

Indirect Adaptive Model Predictive Control Supervised by Fuzzy Logic

Jerry MAMBOUNDOU, Nicolas LANGLOIS

Automatic Control & Systems Research Group
Institut de Recherche en Systèmes Electroniques EMbarqués, IRSEEM
Saint Etienne du Rouvray, France
{jerry.mamboundou, nicolas.langlois}@esigelec.fr

Abstract—In order to get correct results using the model predictive control, one must find the suitable values of its tuning parameters. In many case this task is achieved by empirical methods. It's sufficient to control a physical system which internal parameters remain the same. The chosen controller parameters continue to provide satisfactory performances against small system variations. They become quickly inappropriate when strong variations occur in the process. One of the classical approaches consists to identify online the system model to update its controller based on this model. This solution doesn't ensure always best results. In this paper, we show how one can perform a suitable control to a system with variable parameters applying a fuzzy-logic-supervised predictive control. The fuzzy logic supervisor fulfills the online tuning of the predictive control parameters. Thus, we carry out a comparison between these two strategies.

Keywords- MPC, Augmented state space model, Fuzzy Logic Supervisor, LTV system

I. INTRODUCTION

Model predictive control (MPC) is a powerful tool used to control a large range of physical systems. There are currently many industrial applications where the MPC has proved its efficiency having various difficulty complexities [1]. However, these good results depend on the best appropriate values of the MPC parameters which are the horizons of prediction, control and the weighting factor.

Many previous studies on this topic show that the MPC parameters values are chosen offline [2, 3]. Some methods use algorithms based on traditional stability criteria such as the gain and phase margins [2, 3]. Others intend to determine them by online tedious algorithms based on genetic [4] and mixed-integer linear program [5] optimizations. [6] proposes to build the adaptive MPC by achieving the online process identification to update the process model and compute the optimal control sequence after. In [7], the *Indirect Adaptive MPC (IAMPC)* approach is proposed. Nevertheless, when certain process parameters vary strongly in the time, this method is not able to respect the desired performances.

In this paper we will intend to overcome its limits thanks to the fuzzy-logic. Our approach determines the suitable values

of the MPC parameters thanks to a fuzzy logic supervisor which input variables are the measurable performance criteria. From [8], we elaborate this control strategy in which predictive control is based on a state-space model and the supervisor gets a decision logic less complex. In addition, we reformulate two of the above criteria to reduce the computational load.

The paper is outlined as follows: section II presents the classical *IAMPC* while section III highlights the *IAMPC supervised by Fuzzy Logic*. In section IV, we carry out a comparative study of the obtained performances by applying both approaches on a Linear Time-Variant (LTV) system.

II. INDIRECT ADAPTIVE MODEL PREDICTIVE CONTROL

A. Discrete-time augmented state-space model

Let us consider a single-input and single-output (SISO) system represented by its discrete-time state-space model

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k), \\ y(k) &= C_d x(k), \end{aligned} \quad (1)$$

where k is the sampling instant assumed to be positive. [3] introduces the differences of the state vector and control variable such as

$$\begin{aligned} \Delta x(k+1) &= x(k+1) - x(k), \\ \Delta u(k+1) &= u(k+1) - u(k). \end{aligned}$$

Let us rewrite the state equation in (1) as follows

$$\begin{aligned} \Delta x(k+1) &= A_d (x(k) - x(k-1)) + B_d (u(k) - u(k-1)) \\ &= A_d \Delta x(k) + B_d \Delta u(k). \end{aligned} \quad (2)$$

Which leads to create the new state variable vector defined by

$$x_a(k) = [\Delta x(k) \quad y(k)]^T.$$

When one includes $\Delta x(k+1)$ in the second equation of (1), the measurement equation becomes

$$\begin{aligned} y(k+1) - y(k) &= C_d (x(k+1) - x(k)) = C_d \Delta x(k+1) \\ &= C_d A_d \Delta x(k) + C_d B_d \Delta u(k) \end{aligned} \quad (3)$$

The combinason of (1) and (3) leads to the following discrete-time state-space model:

$$\begin{aligned} x_a(k+1) &= A x_a(k) + B \Delta u(k) \\ y(k) &= C x_a(k), \end{aligned} \quad (4)$$

where

$$\begin{aligned} x_a(k+1) &= \begin{bmatrix} \Delta x(k+1) \\ y(k+1) \end{bmatrix}, x_a(k) = \begin{bmatrix} \Delta x(k) \\ y(k) \end{bmatrix}, \\ A &= \begin{bmatrix} A_d & O_d^T \\ C_d A_d & 1 \end{bmatrix}, B = \begin{bmatrix} B_d \\ C_d B_d \end{bmatrix}, C = [O_d \quad 1], \\ O_d &= [0 \quad 0 \quad \dots \quad 0]. \end{aligned}$$

O_d is a row matrix of size n_A such as $(n_A \times n_A)$ are the dimensions of square matrix A_d . The set of matrices (A, B, C) forms the discrete-time augmented state-space model [3].

B. Predictive control formulation

Let be HI, HP, HC , and λ the horizons of initialization, prediction, control and the weighting factor respectively. The control sequence ΔU and the sets of desired Y_{des} and predicted $\hat{Y}$ outputs are represented under a vector form as follows:

$$\Delta U^T = [\Delta u(k) \quad \Delta u(k+1) \quad \dots \quad \Delta u(k+HC-1)],$$

$$Y_{des}^T = [y_{des}(k+1) \quad y_{des}(k+2) \quad \dots \quad y_{des}(k+HP)],$$

$$\hat{Y}^T = [\hat{y}(k+1|k) \quad \hat{y}(k+2|k) \quad \dots \quad \hat{y}(k+HP|k)],$$

To obtain the control signal, we need to minimize the following cost function [3]:

$$J = (Y_{des} - \hat{Y})^T (Y_{des} - \hat{Y}) + \Delta U^T \bar{R} \Delta U, \quad (5)$$

where $\bar{R} = \lambda I$ and I is the identity matrix of dimension $(HP \times HP)$. It's possible to express the vector $\hat{Y}$ such as

$$\hat{Y} = F x_a(k) + \Phi \Delta U, \quad (6)$$

where

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{HP} \end{bmatrix}, \Phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2 B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ CA^{HP-1} B & CA^{HP-2} B & CA^{HP-3} B & \dots & CA^{HP-HC} B \end{bmatrix}$$

Putting (6) in (5), we obtain a complete form of the criterion which the partial derivative with respect to ΔU gives :

$$\frac{\partial J}{\partial \Delta U} = -2\Phi^T (Y_{des} - F x_a(k)) + 2(\Phi^T \Phi + \bar{R}) \Delta U. \quad (7)$$

We find the optimal control sequence ΔU by solving the following equation:

$$\frac{\partial J}{\partial \Delta U} = 0, \quad (8)$$

what leads to

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (Y_{des} - F x_a(k)). \quad (9)$$

According to the receding horizon principle, the control signal applied to the process will be:

$$\begin{aligned} u(k) &= u(k-1) + \Delta U \\ u(k) &= u(k-1) + (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (Y_{des} - F x_a(k)) \end{aligned} \quad (10)$$

The state vector $x_a(k)$ constitutes the current plant information required to estimate the optimal control signal.

C. Recursive least square method

We illustrate the online model identification [9] by writing the discrete-time model of the plant such as

$$y_{mod}(k) = -\sum_{i=1}^{n_1} a_i y(k-i) + \sum_{i=1}^{n_2} b_i u(k-d-i) \quad (11)$$

where a_i and b_i are the unknown parameters. The model output $y_{mod}(k)$ can be also written under the form of a scalar product between the unknown parameter vector

$$\theta^T = [a_1, \dots, a_{n_1}, b_1, \dots, b_{n_2}]$$

and the vectors of measurements called the plant model regressor vector

$$\phi(k) = \begin{bmatrix} -y(k-1) \\ \vdots \\ -y(k-n_1) \\ u(k-d-1) \\ \vdots \\ u(k-d-n_2) \end{bmatrix}.$$

What leads to

$$y_{\text{mod}}(k) = \theta^T \phi(k). \quad (12)$$

One can define now the prediction error (a priori) as

$$\varepsilon^o(k) = y(k) - \theta^T(k-1)\phi(k-1). \quad (13)$$

The minimization of the cost function

$$J_\theta = \sum_{i=1}^k \left(\varepsilon^o(k) \right)^2 \quad (14)$$

gives the optimal parameter vector

$$\theta(k) = P(k) \sum_{i=1}^k y(i)\phi(i-1) \quad (15)$$

where

$$P(k) = \left[\sum_{i=1}^k \phi(i-1)\phi^T(i-1) \right]^{-1}.$$

The recursive least squares (RLS) steps can be expressed by

$$\varepsilon(k+1) = \frac{y(k) - \theta^T(k)\phi(k)}{1 + \phi(k)^T P(k)\phi(k)} \quad (16)$$

$$P(k+1) = P(k) - \frac{P(k)\phi(k)\phi(k)^T P(k)}{1 + \phi(k)^T P(k)\phi(k)} \quad (17)$$

$$\theta(k+1) = \theta(k) + P(k+1)\phi(k)\varepsilon(k+1). \quad (18)$$

D. Parameter adaptation algorithm

The recursive formula of the adaptation gain $P(k+1)$ given by (17) is generalized by introducing two weighting sequences $\lambda_1(k)$ and $\lambda_2(k)$, as indicated below:

$$P(k+1) = \frac{1}{\lambda_1(k)} \left[P(k) - \frac{P(k)\phi(k)\phi(k)^T P(k)}{\alpha(k) + \phi(k)^T P(k)\phi(k)} \right] \quad (19)$$

where

$$0 < \lambda_1(k) \leq 1, \quad 0 < \lambda_2(k) \leq 2, \quad P(0) > 0 \text{ and } \alpha(k) = \frac{\lambda_1(k)}{\lambda_2(k)}.$$

To improve the results of the previous identification method, [9] associates it to the constant trace technique as parameter adaptation algorithm (PAA). The principle is to move in the optimal direction of the RLS by maintaining the trace of matrix P approximately constant.

Imposing the ratio $\alpha(k)$, the two factors $\lambda_1(k)$ and $\lambda_2(k)$ are automatically calculated at each step to ensure

$$\text{tr}(P(k+1)) = \text{tr}(P(k)) = \text{tr}(P(0)) = (n_1 + n_2)P_0 \quad (20)$$

in which GI is the initial gain and $P(0) = P_0 I$ with I the identity matrix of dimension $((n_1 + n_2) \times (n_1 + n_2))$.

Fig. 1 helps us to visualize and understand well the *IAMPC* strategy wholeness. For simplicity reasons, we include the PAA in the block named RLS.

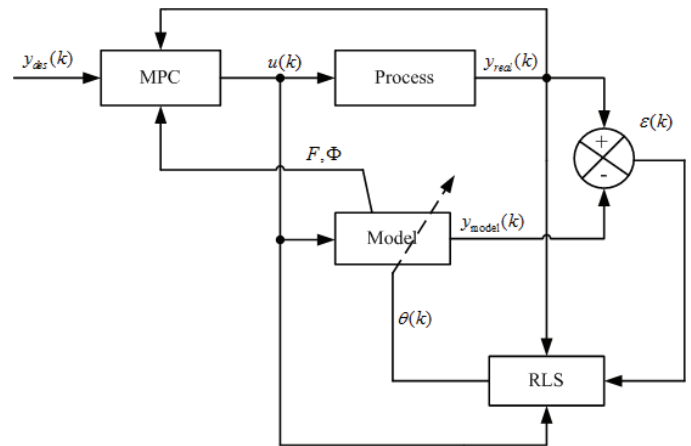


Figure 1. Equivalent structure of the IAMPC.

III. INDIRECT ADAPTIVE MODEL PREDICTIVE CONTROL SUPERVISED BY FUZZY LOGIC

Inspired by [8], it is possible to modify directly the tuning parameters of the predictive controller according to precise performance criteria. We consider four criteria estimated from controlled plant and model data. These four criteria are the inputs of the fuzzy-logic supervisor and the MPC parameters its outputs.

A. Performance criteria

We choose to evaluate the controlled plant rapidity by the controlled output convergence time (OCT). Considering the controlled output y_{cont} as a percentage γ of the desired output y_{des} , we can establish a relative error threshold (RET). It is defined such as

$$RET = \frac{y_{cont} - y_{des}}{y_{des}} = \frac{y_{des}(\gamma - 1)}{y_{des}} = \gamma - 1$$

OCT is the time interval from which the below inequality is verified:

$$\left| (y_{cont} - y_{des}) / y_{des} \right| \leq RET, \quad (21)$$

Subsequently, it remains close to 0, if y_{cont} approaches y_{des} . It remains equal to -0.01, if y_{cont} is not close to y_{des} .

The controlled process stability is measured by the stability degree indicator (SDI). It is the difference between the radius of the unit circle and the modulus of the pole most remote from the unit circle center. What gives

$$SDI(k) = 1 - \max(|p_1|, |p_2|, \dots, |p_{n_A}|), \quad (22)$$

where $p_1, p_2, \dots, p_{n_A}$ are the eigenvalues of matrix A_d . When $0 < SDI < 1$, the system is stable. It will be in limit of stability if $SDI = 0$. It will become unstable if $SDI < 0$.

The control signal variance (VARU) allows observing the mean value of the square deviations of $u(k)$ from its average. It's expressed by

$$VARU(k) = \overline{u(k)^2} - \left[\overline{u(k)} \right]^2 \quad (23)$$

with

$$\overline{u(k)^2} = \overline{u(k-1)^2} + \frac{1}{2} (u(k)^2 - u(k-1)^2),$$

and

$$\left[\overline{u(k)} \right]^2 = \left[\overline{u(k-1)} + \frac{1}{2} (u(k) - u(k-1)) \right]^2.$$

The trace of $\Phi^T \Phi$ provides an excellent tool to evaluate the computation conditioning of the optimal control signal [2, 8].

B. Normalization

Generally speaking, there is a difference between the real values of the mentioned performance criteria and the admissible values of the fuzzy input variables in their respective discourse universes from 0 to 1. One notices the same difference for the MPC parameters which are the fuzzy output variables. Thereby, one needs to normalize the values of these criteria and the obtained MPC horizons.

Let us add the subscript n to the criteria abbreviations OCT_n , SDI_n , $VARU_n$, $trace(\Phi^T \Phi)_n$ as their normalized values. Assigning them maximal and minimal values that fixe the expected performance ranges, SDI_n is evaluated by the simple operation (available for the other criteria)

$$SDI_n(k) = (SDI(k) - SDI_{min}) / SDI_{max}. \quad (24)$$

While the normalized supervisor outputs are designated by HP_n and HC_n respectively for HP and HC . Supposing they have maximal and minimal values that include the suitable horizons, the proposed control horizon is

$$HC(k) = round[HC_n(k)(HC_{max} - HC_{min}) + HC_{min}], \quad (25)$$

where *round* is the function which rounds the real number to its nearest integer. We use the same calculus for HP . [8] shows how one can find the right numerical range of HP

$$HP_{min} = 2 \sup \{n_2 + d; n_1 + 1\},$$

$$HP_{max} = HP_{min} + 10.$$

And those of HC

$$HC_{min} = n_1,$$

$$HC_{max} = HP - HI.$$

C. Fuzzy logic supervisor

The purpose of the supervisor is to propose the appropriate values of HP , HC and λ depending on the plant variations. This supervisor is made of three sub-systems.

The first suggests the HP value based on the variance of $u(k)$ and the controlled OCT. The second gives the HC value according to the previous criteria. The third offers the λ value

in function of the trace of $\Phi^T \Phi$ and the controlled SDI. We illustrate this on fig. 2.

To be able to make proposals, the fuzzy logic supervisor involves the if-then rules using the Mamdani's fuzzy inference system. Thus, each linguistic input has three fuzzy sets and each linguistic output owns five fuzzy sets. All the membership functions are triangular. Tab. I, II and III present the inference tables corresponding to each supervisor sub-system.

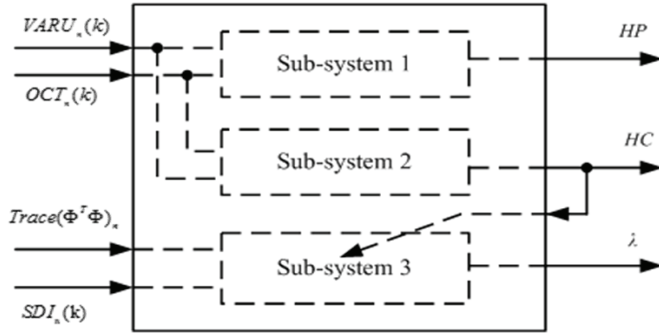


Figure 2. Sub-systems of the fuzzy logic supervisor.

TABLE I. TABLE OF RULES DEFINING HP

		$OCT_n(k)$		
		<i>Big</i>	<i>Middle</i>	<i>Small</i>
$VARU_n(k)$	<i>Small</i>	Very Small	Small	Middle
	<i>Middle</i>	Small	Middle	Big
	<i>Big</i>	Middle	Big	Very Big

TABLE II. TABLE OF RULES DEFINING HC

		$OCT_n(k)$		
		<i>Big</i>	<i>Middle</i>	<i>Small</i>
$VARU_n(k)$	<i>Small</i>	Middle	Big	Very Big
	<i>Middle</i>	Small	Middle	Big
	<i>Big</i>	Very Small	Small	Middle

TABLE III. TABLE OF RULES DEFINING λ

		$SDI_n(k)$		
		<i>Big</i>	<i>Middle</i>	<i>Small</i>
$\frac{\text{Trace}(\Phi^T \Phi)_n}{HC}$	<i>Small</i>	Middle	Small	Very Small
	<i>Middle</i>	Big	Middle	Small
	<i>Big</i>	Very Big	Big	Middle

D. Control architecture

The elaborated control architecture consists of the *IAMPC* control law enhanced by the fuzzy logic supervisor (FLS). To fulfill it this global strategy to a process, one requires to coordinate the activations of each task. The modeling error $\varepsilon(k)$ compares the real and the model outputs to guarantee

good predictions. The model will track well the plant output when this error remains null. Otherwise, we take into account two adaptation cases through two positive thresholds called S_1 and S_2 ($S_1 < S_2$).

Firstly, the condition $\varepsilon(k)$ upper than or equals to S_1 activates the RLS method enhanced by the PAA. It updates the former model parameters.

Secondly, when $\varepsilon(k)$ becomes greater than or equals to S_2 , we activate the supervisor while the first adaptation still works (see fig. 3). This second updating suggests the best appropriate MPC parameters. As soon as $\varepsilon(k)$ becomes lower than or S_2 , we take back the original values of HP , HC and λ .

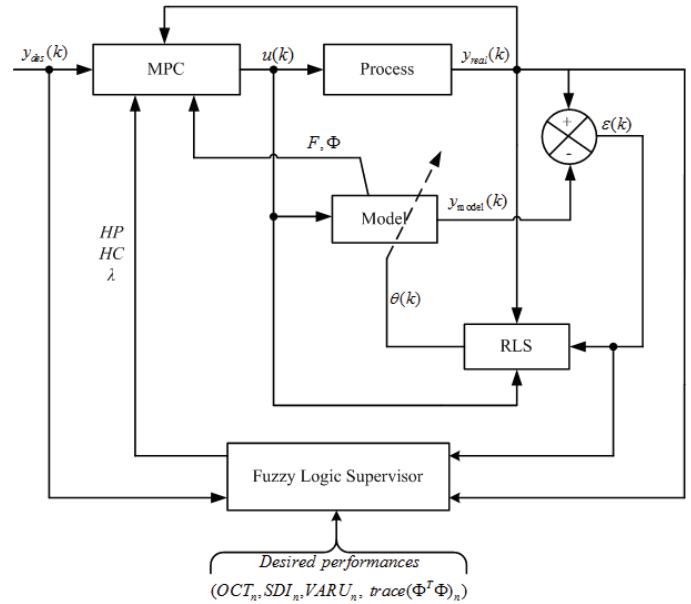


Figure 3. Equivalent structure of the IAMPC with fuzzy logic supervisor.

IV. APPLICATION AND COMPARISON

In this section we will focus on an academic application. Both detailed control strategies will be look at the overshoot or undershoot stability and rapidity. All simulations are derived via Matlab Simulink R2006b. For the computing environment set up, the solver used is "Discrete (no continuous states)" having a fixed step equals to 5 ms.

A. System description

The system considered is an electrical application that takes into account a flexible transmission with important variations of load inertia, which corresponds to typical applications in the machine tool area [10]. The transmission and load are modeled as a second order transfer function given in the continuous-time domain by:

$$H(s) = \frac{K}{J_n s^2 + f_n s + C} \quad \text{with} \quad \begin{cases} K = 2, J_n = 0.002 \\ f_n = 0.006, C = 2 \end{cases} \quad (26)$$

This SISO model is characterized by its natural pulsation and its damping factor:

$$\omega_n = \sqrt{C/J} = 31.623 \text{ rad/s}, \quad \xi = f/(2\omega_n J) = 0.4743.$$

With a sampling period T_s of 5 ms, the discrete-time model coming from (26) including the digital to analogue and analogue to digital converters is given by:

$$H(q^{-1}) = \frac{0.01187q^{-1} + 0.01129q^{-2}}{1 - 1.838q^{-1} + 0.8607q^{-2}}. \quad (27)$$

From [2], we set the original MPC parameters that produce correct results: $HI = 1$, $HC = 1$, $HP = 8$ and $\lambda = 0.42$. This set up respects the below performance criteria during the process control:

$$0 \leq OCT \leq 1, \quad 0 \leq SDI \leq 1, \quad 0 \leq VARU \leq 0.2, \quad |Overshoot| \leq S_1 \leq S_2, \\ 0 \leq trace(\Phi^T \Phi) \leq 1.5, \quad S_1 = 2.5\%, \quad S_2 = 5\%.$$

How does the controlled plant react against the load inertia changes for a square reference signal (fig. 4)?

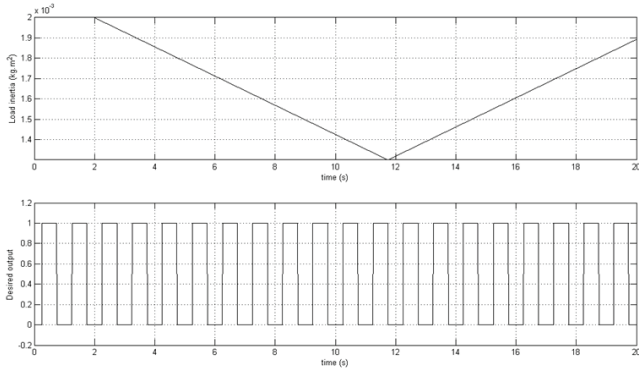


Figure 4. J profile and desired output.

B. Simulation results

1) *Results with IAMPC* : the plot of MPC output in fig. 5 shows us that the original controller MPC has difficulty in tracking correctly the reference signal. From 8 s to 16 s, where J reaches its lower value, the absolute overshoot is strictly greater than 5% and sometimes greater than 10% (fig. 6). In the same time, this gap increasing to $\varepsilon(k)$ activates the plant model reconfiguration (see fig. 7).

The *IAMPC* gets through to enhance the incapacities of MPC. One notices the reduction of the absolute overshoot, a smaller OCT in fig. 8 than before which proves that the process became quicker. *IAMPC* control signal curve has decreasing and rising peaks which looks like the J profile. While MPC control signal keeps the same aspect (fig. 9).

However, an important peak of 1.21 is located around 11 s and 12 s when we use the first adaptive control law. We must solve this problem.

2) *Results with fuzzy-logic-supervised IAMPC*: Considering fig. 6 as the zoom of *IAMPC* output, one can observe various sudden peaks. They are limited to 5% or 6% except between 11 s and 12 s. One visible effect of *fuzzy-logic-supervised IAMPC* is the drastic cancellation of these indisirable elements. Which leads to the smallest absolute overshoot (fig. 11). In fact, there is a modulus increase of the current control signal according to *IAMPC* control. The control signals plot shows that maximal values of control signal using FLS reach approximatively 4.3 (fig. 12).

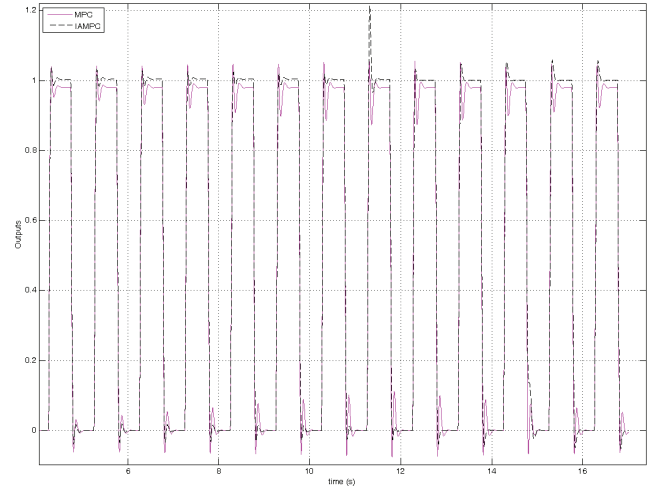


Figure 5. Output results of MPC vs IAMPC.

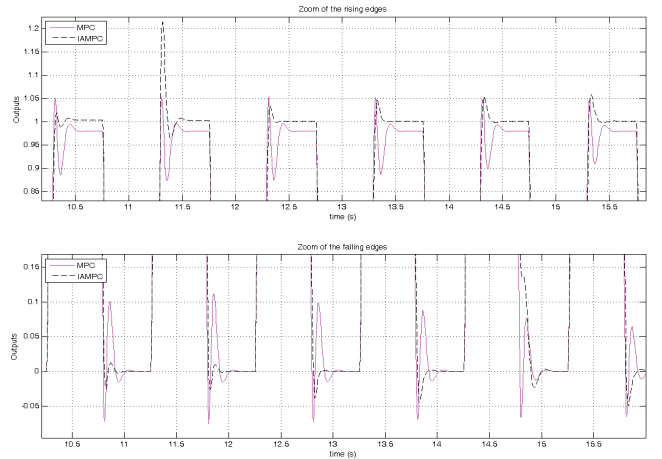


Figure 6. Output zooms of MPC vs IAMPC.

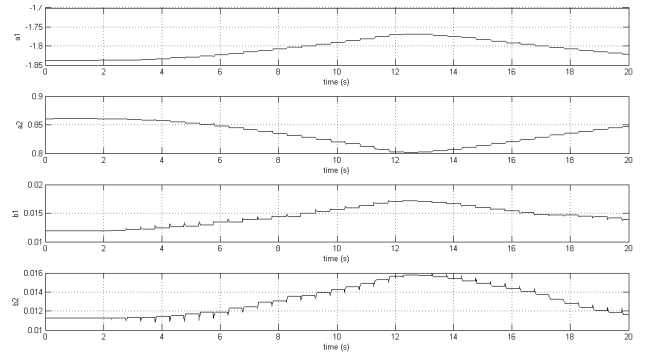


Figure 7. New estimated parameter vector.

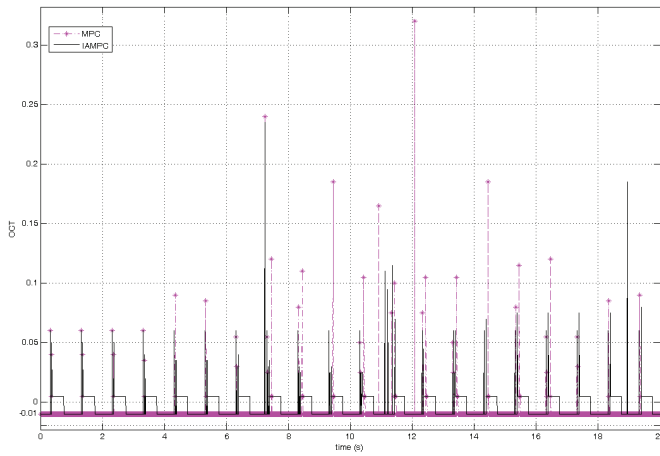


Figure 8. OCT results of MPC vs IAMPC.

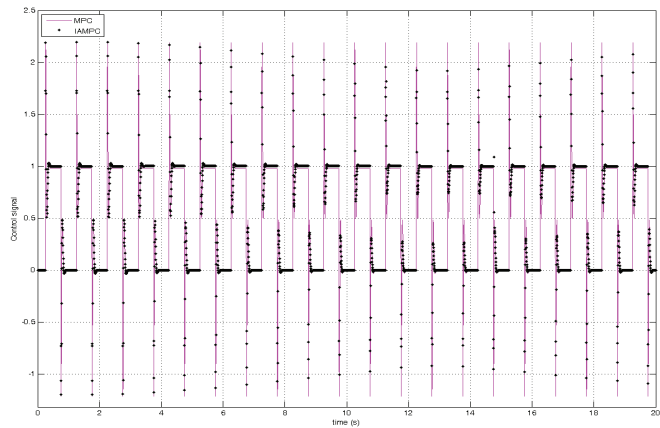


Figure 9. Control signals of MPC vs IAMPC.

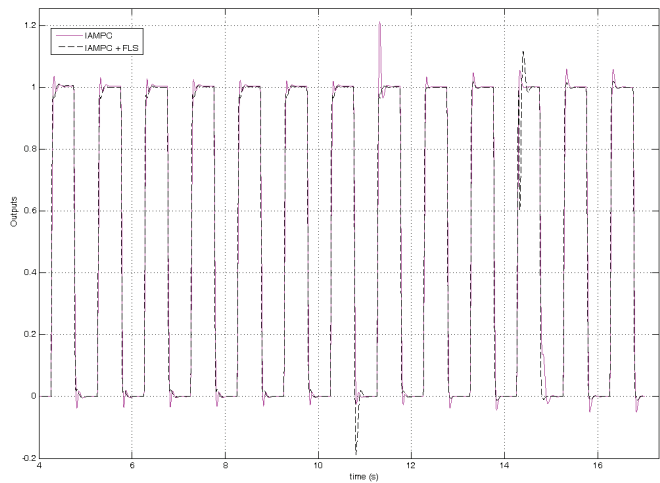


Figure 10. Output results of IAMPC vs IAMPC with FLS.

Concerning the rapidity criterion, the approach based FLS leads to a plant as quick as the process using the *IAMPC* strategy. In the above OCT plots (fig. 13), *IAMPC* proposes sometimes lower values than *IAMPC* enhanced by FLS. Otherwise, it's the opposite e.g near 7 s and from 15 s to 20 s. We perform the same analysis for SDI criterion.

During the first period i.e up to 13 s, *IAMPC* enhanced by FLS provides a better stability, while *IAMPC* strategy takes the advantage after (fig. 14).

This comment recalls the traditional dilemma between rapidity and stability: fluctuations of one affects negatively the other.

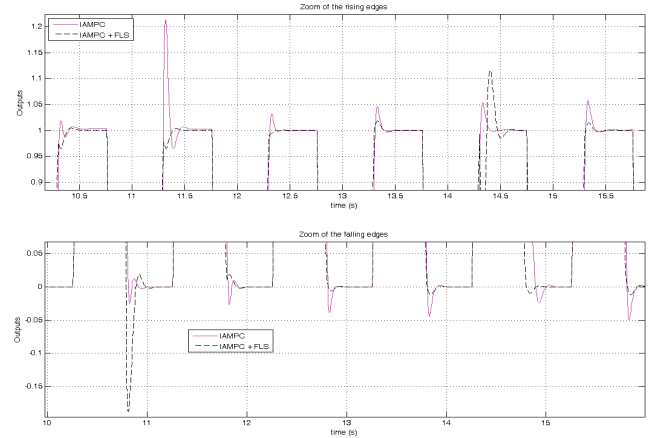


Figure 11. Output zoom of IAMPC vs IAMPC with FLS.

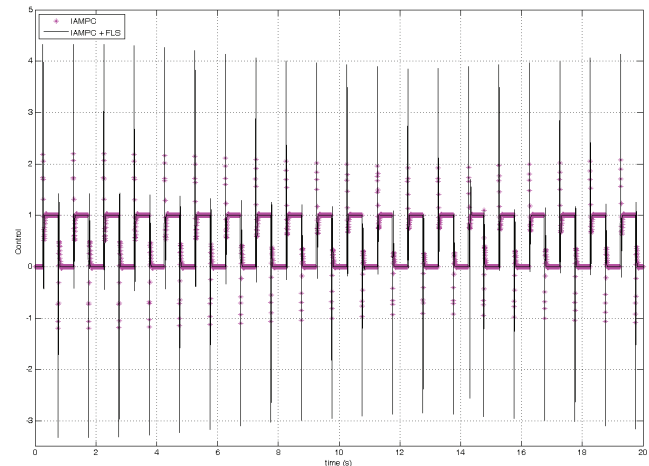


Figure 12. Control signals of IAMPC vs IAMPC with FLS.

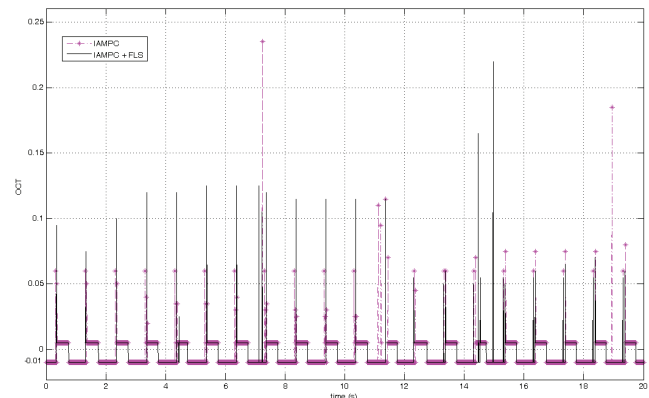


Figure 13. OCT results of IAMPC vs IAMPC with FLS.

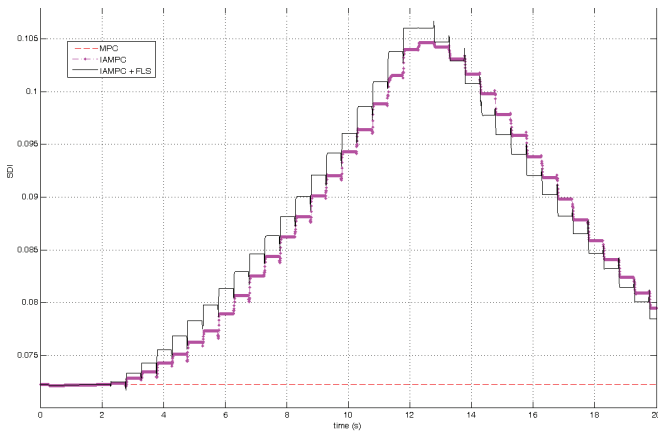


Figure 14. SDI results of IAMPC vs IAMPC with FLS.

Let us note that both reconfiguration strategies improve substantially the original stability getting an indicator closer to zero than theirs results.

All these improvements are due to the double capacity of *fuzzy-logic-supervised IAMPC* to conjugate in parallel the modifications of HC , HP , λ (see fig. 15) with the online model identification. Nevertheless, even if the irregular peak of *IAMPC* in [11 s, 12 s] disappeared, two particular peaks (resp. -0.19 and 1.12) appear at 10.8 s and 14.4 s.

V. CONCLUSION

This paper presents an indirect adaptive MPC using fuzzy logic supervisor for which two simultaneous adaptations can be activated automatically. The first is based on the discrete-time augmented model reconfiguration thanks to RLS combining PAA. While the second provides suitable values of HP , HC and λ via a logic decision carried by the proposed supervisor.

Important abilities of IAMPC enhanced by FLS are highlighted when a Linear Time-Variant system is confronted to strong parameter variations. It gets through to repulse several limits encountered with MPC and IAMPC. The majority of performance criteria are respected. Nonetheless, two undesirable peaks appeared even if their values are lower than previously. It seems possible to eliminate these anomalies by imposing inequality constraints on the control and output signals. After that, it would be also very interesting to study how we can implement and test this enhanced control law on a real-time application.

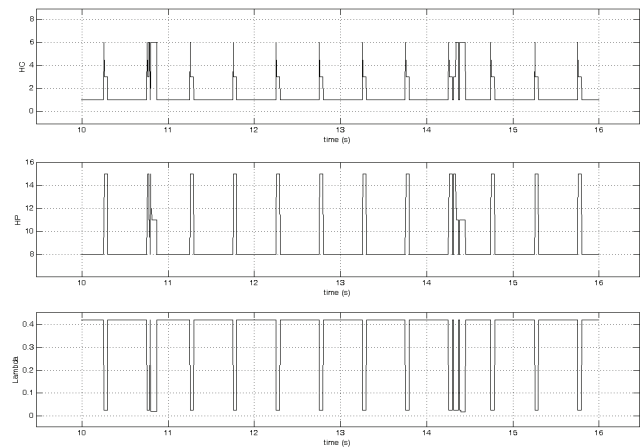


Figure 15. Variations of MPC parameters with FLS from 10 s to 16 s.

ACKNOWLEDGMENT

The authors gratefully thank “Région Haute Normandie” for financially supporting this work and its forthcoming application to diesel engine control in the frame of the ORIANNE (Outil numéRIque pour le mAquettage de foNctions de coNtrôle motEur) project labelled by the French competitiveness clusters Mov’eo and Aerospace valley.

REFERENCES

- [1] S. J. Qina and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Control Engineering Practice*, vol. 11, pp. 733–764, 2003.
- [2] P. Boucher and D. Dumur, *La Commande Prédictive*. TECHNIP, 1996.
- [3] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*. Springer, 2009.
- [4] R. Haber, U. Schmitz, and R. Bars, “Optimal choice of horizons for predictive control by using genetic algorithms,” *Annual Reviews in Control*, vol. 28, pp. 53–58, 2004.
- [5] A. Richards and J. P. How, “Robust variable horizon model predictive control for vehicle maneuvering,” *Int. J. Robust Nonlinear Control*, vol. 16, p. 333–351, 2006.
- [6] K. Watanabet, K. Ikeda, T. Fukudas, and S. Tzafestasfl, “Adaptive generalized predictive control using a state-space approach,” *International Workshop on IROS*, vol. 3, pp. 1609 – 1614, Nov. 3-5, 1991.
- [7] L. Chisci, L. Giarri, and E. Mosca, “Indirect and implicit adaptive predictive control of the benchmark plant,” *Automatica*, vol. 30 No 4, pp. 577–584, 1994.
- [8] R. B. Abdenmour, P. Borne, M. Ksouri, and F. M’sahli, *Identification et commande numérique des procédés industriels*. TECHNIP, 2001.
- [9] I. D. Landau and G. Zito, *Digital Control Systems: Design, Identification and Implementation*. Springer, 2006.
- [10] D. Dumur, P. Boucher, and G. Ramond, “Direct adaptive generalized predictive control. Application to motor drives with flexible modes,” *Annals of the CIRP*, vol. 49/1/2000, pp. 271–274, August 2000.