

On Fractional Predictive PID Controller Design Method

Emmanuel Edet*. Reza Katebi.**

* *Technology and Innovation Centre, Level 4, Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, G1 1RD. UK. (e-mail: emmanuel.edet@strath.ac.uk).*

** *Industrial Control Centre, Royal College Building, Electronic and Electrical Engineering Department, University of Strathclyde, 204 George Street, Glasgow, G1 1XW. UK. (Tel: +441415484297; e-mail: m.r.katebi@strath.ac.uk)*

Abstract: A new method of designing fractional-order predictive PID controller with similar features to model based predictive controllers (MPC) is considered. A general state space model of plant is assumed to be available and the model is augmented for prediction of future outputs. Thereafter, a structured cost function is defined which retains the design objective of fractional-order predictive PI controller. The resultant controller retains inherent benefits of model-based predictive control but with better performance. Simulations results are presented to show improved benefits of the proposed design method over dynamic matrix control (DMC) algorithm. One major contribution is that the new controller structure, which is a fractional-order predictive PI controller, retains combined benefits of conventional predictive control algorithm and robust features of fractional-order PID controller.

Keywords: Fractional-order PI, Dynamic matrix control, Model-based predictive controller

1. INTRODUCTION

The idea of incorporating future set-point into control formulation has gained popularity in various forms giving rise to different variants of model-based predictive control (MPC) methods. These MPC algorithms include: generalised predictive control (GPC), dynamic matrix control (DMC), model algorithmic control (MAC) or finite spectrum assignment (FSA). The central feature is the same which is the incorporation of future set – point in the control law (Camacho & Bourdons, 1998). However, each algorithm differ slightly in the structure of the objective function being optimised and in the type of model used for prediction of future set point.

Constraints handling and inherent multi-input multi-output (MIMO) capability are two attractive benefits of using predictive control method. The goal of the design procedure is to realise an optimal or sub-optimal gains of controller that guarantees excellent tracking performance (Wang, 2009). In this paper, attractive features of both MPC and fractional-order PID (FOPID) controllers are combined to realise a hybrid controller design algorithm with improved performance. A single input single output (SISO) process is selected to test the performance of proposed controller and results can be directly extended to multivariable control problems in a straightforward way by modifying dimensions of relevant matrices.

Historically, fractional order control systems have been of interest to researchers for many years. This is because these controllers, when properly tuned, have been found to yield better performance compared to integer-order controllers under similar conditions (Li & Chen, 2014). Several authors have recorded tremendous progress in utilising fractional order calculus to design control systems (Monje, et al., 2008; Padula & Visioli, 2011; Zhuang, et al., 2008). Improved performance is partly down to extra flexibility granted by the non-restriction

of derivative orders implying that more conflicting design objectives can be achieved. For instance, a conventional integrator (with integer order) is sufficient for rejection of steady state error. However, with the fractional case, extra tuning parameters are available for meeting more frequency domain-based stability specifications.

As a result of such comparative benefits, many design methods have been proposed over the years for fractional-order PID controllers, especially, SISO process applications designed in continuous time domain. One class of design methods is based on direct synthesis. FOPID controller parameters can be computed directly in frequency domain to meet some desired specifications such as phase margin, constant phase at gain cross over frequency and other sensitivity constraints (Luo, et al., 2010; Luo & Chen, 2009; Vale´rio & Costa, 2006). Other methods of designing FOPID controllers include optimization based methods such as linear matrix inequality (LMI) and evolutionary algorithms (Lee & Chang, 2010; Song, et al., 2011).

Considering discrete FOPID controllers with predictive capability, there is no standardised design or tuning method compared to conventional PID controllers. In this paper, a new fractional order predictive PI (FOPPI) controller is proposed with hybrid benefits. Future set-point is incorporated in formulating the control law. It also retains both constraints handling capability of MPC controller and robust features of FOPID controllers. Some useful definitions are presented next.

1.1 Fractional Order Derivatives

Consider a differential equation with fractional order (α):

$$y(t) = D^\alpha u(t) + u(t)$$

Two important fractional derivative definitions are stated:

- Grunwald-Letnikov (GL) definition
- Riemann-Liouville Definition.

According to Riemann-Liouville:

$$D^\alpha f(t) = \frac{d^n}{dt^n} \left(\frac{1}{\Gamma(n-\alpha)} \int_0^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau \right) \quad (1)$$

where $t > 0$; $n-1 < \alpha < n$; $n \in \mathbb{R}$ (Podlubny, 1999).

An alternative definition is Grunwald-Letnikov (GL):

$$D^\alpha f(t)_{t=k\tau} = \lim_{\tau \rightarrow 0} \tau^{-\alpha} \sum_{j=0}^k (-1)^j \binom{\alpha}{j} f(k\tau - j\tau) \quad (2)$$

where: $t > 0$; $n-1 < \alpha < n$; $n \in \mathbb{R}$

$\binom{\alpha}{j}$ represents binomial coefficient.

In continuous form, Laplace transform has been used directly to describe FOPID controllers as given in (3).

$$C(s) = k_p + \frac{k_I}{s^\lambda} + k_D s^\mu \quad (3)$$

where: λ and μ are the integral and derivative orders. k_p , k_I and k_D are proportional, integral and derivative gains respectively.

However, in discrete time domain, different approximation methods are sometimes used to approximate fractional integral and derivative terms. These approximation methods are based on sampling continuous time signal and discretising it at each sampling instant. Examples are Tustin approximation, forward Euler approximation, backward Euler approximation, Taylor series approximation and the long memory discrete time PID - LDPID. Although the GL definition results in infinite series, an approximated form given in (4) is used extensively by several authors for deriving numerical solution (Zhuang, et al., 2008).

$${}_{(t-L)}D_t^\alpha f(t) = \tau^{-\alpha} \sum_{j=0}^{N(t)} b_j f(t - j\tau) \quad (4)$$

where: b_j is binomial coefficient.

$$b_0 = 1; b_1 = \mu b_0; b_2 = \left(1 - \frac{1+\mu}{2}\right) b_1; b_j = \left(1 - \frac{1+\mu}{j}\right) b_{j-1}.$$

L = memory length; $N(t) = \min \left\{ \left\lceil \frac{t}{\tau} \right\rceil, \left\lfloor \frac{L}{\tau} \right\rfloor \right\}$ (Zhuang, et al., 2008).

In this work, the same approach is followed to define a discrete fractional-order PI controller which is only used to formulate cost function for the proposed FOPPI controller.

1.2 Discrete Time Fractional-order PI controller

Consider the fractional order PI controller:

$$u(t)_{pi} = k_p e(t) + k_i D_t^{-\mu} e(t)$$

The discrete time form can be expressed as given below in (5).

$$u_{pi}(k) = k_p e(k) + k_i T_s^\mu \sum_{j=0}^k b_j e(k-j) \quad (5)$$

where: T_s = sampling time. Other terms are as defined in (4).

Guo, et al.,(2010) developed a combined DMC and FOPID design method with some benefits. Similarly, in this paper, an incremental form of fractional-order PI algorithm is used to formulate objective function but without any derivative term in the cost function.

$$\Delta u_{pi}(k) = u_{pi}(k) - u_{pi}(k-1).$$

$$u_{pi}(k-1) = k_p e(k-1) + k_i T_s^\mu \sum_{j=0}^{k-1} b_j e(k-1-j) \quad (6)$$

$$\Rightarrow u_{pi}(k) - u_{pi}(k-1) = k_p (e(k) - e(k-1))$$

$$+ k_i T_s^\mu \left(\sum_{j=0}^k b_j e(k-j) - \sum_{j=0}^{k-1} b_j e(k-1-j) \right)$$

$$\therefore \Delta u_{pi}(k) = k_p (e(k) - e(k-1))$$

$$+ k_i T_s^\mu \left(e(k) - \sum_{j=1}^k \frac{1+\mu}{j} b_{j-1} e(k-j) \right) \quad (7)$$

$$\Delta u_{pi}(k) = k_p \Delta e(k) + k_i T_s^\mu \left(e(k) - \sum_{j=1}^k B_j e(k-j) \right) \quad (8)$$

where: $B_j = \frac{1+\mu}{j} b_{j-1}$. Let: $K_I = k_i T_s^\mu$

Equation (8) is therefore re-written:

$$\Delta u_{pi}(k) = k_p \Delta e(k) + K_I \left(e(k) - \sum_{j=1}^k B_j e(k-j) \right) \quad (9)$$

Two distinct components are evident in (9) namely: a proportional term and a fractional integral component. These two terms are used to formulate a new cost function which is minimised in order to obtain the proposed FOPPI controller. However, $\Delta u_{pi}(k)$ is not implemented directly as given in (9). Direct implementation of FOPID controller has to be band limited. Instead, a model-based predictive control framework is used for implementation of the fractional order predictive PI controller. Consequently, control signal limits are defined inherently using constraint handling feature of MPC.

2. THE PROPOSED ALGORITHM

2.1 A Brief Review of Model Predictive Control Algorithm

Consider a SISO process described by the state space equation in (10):

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \quad (10)$$

where: u is the control input; y is the output; x is the state variable vector and matrices A , B , C , and D are system matrix, input control matrix, output matrix and feedthrough matrix.

Equation (10) is usually used (without the D term) for DMC derivation. This is because in receding horizon control, current information of the plant is required for prediction and control implying that the input $u(k)$ cannot directly affect the output $y(k)$ at the same time. The state space model is rewritten:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \quad (11)$$

The standard state space model in (11) is augmented in order to incorporate integral action. Incremental model is given as:

$$\begin{aligned} \Delta u(k) &= u(k) - u(k-1) \\ x(k+1) - x(k) &= \Delta x(k+1) = A\Delta x(k) + B\Delta u(k) \\ \Delta y(k+1) &= y(k+1) - y(k) = Cx(k+1) - Cx(k) \\ \Rightarrow \Delta y(k+1) &= C\Delta x(k+1) \\ \Delta y(k+1) &= CA\Delta x(k) + CB\Delta u(k) \end{aligned}$$

The augmented matrix equation obtained is given in (12):

$$\begin{aligned} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{bmatrix} A & 0^T \\ CA & I \end{bmatrix} \begin{pmatrix} \Delta x(k) \\ y(k) \end{pmatrix} + \begin{bmatrix} B \\ CB \end{bmatrix} \Delta u(k) \\ y(k) &= [0 \quad 1] \begin{pmatrix} \Delta x(k) \\ y(k) \end{pmatrix} \end{aligned} \quad (12)$$

where: $x_1 = \Delta x(k+1)$; $x_2 = \Delta y(k+1)$.

This augmented state space model is used for prediction of future states. Let the prediction horizon be represented as ' p ' and control horizon be ' N '. Given that $p \geq N$, future states can be predicted as follows:

$$\begin{aligned} \hat{x}(k+1) &= Ax(k) + B\Delta u(k) \\ \Rightarrow \hat{x}(k+2) &= A^2x(k) + AB\Delta u(k) + B\Delta u(k+1) \\ \therefore \hat{x}(k+p) &= A^p x(k) + A^{p-1}B\Delta u(k) + \dots \\ &+ A^{p-N}B\Delta u(k+N-1) \end{aligned} \quad (13)$$

Similarly, the future output variables can be predicted using:

$$\begin{aligned} \hat{y}(k+1) &= CAx(k) + CB\Delta u(k) \\ \Rightarrow \hat{y}(k+2) &= CA^2x(k) + CAB\Delta u(k) + CB\Delta u(k+1) \\ \therefore \hat{y}(k+p) &= CA^p x(k) + CA^{p-1}B\Delta u(k) + \dots \\ &+ CA^{p-N}B\Delta u(k+N-1) \end{aligned} \quad (14)$$

The prediction equation is therefore given by:

$$\hat{Y}_p(k+1) = F_p \hat{X}(k) + G\Delta u(k) \quad (15)$$

$$\text{where: } F_p = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^p \end{bmatrix}; G = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ CA^{p-1}B & CA^{p-2}B & CA^{p-3}B & \dots & CA^{p-m}B \end{bmatrix}_{p \times m}$$

The cost function to be minimised is given by J:

$$J = \sum_{i=1}^p \|r(k+i) - y(k+i)\|_2 Q + \sum_{j=1}^N \|\Delta U(k+j-1)\|_2 R \quad (16)$$

where: R = weighting vector for the control effort and Q = state weighting matrix. A standard MPC solution exist for this optimisation problem assuming no input constraint and taking derivative of J . Therefore:

$$\Delta U_{mpc} = l(G^T G + \lambda I)^{-1} G^T e(k) \quad (17)$$

where: $e(k)$ represents error at sampling instant ' k '

$$l = [1, 0, 0, \dots, 0]^T$$

This is the standard MPC control technique with feedback achieved through receding horizon control (Miller, et al., 1999). In contrast, the cost function given in (16) is changed to obtain a new fractional-order predictive PI controller design method. Both proportional error and integral error components are introduced in the new cost function. Implementation of the derived FOPPI controller is by receding horizon technique as only the first column of control input is applied to the process. At a new sampling instant, all prediction matrix coefficients are re-calculated.

2.2 Derivation of Proposed Control Law

In this section, the proposed control law is derived. At a sampling instant k , within a prediction horizon p , the aim of the control system is to bring the future predicted output $\hat{Y}(k+1)$ as close as possible to the expected set-point signal $-Y_d(k+1)$ assuming the set point signal remains constant in the optimization window. The task is therefore reduced to finding the optimum (or best) control parameter vector ΔU such that an error function between the set-point and the predicted output is minimized. The state space model earlier augmented is used to predict the output as explained in previous section.

Given: P = prediction horizon and N = control horizon

The structured quadratic cost function to be minimised is formed directly from (9) as shown:

$$J = \sum_{k=1}^p \left[k_p (\Delta e(k+1))^2 + K_I (e(k+1))^2 \right] + \gamma \sum_{k=1}^N \Delta u(k)^2 \quad (18)$$

where: $\gamma =$ control signal weight.

Let $Y_f(k+1)$ = forced output signal due to the control input.

The future output after p steps is the sum of free output response and forced output as shown:

$$\hat{Y}(k+1) = \hat{Y}_p(k+1) + Y_f(k+1)$$

The error signal is the difference between the desired set-point and the predicted output:

$$E(k+1) = Y_d(k+1) - \hat{Y}(k+1)$$

Given that the forced output is described as:

$$Y_f(k+1) = G\Delta u(k)$$

$$\therefore E(k+1) = Y_d(k+1) - \hat{Y}_p(k+1) - G\Delta u(k) \quad (19)$$

Re-write the cost function given in (18) as vectors:

$$J = k_p \Delta e^T \Delta e + K_I e^T e + K_I \sum_{j=1}^k B_j e_j^T e_j + \gamma \Delta u^T \Delta u \quad (20)$$

Error signals can be expressed using (19):

$$e = Y_d - \hat{Y}_p - G\Delta u; \quad \Delta e = \Delta Y_d - \Delta \hat{Y}_p - G_p \Delta u$$

$$e_j = Y_{dj} - \hat{Y}_{pj} - G_j \Delta u$$

where coefficient matrices are defined as given below:

$$G = \begin{bmatrix} g_1 & 0 & 0 & 0 & 0 \\ g_2 & g_1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ g_m & g_{m-1} & g_{m-2} & \cdots & g_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ g_p & g_{p-1} & g_{p-2} & \cdots & g_{p-m+1} \end{bmatrix}_{pxm}$$

Elements of matrix G are as derived previously:

$$G = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ CA^{p-1}B & CA^{p-2}B & CA^{p-3}B & \cdots & CA^{p-m}B \end{bmatrix}_{pxm}$$

Matrices (G_p and G_j) are derived from matrix- G :

$$G_p = \begin{bmatrix} g_1 & 0 & \cdots & 0 \\ g_2 - g_1 & g_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_p - g_{p-1} & g_{p-1} - g_{p-2} & \cdots & g_{p-m+1} - g_{p-m} \end{bmatrix}_{pxm}$$

$$G_j = [G_1, G_2, \dots, G_k]; \quad j=1,2,3, \dots, k. \quad G_k = 0$$

$$G_1 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ g_1 & 0 & 0 & \cdots & 0 \\ g_2 & g_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{p-1} & g_{p-2} & \cdots & \cdots & g_{p-m} \end{bmatrix}_{pxm}$$

$$G_2 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ g_1 & 0 & 0 & \cdots & 0 \\ g_2 & g_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{p-2} & g_{p-3} & \cdots & \cdots & g_{p-m-1} \end{bmatrix}_{pxm}$$

Substituting for error signals in J as in (19):

$$\begin{aligned} \Rightarrow J &= k_p [\Delta Y_d - \Delta \hat{Y}_p - G_p \Delta u]^T [\Delta Y_d - \Delta \hat{Y}_p - G_p \Delta u] \\ &+ K_I [Y_d - \hat{Y}_p - G \Delta u]^T [Y_d - \hat{Y}_p - G \Delta u] \\ &+ \sum_{j=1}^k k_j [Y_{dj} - \hat{Y}_{pj} - G_j \Delta u]^T [Y_{dj} - \hat{Y}_{pj} - G_j \Delta u] + \gamma \Delta u^T \Delta u \end{aligned}$$

The minimum point can be found by finding the gradient of J with respect to Δu .

$$\frac{\partial J}{\partial \Delta u} = 0;$$

$$\begin{aligned} \therefore & (k_p G_p^T \Delta Y_d + k_p G_p) (\Delta Y_d - \Delta \hat{Y}_p) + (K_I G^T + K_I G) (Y_d - \hat{Y}_p) \\ & + \sum_{j=1}^k k_j (G_j^T + G) (Y_{dj} - \hat{Y}_{pj}) \\ & = 2\gamma \Delta u + \Delta u (2k_p G_p^T G_p) + \Delta u (2K_I G^T G) + 2\Delta u \sum_{j=1}^k (k_j G_j^T G_j) \end{aligned}$$

$$\text{where: } k_j = K_I \frac{1+\mu}{j} b_{j-1} = K_I B_j$$

Solving for optimum control increment:

$$\begin{aligned} \Delta u(k) &= k_p G_p^T l \phi (\Delta Y_d - \Delta \hat{Y}_p) + K_I G^T l \phi (Y_d - \hat{Y}_p) \\ &+ \sum_{j=1}^k k_j G_j^T l \phi (Y_{dj} - \hat{Y}_{pj}) \end{aligned} \quad (21)$$

where: $l = [1, 0, 0, \dots, 0]^T$

$$\phi = \left(k_p G_p^T G_p + K_I G^T G + \sum_{j=1}^k (k_j G_j^T G_j) + \gamma I \right)^{-1}$$

Equation (21) describes the new FOPPI controller. PI parameters (k_p and k_i) are provided for tuning.

3. HANDLING OF CONSTRAINTS

Constraints can be readily formulated into the cost function thereby turning the control design problem into a quadratic programming problem (Katebi & Moradi, 2001). Several quadratic programming routines are available for solving constrained optimisation problem. The most common constraint is the constraint on input control signal amplitude.

$$\text{e.g. } 0 \leq u(k) \leq 0.8$$

This type of constraint can be defined to handle input saturation problems. Other types of constraints include:

- Constraint on the incremental variation of the input control signal (e.g. $-0.08 \leq \Delta u(k) \leq 0.08$)
- Constraint on output signal (e.g. $-1.0 \leq y(k) \leq 1.0$).

4. SIMULATION EXAMPLES

4.1 Example 1 – Double Integrator Process Control

Consider a double integrator plant given by:

$$x(k+1) = Ax(k) + Bu(k)$$

$$y(k) = Cx(k)$$

$$\text{where: } A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}; B = \begin{bmatrix} 0.5 \\ 1 \\ 0.5 \end{bmatrix}; C = [0 \quad 0 \quad 1].$$

Let the sampling period be 0.01s and prediction horizon chosen to be 7. The number of optimization step chosen to be 7 and all weighting matrices chosen as unity identity matrices of proper dimension. Small values of k_p and k_i can be used to tune the plant. Here, k_i and k_p are 0.00351 and 0.7×10^{-4} with fractional order set to 0.7 and optimization is unconstrained. Simulation of step response is illustrated in Fig. 1.

$$\Delta U = 10^{-17} x \begin{bmatrix} -0.502 & -0.502 & -0.251 & -0.1255 & -0.0628 & -0.0157 & -0.002 \\ 0.1391 & 0.1391 & 0.0695 & 0.0348 & 0.0174 & 0.0043 & 0.0005 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0.002 & 0.002 & 0.001 & 0.0005 & 0.0003 & 0.0001 & 0.0000 \end{bmatrix}$$

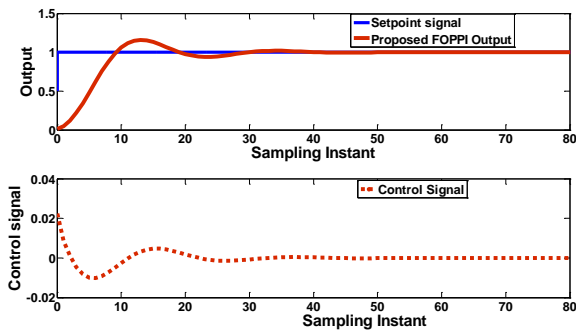


Fig. 1. Step response diagram showing the output signal rising to the unit step reference with control input in blue.

4.2 Example 2 – Non-Minimum Phase Control Comparison

Consider a non-minimum phase process control example where the model is described using state space with matrices:

$$A = \begin{bmatrix} -0.0217 & -0.3141 \\ 0.3141 & 0.7630 \end{bmatrix}; B = \begin{bmatrix} 0.3141 \\ 0.2364 \end{bmatrix}; C = [-1 \quad 2].$$

(Uren & Schoor, 2011)

The sampling period T_s is selected as 1s. Prediction horizon is chosen to be 7 and fractional order of 0.9. The number of optimization steps is chosen to be 7 and all weight matrices chosen as unity identity matrices of proper dimension. Obtained control matrix is given:

$$\Delta U = 10^{-12} x \begin{bmatrix} -0.5539 & -0.7005 & -0.6584 & -0.5142 & -0.3281 & -0.149 & -0.0252 \\ 0.1401 & 0.1772 & 0.1666 & 0.1301 & 0.0830 & 0.0377 & 0.0065 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0.002 & 0.0003 & 0.0003 & 0.0002 & 0.0001 & 0.0001 & 0.0000 \end{bmatrix}$$

With k_i and k_p equal to 0.15, excellent set-point tracking is achieved as shown in Fig.2. In order to demonstrate comparative benefit, DMC is used as a baseline (Uren & Schoor, 2011). It was properly tuned for this same plant by the authors using $N_p = 20$; $N_c = 3$; T_s is selected as 1 and weight $R = 0.9I$. The resultant (incremental) DMC control:

$$\Delta U_{mpc} = 10^{-13} x \begin{bmatrix} 0.1503 \\ 0.0190 \\ 0.0479 \end{bmatrix}$$

Applied control signal is updated ($u_1 = u_0 + \Delta u_1$). Fig. 3 shows disturbance rejection property of these two controllers. A 25% input disturbance is introduced at 200s. It can be observed that the control input of the proposed fractional predictive controller falls within the range: $0 < u < 0.8$.

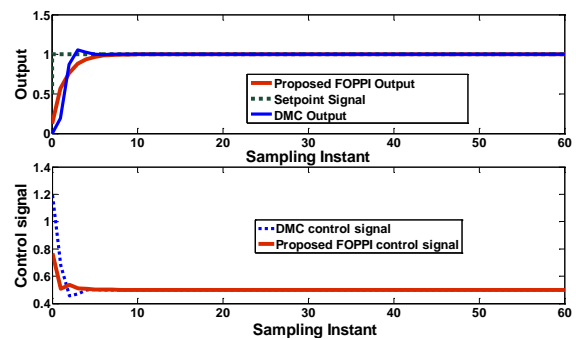


Fig. 2. Step response diagram showing the proposed output (brown) rising to the unit step reference with zero overshoot compared to DMC output (blue).

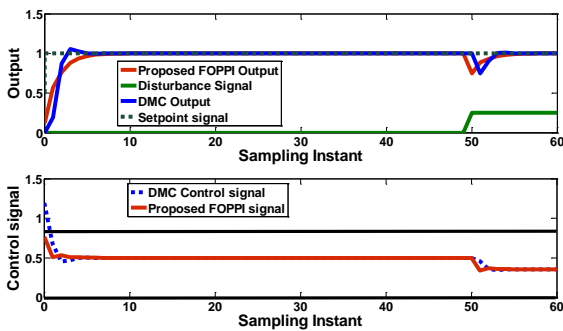


Fig. 3. Disturbance rejection comparison – Both methods have similar settling time of 5s after 25% disturbance (green) is introduced at $k=50$ s. The proposed method (brown) has zero overshoot.

6. CONCLUSIONS

It can be observed from simulation examples that the new fractional predictive control design method performs better than dynamic matrix control algorithm without any increased computational overhead. Also, control effort used to achieve this result is smaller therefore meeting any input constraints more easily. The significance of the proposed design is in simplicity of tuning. While the algorithm can be computed in some programmable or computerised fashion, the entire design procedure can be automated to the point of pushing k_p and k_I knobs. This is expected to be attractive to industrial practitioners.

REFERENCES

- Camacho, E. & Bourdons, C., 1998. *Model Predictive Control*. London: Springer-Verlag.
- Guo, W., Wen, J. & Zhuo, W., 2010. *Fractional-order PID Dynamic Matrix Control Algorithm based on Time Domain*. Jinan, China, IEEE.
- Katebi, M. & Moradi, M., 2001. Predictive PID Controllers. *IEE Proceeding - Control Theory Applications*, 148(6), pp. 478-487.
- Lee, C.-H. & Chang, F.-K., 2010. Fractional-order PID controller optimization via improved electromagnetism-like algorithm. *Expert Systems with Applications*, 37(12), pp. 8871-8878.
- Li, Z. & Chen, Y., 2014. *Ideal, Simplified and Inverted Decoupling of Fractional order TITO Processes*. Cape Town, IFAC.
- Luo, Y., Chenc, Y. Q., Wang, C. Y. & Pi, Y. G., 2010. Tuning fractional order proportional integral controllers for fractional order system. *Journal of Process Control*, 20(7), pp. 823-831.
- Luo, Y. & Chen, Y., 2009. Fractional order [proportional derivative] controller for a class of fractional order systems. *automatica*, Volume 45, pp. 2446-2450.
- Miller, R., Shaha, S., Wooda, R. & Kwokb, E., 1999. Predictive PID. *ISA Transactions*, Volume 38, pp. 11-23.
- Monje, C., Vinagre, B., Vicente, F. & Chen, Y., 2008. Tuning and auto-tuning of fractional order controllers for industry applications. *Control engineering practice*, 16(7), pp. 798-812.
- Padula, F. & Visioli, A., 2011. Tuning rules for optimal PID and fractional order PID controllers. *Journal of Process Control*, 21(1), pp. 69-81.
- Podlubny, I., 1999. Fractional-Order Systems and PID Controllers. *IEEE Transactions on Automatic Control*, 44(1), pp. 208-214.
- Song, X., Chen, Y., Tejado, I. & Vinagre, B. M., 2011. *Multivariable fractional order PID controller design via LMI approach*. Milano, Italy, International Federation of Automatic Control.
- Uren, K. & Schoor, G., 2011. *Predictive PID control of non-minimum phase systems*, Potchefstroom: Intech.
- Vale'rio, D. & Costa, J. S. d., 2006. Tuning of fractional PID controllers with Ziegler Nichols Type Rules. *Signal Processing*, 86(10), pp. 2771-2784.
- Wang, L., 2009. *Model Predictive Control System Design and Implementation with Matlab*. London: Springer Verlag.
- Zhuang, D., Yu, F. & Lin, Y., 2008. Evaluation of a Vehicle Directional Control with a Fractional Order PD Controller. *International Journal of Automotive Technology*, 9(6), pp. 679-685.