The Journal of Engineering and Exact Sciences – jCEC, Vol. 05 N. 05 (2019)
journal homepage: https://periodicos.ufv.br/ojs/jcec
doi: 10.18540/jcecvl5iss5pp0408-0414
OPEN ACCESS – ISSN: 2527-1075

FAST ADAPTIVE GENERALIZED PREDICTIVE CONTROL FOR SYSTEMS WITH VARIABLE PARAMETERS

S. B. ROVEA, R. C. C. FLESCH*

[1] Federal University of Santa Catarina, Department of Automation and Systems, Florianópolis, Santa Catarina, Brazil

*Corresponding author. Federal University of Santa Catarina, Department of Automation and Systems, Florianópolis, Santa Catarina, Brazil, Phone: +55 (48) 3721-7600
e-mail address: rodolfo.flesch@ufsc.br (Rodolfo C. C. Flesch).*

ARTICLE INFO

Article history:
Received 2019-10-02
Accepted 2019-12-20
Available online 2019-12-20

palavras-chave
Controle Preditivo Generalizado
Controle Adaptativo
Controle Digital
Sistemas Embarcados

keywords
Generalized Predictive Control
Adaptive Control
Digital Control
Embedded Systems

ABSTRACT

This paper proposes a fast predictive control structure with online model update according to process parametric variations. The proposed controller is based on the Generalized Predictive Control (GPC) algorithm, but it integrates the recursive least squares identification method with a variable forgetting factor to estimate at each iteration the parameters of a linear structure model used for multi-step ahead prediction. For a system with constraints on the process variables, the resulting optimization problem of GPC is solved using quadratic programming based on the Alternate Direction Method of Multipliers, which allows the control signal to be obtained with small computational effort. In order to validate the proposed algorithm an experimental case study that considers the speed control of a direct current motor and the proposed controller embedded in a microcontroller STM32F303K8T6 is presented. Experimental results use as baseline the GPC with fixed model parameters and show that the proposed fast adaptive predictive control structure is able to keep almost the same transient response for all the considered operating points of the motor, while GPC presents high oscillations at operating conditions far from the one used to obtain the nominal model. Even though the proposed controller needs to solve two optimization problems at each sampling instant, it can run about 60 times in a second in the microcontroller used in this study.

1. INTRODUCTION

The Model Predictive Control (MPC) paradigm was introduced and has gained importance in the past few decades both in industry and academia (Clarke *et al*., 1987). MPC is a control strategy characterized by using an explicit process model to formulate an optimization problem whose objective is to minimize a cost function. In general, the aim is to minimize the distance between the predicted process output trajectory and the future reference trajectory and, at the same time, penalize the control signal effort. The model used in MPC can be linear or nonlinear. Although nonlinear models result in a better representation of the static and dynamic process characteristics, nonlinear approaches increase the computational effort, thus making it difficult to solve the optimization problem within the sampling time (Camacho and Bordons, 2007).

Generally, MPC is implemented using a linear model with fixed parameters. However, this approach is valid only for a specific system operating condition or when the process is not time-varying. An alternative is to continually estimate the plant model in order to improve the quality of closed-loop control, especially when the process presents parametric variations. This approach also allows the controller to be used without the need for identifying the system dynamics *a priori*, which can speed up the deployment of new MPC controllers.

The literature presents many studies which propose MPC formulations with online system identification. In a recent study, Zhu and Xiaohua (2016) proposed an adaptive updating law for estimating the parameters of a discrete-time linear model combined with constrained MPC. Radecki and Hencey (2017) presented a predictive thermal control with an online model estimation based on the unscented Kalman filter (UKF) for parameter and disturbance estimation. Fesharaki, Kamali and Sheikholeslam (2017) proposed a new approach to adapt at each sampling instant the model of a direct current motor in order to improve performance in the closed-loop predictive control. Jabbour and Mademlis (2019) proposed a new method that can automatically tune and adjust the controller parameters according to the operating point at which the plant used for prediction in an MPC operates. Koo *et al*. (2019) presented a self-tuning adaptive structure combining the recursive least squares algorithm based on Kalman filter with MPC strategy to control the electron density of plasma in real time. A new design of adaptive MPC for wind speed control with uncertainties in system parameters is proposed in Elsisi (2019). Yang *et al.* (2019) presented an indoor climate predictive control with an adaptation scheme that incorporates an online estimation of uncertainty coefficients in the model of the plant. Although these works combine recursive identification methods integrated with the resulting MPC problem to solve linear model parameters change, they consider a conventional state-space formulation. One drawback of using these formulations is associated with the need for a state observer when not all the states used for the model estimation are measured. In addition, in industry it is more common to find MPCs which make use of different representations, such as Generalized Predictive Control (GPC), which considers predictions based on the controlled autoregressive integrated moving average model, and Dynamic Matrix Control (DMC), which considers a model based on the step-response coefficients of the plant. Thus, most of the formulations with online identification from literature cannot be directly used in some practical applications which already exist in industry.

Although most of MPC applications are limited to the process industry, which is characterized by slow dynamics, it is being more and more used in different applications, including those with fast dynamics. Thus, efficient computation of the resulting optimization problem taking into account constraints on the process must be considered. Short (2012) presented a time-varying parameter identification scheme for GPC algorithm implemented in an embedded system, but the method is limited to plants which can be described as First-Order Plus Dead Time (FOPDT) models subjected to constraints of amplitude and rate on the plant input. Li et al. (2015) presented a framework to reduce the online computational burden associated with constrained MPC in order to compute in real-time an efficient fuel economy applied to cruise control of vehicles. Zheng, Negenborn, and Lodewijks (2017) also presented an efficient algorithm which can solve in parallel the optimization problem of MPC for a distributed cooperative system. The fast approach is implemented using the Alternate Direction Method of Multipliers (ADMM) in order to improve convergence rates. Saraf and Bemporad (2017) introduced a fast method to solve the MPC problem in real time for multivariable discrete-time linear systems under input and output constraints. Even though these works can integrate MPC with efficient computation of the control signal under constraints, they are not represented in GPC and DMC formulations. Recently, Peccin *et al*. (2018) proposed a solver based on ADMM which efficiently computes the control signal for GPC and DMC formulations, but the method considers plants with constant coefficients.

This work is an extension of Rovea and Flesch (2019) and proposes an MPC formulation which uses the recursive least squares (RLS) method to update at each sampling time the model of the plant used by the GPC algorithm for prediction. In this case, the model used for prediction is linear at each sampling instant, which allows the use of fast computation techniques even if the model is not constant from sample to sample. Since the model identification is done based on closed-loop data, the level of excitation of the process input is determined and the variable forgetting factor of the plant identification is automatically tuned online. The resulting quadratic programming problem with process constraints is efficiently solved using ADMM. As a result, the proposed controller is able to be used in plants which can be represented as a linear system in the parameters of any order with variable coefficients, subjected to any type of affine constraints, and with fast dynamics. The proposed controller was implemented in a microcontroller and experimentally evaluated.

The rest of this paper is organized as follows. Section 2 presents the proposed fast adaptive generalized predictive control structure and its formulation. Section 3 presents the main algorithm to implement the proposed method in an embedded system. Section 4 presents the experimental case study of speed control of a direct current motor. Section 5 summarizes the conclusions of the present work.

2. PROPOSED FAST ADAPTIVE GENERALIZED PREDICTIVE CONTROL

An overview of the proposed Fast Adaptive Generalized Predictive Control (FAGPC) structure is shown in Figure 1. It uses a linear model of the plant, based on a transfer function, to predict the future behavior of the system output, but the coefficients of the model are adjusted at each sampling instant based on the past behavior of the plant. For that purpose, an RLS estimator with forgetting factor is used. Finally, ADMM is used to determine in a quick manner a sub-optimal control action to be applied to the plant input. In Figure 1, for a certain time instant *k*, *w* is the reference, *Δu* is the control increment, *u* is the control signal, *d* is a disturbance, *y* is the output of the plant, and z^{-1} represents a unit delay. Details about the main blocks of the proposed structure are presented in the sections below.

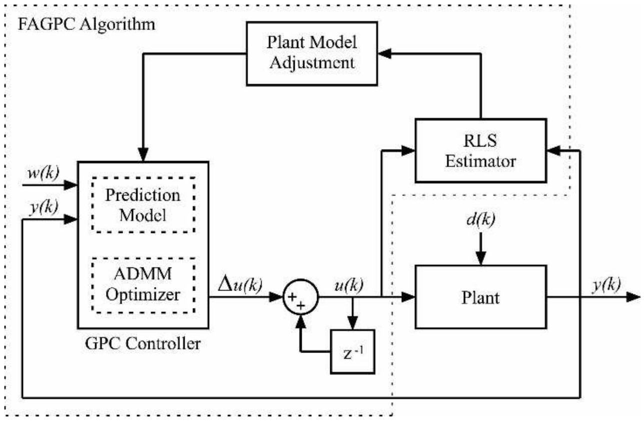

Figure 1 – Block diagram of the proposed fast adaptive control system.

2.1 GPC Formulation

The basic idea of GPC operation is to minimize a cost function that considers reference tracking error and control effort, which is given in (1) for the SISO case (Normey-Rico and Camacho, 2007; Camacho and Bordons, 2007),

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \delta(j)[\hat{y}(k+j|k) - w(k+j)]^2 + \sum_{j=1}^{N_u} \lambda(j)[\Delta u(k+j-1)]^2, \qquad (1)$$

where $\hat{y}(k+j|k)$ represents the prediction of the plant output for $k+j$ obtained with information up to the time instant k, $w(k+j)$ is the future reference, $N_1 = d+1$ and $N_2 = d+N$ are the minimum and maximum values of the prediction horizon, N_u is the control horizon, $\Delta u(k+j-1)$ represents each of the control increments within the control horizon, and $\delta(j)$ and $\lambda(j)$ are the weighting sequences of the reference tracking and control effort, respectively. The prediction sequence over the prediction horizon can be grouped in a vector as

$$\hat{\boldsymbol{y}}(k) = [\hat{y}(k+N_1|k) \ldots \hat{y}(k+N_2|k)]^T, \qquad (2)$$

which can be split into a forced and a free response, as in Equation (3)

$$\hat{\boldsymbol{y}}(k) = \boldsymbol{G}(k)\boldsymbol{\Delta u}(k) + \boldsymbol{f}(k), \qquad (3)$$

where $\boldsymbol{G}$ is the dynamic step response matrix, $\boldsymbol{f}$ is the free response of the system, and $\boldsymbol{\Delta u}$ is a vector of the control increments in the control horizon, defined in Equation (4)

$$\boldsymbol{\Delta u}(k) = [\Delta u(k|k) \ldots \Delta u(k+N_u-1|k)]^T. \qquad (4)$$

By substituting (3) and (4) in (1), it is possible to write the GPC problem as a quadratic programming problem, written as in Equation (5)

$$J = \min_{\boldsymbol{\Delta u}} \frac{1}{2} \boldsymbol{\Delta u}^T \boldsymbol{H} \boldsymbol{\Delta u} + \boldsymbol{b}^T \boldsymbol{\Delta u} \qquad (5)$$

$$\text{s.t. } \boldsymbol{R}\boldsymbol{\Delta u} \le \bar{\boldsymbol{r}},$$

where $\boldsymbol{H} = 2(\boldsymbol{G}^T\boldsymbol{Q}_\delta\boldsymbol{G} + \boldsymbol{Q}_\lambda)$, $\boldsymbol{b}^T = 2(\boldsymbol{f} - \boldsymbol{w})^T\boldsymbol{G}$, $\boldsymbol{Q}_\delta = \text{diag}(\delta(j)) \in \mathbb{R}^{N \times N}$, $\boldsymbol{Q}_\lambda = \text{diag}(\lambda(j)) \in \mathbb{R}^{N_u \times N_u}$, $\boldsymbol{R} \in \mathbb{R}^{N_r \times N_u}$, $\bar{\boldsymbol{r}} \in \mathbb{R}^{N_r}$, and N_r is the number of modeled constraints. The problem in (5) is solved at each iteration to determine the optimal sequence of control increments and the first element of this sequence is used in the plant.

2.2 RLS Estimator

The GPC predictions of the system output computed by Equation (2) consider a discrete-time model of the plant which can be written as:

$$G(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n_b} z^{-n_b}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n_a} z^{-n_a}}, \qquad (6)$$

where $B(z^{-1})$ and $A(z^{-1})$ are polynomials with degree n_b and n_a , respectively. RLS is used to estimate the polynomial coefficients $b_1, b_2, \ldots, b_{n_b}$ and $a_1, a_2, \ldots, a_{n_a}$, which are grouped in a vector of estimated parameters $\hat{\boldsymbol{\theta}} \in \mathbb{R}^{n_a+n_b}$ as:

$$\hat{\boldsymbol{\theta}} = [-a_1 \; -a_2 \ldots -a_{n_a} \;\; b_1 \;\; b_2 \ldots b_{n_b}]^T. \qquad (7)$$

For a given estimate of the parameters done at instant $k-1$, it is possible to obtain the estimate error for one-step-ahead prediction of $y(k)$ as:

$$\varepsilon(k) = y(k) - \boldsymbol{x}^T(k)\hat{\boldsymbol{\theta}}(k-1), \qquad (8)$$

where $\boldsymbol{x} \in \mathbb{R}^{n_\theta}$ is a vector of regressors which contains the past values for *u* and *y* defined as:

$$\boldsymbol{x}^T = [-y(k-1) \ldots -y(k-n_a) \;\; u(k-1) \ldots u(k-n_b)]. \qquad (9)$$

Based on the quality of the estimate, measured in Equation (8), the parameter vector can be updated as:

$$\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \boldsymbol{K}(k)\varepsilon(k), \qquad (10)$$

where $\boldsymbol{K} \in \mathbb{R}^{n_\theta}$ is the gain vector of RLS algorithm, defined as (Aguirre, 2015):

$$\boldsymbol{K}(k) = \frac{\boldsymbol{L}(k-1)\boldsymbol{x}(k)}{\alpha(k) + \boldsymbol{x}^T(k)\boldsymbol{L}(k-1)\boldsymbol{x}(k)}. \qquad (11)$$

In this definition, matrix $\boldsymbol{L} \in \mathbb{R}^{n_\theta \times n_\theta}$ is a modified version of the covariance matrix:

$$\boldsymbol{P}(k) = \frac{\boldsymbol{L}(k-1) - \boldsymbol{K}(k)\boldsymbol{x}^T(k)\boldsymbol{L}(k-1)}{\alpha(k)}, \qquad (12)$$

which avoids that the trace of $\boldsymbol{P} \in \mathbb{R}^{n_\theta \times n_\theta}$ assumes small values or, in other words, that the RLS estimator becomes insensitive to parameter variations (Aguirre, 2015). Thus, after $\boldsymbol{P}$ is obtained by using Equation (12), a new matrix is defined as:

$$\boldsymbol{L}(k) = \begin{cases} \boldsymbol{P}(k) + \boldsymbol{Q}(k), \text{if } \operatorname{tr}[\boldsymbol{P}(k)] < \rho_0 \\ \qquad \boldsymbol{P}(k), \text{ else} \end{cases}, \qquad (13)$$

where a positive semidefinite diagonal matrix:

$$\boldsymbol{Q}(k) = \begin{bmatrix} q_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & q_{n_\theta} \end{bmatrix} \tag{14}$$

is added to $\boldsymbol{L}$ at each sampling instant only if the trace of $\boldsymbol{P}$, $\mathrm{tr}[\boldsymbol{P}(k)]$, is smaller than a minimal trace defined by the user, ρ_0 . Matrix $\boldsymbol{Q} \in \mathbb{R}^{n_\theta \times n_\theta}$ is composed of coefficients, $q_1, q_2, \dots, q_{n_\theta}$, that must be chosen according to the variation of each estimated parameter in $\widehat{\boldsymbol{\theta}}$. If an estimated parameter is expected to have large changes, then a large value for $\boldsymbol{Q}$ coefficients should be considered. Otherwise, values should be small, since large values insert an abrupt gain for the change of the estimated parameters.

In Equations (11) and (12), α is known as a variable forgetting factor and it is used to define the relative importance of new and past data on the estimates. In this work α is calculated at each sampling instant as:

$$\alpha(k) = \begin{cases} \frac{\mathrm{tr}[\boldsymbol{P}(k)]}{\rho_0}, \text{if } \mathrm{tr}[\boldsymbol{P}(k)] < \rho_0 \\ \quad 1, \text{ else} \end{cases}, \tag{15}$$

where if $\mathrm{tr}[\boldsymbol{P}(k)]$ is smaller than ρ_0, α assumes the value of $\mathrm{tr}[\boldsymbol{P}(k)]/\rho_0$, and the RLS estimator gives more attention to current values than past system history values.

3. ALGORITHM

From the base formulation developments presented in section 2, the implementation of the proposed FAGPC is summarized as in Algorithm 1.

Algorithm 1 Fast Adaptive Generalized Predictive Control

```
 1: Initialize:
        $\widehat{\boldsymbol{\theta}}$, $\boldsymbol{L}$, $\boldsymbol{Q}$, $\alpha$, $\rho_0$, $N$, $N_u$, $\boldsymbol{Q}_\lambda$, $\boldsymbol{Q}_\delta$, $y$, $\Delta\boldsymbol{u}$
 2: $k = 1$
 3: while $k \geq 1$ do
 4:     Measure $y(k)$
 5:     Form $\boldsymbol{x}(k)$
 6:     Compute $\varepsilon(k), \boldsymbol{K}(k), \widehat{\boldsymbol{\theta}}(k), \boldsymbol{P}(k)$
 7:     Compute $\boldsymbol{f}(k)$ and $\boldsymbol{G}(k)$
 8:     Form $\boldsymbol{w}(k)$
 9:     Compute $\boldsymbol{H}(k), \boldsymbol{b}^T(k), \boldsymbol{R}(k), \bar{\boldsymbol{r}}(k)$
10:     Run ADMM optimizer
11:     Obtain $\boldsymbol{\Delta u}(k)$
12:     Apply $u(k) = u(k-1) + \Delta u(1)$
13:     if $\mathrm{tr}[\boldsymbol{P}(k)] < \rho_0$ then
14:         $\boldsymbol{L}(k) = \boldsymbol{P}(k) + \boldsymbol{Q}(k)$
15:         $\alpha(k) = \mathrm{tr}[\boldsymbol{P}(k)]/\rho_0$
16:     else
17:         $\boldsymbol{L}(k) = \boldsymbol{P}(k)$
18:         $\alpha(k) = 1$
19:     end if
20:     Update Variables
21:     $k = k + 1$
22:     wait for $T_s$
23: end while
```

In line 1 both the estimator and GPC parameters are declared. The code defined from line 3 to 23 is repeated at each sampling instant. First, the system output $y(k)$ is measured. Then, the vector of regressors $\boldsymbol{x}(k)$ is formed using the current and previous values for y and the previous values for u. Line 6 can be computed to obtain the one step-ahead prediction error of the estimator $\varepsilon(k)$, the gain vector $\boldsymbol{K}(k)$, the estimated parameters $\widehat{\boldsymbol{\theta}}(k)$ and the covariance matrix $\boldsymbol{P}(k)$. Note that $\widehat{\boldsymbol{\theta}}(k)$, as also $\boldsymbol{P}(k)$, is computed in the first iteration using the initial values declared in line 1 of the Algorithm 1. There are some rules of thumb to initialize $\widehat{\boldsymbol{\theta}}(0)$ and $\boldsymbol{L}(0)$ since they represent the direction and velocity of the convergence of the parameters. The initial values for both depend on the quality of the *a priori* information about the process dynamics. If nothing is known about the system parameters, the initial condition of the vector of parameters is usually assumed null, i.e. $\widehat{\boldsymbol{\theta}}(0) = \boldsymbol{0}$, and $\boldsymbol{L}(0)$ is typically chosen in the range $10^3\boldsymbol{I} \leq \boldsymbol{L}(0) \leq 10^7\boldsymbol{I}$, where $\boldsymbol{0} \in \mathbb{R}^{n_\theta}$ is a vector of zeros and $\boldsymbol{I} \in \mathbb{R}^{n_\theta \times n_\theta}$ is the identity matrix. Otherwise, if there is reliable *a priori* information about the process, $\widehat{\boldsymbol{\theta}}(0)$ should be initialized with the known parameters and the trace of $\boldsymbol{L}(0)$ must be small, normally chosen as $\boldsymbol{L}(0) \leq 10\boldsymbol{I}$.

Once the model is updated, the predictions used by the GPC controller are computed (line 7 in Algorithm 1). The vector of free response elements, $\boldsymbol{f}(k)$, and the dynamic step response matrix, $\boldsymbol{G}(k)$, are calculated with the model of the process updated by the estimated parameters in $\widehat{\boldsymbol{\theta}}(k)$. Then, the vector of future references $\boldsymbol{w}(k)$ is formed and Equation (5) is solved using a custom developed ADMM optimizer, as shown in line 10. As a result, a sub-optimal future control increment sequence $\boldsymbol{\Delta u}(k)$ is obtained, taking into account constraints in the process. The control signal $u(k)$ is computed with its past value added with the first value of vector $\boldsymbol{\Delta u}(k)$, which contains the control increment for k. This strategy of control is also called receding horizon control, where only the first control increment of the calculated future sequence is applied to the plant input and the whole process is repeated at the next sampling instant. Finally, $\boldsymbol{L}(k)$ and $\alpha(k)$ are updated according to the trace of covariance matrix $\mathrm{tr}[\boldsymbol{P}(k)]$, the variables are updated for the next iteration and the period required for completing the sampling time is waited.

4. EXPERIMENTAL CASE STUDY

This section presents an experimental case study to evaluate the proposed FAGPC algorithm. Figure 3 shows a picture of a didactic plant used, which is composed of a direct current (DC) motor, *M*, mechanically coupled to a tachogenerator, T_a, a power drive circuit, and a microcontroller.

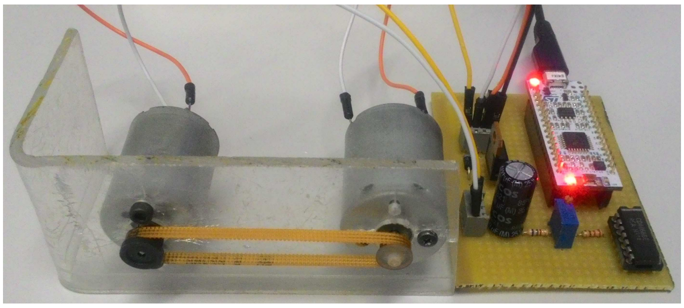

Figure 3 – Picture of the didactic plant used for the experimental case study.

A schematic view of the plant is shown in Figure 4. A bipolar junction transistor, Q_a, is used to drive the motor. Since *u* is a Pulse Width Modulation (PWM) signal, the driving transistor just operates in two modes: saturated or cut. The feedback signal, T_a, is filtered by the capacitor *C* and a voltage divider (resistors *R3* and *R4*) is used to adjust the magnitude of the output signal to the range supported by the microcontroller. The clock frequency of the microcontroller processor is configured as 64 MHz. The switch S is used to apply a step disturbance in the output voltage T_a. A microcontroller output is used to trigger the disturbance, which allows a better performance comparison between the proposed controller and the classical GPC since the same disturbances can be used in both tests.

The values of the components used in this experimental case study are: *R1* = *R5* = 220 Ω; *R2* = 2.2 kΩ; *R3* = 6.8 kΩ; *R4* = 10 kΩ; Q_a = TIP41C; *D* = 1N4007; *C* = 1000 µF; *M* = 12 W (@ 12 V); T_a = 9 V; V_{in} = 12 V; *S* = CD4066B; *Microcontroller* = STM32F303K8T6 / ARM® Cortex® M4 72 MHz / 64 KB Flash / 16 KB SRAM.

In order to evaluate the performance of the closed-loop system controlled by both the proposed approach and traditional GPC, the Integral Absolute Error (IAE), as given in Equation (16), was used.

$$IAE = \sum_{k=1}^{n_{max}} |w(k) - y(k)|, \tag{16}$$

where n_{max} is the maximum number of iterations of the algorithm in a given test scenario.

In this case study, the plant was modeled as a first-order system

$$G(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_1 z^{-1}}{1 + a_1 z^{-1}}. \tag{17}$$

The coefficients of the discrete-time transfer function in Equation (17) were initialized based on experimental tests in open-loop between 20 % and 40 % of the full speed of the DC motor as $\hat{\boldsymbol{\theta}}(0) = [-a_1 \; b_1]^T$, were $b_1 = 0.2$ and $a_1 = 0.7$.

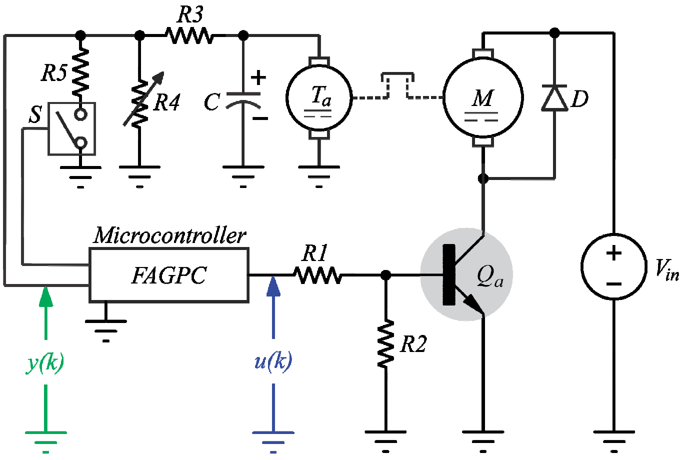

Figure 4 – General scheme of the system for the experimental case study.

The RLS is used to estimate the coefficients b_1 and a_1 at each sampling instant based on data measured up to that time instant. The initial values for RLS estimator were assumed as $\boldsymbol{P}(0) = 10\boldsymbol{I}_{2x2}$, $\alpha(0) = 0.95$, $q_1 = 0.01$, $q_2 = 0$ and $\rho_0 = 10^{-7}$. The prediction and control horizons were defined as N = 5 and N_u = 3, respectively. Constant weighting factors δ = 1 and λ = 1.5 were considered in all experiments, since this tuning results in a good balance between reference-tracking dynamics and smoothness of the control signal. Constraints in the control increment and the absolute value of the control signal are considered as: $\Delta u_{max} = \pm 20$ %, $u_{max} = 100$ %, and $u_{min} = 0$ %.

The general scenario proposed for experimental testing of the behavior of the process output consists in applying a series of step reference changes which take the plant to different operating points. In all cases, the proposed controller is compared with the regular implementation of GPC (with a constant plant model used for prediction).

Figure 5 shows the closed-loop responses for reference tracking. It can be noted that in the initial samples the system response with the proposed controller does not present a satisfactory transient compared with the response obtained by GPC. This happens mainly because the initial values assumed for the model were not a good representation of the plant at the initial operating point, and the number of samples used for estimating the coefficients of the plant model was too small, so there was a large change in the plant model used for prediction. After this first transient it is possible to observe that the proposed controller has better dynamic response than the traditional GPC. In addition, the proposed controller is able to keep almost the same shape of response during step reference changes at different operating points, while the response obtained by GPC strongly depends on the operating point. The IAE values for GPC and the proposed controller for reference tracking (Figure 5) are, respectively, 1400.4 % and 1196.3 %. This index reinforces the better response obtained by the proposed controller. Even though the proposed controller presents a good dynamic behavior, its computational burden is compatible with the dynamics required to control a DC motor, since the minimum, maximum and average execution times for one iteration are $T_{s_{min}} = 2.0$ ms, $T_{s_{max}} = 16.4$ ms and $T_{s_{avg}} = 2.2$ ms, respectively.

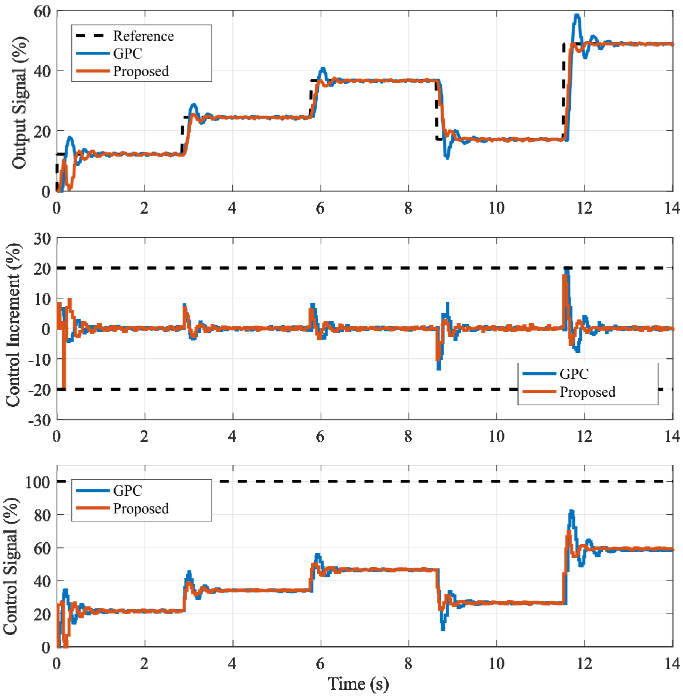

Figure 5 – Experimental results of the proposed controller compared with GPC for closed-loop reference tracking.

Figure 6 shows the parameters estimated by the proposed method for all operating points considered in Figure 5. Notice that if the plant model is stable, its steady-state gain is defined as

$$K_p = \frac{b_1}{1 + a_1}. \tag{18}$$

The same initial model parameters were considered in both controllers (GPC and proposed), but after a short transient, the estimated parameters converge to the corresponding operating point of the process. During this first transient of the RLS estimator, the quality of the output prediction used by the controller to define the control action was not good, which translated into additional oscillations in the closed-loop response, as shown in Figure 5.

Figure 7 presents the closed-loop response for disturbance rejection. The system was put to operate at a constant reference of approximately 61 % and a series of step disturbances were applied. In the first and second transients of Figure 7 a disturbance $d \cong -12$ % was inserted and removed at the plant output. In the third and fourth transients $d \cong 19$ % was inserted and removed.

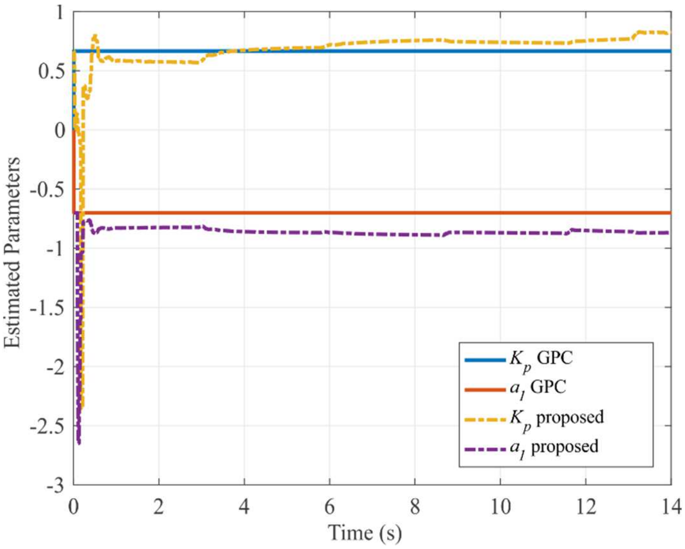

Figure 6 – Estimated parameters of the discrete-time model given in Equation (17).

It can be seen in Figure 7 that both controllers were able to reject the disturbances, but the proposed controller presented a faster response because the values assumed in the model of traditional GPC were obtained at a different operating point. The proposed controller again presented better IAE index: 855.1 % against 1146.4 % presented by GPC. The minimum, maximum and average sampling periods in the test of Figure 7 are, respectively, $T_{s_{min}} = 2.0$ ms , $T_{s_{max}} = 6.4$ ms and $T_{s_{avg}} = 2.2$ ms.

In addition to presenting better dynamic performance, the proposed controller resulted in smoother control signals for all the considered operating points and resulted in almost the same transient response characteristics for all the operating points, while the transient response of GPC is different for each considered condition. Thus, this case study shows that the proposed method can be used not only to estimate unknown coefficients of a plant model used in an MPC implementation, but also to allow the use of a computationally efficient MPC implementation to control nonlinear plants. Regular MPC implementations for nonlinear models require a nonlinear program to be solved at each iteration, which can limit their applicability to plants with fast dynamics. The proposed controller allows the use of linear MPC formulations by constantly updating the (linear) prediction model to be a good representation of the plant dynamic behavior for a given operating point.

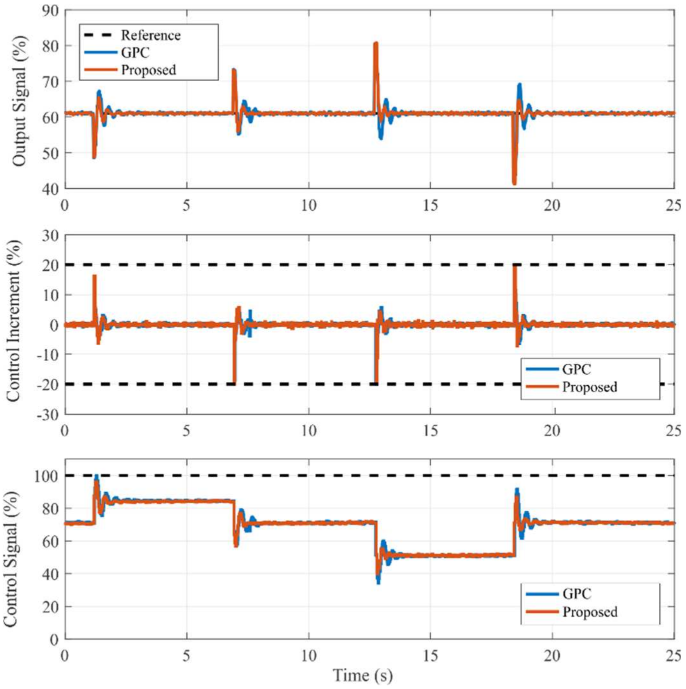

Figure 7 – Experimental results of the proposed controller compared with GPC for disturbance rejection.

5. CONCLUSION

This work proposes a controller based on the GPC algorithm which includes an RLS identification method to estimate online the parameters of the linear discrete-time model of the plant. The main goal of the proposed controller structure is to adjust the coefficients of the model at each sampling instant for different operating points, so that the proposed controller can be used in plants with variable parameters, constraints, and fast dynamics. In this work, the proposed adaptive algorithm considered a variable forgetting factor updated online according to the quality of the closed-loop data used for identification, which was determined based on the trace of the covariance matrix. Thus, when the system is considered not to have a persistent excitation, the plant is not updated to avoid identification problems.

An experimental case study showed that the proposed controller presented a good representation of the plant and a satisfactory closed-loop transient for all the considered operating points and for both reference changes and disturbance rejection. The IAE index values proved that the proposed controller presented better and faster closed-loop dynamic behavior than the traditional formulation of GPC. In addition, the obtained computational burden results show that the proposed controller can be embedded in a microcontroller.

Future work will focus on the expansion to the case of Multiple-Input Multiple-Output (MIMO) plants and the application of the proposed controller to systems with more pronounced nonlinearities.

6. ACKNOWLEDGMENT

This work was supported in part by the National Council for Scientific and Technological Development (CNPq/Brazil) under Grant 309244/2018-8 and in part by the Coordenação de

Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

REFERENCES

AGUIRRE, L. A. Introdução à Identificação de Sistemas: Técnicas Lineares e Não Lineares. 4 ed., Belo Horizonte: Editora UFMG, 2015.

CAMACHO, E. F.; BORDONS, C. Model Predictive Control. London: Springer, 2007.

CLARKE, D. W.; MOHTADI, C., TUFFS, P. S. Generalized Predictive Control – Part I: The Basic Algorithm. **Automatica**, vol. 23, no. 2, p. 137-148, 1987.

ELSISI, M. New design of adaptive model predictive control for energy conversion system with wind torque effect. **Journal of Cleaner Production**, vol. 240, p. 1-8, 2019.

FESHARAKI, J. S.; KAMALI, M., SHEIKHOLESLAM, F. Adaptive tube-based model predictive control for linear systems with parametric uncertainty. **The Institution of Engineering and Technology**, vol. 11, no. 17, p. 2947-2953, 2017.

JABBOUR, N.; MADEMLIS, C. Online parameters estimation and autotuning of a discrete-time model predictive speed controller for induction motor drives. **IEEE Transactions on Power Electronics**, vol. 34, no. 2, p. 1548-1559, 2019.

KOO, J.; PARK, D.; RYU, S.; KIM, G.; LEE, Y. Design of a self-tuning adaptive model predictive controller using recursive model parameter estimation for real-time plasma variable control. **Journal of Computers and Chemical Engineering**, vol. 123, p. 126-142, 2019.

LI, S. E.; JIA, Z.; LI, K.; CHENG, B. Fast online computation of a model predictive controller and its application to fuel economy–oriented adaptive cruise control. **IEEE Transactions on Intelligent Transportation Systems**, vol. 16, no. 3, p. 1199-1209, 2015.

NORMEY-RICO, J. E.; CAMACHO, E. F. Control of Dead-Time Processes. London: Springer, 2007.

PECCIN, V. B.; LIMA, D. M.; FLESCH, R. C. C.; NORMEY-RICO J. E. Implementação de GPC e DMC para sistemas rápidos usando ADMM. Congresso Brasileiro de Automática - CBA, p. 1-8, 2018.

RADECKI, P.; HENCEY, B. Online Model Estimation for Predictive Thermal Control of Buildings. **IEEE Transactions on Control Systems Technology**, vol. 25, no. 4, p. 1414-1422, 2017.

ROVEA, S. B.; FLESCH, R. C. C. Practical Implementation of Fast Adaptive Generalized Predictive Control for Systems with Variable Parameters. Congresso Brasileiro de Instrumentação, Sistemas e Automação – COBISA, Campinas, São Paulo, 14-16 May, p. 1-7, 2019.

SARAF, N.; BEMPORAD, A. Fast model predictive control based on linear input/output models and bounded-variable least squares. IEEE 56th Annual Conference on Decision and Control (CDC), p. 1919-1924, 2017.

SHORT, M. Input-constrained adaptive GPC for simple industrial plant. 18th International Conference on Automation and Computing (ICAC), p. 1-6, 2012.

ZHENG, H.; NEGENBORN, R. R.; LODEWIJKS, G. Fast ADMM for distributed model predictive control of cooperative waterborne AGVs. **IEEE Transactions on Control Systems Technology**, vol. 25, no. 4, p. 1406-1413, 2017.

YANG, S.; WAN, M. P.; CHEN, W.; FENG, B. N.; ZHAI, D. An adaptive robust model predictive control for indoor climate optimization and uncertainties handling in buildings. **Building and Environment**, vol. 163, p. 1-18, 2019.

ZHU, B.; XIAOHUA, X. Adaptive model predictive control for unconstrained discrete-time linear systems with parametric uncertainties. **IEEE Transactions on Automatic Control**, vol. 61, no. 10, p. 3171-3176, 2016.