



Adaptive Tuning of Model Predictive Control Parameters Based on Analytical Results

T. Gholaminejad^{1*}, A. Khaki-Sedigh¹, P. Bagheri²

¹ Department of Electrical Engineering, K. N. Toosi University of Technology, Tehran, Iran

² Control Engineering Department Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran

ABSTRACT: In dealing with model predictive controllers (MPC), controller tuning is a key designing step. Various tuning methods are proposed in the literature which can be categorized as heuristic, numerical and analytical methods. Among the available tuning methods, analytical approaches are more interesting and useful. This paper is based on a proposed analytical MPC tuning approach for plants which can be approximated by first-order plus dead-time models. The performance of such methods fails to deal with unknown or time-varying parameter plants. To overcome this problem, adaptive MPC tuning strategies are practical alternatives. The adaptive MPC tuning approach proposed in this paper is based on on-line identification and analytical tuning formulas. Simulation results are used to show the effectiveness of the proposed methodology. Also, a comparison of the proposed adaptive tuning method with a well-known online tuning method is presented briefly which shows the superiority of the proposed adaptive tuning method.

Review History:

Received: 10 November 2016

Revised: 2 June 2017

Accepted: 11 June 2017

Available Online: 16 June 2017

Keywords:

Adaptive model predictive control
analytical tuning
first order plus dead time models

1- Introduction

Model Predictive Control (MPC) is a well-established control technique used in advanced process control systems. Successful applications of MPC have been reported in many papers [1, 2]. MPC has several tuning parameters that must be appropriately tuned for acceptable closed-loop performances. Conventional tuning parameters of MPC are the prediction and control horizons and the weighted matrices. These parameters can significantly influence both the closed-loop performance and stability characteristics and therefore have been extensively studied in many research papers [3]. For example, Dynamic Matrix Control (DMC) tuning equations are developed in [4, 5] based on the analysis of variance. There is a tendency for closed-form tuning equations based on analytical methods, since they can be easily implemented and are also useful in the closed-loop analysis. In [6], an analytical MPC tuning approach is presented. Considering First Order Plus Dead Time (FOPDT) model of the real system, the closed-loop transfer function is derived, and it is shown that the maximum achievable performance can be obtained by the control horizon of one and two. Also, this approach is extended to unstable plants with the fractional delay [7] and multivariable plants [8]. However, these methods are inappropriate for unknown or time-varying plants. Adaptive tuning algorithms are suitable for these cases. In [9], an intuitive on-line MPC tuning strategy is presented, where a linear approximation between the output predictions and the MPC tuning parameters is utilized. In this set-up, the predicted outputs and the weights on the predicted outputs error and control action are related by sensitivity expressions which have been employed for automatic online adjustments of the parameters. In [9], a fuzzy logic-based auto-tuning

strategy is provided for model predictive controllers, and in [15] a similar strategy is suggested for non-linear MPC. In [16], a numerical auto-tuning algorithm is developed for DMC without constraint considerations. In this approach, Genetic Algorithm (GA) and Multi-Objective Fuzzy Decision-Making (MOFDM) strategies are combined to determine the optimal tuning parameters, where an objective function is calculated for each tuning parameter set. Then, based on the MOFDM algorithm, the best two sets of tuning parameters enter the next GA iteration while the worst set is eliminated. At the end of the GA cycle, the optimal tuning parameters are selected. In [17], it was proposed to auto-tune the GPC based on the characteristics of a second-order process such as rise time, settling time and overshoot. In this scheme, the auto-tuned GPC was used in a programmable logic controller. Finally, [14] proposed an MPC auto-tuning method to achieve the minimum variance output. This requirement is performed by forcing the MPC system to track its optimal closed-loop bandwidth. In this paper, an adaptive tuning algorithm for MPC based on the results of [6] is developed. It is shown that the proposed method can be used for all “unknown”, “time-varying”, “stable” systems with arbitrary order which can be approximated by an FOPDT model in an acceptable approximation. Note that many industrial processes can be sufficiently described by FOPDT models [18]. Also note that it is supposed that the variation of the systems is such that the robust control rules are not efficient.

The present paper is organized as follows. The MPC formulation and the tuning strategy given by [6] are briefly introduced in section II. A short study of the Recursive Least Square (RLS) method is provided in section III. Then, the proposed adaptive MPC tuning algorithm is developed and tested via simulation examples. Then, a comparison of the

Corresponding author, E-mail: 9205044@ee.kntu.ac.ir

proposed adaptive tuning method with a well-known online tuning method is presented briefly. Finally, the conclusion of the paper is given.

2- Analytically MPC tuning approach [6]

In [6], a standard structure in state-space form has been presented to formulate MPC tuning problem for FOPDT model of a real plant. In this section, this approach is briefly studied. Consider the following stable FOPDT model of a plant,

$$G_p(s) = \frac{ke^{-\theta_d s}}{\tau s + 1} \tag{1}$$

using the sampling time T_s , the discrete time model is:

$$G_p(z^{-1}) = \frac{k(1-a)z^{-d-1}}{1-az^{-1}} \tag{2}$$

where $a = e^{-T_s/\tau}$ and the dead time is considered to be an integer multiple of the sampling time, i.e. $d = \theta_d/T_s$. The typical finite optimal control problem for MPC is

$$\begin{aligned} \min_{\Delta u(n)} & (\mathbf{w}(n) - \mathbf{y}_p(n))^T \mathbf{Q}(\mathbf{w}(n) - \mathbf{y}_p(n)) + (\Delta \mathbf{u}(n))^T \mathbf{R}(\Delta \mathbf{u}(n)) \\ \text{s.t. } & u_{\min} \leq u(n+i|n) \leq u_{\max}, \quad i = 0, 1, \dots, M-1 \\ & y_{\min} \leq y_p(n+j|n) \leq y_{\max}, \quad j = N_1, N_1+1, \dots, N_2 \end{aligned} \tag{3}$$

where

$$\begin{aligned} \mathbf{w}(n) &= [w(n) \ w(n) \ \dots \ w(n)]_{P \times 1}^T \\ \mathbf{y}_p(n) &= \begin{bmatrix} \hat{y}_p(n+N_1|n) \\ \hat{y}_p(n+N_1+1|n) \\ \vdots \\ \hat{y}_p(n+N_2|n) \end{bmatrix}_{P \times 1}, \quad \Delta \mathbf{u}(n) = \begin{bmatrix} \Delta u(n) \\ \Delta u(n+1) \\ \vdots \\ \Delta u(n+M-1) \end{bmatrix}_{M \times 1} \end{aligned} \tag{4}$$

$$\mathbf{Q} = \text{diag}\{q_1, q_2, \dots, q_P\}, \quad \mathbf{R} = k^2(1-a)^2 \text{diag}\{r_1, r_2, \dots, r_M\}$$

and $N_1 = d+1$, $N_2 = d+P$; P is the prediction horizon, M is the control horizon. $w(n)$ is the desired reference signal; $\hat{y}_p(\cdot|n)$ is the predicted value of the plant output at the instance n and $\Delta u(n)$ is the control effort at time n . Also \mathbf{Q} and \mathbf{R} are the weight matrices of the cost function (3), $q_1 = 1$ and $\text{diag}\{\}$ is a diagonal matrix. It is known in the MPC tuning problem that the weight matrices are more dominant on system performance than other tuning parameters [19]. Thus, the weights on the cost function (3), namely, q_2, q_3, \dots, q_P and r_1, r_2, \dots, r_M are chosen as the tuning parameters. In [6], using the state-space representation of the augmented model with an integrator, the future values of the model output are as follows (Note that the integrator must be considered in order to guarantee the elimination of the step disturbance and steady-state error for step input).

$$\mathbf{y}(n) = \mathbf{F}\mathbf{x}(n) + \mathbf{S}\Delta \mathbf{u}(n) \tag{5}$$

where

$$\begin{aligned} \mathbf{F} &= [\mathbf{F}_1 \ \mathbf{F}_2 \ \dots \ \mathbf{F}_d \ \mathbf{F}_{d+1} \ \mathbf{F}_{d+2}] \\ \mathbf{F}_1 = \mathbf{F}_2 = \dots = \mathbf{F}_d &= \mathbf{0}_{P \times 1}, \quad \mathbf{F}_{d+2} = \mathbf{I}_{P \times 1} \\ \mathbf{F}_{d+1} &= a \begin{bmatrix} 1 \\ 1+a \\ \vdots \\ 1+a+\dots+a^{P-2}+a^{P-1} \end{bmatrix}_{P \times 1}, \quad \mathbf{y}(n) = \begin{bmatrix} \hat{y}(n+N_1|n) \\ \hat{y}(n+N_1+1|n) \\ \vdots \\ \hat{y}(n+N_2|n) \end{bmatrix}_{P \times 1} \\ \mathbf{S} &= k(1-a) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1+a & 1 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1+a+\dots+a^{P-1} & 1+a+\dots+a^{P-2} & \dots & 1+a+\dots+a^{P-M} \end{bmatrix}_{P \times M} \end{aligned} \tag{6}$$

where $\hat{y}(\cdot|n)$ is the predicted value of the model output calculated at the instance n and $\mathbf{1} = [1 \ 1 \ \dots \ 1]^T$. In the case of no active constraints, the optimal control effort solution is

$$\begin{aligned} \Delta \mathbf{u}(n) &= (\mathbf{R} + \mathbf{S}^T \mathbf{Q} \mathbf{S})^{-1} \mathbf{S}^T \mathbf{Q} (\mathbf{1}_{P \times 1} w(n) \\ &- \mathbf{F}_{d+1} \Delta \hat{\mathbf{y}}_p(n+d) - \mathbf{1}_{P \times 1} \hat{\mathbf{y}}_p(n+d)) \end{aligned} \tag{7}$$

where $\hat{\mathbf{y}}_p(n+d)$ is the real output prediction. We have

$$\begin{aligned} \hat{y}_p(n+d) &= a^d y(n) + k(1-a)(a^{d-1}u(n-d) + \dots \\ &+ au(n-2) + u(n-1)) + b(n) \end{aligned} \tag{8}$$

and

$$b(n) = y_p(n) - y(n), \tag{9}$$

where $b(n)$ can be considered as the disturbance or uncertainty term. In [6], two gains have been defined as

$$\begin{aligned} \mathbf{K}_x &= [\mathbf{K}_{x1} \ \mathbf{K}_{x2} \ \dots \ \mathbf{K}_{xM}]^T = (\mathbf{R} + \mathbf{S}^T \mathbf{Q} \mathbf{S})^{-1} \mathbf{S}^T \mathbf{Q} \mathbf{F}_{d+1} \\ \mathbf{K}_y &= [\mathbf{K}_{y1} \ \mathbf{K}_{y2} \ \dots \ \mathbf{K}_{yM}]^T = (\mathbf{R} + \mathbf{S}^T \mathbf{Q} \mathbf{S})^{-1} \mathbf{S}^T \mathbf{Q} \mathbf{1}_{P \times 1} \end{aligned} \tag{10}$$

Finally, the control signal at the time n is computed by

$$\Delta u(n) = \mathbf{K}_{y1} (w(n) - \hat{y}_p(n+d)) - \mathbf{K}_{x1} \Delta \hat{\mathbf{y}}_p(n+d) \tag{11}$$

Note that according to the predictive control rules, since the control effort at time n is sent to the process while the next control efforts calculated are rejected, in (11) the first components of the feasible gains \mathbf{K}_x and \mathbf{K}_y are only used. Then, by introducing the feasible gain concept, tuning formulas can be derived for stable systems. Furthermore, the area of feasible gains is characterized with related theorems in [6], and it is shown that the closed-loop plant with control horizon of one or two can achieve any feasible performance that can be delivered by MPC for a FOPDT model.

For the control horizon of one, by selecting r and q_P in (4) as tuning parameters, desired feasible gains \mathbf{K}'_{xd1} and \mathbf{K}'_{yd1} satisfy the following inequalities [6],

$$0 < \mathbf{K}'_{yd1}, \quad 0 < \mathbf{K}'_{xd1} < a, \quad \frac{1}{a} \frac{1}{X_1} < \frac{\mathbf{K}'_{yd1}}{\mathbf{K}'_{xd1}} < \frac{1}{a} \frac{Y_2}{X_2}, \tag{12}$$

where

$$\mathbf{K}'_{x1} = k(1-a)\mathbf{K}_{x1}, \quad \mathbf{K}'_{y1} = k(1-a)\mathbf{K}_{y1}, \tag{13}$$

and tuning formulas for achieving these gains are [6]

$$q_P = \frac{Y_2 - \frac{\mathbf{K}'_{yd1}}{\mathbf{K}'_{xd1}} a X_2}{X_1 \left(\frac{\mathbf{K}'_{yd1}}{\mathbf{K}'_{xd1}} a X_1 - 1 \right)}, \quad r = \frac{a - \mathbf{K}'_{xd1}}{\mathbf{K}'_{xd1}} (X_2 + q_P X_1^2), \tag{14}$$

where

$$\begin{aligned} X_1 &= 1 + a + \dots + a^{P-1}, \\ X_2 &= 1 + (1+a)^2 + (1+a+a^2)^2 + \dots + (1+a+\dots+a^{P-2})^2, \\ Y_2 &= 1 + (1+a) + (1+a+a^2) + \dots + (1+a+\dots+a^{P-2}). \end{aligned} \tag{15}$$

It is clearly observed that the proposed MPC tuning method [6] is effective when the plant model is known. However, in the case of unknown or time-varying plants, tuning formulas given by [6] might not guarantee the desired performance or stability of the closed-loop plant. A solution for this problem is an online estimation of the model parameters and using the estimated parameters in the tuning equations in each sample time.

3- Adaptive MPC Tuning Algorithm

Online identification of dynamic systems provides the model of system at each sampling time. This estimated model can then be used in the MPC tuning rules.

3- 1- RLS Formulation

Consider the following model of the system

$$A(z^{-1})y(n) = B(z^{-1})u(n),$$

where

$$A(z^{-1}) = 1 + a_1z^{-1} + \dots + a_lz^{-l},$$

$$B(z^{-1}) = b_1z^{-1} + b_2z^{-2} + \dots + b_mz^{-m}.$$

This model can be written as

$$y(n) = \Phi^T(n-1)\theta, \quad (\text{Regression Model})$$

where

$$\Phi(n-1) = [-y(n-1) \ -y(n-2) \ \dots \ -y(n-l) \ u(n-1) \ \dots \ u(n-m)]^T$$

$$\theta = [a_1 \ a_2 \ \dots \ a_l \ b_1 \ \dots \ b_m]^T$$

Then, RLS algorithm can be stated briefly as follows

$$\hat{\theta}(n) = \hat{\theta}(n-1) - \mathbf{K}(n)[y(n) - \Phi^T(n)\hat{\theta}(n-1)] \quad (16)$$

where

$$\mathbf{K}(n) = \mathbf{P}(n-1)\Phi(n)[\mathbf{I} + \Phi^T(n)\mathbf{P}(n-1)\Phi(n)]^{-1} \quad (17)$$

and

$$\mathbf{P}(n) = [\mathbf{I} - \mathbf{K}(n)\Phi^T(n)]\mathbf{P}(n-1) \quad (18)$$

As for an unknown time-varying plant the polynomials $A(z^{-1})$ and $B(z^{-1})$ and, thus, the parameters vector θ are unknown and just the vector of inputs and outputs $\Phi(n)$ and the output $y(n)$ are known, by selecting the proper initial conditions (initial θ and the covariance matrix \mathbf{P}), the unknown parameters vector θ can be obtained through the relations (16), (17) and (18).

Note that in this paper, the time delay is estimated via the proposed method in [16].

3- 2- Application of RLS in Tuning formulas Proposed in [6]
Consider the following arbitrary order time-varying real plant:

$$G_p(z^{-1}) = z^{-d-1} \frac{\hat{B}(z^{-1})}{\hat{A}(z^{-1})}, \quad (19)$$

where supposed that it can be approximated by FOPDT model (2), and $\hat{B}(z^{-1})$ and $\hat{A}(z^{-1})$ are the polynomials which are estimated via the RLS method mentioned in part 3-1 which are

$$\hat{B}(z^{-1}) = \hat{k}(1-\hat{a}), \quad \hat{A}(z^{-1}) = 1 - \hat{a}z^{-1}.$$

Thus, the output of plant (19) can be defined as

$$y_p(n) = z^{-d-1} \frac{\hat{B}(z^{-1})}{\hat{A}(z^{-1})} u(n). \quad (20)$$

To obtain the desired closed-loop response, the predictive control signal (11) can be used, which can be rewritten as

$$u(n) = K_{y1} \frac{1}{1-z^{-1}} w(n) - \frac{1}{1-z^{-1}} (K_{y1} + K_{x1}(1-z^{-1})) \hat{y}_p(n+d). \quad (21)$$

Generally, the control signal is

$$u(n) = \frac{T(z^{-1})}{R(z^{-1})} w(n) - \frac{S(z^{-1})}{R(z^{-1})} \hat{y}_p(n+d), \quad (22)$$

where the predicted output $\hat{y}_p(n+d)$ can be obtained by (8). In the above relation, we have

$$R(z^{-1}) = 1 - z^{-1}, \quad T(z^{-1}) = K_{y1},$$

$$S(z^{-1}) = (K_{x1} + K_{y1}) - K_{x1}z^{-1}. \quad (23)$$

Now, from (20) and (22), we have

$$y_p(n) = \frac{z^{-d-1}B(z^{-1})T(z^{-1})}{A(z^{-1})R(z^{-1}) + S(z^{-1})B(z^{-1})z^{-1}} w(n) \quad (24)$$

Substituting $R(z^{-1})$, $T(z^{-1})$ and $S(z^{-1})$ in (24) gives

$$y_p(n) = \frac{z^{-d-1}\hat{k}(1-\hat{a})K_{y1}}{\Delta_{cl}} w(n), \quad (25)$$

where

$$\Delta_{cl} = 1 + (-1 - \hat{a} + \hat{k}(1-\hat{a})K_{x1} + \hat{k}(1-\hat{a})K_{y1})z^{-1} + (\hat{a} - \hat{k}(1-\hat{a})K_{x1})z^{-2}. \quad (26)$$

Now, consider the following desired model

$$G_d(s) = \frac{w_n^2 e^{-\theta s}}{s^2 + 2\zeta w_n s + w_n^2}, \quad (27)$$

where ζ_d and w_n are the damping ratio and the natural frequency of the desired system, respectively.

Please note that since in this method the plant is approximated by an FOPDT model and due to the structure of the predictive control signal (21), the closed-loop system is a second-order model similar to (27). Also, due to the relation of the desired conditions, such as settling time, rise time and overshoot, and the coefficients of transfer function (27), it can be concluded that for each desired output, the desired model (27) can be exactly specified. Also, note that usually, desired conditions of output can be approximated through a second-order time-delay model. But if in a case, a second-order time-delay model is not an accurate approximation, the proposed method in this paper is not usable. The corresponding discrete time model of (27) is

$$G_d(z^{-1}) = \frac{w_n e^{-\zeta w_n T_s} \sin(w_n \sqrt{1-\zeta^2} T_s) z^{-d-1}}{\sqrt{1-\zeta^2} (1 - 2e^{-\zeta w_n T_s} \cos(w_n \sqrt{1-\zeta^2} T_s) z^{-1} + e^{-2\zeta w_n T_s} z^{-2})} \quad (28)$$

which can be rewritten as

$$G_d(z^{-1}) = \frac{z^{-d-1} B_d(z^{-1})}{A_d(z^{-1})}. \quad (29)$$

Hence, the desired output is

$$y_d(n) = \frac{z^{-d-1}B_d(z^{-1})}{A_d(z^{-1})}w(n). \quad (30)$$

Now, according to (24) and (30), the closed-loop system and the desired model have the same responses if

$$B(z^{-1})T(z^{-1}) = B_d(z^{-1}), \quad (31)$$

$$A(z^{-1})R(z^{-1}) + S(z^{-1})B(z^{-1})z^{-1} = A_d(z^{-1}).$$

Also,

$$\hat{B}(z^{-1}) = \hat{b}_0, B_d(z^{-1}) = b_d, \quad (32)$$

where

$$\hat{b}_0 = \hat{k}(1-\hat{a}). \quad (33)$$

Furthermore, according to (25), (26), (30) and (31), the polynomial $A_d(z^{-1})$ is obtained as

$$A_d(z^{-1}) = 1 + a_{d1}z^{-1} + a_{d2}z^{-2} = 1 + (-1-\hat{a} + \hat{k}(1-\hat{a})K_{x1} + \hat{k}(1-\hat{a})K_{y1})z^{-1} + (\hat{a} - \hat{k}(1-\hat{a})K_{x1})z^{-2}. \quad (34)$$

Thus, according to (32) to (34), the following relations for K_{x1} and K_{y1} are derived

$$K_{x1} = (\hat{a} - a_{d2}) / \hat{b}_0, K_{y1} = b_d / \hat{b}_0. \quad (35)$$

Now since the model reference parameters a_{d2} and b_d are known and the system parameters are estimated in each sampling time through RLS algorithm, the feasible gains K_{x1} and K_{y1} are obtained from (35) in each sampling time. Note that after identifying the time-varying system parameters, if the obtained gains from (35) were not feasible, i.e. they did not satisfy (12), we can change the model reference parameters a_{d2} and b_d within the desired performance characteristics. It can be obtained in a way that to reach feasible gains K_{x1} and K_{y1} we solve multiple LMIs [15]. These LMIs can be stated from (12) as it follows.

1. $F_1(\mathbf{x}): a - K'_{x1} > 0,$
2. $F_2(\mathbf{x}): K'_{x1} > 0,$
3. $F_3(\mathbf{x}): K'_{y1} > 0,$
4. $F_4(\mathbf{x}): -aK'_{y1}X_2 + K'_{x1}Y_2 > 0,$
5. $F_5(\mathbf{x}): -K'_{x1} + aK'_{y1}X_1 > 0.$

Thus, we have

$$F(\mathbf{x}) := \begin{bmatrix} F_1(\mathbf{x}) & 0 & 0 & 0 & 0 \\ 0 & F_2(\mathbf{x}) & 0 & 0 & 0 \\ 0 & 0 & F_3(\mathbf{x}) & 0 & 0 \\ 0 & 0 & 0 & F_4(\mathbf{x}) & 0 \\ 0 & 0 & 0 & 0 & F_5(\mathbf{x}) \end{bmatrix} > 0,$$

where a, X_1, X_2 and Y_2 vary in each sampling time and \mathbf{x} is the vector of unknown feasible gains K'_{x1} and K'_{y1} . In other words, for each sampling time, the above linear matrix inequalities are solved through the LMI toolbox in MATLAB. Then, the tuning parameters in (14) can be obtained from the resulted gains. The steps involved in the proposed adaptive tuning method of MPC can be summarized as the following algorithm,

1. Obtain the parameters of the approximated FOPDT

discrete model using RLS.

- Estimate the dead time of the process using the method proposed in [16].
 - Apply the obtained delay to the unknown plant (19) and identify the approximated FOPDT model parameters using RLS (equations (16), (17) and (18)).
2. Calculate the gains in (35) for the estimated parameters of step 1.
 - If the calculated gains of (35) are feasible, go to the next step; otherwise, obtain the desired gains by solving the LMIs (36).
 - If the obtained gains from solving (36) do not generate the model reference parameters a_{d2} and b_d within the desired performance characteristics according to (35), the obtained gains are not feasible and the desired response is not reachable.
 3. Obtain the control signal and apply it to the system.
 4. Repeat the above steps for each sampling time.

The general structure of the proposed adaptive MPC tuning strategy is shown in Fig. 1.

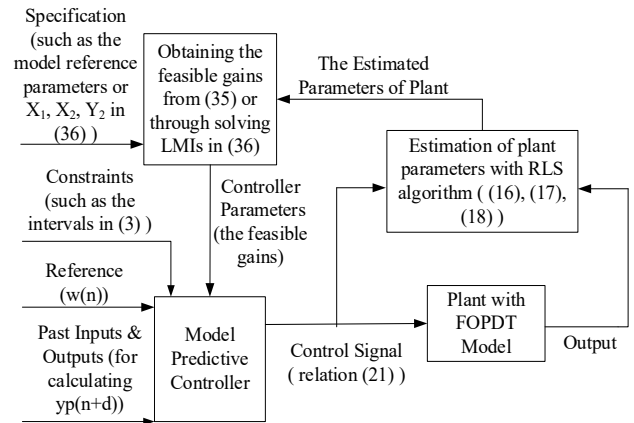


Fig. 1. Block diagram of the proposed method for adaptive tuning of MPC.

4- Simulation Results

In this section, three examples are used to verify the proposed adaptive tuning method. In the first example, a FOPDT plant is used with varying parameters. In the second example, the nonlinear pH process is considered and, finally, in the third example, a second-order plant whose parameters vary is considered.

4- 1- Example 1

Consider the following FOPDT process [6]

$$G_p(s) = \frac{1.5e^{-4s}}{10s+1}.$$

Let $T_s = 1$ sec. The related discrete transfer function is

$$G_p(z^{-1}) = \frac{0.1425z^{-5}}{1-0.905z^{-1}}.$$

Let the desired closed-loop performance has a maximum overshoot of 10% and a settling time of 14 sec up to 24 sec.

A transfer function that satisfies these requirements can be

$$G_d(z^{-1}) = \frac{0.1z^{-5}}{1 - 1.505z^{-1} + 0.6048z^{-2}},$$

which is selected as the desired reference model. It is assumed that the plant parameters are unknown. Assume a 30% change in the parameter k at the time 300 sec and also a 50% change for parameter a at the time 800 sec occur. First, we apply the non-adaptive MPC tuning [6]. Fig 2 shows the results. We can see that although for the 30% variation of the parameter k in 300 sec, tracking is acceptable (and only we have a small overshoot for a short time) but for 50% variation of parameter a in 800 sec, the non-adaptive MPC cannot track the reference signal.

Now, we apply the proposed adaptive MPC tuning strategy, and the results are shown in Figs 3, 4 and 5. In Fig. 3 it is shown that, the parameters of the system have converged to their real values after few seconds. Also, it can be seen that in Figs. 3a and 3b, the sudden variation of each parameter k and a temporarily affects the other one. In this example, forgetting factor is selected as 0.96. In Fig. 4 the controller parameters are shown and we can see that they have positive values and are therefore acceptable. Since parameters r and q_p according to (14) only include the system parameter a , they have not been affected by the change of parameter k , and only their estimations were affected within the transient time. Note that in this example, because the model of the plant is known (while in real applications the model is unknown with unknown variation), we can obtain the accurate controller parameters. Thus, in Fig 4, it has been shown that through the proposed adaptive MPC tuning method, the accurate controller parameters can be followed by any variation in the plant model parameters. Fig. 5 shows the closed-loop responses for the proposed adaptive MPC tuning method. The efficiency of the proposed method is shown in this figure and it can be seen that the sudden variation in 300 and 800 sec have been handled with very

small overshoots.

4- 2- Example 2 (The Nonlinear pH Process)

Now, the proposed method is tested via the nonlinear pH process [17]. This process is a good benchmark to evaluate controllers. The dynamic and static equations of pH process can be found in [17]. Assume the following FOPDT model for the pH process in an operating point

$$G_P(s) = \frac{0.9e^{-30s}}{100s + 1} \xrightarrow{T_s=0 \text{ sec}} G_P(z^{-1}) = \frac{0.08565z^{-4}}{1 - 0.9048z^{-1}}.$$

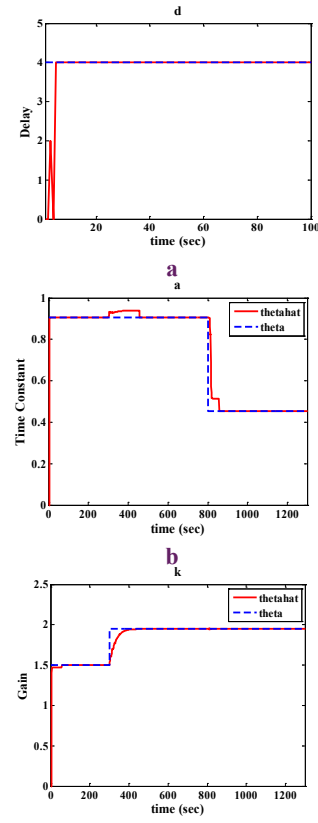


Fig 3. The estimated parameters of d , a and k , a: time delay d , b: time constant a , c: gain k .

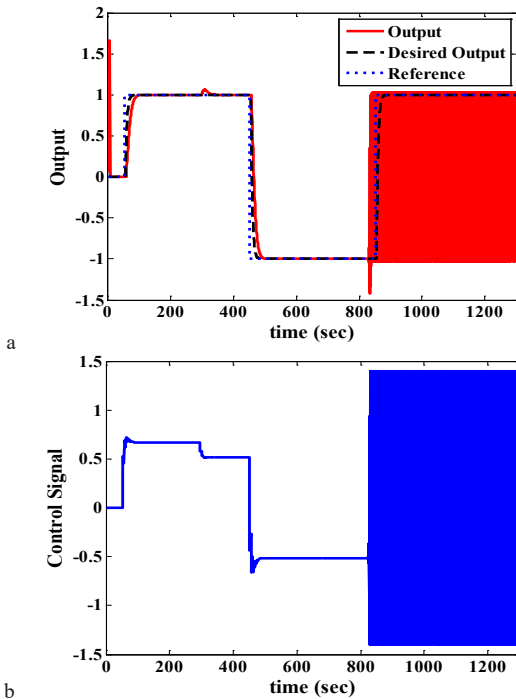


Fig 2. The closed-loop responses in example 1 (tuning method of [6]), a: Closed-loop Output, b: Control Signal.

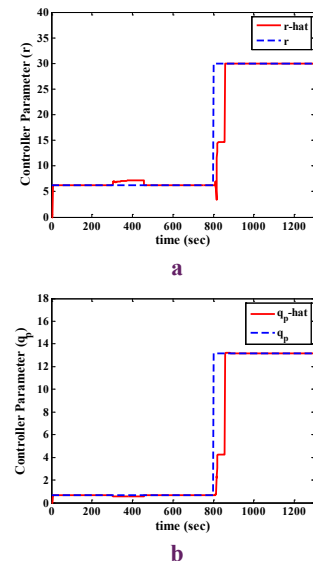


Fig 4. The estimated MPC tuning parameters, a: predictive controller parameter r , b: predictive controller parameter q_p .

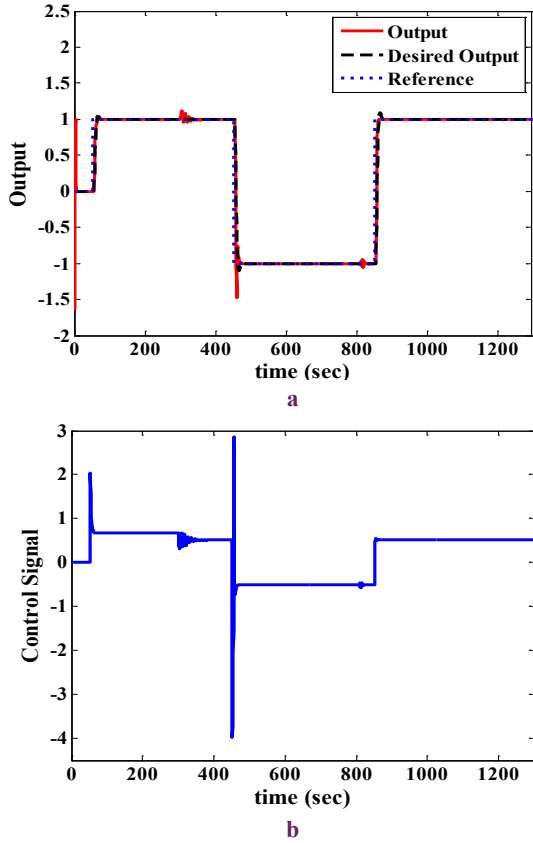


Fig 5. The closed-loop responses in example 1 (Adaptive tuning method), a: Closed-loop output, b: Control Signal.

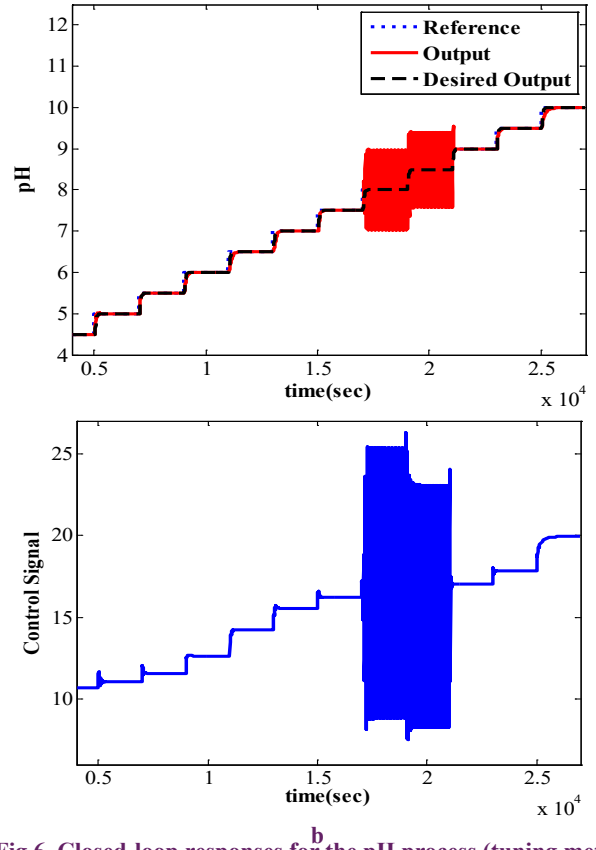


Fig 6. Closed-loop responses for the pH process (tuning method [6]), a: Closed-loop output, b: Control Signal.

Let K'_{xd1} and K'_{yd1} be selected as 0.7 and 0.148, respectively, using the results of [6]. Fig. 6 shows the system output and the control signal in the case of non-adaptive tuning based on [6] and it can be seen that for the above constant FOPDT model, the non-adaptive MPC proposed in [6] is unable to control the neutralization pH process for the pH values 8 and 8.5. Now consider the proposed tuning method. It should be noted that the indirect RLS method for adapting this process is sensitive to the system parameters initial values. Fig. 7 shows the closed-loop responses of the adaptive proposed tuning method. It is shown that tracking performance is good. The effectiveness of the proposed tuning strategy is evident after comparing Figs. 6 and 7, and the control problem at pH values 8 and 8.5 in the non-adaptive MPC is resolved.

4- 3- Example 3 (Higher Order System)

In this example, a high-order plant is considered [19],

$$G_p(s) = \frac{e^{-50s}}{(150s+1)(25s+1)},$$

where the corresponding discrete-time transfer function is

$$G_p(z^{-1}) = z^{-4} \frac{0.004378 + 0.02879z^{-1} + 0.005041z^{-2}}{1 - 1.482z^{-1} + 0.5203z^{-2}} \rightarrow$$

$$G_p(z^{-1}) = z^{-4} \frac{b_1 + b_2z^{-1} + b_3z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}}.$$

A proper approximated FOPDT model for this plant is given in [6] which shows that this high-order plant can be approximated by a FOPDT model with an acceptable error.

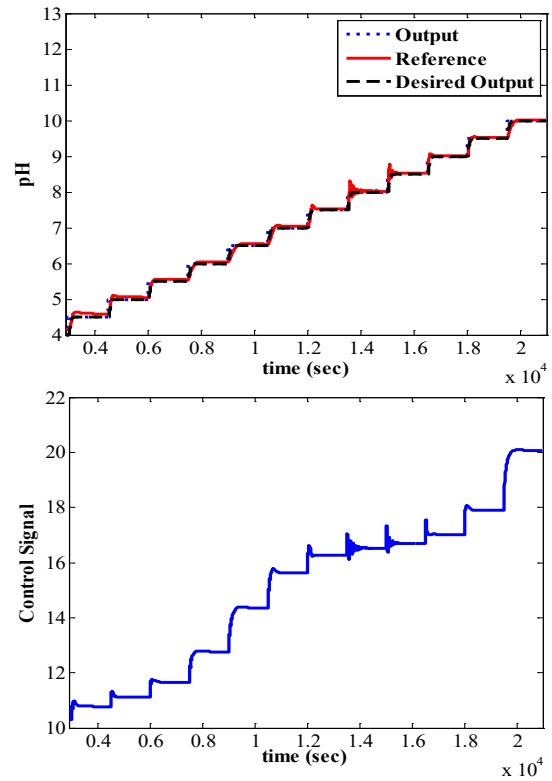


Fig 7. Closed-loop responses for pH process (Adaptive tuning method), a: Closed-loop output, b: Control Signal.

Hence, in the proposed adaptive MPC method an estimated FOPDT model is derived. It is assumed that all parameters varies as: a_1 multiplies in 0.9 at 800 sec., a_2 multiplies in 1.05 at 1200 sec., b_1 multiplies in 5 at 1600 sec., b_2 multiplies in 10 at 2100 sec., and b_3 multiplies in 0.05 at 2600 sec.

Fig. 8 shows the system output and the control signal for the non-adaptive tuning method of [6] and Fig. 9 shows the same plots for the proposed adaptive MPC tuning method in this paper. It is seen that for the variation of the denominator parameter a_1 at 800 sec., the response of the non-adaptive method is slow compared to the same response of the adaptive method. Also, at 2100 sec., with the variation of the parameter b_2 , the response of the non-adaptive method is unstable while the same response for the adaptive method is stable and yields a good tracking after some seconds. Thus, the effectiveness of the proposed method can be concluded for this high order time-varying plant.

4-4- Comparison of the proposed adaptive tuning method with a well-known online MPC tuning method

In this section, in order to investigate the advantages of the proposed tuning method, a comparison between the proposed method and an online tuning method [9] is presented. For example, consider that a 30% decrement in the parameter a and a 50% decrement in the parameter k occur at 800 sec. and also a 5% increment in the parameter a and a 20% increment in the parameter k occur at 1800 sec. Fig. 10 shows the

result of the comparison between the online tuning method in [9] with the proposed adaptive tuning method in this paper. It shows that the reaction of the online tuning method to the system dynamic changes is more than the proposed method in this paper and the obtained response from the method of [9] is not desirable. Also, note that in the proposed online tuning method in [9], the initial plant model must be available and the online method of [9] can be affected just by unpredicted disturbances and unknown model parameters variations, while the proposed method in this paper does not need the initial plant model.

Note that the proposed method in this paper can be used either in the case that the plant model is unknown and time-varying or in the case that an unknown disturbance is applied to input or output. In these two cases, the new parameters of the FOPDT model are identified in each sampling time and for the desired closed-loop output, the tuning parameters of the predictive controller are obtained.

5- Conclusion

In this paper, an adaptive MPC tuning method was proposed based on the results of an available analytical MPC tuning method. Using RLS algorithm, we tried to update the tuning and controller parameters. Furthermore, an idea of the LMI was implemented when the obtained gains from the RLS algorithm were not feasible. Thus, in an algorithm, the proposed adaptive MPC tuning method was provided.

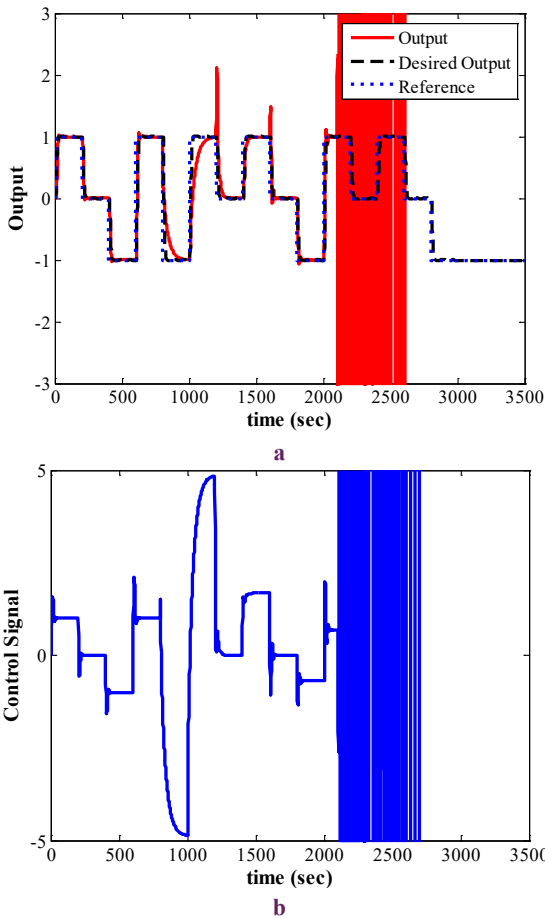


Fig 8. Closed-loop responses for the second-order plant in example 3 (tuning method [6]), a: Closed-loop Output, b: Control Signal.

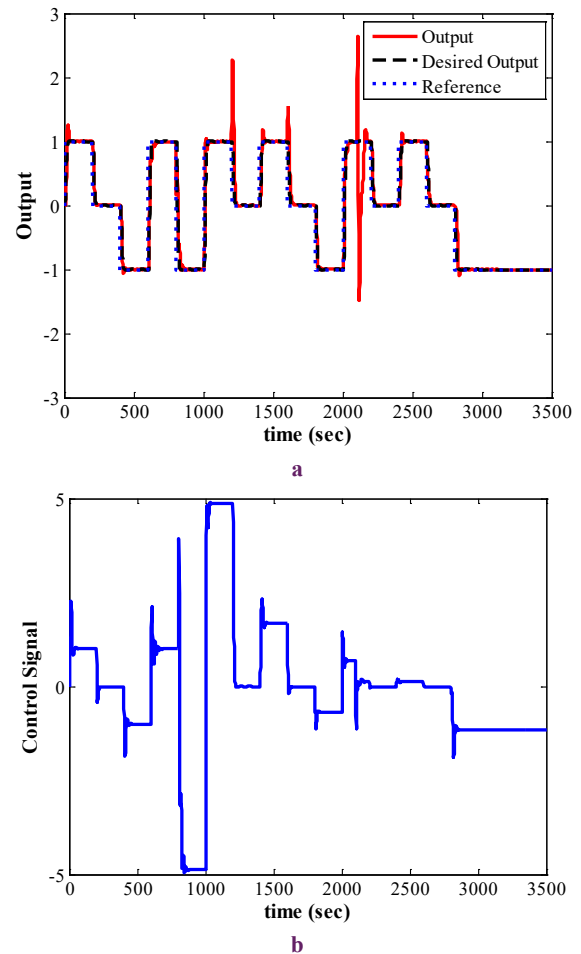


Fig 9. Closed-loop responses for the second-order plant in example 3 (proposed adaptive MPC tuning method in this paper), a: Closed-loop Output, b: Control Signal.

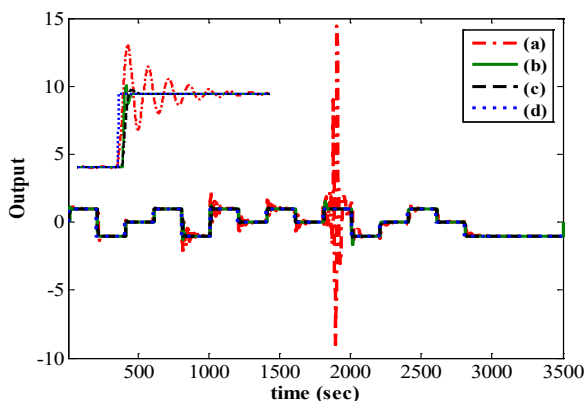


Fig 10. Closed-loop outputs for the FOPDT process of example A: (a) Output for the method of [9], (b) Output for the adaptive tuning method proposed in this paper, (c) Desired Output, (d) Reference input.

Simulation results evaluated the effectiveness of the proposed method. Finally a comparison of the proposed adaptive tuning method with a well-established online tuning method was briefly done.

In this paper, it was shown that the proposed adaptive tuning method can be used for all unknown time-varying arbitrary order systems which can be approximated by a FOPDT model. However, the proposed method is inefficient when the feasible gains cannot be reached for a special desired output. Also, since in the proposed method, the parameters of a FOPDT model must be estimated in each sampling time and the mentioned multiple LMIs must be solved, the calculation time for this method shall be reduced in a direct adaptive tuning case which will be considered in future studies by the authors.

References

- [1] M.L. Darby, M. Nikolaou, MPC: Current practice and challenges, 20(4) (2012) 328-342.
- [2] D.Q. Mayne, Model predictive control: Recent developments and future promise, 50(12) (2014) 2967-2986.
- [3] J.L. Garriga, M. Soroush, Model predictive control tuning methods: A review, 49(8) (2010) 3505-3515.
- [4] P. Bagheri, A. Khaki-Sedigh, Tuning of dynamic matrix controller for FOPDT models using analysis of variance, 44(1) (2011) 12319-12324.
- [5] P. Bagheri, A. Khaki-Sedigh, Robust tuning of dynamic matrix controllers for first order plus dead time models, 39(22) (2015) 7017-7031.
- [6] P. Bagheri, A. Khaki-Sedigh, K.J.I.C.T. Sedigh, Applications, Analytical approach to tuning of model predictive control for first-order plus dead time models, 7(14) (2013) 1806-1817.
- [7] P. Bagheri, A. Khaki-Sedigh, Closed form tuning equations for model predictive control of first-order plus fractional dead time models, 13(1) (2015) 73-80.
- [8] P. Bagheri, A. Khaki-Sedigh, An analytical tuning approach to multivariable model predictive controllers, 24(12) (2014) 41-54.
- [9] A. Al-Ghazzawi, E. Ali, A. Nouh, E. Zafiriou, On-line tuning strategy for model predictive controllers, 11(3) (2001) 265-284.
- [10] E. Ali, A. Al-Ghazzawi, On-line tuning of model predictive controllers using fuzzy logic, 81(5) (2003) 1041-1051.
- [11] E. Ali, Heuristic on-line tuning for nonlinear model predictive controllers using fuzzy logic, 13(5) (2003) 383-396.
- [12] J. Van der Lee, W. Svrcek, B. Young, A tuning algorithm for model predictive controllers based on genetic algorithms and fuzzy decision making, 47(1) (2008) 53-59.
- [13] G. Valencia-Palomo, J. Rossiter, Programmable logic controller implementation of an auto-tuned predictive control based on minimal plant information, 50(1) (2011) 92-100.
- [14] Q.N. Tran, J. Scholten, L. Ozkan, A. Backx, A model-free approach for auto-tuning of model predictive control, 47(3) (2014) 2189-2194.
- [15] S. Boyd, L. El-Ghaoui, E. Feron, V. Balakrishnan, Linear matrix inequalities in system and control theory, 85(4) (1997) 698-699.
- [16] C. Lynch, G. Dumont, Control loop performance monitoring, 4(2) (1996) 185-192.
- [17] M.A. Henson, D.E. Seborg, Adaptive nonlinear control of a pH neutralization process, 2(3) (1994) 169-182.
- [18] C. Bordons, E.F. Camacho, A generalized predictive controller for a wide class of industrial processes, 6(3) (1998) 372-387.
- [19] R. Shridhar, D.J. Cooper, A tuning strategy for unconstrained SISO model predictive control, 36(3) (1997) 729-746.

Please cite this article using:

T. Gholaminejad, A. Khaki-Sedigh, P. Bagheri, Adaptive Tuning of Model Predictive Control Parameters Based on Analytical Results, *AUT J. Mod. Simul.*, 50(2) (2018) 109-116.
DOI: 10.22060/miscj.2017.12143.5005

