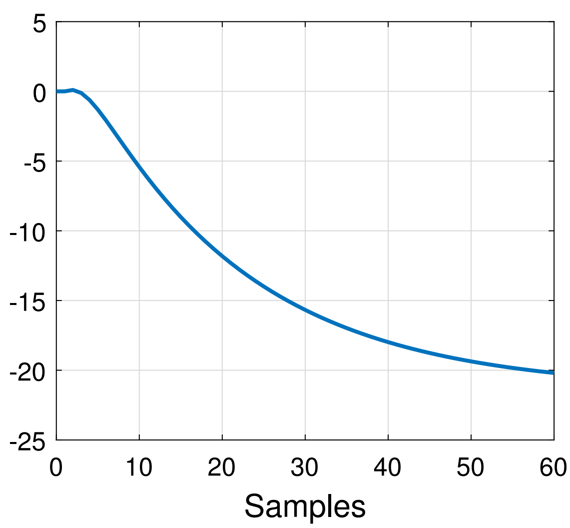

Fig. 4. Open-loop step response for system (23).

The coincidence horizon is taken to be 30 in lieu of the very slow pole at 0.95. The open-loop response is given in figure 4. The closed-loop responses for different choices of θ are shown in figure 5. Once again it is evident that the required speed up has been achieved accurately and thus the proposed tuning parameter of θ is intuitive and easy to use.

C. Disturbance rejection

For completeness, this section illustrates that the benefits are retained by the loop and thus apply, for example during disturbance rejection. Figure 6 shows the disturbance rejection with system (23); it is clear that the SPEED-UP has

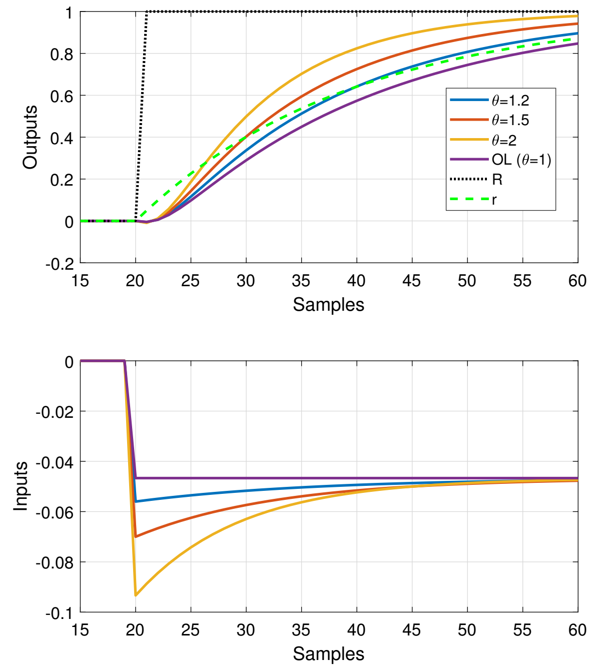

Fig. 5. Closed-loop responses for system (23) with various θ.

been retained.

V. Conclusions and Future Work

This paper has proposed a totally different conceptually approach to PFC algorithms, that is where tuning is based on relative rather than absolute statements. The advantage of using relative statements is that it is possible to enable an intuitive tuning parameter, here denoted as SPEED-UP: how much faster, or slower, than open-loop do you want to be? It is also noticeable that the proposed approach moves away form the traditional control law definition around (4) and thus avoids issues linked to inconsistent use of the target information [17].

As compared to traditional PFC approaches and indeed the many modifications proposed in the recent literature, the tuning parameter here seems to behave far more consistently so that the user achieves the desired behaviour; this is evident from figures 2-5 where the initial input over or under actuates to the required degree. It should be emphasised however, that this approach is not effective with under-damped systems.

A core conceptual point within this paper is that it builds on work [16] which used a Laguerre formulation for the input parameterisation. This is essential as it means that the predicted input moves smoothly from its initial over-actuation to the required steady-state thus giving consistency between predictions and closed-loop behaviour, something that conventional PFC cannot give.

References

[1] E.F. Camacho and C. Bordons, Model Predictive Control. Springer, 1999.

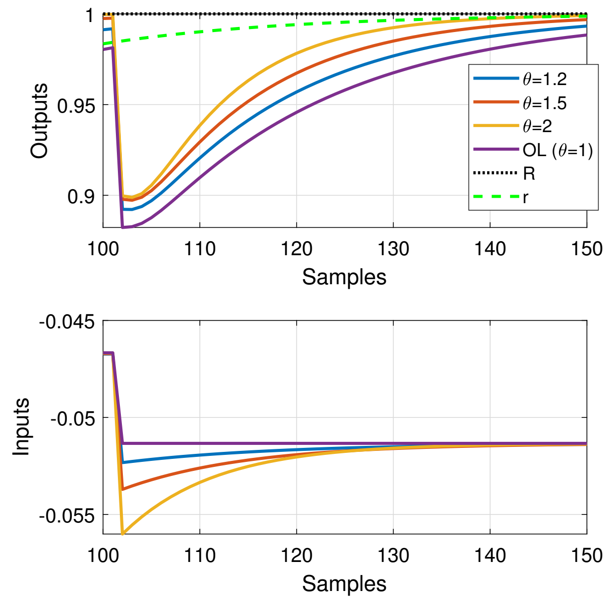

Fig. 6. Closed-loop disturbance rejection for system (23) with various θ.

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

[3] D.Q. Mayne, Model predictive control: Recent developments and future promise, Automatica, vol. 50, no. 12, pp. 2967-2986, Dec 2014.

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

[17] J.A. Rossiter and M. Abdullah, Improving the use of feedforward

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

[6] J. Richalet and D. O'Donovan, Predictive Functional Control: Principles and Indus- trial Applications. Springer Science and Business Media, 2009.

[7] J. Rossiter and R. Haber, The effect of coincidence horizon on predictive functional control, Processes, vol. 3, no. 1, pp. 25-45, 2015.

[8] J. Rossiter, Input shaping for pfc: how and why?" Journal of control and decision, vol. 3, no. 2, pp. 105-118, 2016.

[9] J.A. Rossiter and M.S. Aftab, M.S. A Comparison of Tuning Methods for Predictive Functional Control. Processes 2021, 9, 1140. https://doi.org/10.3390/pr9071140.

[10] J. A. Rossiter and M. Abdullah, A new paradigm for predictive functional control to enable more consistent tuning, in 2019 American Control Conference (ACC). IEEE, Jul 2019.

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

[12] M. Abdullah and J. A. Rossiter, Using laguerre functions to improve the tuning and performance of predictive functional control, International Journal of Control, pp.1-13, 2019.

[13] M.S. Aftab and J.A. Rossiter, (Accepted: 2021) Pre-stabilised predictive functional control for open-loop unstable dynamic systems, 7th IFAC Conference on Nonlinear Model Predictive Control 2021, 11-14 Jul 2021, Virtual conference.

[14] Z. Zhang, J. A. Rossiter, L. Xie, H. Su, Predictive functional control for integrator systems, Journal of the Franklin Institute, Volume 357, Issue 7, 2020, Pages 4171-4186, ISSN 0016-0032, https://doi.org/10.1016/j.jfranklin.2020.01.026.

[15] K.R. Muske and J.B. Rawlings, Model predictive control with linear models, AIChE Journal, 39(2), 262–287, 1993.

[16] M. Abdullah and J. A. Rossiter, Utilising Laguerre function in predictive functional control to ensure prediction consistency, 2016 UKACC 11th International Conference on Control (CONTROL), 2016, pp. 1-6, doi: 10.1109/CONTROL.2016.7737639.

in predictive functional control to improve the impact of tuning, International Journal of Control, 2020, ISSN 0020-7179.