Adaptive Inverse Control Based on Second Order Volterra Model via Modified Variable Step Size Fractional LMS Algorithm

Rodrigo Possidônio Noronha
Department of Electrical Engineering
Federal Institute of Education, Science and Technology of Maranhão
Imperatriz, Brazil
rodrigo.noronha@ifma.edu.br

***Abstract*—This work proposes a new methodology of nonlinear Direct Adaptive Inverse Control (DAIC). In this methodology, the controller is based on the Volterra model. To obtain a better interpretability and decrease the complexity of mathematical analysis, the Volterra model was truncated up to the second order kernel. The weight vector of Volterra model was estimated through the Modified Variable Step Size Least Mean Square (MVSS-FLMS) algorithm. Since the Volterra model is nonlinear, the proposed methodology was evaluated on a dynamic system represented through of a Nonlinear AutoRegressive with eXogenous inputs (NARX) model in the presence of a disturbance signal.**

***Index Terms*—Adaptive Inverse Control, Fractional LMS, Inverse Model, Nonlinear Control, Volterra Series.**

I. Introduction

The Direct Adaptive Inverse Control (DAIC) technique proposed by [1] aims to control the plant inverse dynamics using a quasi-feedforward configuration. DAIC is a control technique based on inverse model identification. In the absence of uncertainty, the controller is equal to the plant inverse model. Usually, the DAIC methodologies have been formulated with the controller mathematical representation based on classes of linear models [1]–[4]. Although the DAIC with the controller mathematically defined by a linear representation performs well in tracking the inverse dynamics of linear plants, in the presence of nonlinearity the performance of tracking the plant inverse dynamics will be unsatisfactory.

One possible solution for tracking of the inverse dynamics of nonlinear plants is through mathematical representations of classes of nonlinear model. However, depending on the complexity of the nonlinear model, it is quite difficult to perform the design and practical implementation of a controller based on a nonlinear model [5]–[7]. In order to achieve greater interpretability and less complexity in nonlinear DAIC design, it is necessary to impose constraints on the nonlinear model obtained [8], [9].

One class of mathematical representation of nonlinear models used in many different applications are the nonlinear models based on Volterra series [10], [11]. A Volterra model can be defined as a generalization of the impulse response of a dynamic system, composed by the combination of linear terms and nonlinear terms of polynomial type [12]. Some examples of application areas of Volterra models: predictive control [13], [14], time series prediction [15]–[17], classification [18], [19], fault diagnosis [20], [21], and others. To obtain increased interpretability and decreased mathematical complexity of the nonlinear DAIC, the order of Volterra model, in this work, has been truncated up to the second order kernel. Thus, the controller is represented mathematically by a second order Volterra model.

According to [3], since the weight vector of Volterra model are linear, then it is possible to use algorithms based on stochastic gradient to estimate them. In the context of adaptive algorithms based on stochastic gradient, in [22] a new version of LMS algorithm based on the Rieman-Liouville fractional derivative definitions was proposed, such that the order of the derivative of the cost functional is fractional, titled Fractional Least Mean Square (FLMS). In this present work, the estimation of the weight vector of Volterra model was performed through the Modified Variable Step Size FLMS (MVSS-FLMS) algorithm, proposed in [23]. According to [23], the MVSS-FLMS algorithm is an improvement of FLMS algorithm, such that was proposed the use of the variable step size to obtain a better learning rate and, consequently, a fast convergence speed and a small steady-state Mean Square Error (MSE) during the update of the estimate of the weight vector of Volterra model. Since the Volterra model is able to represent nonlinear dynamics of polynomial type, the proposed control methodology was evaluated in a plant represented through a Nonlinear AutoRegressive with eXogenous inputs (NARX) model in the presence of a disturbance signal. This paper is organized with the following sections: in Section II, the mathematical formulations for the nonlinear model based on Volterra series and MVSS-FLMS algorithm are presented; in Section III, the mathematical formulations for the nonlinear DAIC are presented; in Section IV, the computational results obtained through the evaluation of the proposed control methodology in the presence of a disturbance signal are presented.

II. Nonlinear Volterra Model

According to [10], a Volterra model is a generalization of impulse response of dynamic systems, composed of linear terms and nonlinear terms of polynomial type. According to [12], the linear terms of Volterra model are due to the convolution of the input signal $u(k)$ with the impulse response function (first order Volterra kernel); the nonlinear terms are due to convolution the multiplications between the delayed versions of the input signal $u(k)$ with the impulse response function (higher order Volterra kernel). The general form of Volterra model is given by:

$$y(k)=\sum_{m=1}^{+\infty}\cdots\sum_{n_m=-\infty}^{+\infty}\theta_m\left(n_1,\ldots,n_m\right)\prod_{i=1}^{m}u\left(k-n_i\right), \tag{1}$$

where m is the degree of nonlinearity of Volterra model and $\theta_m\left(n_1,n_2,\ldots,n_m\right)$ is the m-th order Volterra kernel. It is important to note that the higher the order of a Volterra model, the better the model can perform in representing complexities [12], [24], [25]. However, the higher the complexity of the model the more complex it will be to estimate the weights of the m-th order Volterra kernel, of mathematical analysis and controller design. A common alternative to work with Volterra models is truncating it up to the second order. Even with a truncated order Volterra model, it is possible to represent nonlinear dynamics of polynomial type [26]. In addition, through a truncated order Volterra model, it is easier to work on applications involving nonlinear system identification, nonlinear system control, nonlinear filtering, and others. Thus, the second-order Volterra model is given by:

$$\begin{aligned}y(k)=&\sum_{n_1=0}^{N_1}\theta_1\left(n_1\right)u\left(k-n_1\right)\\&+\sum_{n_1=0}^{N_2}\sum_{n_2=0}^{N_2}\theta_2\left(n_1,n_2\right)u\left(k-n_1\right)u\left(k-n_2\right),\end{aligned} \tag{2}$$

where $\theta_1(n_1)$ is the first order Volterra kernel and $\theta_2(n_1,n_2)$ is the second order Volterra kernel.

A. Estimation of the Weight Vector of a Second Order Volterra Model

To reduce the complexity of Volterra model, in (2) it was set that $N_1=N_2=2$. Then, the general form of a second order Volterra series is given by:

$$\begin{aligned}y(k)=&\sum_{n_1=0}^{2}\theta_1\left(n_1\right)u\left(k-n_1\right)\\&+\sum_{n_1=0}^{2}\sum_{n_2=0}^{2}\theta_2\left(n_1,n_2\right)u\left(k-n_1\right)u\left(k-n_2\right),\end{aligned} \tag{3}$$

According to [26], since the weight vector of Volterra are linear, it is possible to use the FLMS algorithm to estimate them. According to [12], the update of the estimate of the weight vector of Volterra model through of FLMS algorithm is given by:

$$\mathbf{\Theta}(k+1)=\mathbf{\Theta}(k)-\mu\frac{\partial J(k)}{\partial\mathbf{\Theta}(k)}-\mu_f\left(\frac{\partial J(k)}{\partial\mathbf{\Theta}(k)}\right)^{v}J(k), \tag{4}$$

where $J=\nabla_{\mathbf{\Theta}(k)}(e^2(k))$ is the cost functional described as $e^2(k)=(\bar{y}(k)-y(k))^2$, v is the fractional order, μ and u_f are the step sizes, $\bar{y}(k)$ is the system output signal and $y(k)$ is the Volterra model output signal. It is important to note that the weight vector $\mathbf{\Theta}(k)\in\mathbb{R}^{12\times1}$ and regressors vector $\mathbf{U}(k)\in\mathbb{R}^{12\times1}$ are given, respectively, by:

$$\mathbf{\Theta}(k)=\begin{bmatrix}\theta_1(0)\\\vdots\\\theta_1(2)\\\theta_2(0,0)\\\theta_2(0,1)\\\vdots\\\theta_2(2,1)\\\theta_2(2,2)\end{bmatrix}\quad\mathbf{U}(k)=\begin{bmatrix}u(k)\\\vdots\\u(k-2)\\(u(k))^2\\u(k)u(k-1)\\\vdots\\u(k-2)u(k-1)\\(u(k-2))^2\end{bmatrix}. \tag{5}$$

According to [27], through the chain rule, the partial derivative $\frac{\partial J(k)}{\partial\mathbf{\Theta}(k)}$ is given by:

$$\frac{\partial J(k)}{\partial\mathbf{\Theta}(k)}=\frac{\partial J(k)}{\partial e(k)}\frac{\partial e(k)}{\partial y(k)}\frac{\partial y(k)}{\partial\mathbf{\Theta}(k)}, \tag{6}$$

which can be rewritten as:

$$\frac{\partial J(k)}{\partial\mathbf{\Theta}(k)}=-e(k)\mathbf{U}(k). \tag{7}$$

According to [27], through the chain rule, the partial derivative $\left(\frac{\partial}{\partial\mathbf{\Theta}(k)}\right)^{v}J(k)$ is given by:

$$\left(\frac{\partial}{\partial\mathbf{\Theta}(k)}\right)^{v}J(k)=\frac{\partial J(k)}{\partial e(k)}\frac{\partial e(k)}{\partial y(k)}\left(\frac{\partial}{\partial\mathbf{\Theta}(k)}\right)^{v}y(k). \tag{8}$$

Definition 1 [28]: *The fractional derivative of Riemann-Liouville of a function f is given by:*

$$(D^v f)(t)=\frac{1}{\Gamma(k-v)}\left(\frac{d}{dt}\right)^{v}\int_0^t f(\tau)(t-\tau)^{k-v-1}d\tau, \tag{9}$$

$$(D^v f)(t-\alpha)^{\alpha}=\frac{\Gamma(1+\alpha)}{\Gamma(1+\alpha-v)}(t-\alpha)^{\alpha-v} \tag{10}$$

where $1+\alpha-v>0$, $t>0$ and D is the differential operator. The Gamma function $\Gamma(v)$ is defined as [29]:

$$\Gamma(v)=\int_0^{\infty}t^{v-1}e^{-t}dt. \tag{11}$$

According to [23], using (7) and (10), (8) is rewritten as follows:

$$\left(\frac{d}{d\mathbf{\Theta}(k)}\right)^{v}J(k)=-e(k)\mathbf{U}(k)\frac{(\mathbf{\Theta}(k))^{1-v}}{\Gamma(2-v)}. \tag{12}$$

Using (7) and (12), (4) is rewritten as follows [23]:

$$\mathbf{\Theta}(k+1)=\mathbf{\Theta}(k)+\mu e(k)\mathbf{U}(k)+\mu_f e(k)\mathbf{U}(k)\frac{(\mathbf{\Theta}(k))^{1-v}}{\Gamma(2-v)}. \tag{13}$$

As proposed in [23], to obtain the MVSS-FLMS algorithm, it is considered that $\mu_f = \mu\Gamma(2-f)$. Due to this consideration, (13) is rewritten as follows:

$$\begin{aligned}\mathbf{\Theta}(k+1) &= \mathbf{\Theta}(k) + \mu e(k)\mathbf{U}(k) + \mu e(k)\mathbf{U}(k)(\mathbf{\Theta}(k))^{1-v} \\ &= \mathbf{\Theta}(k) + \mu e(k)\mathbf{U}(k)\left[1 + (\mathbf{\Theta}(k))^{1-v}\right]\end{aligned} \tag{14}$$

After obtaining (14), it is necessary to make the step size μ variable, such that the update of μ is given by [23]:

$$\mu(k+1) = c\left[\mu(k)(0.1 - \mu(k))\exp(-e(k)e(k-1))\right] \tag{15}$$

where c is a scaling parameter, $\mu(k)$ is the step size obtained at time k and exp($\bullet$) is the exponential function. Thus, the equation for updating the weight vector described in (14) is rewritten as:

$$\mathbf{\Theta}(k+1) = \mathbf{\Theta}(k) + \mu(k)e(k)\mathbf{U}(k)\left[1 + (\mathbf{\Theta}(k))^{1-v}\right] \tag{16}$$

III. Nonlinear DAIC

In this work, the relationship between the plant output signal $y(k)$ and control signal $u(k)$ is given by:

$$y(k) = \mathrm{P}[y(k-1), \ldots, y(k-n_y), u(k), \ldots, u(k-n_u)], \tag{17}$$

it is important to note that the plant model P[$\bullet$] is a nonlinear functional defined as a second degree NARX model with $n_y = 2$ and $n_u = 1$. Thus, (17) can be rewritten as:

$$\begin{aligned} y(k) &= c_{0,0} + \sum_{n_1=1}^{2} c_{1,0}(n_1)y(k-n_1) \\ &+ \sum_{n_1=1}^{2} c_{1,0}(n_1)u(k-n_1) \\ &+ \sum_{n_1=1}^{2}\sum_{n_2=1}^{2} c_{2,0}(n_1,n_2)y(k-n_1)y(k-n_2) \\ &+ \sum_{n_1=1}^{2}\sum_{n_2=1}^{2} c_{1,1}(n_1,n_2)y(k-n_1)u(k-n_2) \\ &+ \sum_{n_1=1}^{2}\sum_{n_2=1}^{2} c_{0,2}(n_1,n_2)u(k-n_1)u(k-n_2), \end{aligned} \tag{18}$$

In Fig. 1 it is presented the block diagram for the nonlinear DAIC. It is important to note that the controller, defined by a second order Volterra model, is represented by the nonlinear functional $\hat{\mathrm{C}}[\bullet]$. In addition, it is important to note that the controller weight vector is estimated as a function of the reference error $e_{ref}(k)$ and estimation error of the plant model $\hat{\mathrm{P}}[\bullet]$, as proposed by [3] for the linear DAIC. Due to this, the nonlinear DAIC performs well in tracking the plant reverse dynamics and, consequently, of the reference signal $r(k)$.

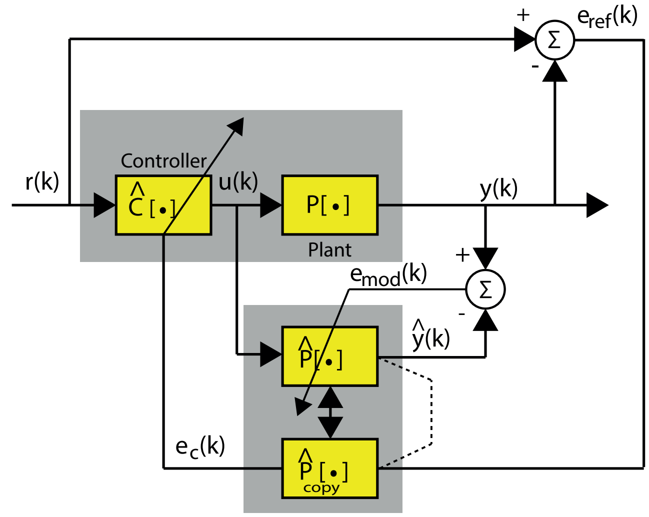

Fig. 1: Block diagram for the nonlinear DAIC.

The control signal $u(k)$ obtained through the controller $\hat{\mathrm{C}}[\bullet]$, represented by a second order Volterra model with $N_1 = N_2 = 2$, is given by:

$$\begin{aligned} u(k) &= \sum_{n_1=0}^{2} \theta_1(n_1)\, e_c(k-n_1) \\ &+ \sum_{n_1=0}^{2}\sum_{n_2=0}^{2} \theta_2(n_1,n_2)\, e_c(k-n_1)\, e_c(k-n_2), \end{aligned} \tag{19}$$

where $\theta_1(n_1)$ is the first order Volterra kernel and $\theta_2(n_1,n_2)$ is the second order Volterra kernel of $\hat{\mathrm{C}}[\bullet]$. The weight vector of $\hat{\mathrm{C}}[\bullet]$ is given by $\mathbf{\Theta}(k) = [\theta_1(0)\ \ldots\ \theta_1(2)\ \theta_2(0,0)\ \theta_2(0,1)\ \ldots\ \theta_2(2,1)\ \theta_2(2,2)]^{\mathrm{T}} \in \mathbb{R}^{12\times 1}$. The weight vector $\mathbf{\Theta}(k)$ is estimated by the MVSS-FLMS algorithm. Developing (19), it is obtained that:

$$\begin{aligned} u(k) &= \theta_1(0)r(k) + \theta_1(1)r(k-1) + \theta_1(2)r(k-2) \\ &+\theta_2(0,0)(r(k))^2 + \theta_2(0,1)r(k)r(k-1) + \ldots \\ &+\theta_2(2,1)r(k-2)r(k-1) + \theta_2(2,2)(r(k-2))^2, \end{aligned} \tag{20}$$

which, in compact form, (20) can be rewritten as follows:

$$u(k) = \mathbf{R}^{\mathrm{T}}(k)\mathbf{\Theta}(k) = \mathbf{\Theta}^{\mathrm{T}}(k)\mathbf{R}(k), \tag{21}$$

where $\mathbf{R}(k) = [r(k)\ r(k-1)\ r(k-2)\ (r(k))^2\ r(k)r(k-1)\ \ldots\ r(k-2)r(k-1)\ (r(k-2))^2]^{\mathrm{T}} \in \mathbb{R}^{12\times 1}$ is the reference signal vector.

For the estimate of the weight vector $\mathbf{\Theta}(k)$ be updated, it is necessary that the update of the estimate of the weight vector $\mathbf{\Pi}(k)$ of the plant model $\hat{\mathrm{P}}[\bullet]$ be performed previously. For this to be possible, it is initially necessary to obtain the estimation error of the weight vector of $\hat{\mathrm{P}}[\bullet]$, given by $e_{mod}(k) = y(k) - \hat{y}(k)$ where $\hat{y}(k)$ is the output signal of $\hat{\mathrm{P}}[\bullet]$. The output signal

$\hat{y}(k)$ of the model $\hat{\mathrm{P}}[\bullet]$, represented by a second order Volterra model with $N_1 = N_2 = 2$, is given by:

$$\hat{y}(k) = \sum_{n_1=0}^{2} \pi_1(n_1)\, u(k-n_1) + \sum_{n_1=0}^{2}\sum_{n_2=0}^{2} \pi_2(n_1,n_2)\, u(k-n_1)\, u(k-n_2), \quad (22)$$

where $\pi_1(n_1)$ is the first order Volterra kernel and $\pi_2(n_1,n_2)$ is the second order Volterra kernel of $\hat{\mathrm{P}}[\bullet]$. The weight vector of $\hat{\mathrm{P}}[\bullet]$ is given by $\mathbf{\Pi}(k) = [\pi_1(0)\ \dots\ \pi_1(2)\ \pi_2(0,0)\ \pi_2(0,1)\ \dots\ \pi_2(2,1)\ \pi_2(2,2)]^{\mathrm{T}} \in \mathbb{R}^{12\times 1}$. The weight vector $\mathbf{\Pi}(k)$ is estimated by the MVSS-FLMS algorithm. Developing (19), it is obtained that:

$$\begin{aligned} \hat{y}(k) &= \pi_1(0)u(k) + \pi_1(1)u(k-1) + \pi_1(2)u(k-2) \\ &+\pi_2(0,0)(u(k))^2 + \pi_2(0,1)u(k)u(k-1) + \dots \\ &+\pi_2(2,1)u(k-2)u(k-1) + \pi_2(2,2)(u(k-2))^2, \end{aligned} \quad (23)$$

which, in compact form, (23) can be rewritten as follows:

$$\hat{y}(k) = \mathbf{U}^{\mathrm{T}}(k)\mathbf{\Pi}(k) = \mathbf{\Pi}^{\mathrm{T}}(k)\mathbf{U}(k), \quad (24)$$

where $\mathbf{U}(k) = [u(k)\ u(k-1)\ u(k-2)\ (u(k))^2\ u(k)u(k-1)\ \dots\ u(k-2)u(k-1)\ (u(k-2))^2]^{\mathrm{T}} \in \mathbb{R}^{12\times 1}$ is the control signal vector.

After obtain $\hat{y}(k)$, it is possible to update of the estimate of the weight vector $\mathbf{\Pi}(k)$ of $\hat{\mathrm{P}}[\bullet]$. After obtained the update of the estimate of the weight vector $\mathbf{\Pi}(k)$, the error used to update the weight vector $\mathbf{\Theta}(k)$ is given by:

$$e_c(k) = \sum_{n_1=0}^{2} \pi_1(n_1)\, e_{ref}(k-n_1) + \sum_{n_1=0}^{2}\sum_{n_2=0}^{2} \pi_2(n_1,n_2)\, e_{ref}(k-n_1)\, e_{ref}(k-n_2), \quad (25)$$

developing (25), it is obtained that:

$$\begin{aligned} e_c(k) &= \pi_1(0)e_{ref}(k) + \pi_1(1)e_{ref}(k-1) \\ &+\pi_1(2)e_{ref}(k-2) + \pi_2(0,0)(e_{ref}(k))^2 \\ &+\pi_2(0,1)e_{ref}(k)e_{ref}(k-1) + \dots \\ &+\pi_2(2,1)e_{ref}(k-2)e_{ref}(k-1) + \pi_2(2,2)(e_{ref}(k-2))^2, \end{aligned} \quad (26)$$

which, in compact form, (26) can be rewritten as follows: $e_c(k) = \mathbf{E}_{ref}^{\mathrm{T}}(k)\mathbf{\Pi}(k) = \mathbf{\Pi}^{\mathrm{T}}(k)\mathbf{E}_{ref}(k)$, where $\mathbf{E}_{ref}(k) = [e_{ref}(k)\ e_{ref}(k-1)\ e_{ref}(k-2)\ (e_{ref}(k))^2\ e_{ref}(k)e_{ref}(k-1)\ \dots\ e_{ref}(k-2)e_{ref}(k-1)\ (e_{ref}(k-2))^2]^{\mathrm{T}} \in \mathbb{R}^{12\times 1}$. The reference error $e_{ref}(k)$ is given by:

$$e_{ref}(k) = r(k) - y(k) \quad (27)$$

IV. Computational Results

In this section, it is performed the evaluation of the proposed methodology of nonlinear DAIC control based on second order Volterra model in a plant described through a NARX model, given by:

$$\begin{aligned} y(k) &= 0.6356y(k-1) + 0.3115y(k-2) + 0.1341y(k-3) \\ &-0.0916y(k-4) - 0.0047u(k-1) + 0.0054u(k-2) \\ &+0.0082u(k-3) - 0.0025y(k-4)u(k-1) + n(k), \end{aligned} \quad (28)$$

where, to evaluate the proposed control methodology in the presence of disturbances, a disturbance signal $n(k)$ was added to (28). In Fig. 2, it is shown the disturbance signal $n(k)$.

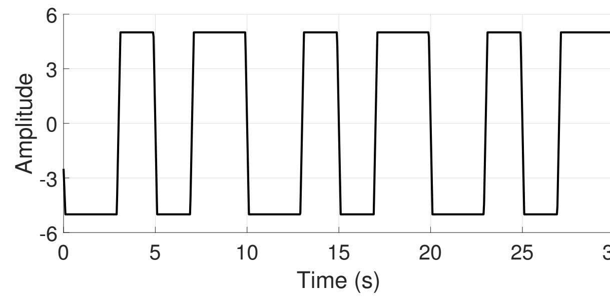

Fig. 2: Disturbance signal $n(k)$.

The sampling period used was set equal to $T_a = 0.025$ s and total simulation time was set equal to 30 s. It is important to note that the speed of evolution of the system dynamics must be compatible with the specified sampling period, to avoid a mismatch of the vectors described in Sections II and III. For the MVSS-FLMS algorithm, the initial value of the step size was set equal to $\mu(1) = 1 \times 10^{-3}$. The value of the fractional order was set equal to $v = 0.6$; the value of c in (15) was set equal to 0.03. The weight vectors $\mathbf{\Theta}(k)$ and $\mathbf{\Pi}(k)$ of Volterra models represented by the nonlinear functionals $\hat{\mathrm{C}}[\bullet]$ and $\hat{\mathrm{P}}[\bullet]$ were defined as follows:

$$\begin{aligned} &\mathbf{\Theta}(k) = [\theta_1(0)\ \theta_1(1)\ \theta_1(2)\ \theta_2(0,0)\ \theta_2(0,1)\ \theta_2(0,2) \\ &\quad \theta_2(1,1)\ \theta_2(1,2)\ \theta_2(2,2)]^{\mathrm{T}} \in \mathbb{R}^{9\times 1} \\ &\mathbf{\Pi}(k) = [\pi_1(0)\ \pi_1(1)\ \pi_1(2)\ \pi_2(0,0)\ \pi_2(0,1)\ \pi_2(0,2) \\ &\quad \pi_2(1,1)\ \pi_2(1,2)\ \pi_2(2,2)]^{\mathrm{T}} \in \mathbb{R}^{9\times 1}, \end{aligned} \quad (29)$$

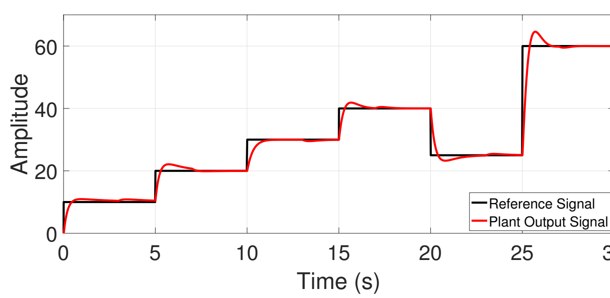

Fig. 3: Plant output signal $y(k)$.

In Fig. 3, it is shown the tracking of the reference signal $r(k)$ developed by the nonlinear DAIC. Due to the use of the variable step size, a satisfactory and fast tracking of the plant inverse dynamics and, consequently, of the reference signal $r(k)$ were obtained. The good performance developed by the nonlinear DAIC designed by the MVSS-FLMS algorithm, even in the presence of the disturbance signal $n(k)$, is also due to estimation of the weight vector $\mathbf{\Theta}(k)$ of $\hat{\mathrm{C}}[\bullet]$ be performed as a function of the reference error $e_{ref}(k)$ and estimation error $e_{mod}(k)$ of $\hat{\mathrm{P}}[\bullet]$.

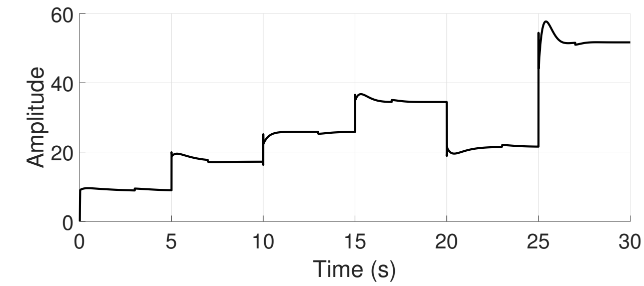

Fig. 4: Control signal $u(k)$.

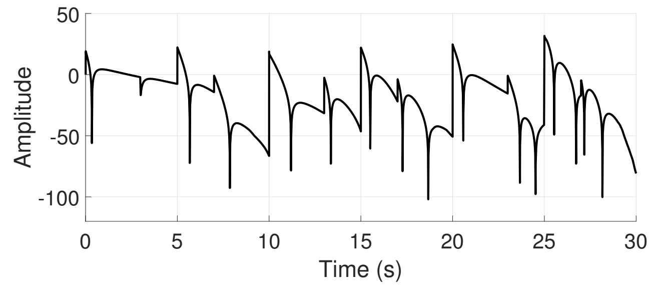

Fig. 7: MSE of $e_{mod}(k)$.

The control signal $u(k)$ developed by the nonlinear DAIC designed by the MVSS-FLMS algorithm is shown in Fig. 4. In Figs. 5 to 7, are shown the MSE of the errors $e_{ref}(k)$, $e_{mod}(k)$ and $e_c(k)$. It is important to note that the errors $e_{ref}(k)$ and $e_{mod}(k)$ are used to update the estimate of the weight vector $\mathbf{\Theta}(k)$ of the controller $\hat{\text{C}}[\bullet]$; the estimation error $e_{mod}(k)$ is used to update the estimate of the weight vector $\mathbf{\Pi}(k)$ of the model $\hat{\text{P}}[\bullet]$. Through MSE of $e_c(k)$, it is possible to note that was obtained a satisfactory and small steady-state MSE, which is due to tracking ability of the plant inverse dynamics and, consequently, of the reference signal $r(k)$ developed by the nonlinear DAIC designed by the MVSS-FLMS algorithm.

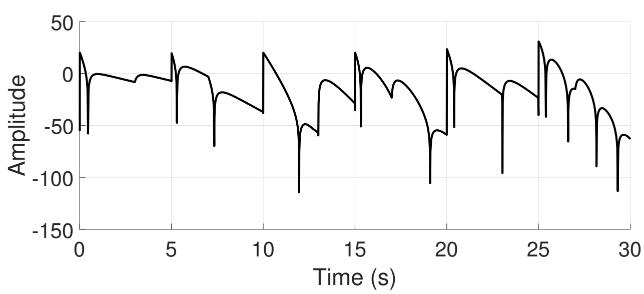

Fig. 5: MSE of $e_{ref}(k)$.

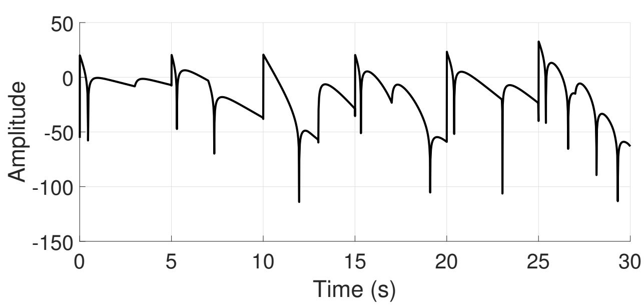

Fig. 6: MSE of $e_c(k)$.

V. Conclusion

Through MVSS-FLMS algorithm, where it was proposed the use of the variable step size for the FLMS algorithm, it was observed that the controller of DAIC satisfactorily tracked the plant inverse dynamics, such that the model plant contains a nonlinearity of polynomial type, even in the presence of the disturbance signal. In addition, it is possible to note that at each change of the amplitude of the disturbance signal and reference signal, the plant output signal still continued to converge to the reference signal.

References

[1] M. A. Shafiq, M. Shafiq, and N. Ahmed, “Closed loop direct adaptive inverse control for linear plants,” *The Scientific World Journal*, vol. 2014, 2014.

[2] M. Shafiq, M. Al Lawati, and H. Yousef, “A simple direct adaptive inverse control structure,” in *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2016, pp. 1–4.

[3] M. Shafiq, M. A. Shafiq, and H. A. Yousef, “Stability and convergence analysis of direct adaptive inverse control,” *Complexity*, vol. 2017, 2017.

[4] A. H. O. Ahmed, “High performance speed control of direct current motors using adaptive inverse control,” *WSEAS Transactions on Systems and Control*, no. 2, pp. 54–63, 2012.

[5] A. H. Ribeiro, K. Tiels, J. Umenberger, T. B. Schön, and L. A. Aguirre, “On the smoothness of nonlinear system identification,” *Automatica*, vol. 121, p. 109158, 2020.

[6] I. Jaffel, O. Taouali, M. F. Harkat, and H. Messaoud, “Kernel principal component analysis with reduced complexity for nonlinear dynamic process monitoring,” *The International Journal of Advanced Manufacturing Technology*, vol. 88, no. 9-12, pp. 3265–3279, 2017.

[7] N.-P. Diamantopoulos, H. Nishi, W. Kobayashi, K. Takeda, T. Kakitsuka, and S. Matsuo, “On the complexity reduction of the second-order volterra nonlinear equalizer for im/dd systems,” *Journal of Lightwave Technology*, vol. 37, no. 4, pp. 1214–1224, 2018.

[8] F. J. Doyle, R. K. Pearson, and B. A. Ogunnaike, *Identification and control using Volterra models*. Springer, 2002.

[9] J. Schoukens and L. Ljung, “Nonlinear system identification: A user-oriented road map,” *IEEE Control Systems Magazine*, vol. 39, no. 6, pp. 28–99, 2019.

[10] R. Mortensen, “Nonlinear system theory: The volterra/wiener approach.” 1983.

[11] K. Li, J.-X. Peng, and G. W. Irwin, “A fast nonlinear model identification method,” *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1211–1216, 2005.

[12] S. B. Shiki, V. Lopes, and S. da Silva, “Identification of nonlinear structures using discrete-time volterra series,” *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 36, no. 3, pp. 523–532, 2014.

[13] S. Chang, X. Wei, F. Su, C. Liu, G. Yi, J. Wang, C. Han, and Y. Che, “Model predictive control for seizure suppression based on nonlinear auto-regressive moving-average volterra model,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 10, pp. 2173–2183, 2020.

[14] A. Khouaja, T. Garna, J. Ragot, and H. Messaoud, “Constrained predictive control of a siso nonlinear system based on third-order s-parafac volterra models,” *Transactions of the Institute of Measurement and Control*, vol. 39, no. 6, pp. 907–920, 2017.

[15] J.-H. Son and Y. Kim, “Probabilistic time series prediction of ship structural response using volterra series,” *Marine Structures*, vol. 76, p. 102928, 2021.

[16] M. Yetkin and Y. Kim, "Time series prediction of mooring line top tension by the narx and volterra model," *Applied Ocean Research*, vol. 88, pp. 170–186, 2019.

[17] Y. Zhang, Y. Rong, C. Yan, J. Liu, and X. Wu, "Kernel estimation of volterra using an adaptive artificial bee colony optimization and its application to speech signal multi-step prediction," *IEEE Access*, vol. 7, pp. 49 048–49 058, 2018.

[18] R. Kapoor, M. Tripathi *et al.*, "Detection and classification of multiple power signal patterns with volterra series and interval type-2 fuzzy logic system," *Protection and Control of Modern Power Systems*, vol. 2, no. 1, pp. 1–10, 2017.

[19] G. Feng, H. Li, J. Dong, and J. Zhang, "Face recognition based on volterra kernels direct discriminant analysis and effective feature classification," *Information Sciences*, vol. 441, pp. 187–197, 2018.

[20] Y. Chen, Y. Chen, and Q. Dai, "Fault diagnosis of roller bearing based on volterra and flexible polyhedron search algorithm," in *International Symposium on Computational Science and Computing*. Springer, 2018, pp. 246–251.

[21] Y. Deng and N. Liu, "Soft fault diagnosis in analog circuits based on bispectral models," *Journal of Electronic Testing*, vol. 33, no. 5, pp. 543–557, 2017.

[22] R. M. A. Zahoor and I. M. Qureshi, "A modified least mean square algorithm using fractional derivative and its application to system identification," *European Journal of Scientific Research*, vol. 35, no. 1, pp. 14–21, 2009.

[23] Y. K. Barapatre, A. Patnaik, and S. Nanda, "A modified variable step size for fractional least mean square algorithm for system identification and its fpga implementation," in *2020 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 2020, pp. 1553–1558.

[24] S. Da Silva, "Non-linear model updating of a three-dimensional portal frame based on wiener series," *International Journal of Non-Linear Mechanics*, vol. 46, no. 1, pp. 312–320, 2011.

[25] S. da Silva, S. Cogan, and E. Foltête, "Nonlinear identification in structural dynamics based on wiener series and kautz filters," *Mechanical Systems and Signal Processing*, vol. 24, no. 1, pp. 52–58, 2010.

[26] A. Chatterjee, "Identification and parameter estimation of a bilinear oscillator using volterra series with harmonic probing," *International Journal of Non-Linear Mechanics*, vol. 45, no. 1, pp. 12–20, 2010.

[27] J. Ahmad, M. Usman, S. Khan, I. Naseem, and H. J. Syed, "Rvp-flms: A robust variable power fractional lms algorithm," in *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*. IEEE, 2016, pp. 494–497.

[28] A. Atangana and J. Gómez-Aguilar, "Numerical approximation of riemann-liouville definition of fractional derivative: from riemann-liouville to atangana-baleanu," *Numerical Methods for Partial Differential Equations*, vol. 34, no. 5, pp. 1502–1523, 2018.

[29] B. Ross, *Fractional calculus and its applications: proceedings of the international conference held at the University of New Haven, June 1974*. Springer, 2006, vol. 457.

[30] B. Widrow and E. Walach, "Adaptive inverse control: A signal processing approach, reissue ed," 2008.