MODEL FREE ADAPTIVE CONTROL UNDER GRADIENT METHOD FOR ESTIMATION OF ENERGY AND PSEUDO-PARTIAL-DERIVATIVE FACTORS

Edwin C. Durandal*, Guilherme A. Rosa*, Rejane de B. Araujo*, Antonio A. R. Coelho*

*Federal University of Santa Catarina
Department of Automation and Systems 88040-900
Florianopolis, SC, Brazil

Emails: edwin.calla@posgrad.ufsc.br, guilherme.americo.rosa@posgrad.ufsc.br, rejane.barros@posgrad.ufsc.br, antonio.arc@ufsc.br

Abstract— The Model Free Learning Adaptive Control (MFLAC) is based on the pseudo-partial-derivative (PPD) calculated from the input and output signals of the system to be controlled and also using a fixed penalty factor that weights the control energy to provide a good behavior to the feedback process. This paper presents a new adaptation algorithm to adjust not only the PPD but also the energy factor (λ), both based on the Classical Gradient method (CG). With this proposal, an alternative control design is obtained for the on-line tuning of both parameters, ensuring good stability and robustness for the closed-loop system. Two numerical simulations on SISO discrete-time nonlinear plants demonstrate the efficiency and superiority of the proposed adaptive controller when using an estimation mechanism to adjust these parameters over the standard MFLAC. Performance indices are used to show the improvement of the behavior of the proposal control algorithm.

Keywords— Model free adaptive control, penalty/energy factor, pseudo-partial-derivative, gradient method, nonlinear system, autotuning.

1 Introduction

At the present time the industry is looking for ways to enhance the performance, quality of production and manufacturing time, among other things. For these industrial aspects, it is necessary to deal with uncertainties, nonlinearities, changes in the process dynamic, unknown plants, unexpected disturbances and/or noise in the feedback systems. A possible control solution for good closed-loop assessment is to use Model Free Adaptive Control (MFAC) (Hou and Wang, 2013; Hou and Jin, 2014).

The classical adaptive control design technique is based on the input and output measurements of the controlled process and sometimes we have no idea of the prior information of the process dynamic, physical process and system order structure that are not well understood. This is called a "black box" problem, where the model parameters are difficult and hard to be identified. From these meaning, the parsimony is important in adaptive control, because if there are too many parameters to estimate, computing becomes excessive and adaptation is slow. In addition, the adaptive control becomes especially important when standard controllers must be retuned in time-varying systems. This motivates our research in model free adaptive approaches that are achieving a favorable outcome in the control theory and experimental issues (Åström and Wittenmark, 1995; Cao and Hou, 2006; Bobál et al., 2006).

When the MFAC is used in a practical application, also robustness is fundamental to be achieved. In the traditional control theory, robustness is the ability to deal with uncertainties or unmodeled dynamics of the plant to be controlled. However, these loop aspects have no relevance to the MFAC synthesis because it is a special type of controller designed without any knowledge of the mathematical model information (Bu et al., 2012).

Model based control algorithms are usually implemented from the knowledge of the process dynamic and its operating point. These techniques can fail to provide satisfactory dynamic results when applied to processes with modeling uncertainties. When it comes to complex systems often physical processes are highly nonlinear or not fully understood, the stability is compromised when the process changes from one operating region to another (Al-Duwaish and Naeem, 2001; Bu et al., 2012).

Nowadays, MFAC is becoming popular in industry through manufactured products and is controlling a variety of industrial loops. Model free adaptive control has been hybridized and replacing the traditional PID and appears as a good closed-loop solution (Silveira et al., 2012). Some characteristics that justify the MFAC are: i) has good performance when dealing with modeling errors and hard control systems, ii) works directly on a single step, iii) allows the user to change the closed-loop dynamic, iv) has more intuitive parameters, v) is not necessary to retune controller specifications (Guang and Wenlong, 2011).

This paper proposes the Model Free Adaptive Control based on the Classical Gradient method (CG-MFAC) algorithm for autotuning the weighting parameters of the energy and the PPD factors, implementing and comparing with the standard Model Free Learning Adaptive Controller (MFLAC) of Cao et al. (2008). Numerical simulations are performed in two nonlinear systems to demonstrate the performance and the efficiency of this new control methodology.

This paper is organized as follows: Section 2 presents the basic definitions of the MFLAC. Section 3 describes the on-line adaptation method based on the Classical Gradient (CG) not only for the energy factor but also for the pseudo-partial-derivate. Simulation results are given in section 4. Conclusions are shown in section 5.

2 MFLAC Design

Consider the direct adaptive control of the following general SISO discrete-time nonlinear system

$$y(k+1) = f\left(y(k), \cdots, y\left(k-n_y\right), u(k), \cdots, u\left(k-n_u\right)\right) \quad (1)$$

where n_y and n_u are the degrees of the output, $y(k)$, and input, $u(k)$, respectively, and $f(\cdot)$ is a general nonlinear function.

The equation (1) can be rewritten as follows:

$$y(k+1) = f\left(Y(k), u(k), U(k-1)\right) \quad (2)$$

where $Y(k)$, $U(k-1)$ are the sets of system outputs and inputs up to sampling period k and $k-1$. For deriving the MFLAC design it is necessary to ensure the following assumptions:

Assumption 1 *Systems* (1) *and* (2) *are observable and controllable by the expected bounded system output signal* $y(k+1)$*, where exists a bounded feasible control input signal which drives the system output equal to the expected output.*

Assumption 2 *The partial derivative of* $f(\cdot)$ *with respect to the control input* $u(k)$ *is continuous.*

Assumption 3 *System* (1) *is a generalized Lipschitz, i.e., satisfying* $|\Delta y(k+1)| \leq b|\Delta u(k)|$ *for any* k *and* $\Delta u(k) \neq 0$*, where* $\Delta y(k+1) = y(k+1) - y(k)$*,* $\Delta u(k) = u(k) - u(k-1)$ *and* b *is a constant. This assumption has certain limitations in the rate of the change of the system output. Obviously, it includes a class of a nonlinear plant.*

Next, the following theorem must be included:

Theorem 1 *For the nonlinear system* (2)*, in order to satisfy Assumptions 1, 2 and 3, there must exist a variable* $\phi(k)$*, called pseudo-partial-derivative (PPD), when* $\Delta u(k) \neq 0$ *we have:*

$$\Delta y(k+1) = \phi(k)\Delta u(k) \quad (3)$$

where $|\phi(k)| \leq b$*.*

Equation (3) is a linear dynamic system with slowly time-varying parameter when $\Delta \mathrm{u}(k) \neq 0$ and the sampling period is small enough. Therefore, in designing the control system, in addition to the condition $\Delta \mathrm{u}(k) \neq 0$ some free adjustable parameters should be added to the input control criterion function. This is to keep the input control rate of changing within acceptable values.

The weighted one-step-ahead control input criterion function, defined by Hou and Huang (1997) and Cao et al. (2008), is used as the starting point for deriving the control law such as

$$\begin{aligned} J(u(k)) = &\left[y_r(k+1) - y(k+1)\right]^2 + \\ &\lambda[u(k) - u(k-1)]^2 \end{aligned} \quad (4)$$

where $y_r(\cdot)$ is the setpoint and the parameter estimation criterion function is given by

$$\begin{aligned} J(\phi(k)) = &\left[y_r(k) - y(k-1) - \phi(k)\Delta \mathrm{u}(k-1)\right]^2 + \\ &\mu\left[\phi(k) - \hat{\phi}(k-1)\right]^2 \end{aligned} \quad (5)$$

with λ and μ positive parameters to be tuned. Substituting equation (3) in (4) and (5) the minimization of both criterion functions gives the control law as

$$u(k) = u(k-1) + \frac{\alpha\phi(k)}{\lambda + \phi(k)^2}\left\{y_r(k+1) - y(k)\right\} \quad (6)$$

and the PPD estimation is calculated as follows:

$$\begin{aligned} \hat{\phi}(k) = &\hat{\phi}(k-1) + \frac{\sigma\Delta \mathrm{u}(k-1)}{\mu + \Delta \mathrm{u}(k-1)^2}\cdot \\ &\left\{\Delta \mathrm{y}(k) - \hat{\phi}(k-1)\Delta \mathrm{u}(k-1)\right\} \end{aligned} \quad (7)$$

The values of α and σ are defined as step-size constant series. These parameters are added in order to generalize them and will be used for the analytical stability proofs (λ is adjusting the energy factor on the incremental control law and can change the loop stability). Computer simulation results show that a suitable choice of λ can improve the performance of the controlled system.

Additional details of the theoretical contents and the mathematical proofs of the MFLAC can be found in Hou and Jin (2014).

3 CG-MFAC Control Algorithm

It is possible to obtain improvements over the MFLAC using an on-line adaption for the energy as well as the pseudo-partial-derivate factors, using the CG-MFAC algorithm. Next, both possibilities are presented separately for a better understanding for the user over the implementation and stability.

3.1 On-line adaption design for λ using CG

The control design from equations (3), (4) and (5) is implemented with an on-line adaptive estimation for the energy factor.

Using equations (3) and (4) with the definition and concepts of the gradient descent algorithm (Boyd and Vandenberghe, 2004; Nocedal and Wright, 2006), the energy factor may be optimized as follows:

$$\lambda(k+1) = \lambda(k) - \beta_{\mathrm{CG1}} \nabla J(u(k)) \tag{8}$$

where β_{CG1} is a step-size constant which determines the stability and converge rate of the algorithm.

The following mathematical operation is done in order to find the derivative part required in equation (8) (adaptive control signal weighting) such as:

$$\begin{aligned} \nabla J(u(k)) &= \frac{\partial J(u(k))}{\partial \lambda(k)} \\ &= \frac{\partial J(u(k))}{\partial u(k)} \frac{\partial u(k)}{\partial \lambda(k)} \end{aligned} \tag{9}$$

Once these operations are calculated in equation (9), it is possible to replace these values in equation (8), and get

$$\lambda(k+1) = \lambda(k) - \beta_{\mathrm{CG1}} \left[A_{\mathrm{CG1}} B_{\mathrm{CG1}} C_{\mathrm{CG1}}\right] \tag{10}$$

where the variables A_{CG1}, B_{CG1} and C_{CG1} are described in equations (11), (12) and (13), respectively:

$$A_{\mathrm{CG1}} = (y_r(k+1) - y(k)) \tag{11}$$

$$\begin{aligned} B_{\mathrm{CG1}} =& y_r(k+1)\phi(k) - y(k)\phi(k) - \\ & \left(\phi(k)^2 + \lambda(k)\right) \Delta \mathrm{u}(k) \end{aligned} \tag{12}$$

$$C_{\mathrm{CG1}} = \frac{2\alpha\phi(k)}{\left(\lambda(k) + \phi(k)^2\right)^2} \tag{13}$$

3.2 On-line adaption design for PPD using CG

Equations (3) and (4) are implemented in this control design together with an on-line adaptive algorithm for evaluating the pseudo-partial-derivative factor.

Now, following the same steps shown in section 3.1, the PPD is optimized as follows:

$$\phi(k+1) = \phi(k) - \beta_{\mathrm{CG2}} \nabla J(u(k)) \tag{14}$$

where β_{CG2} is a step-size constant.

A mathematical operation is done to find out the derivative part required in equation (14) and is given by

$$\begin{aligned} \nabla J(u(k)) &= \frac{\partial J(u(k))}{\partial \phi(k)} \\ &= \frac{\partial J(u(k))}{\partial u(k)} \frac{\partial u(k)}{\partial \phi(k)} \end{aligned} \tag{15}$$

Once the operations are calculated in equation (15), then can be replaced in equation (14) to obtain

$$\phi(k+1) = \phi(k) - \beta_{\mathrm{CG2}} \left[A_{\mathrm{CG2}} B_{\mathrm{CG2}} C_{\mathrm{CG2}}\right] \tag{16}$$

where the variables A_{CG2}, B_{CG2} and C_{CG2} are derived in equations (17), (18) and (19), respectively:

$$A_{\mathrm{CG2}} = (y_r(k+1) - y(k)) \tag{17}$$

$$\begin{aligned} B_{\mathrm{CG2}} =& y_r(k+1)\phi(k) - y(k)\phi(k) - \\ & \Delta \mathrm{u}(k)\phi(k)^2 - \lambda \Delta \mathrm{u}(k) \end{aligned} \tag{18}$$

$$C_{\mathrm{CG2}} = \frac{2\alpha\left(-\lambda + \phi(k)^2\right)}{\left(\lambda + \phi(k)^2\right)^2} \tag{19}$$

4 Simulation Results

The efficiency and performance of the proposed CG-MFAC algorithm with the implementation of an adaptive tuning procedure for energy and PPD factors are analyzed by simulation scenarios in two different discrete-time nonlinear systems, compared against the standard MFLAC.

For the first numerical simulation the plant model, shown in Hou and Jin (2014), is used. The second simulation utilizes the Heat Exchanger (HE) model as described in Al-Duwaish and Naeem (2001).

The parameters of both control design are tuned with the same values in order to analyze and to compare the behavior and effectiveness of the closed-loop nonlinear systems.

4.1 Discrete-time nonlinear system

Consider the nonlinear model as described in equation (20) and used in Hou and Jin (2014):

$$y(k+1) = \frac{y(k)}{1 + y(k)^2} + u(k)^3 \tag{20}$$

where $y(\cdot)$ is the output and $u(\cdot)$ is the input. The static curve of the system is shown in Figure 1.

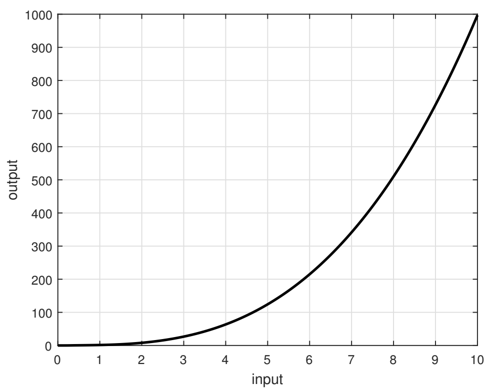

Figure 1: Static characteristic of the discrete-time nonlinear system.

The common parameters of the MFLAC and CG-MFAC are set as follows: $\alpha = 0.5$, $\lambda(1) = 3$,

ISSN 2525-8311

$\phi(1) = 0.5$. It is important to clarify that in the MFLAC the penalty factor λ has a constant value, with $\lambda = \lambda(1)$. For the numerical gradient estimation of the energy and pseudo-partial-derivative factors of the CG-MFAC algorithm the tuned parameters are $\beta_{\text{CG1}} = 0.01$ and $\beta_{\text{CG2}} = 0.001$, respectively. For the PPD of the MFLAC algorithm the parameters are tuned with $\sigma = 0.5$ and $\mu = 0.5$.

Output and control responses are shown in Figure 2, where it is possible to note the speed and smoothness of the convergence of the CG-MFAC over the MFLAC when the magnitude of the reference is varying in different operating regions.

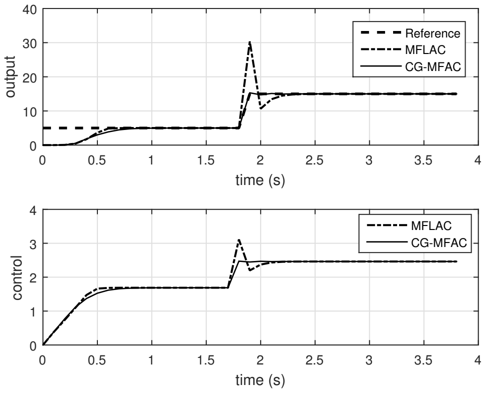

Figure 2: Output and control of discrete-time nonlinear system.

In order to compare and to measure the efficiency of both controllers, the IAE (Integrated Absolute Error) and TVC (Total Variation of Control) performance indices are calculated from 1.5 to 2.5 s. This sampling interval was chosen because it does not carry the initial conditions. For clarity better the comparison, Table 1 shows the significant improvement of the proposed control algorithm. The on-line adaptation of the energy factor is shown in Figure 3 with stable behavior. The evolution of the adaptive parameter, λ, is decreasing over the time and it is providing faster and smoother response of the output system.

Table 1: Performance indices for the controllers.

	MFLAC	*CG-MFAC*
IAE	2.1960	0.0782
TVC	0.2603	0.0847

Figures 4 and 5 show the on-line adaptation of the pseudo-partial-derivative of the MFLAC and CG-MFAC, respectively. Ensuring convergence and stability. The small variation observed in Figure 5 is directly related to the value used in β_{CG2}, this is not the case for the variation of PPD on the standard MFLAC where goes from 0.5 to 12.4, for the same reference tracking.

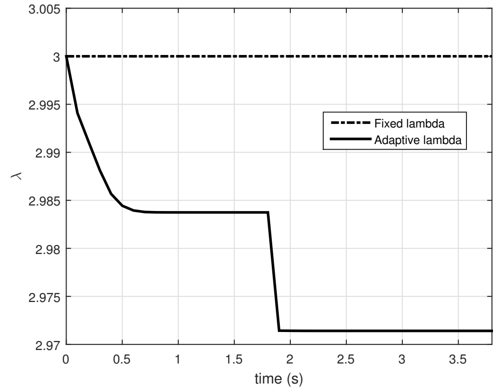

Figure 3: Energy factor of the CG-MFLAC.

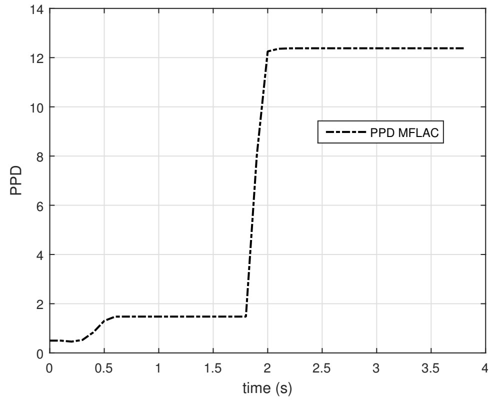

Figure 4: Pseudo-partial-derivative of the MFLAC.

4.2 Heat Exchanger System

The Heat Exchanger process (HE) is modeled as a Hammerstein system with equations (21) and (22) and is presented in Al-Duwaish and Naeem (2001) as follows:

$$x(k) = -31.549u(k) + 41.732u(k)^2 - 24.201u(k)^3 + 68.634u(k)^4 \quad (21)$$

$$y(k) = \frac{0.207z^{-1} - 0.1764z^{-2}}{1 - 1.608z^{-1} + 0.6385z^{-2}} x(k) \quad (22)$$

where $x(k)$ is the static nonlinearity, $u(k)$ is the process flow rate and $y(k)$ is the process exit temperature. The input to the process is constrained between [0,1].

Figure 6 shows the static curve of the HE. The major challenge is to keep the process output as close as possible to the reference when moving the nominal operating point in the nonlinear zone.

Tuning parameters of the MFLAC and CG-MFAC are: $\alpha = 0.15$, $\lambda(1) = 4.5$, $\phi(1) = 0.5$. Recalling that in the case of MFLAC λ has a constant value, with $\lambda = \lambda(1)$. For the numerical gradient estimation of the energy and pseudo-partial-derivative factors the tuning is selected with $\beta_{\text{CG1}} = 0.1$ and $\beta_{\text{CG2}} = 0.001$, respectively. However, for the PPD of the MFLAC algorithm $\sigma = 0.5$ and $\mu = 0.5$.

ISSN 2525-8311

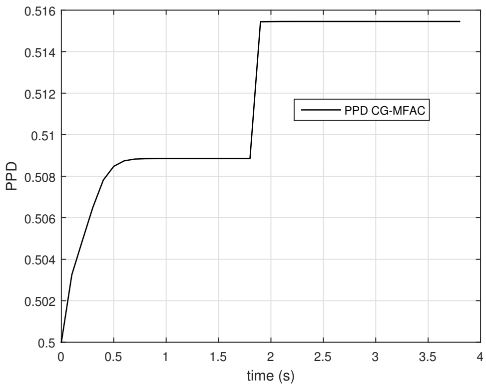

Figure 5: Pseudo-partial-derivative of the CG-MFAC.

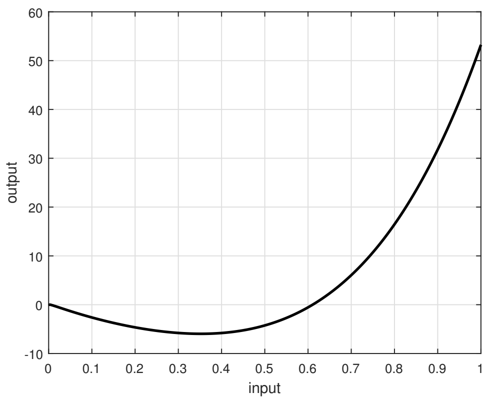

Figure 6: Static characteristic of the heat exchanger.

Output and control signals are shown in Figure 7, where the CG-MFAC converges faster than the original MFLAC, dealing better with the initialization of the controller parameters and the nonlinear characteristic of the closed-loop system. The control effort is almost equal, differing on all the simulations by 17.68%. When both outputs are on a stable point an output disturbance is introduced with magnitude equal to 10% of the maximum value of the reference. It can be noted that the disturbance rejection, for both controllers, is achieved properly. The ITAE (Integral Time Absolute Error) performance index is calculated from 75 to 85 s as shown in Table 2.

In like manner, the efficiency of both controllers are compared as described in section 4.1. IAE and TVC performance indices are calculated throughout the simulation. The reason for this is that the objective for the HE is to track the reference as close as possible as stated in the beginning of this section. Table 2 shows these values, where CG-MFAC achieved better closed-loop results for this objective.

Figure 8 shows the on-line adaptation of the energy weighting, λ, where it is following a pattern according the setpoint switching. This is the reason for the faster convergence of the CG-MFAC

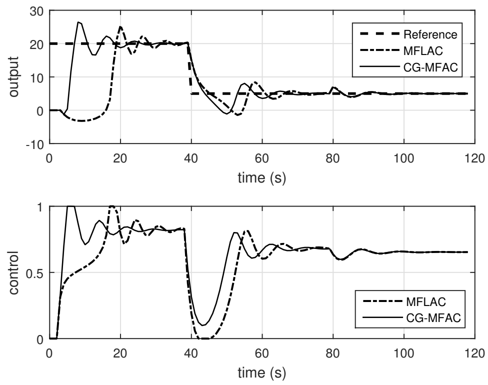

Figure 7: Output and control of HE.

Table 2: Performance indices for the controllers.

	$MFLAC$	$CG\text{-}MFAC$
IAE	3.7508	1.5574
TVC	4.3387	3.9552
$ITAE$	6.2121	5.2787

algorithm. It is possible to observe that the energy factor changes smoothly when the disturbance is introduced.

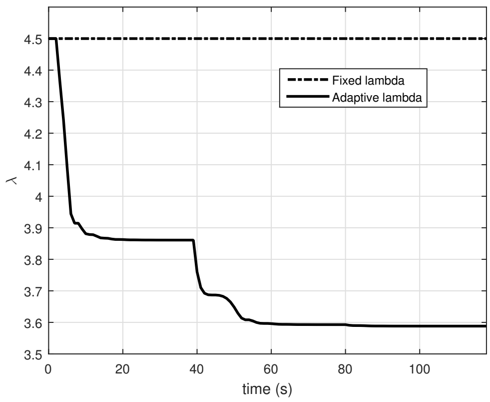

Figure 8: Energy factor of the CG-MFAC.

The on-line adaptation of the pseudo-partial-derivative of the MFLAC and CG-MFAC are shown in Figures 9 and 10, respectively. The PPD value calculated in the CG-MFAC controller is retuned from 0.5 to 0.57 and has fast convergence speed as shown in Figure 10. Under the same condition of reference tracking case the PPD value, calculated via the MFLAC, is moving from 0.5 to 6.45.

5 Conclusions

This paper has proposed an adaptive hybridization for the MFLAC, implementing two factors by using the gradient descent method in order to provide a faster convergence rate and to ensure a closed-loop system stability. The CG-MFAC has

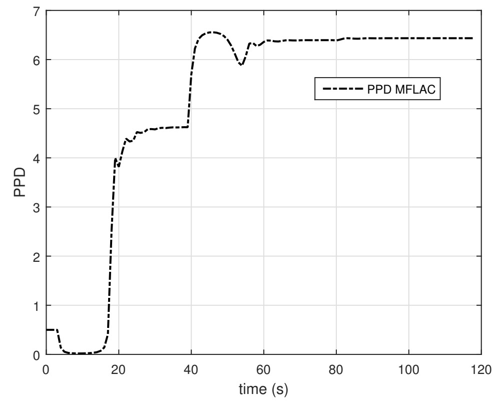

Figure 9: Pseudo-partial-derivative of the MFLAC.

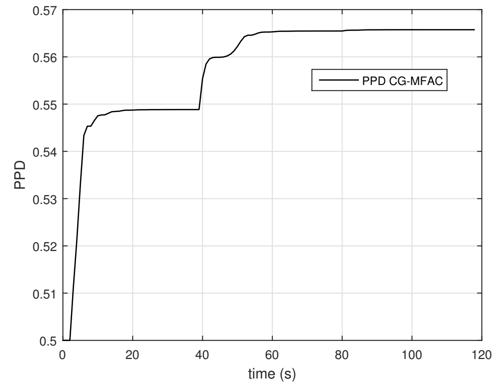

Figure 10: Pseudo-partial-derivative of the CG-MFAC.

more intuitive parameters to be tuned, simplifying and reducing the procedure of the trial and error on the initial parameter settings, which is depending on the desired characteristic of the closed-loop system. This autotuning mechanism successfully improves the convergence, stability and robustness. Numerical results have shown the efficiency of the proposed CG-MFAC design. Although has been not shown, the proposed controller can be used in practical essays.

However, further control essays, including laboratory hardware examples, and stability analysis are required in nonlinear systems with hard complexities for different operating points in order to assess the robustness and the performance of the control algorithm under difficult conditions.

Future work will explore new aspects of the gradient descent method to become faster the convergence of the MFAC algorithm and to improve the closed-loop robustness aspects in various nonlinear systems. In addition, different fitness functions will be considered to derive the control law.

6 Acknowledgements

This work has been supported by CNPq, GPqTCA and PPGEAS/UFSC.

References

Al-Duwaish, H. and Naeem, W. (2001). Nonlinear model predictive control of Hammerstein and Wiener models using genetic algorithms, *Control Applications, 2001.(CCA'01). Proceedings of the 2001 IEEE International Conference on IEEE* pp. 465–469.

Åström, K. J. and Wittenmark, B. (1995). *Adaptive control*, Addison-Wesley.

Bobál, V., Böhm, J., Fessl, J. and Machácek, J. (2006). *Digital Self-tuning Controllers: Algorithms, Implementation and Applications*, Springer Science & Business Media.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*, Cambridge university press.

Bu, X., Yu, F., Hou, Z. and Zhang, H. (2012). Model-free adaptive control algorithm with data dropout compensation, *Mathematical Problems in Engineering* pp. 1–14.

Cao, R., Bai, L. and Hou, Z. (2008). Study on model-free learning adaptive control in permanent magnet linear motor, *Chinese Control and Decision Conference* pp. 2946–2949.

Cao, R.-m. and Hou, Z.-s. (2006). Simulation study on model-free control method in linear motor control system, *Journal of System Simulation* **10**: 045.

Guang, M. and Wenlong, C. (2011). Design and application of model free controller, *Electric Information and Control Engineering (ICEICE), 2011 International Conference on* pp. 2058–2061.

Hou, Z. and Huang, W. (1997). The model-free learning adaptive control of a class of SISO nonlinear systems, *American Control Conference, 1997. Proceedings of the 1997* **1**: 343–344.

Hou, Z. and Jin, S. (2014). *Model free adaptive control: theory and applications*, CRC press.

Hou, Z.-S. and Wang, Z. (2013). From model-based control to data-driven control: survey, classification and perspective, *Information Sciences* **235**: 3–35.

Nocedal, J. and Wright, S. (2006). *Numerical optimization*, Springer Science & Business Media.

Silveira, A. S., Coelho, A. A. R. and Gomes, F. J. (2012). Model free adaptive PID controllers applied to the benchmark PID12, *IFAC Conference on Advances in PID Control, Brescia, Italy* pp. 370–375.