A Novel Tuning Approach for MPC Parameters Based on Artificial Neural Network: An application to FOPDT System

Houssam MOUMOUH * **Nicolas LANGLOIS** ** **Madjid HADDAD** ***

* *NORMANDIE UNIV, UNIROUEN, SEGULA TECHNOLOGIES: 19 rue dArras 92000, Nanterre. France (e-mail:Houssam.moumouh@segula.fr).*
** *NORMANDIE UNIV, UNIROUEN,ESIGELEC, IRSEEM 76100 Rouen. France (e-mail: nicolas.langlois@esigelec.fr)*
*** *SEGULA TECHNOLOGIES: 19 rue dArras 92000, Nanterre. France (e-mail: madjid.haddad@segula.fr)*

Abstract: A successful implementation of Model Predictive Control (MPC) requires appropriately tuned parameters. In this paper an Artificial-Neural-Network (ANN) based approach is presented and detailed in the case of a First Order Plus Dead Time (FOPDT) controllable system. The original part of our approach lies in its capability to tune the MPC parameters using Particle-Swarm-Optimization (PSO) and Online-Sequential-Extreme-Learning-Machine(OS-ELM). This approach allows also to reach efficiently closed-loop stability. The effectiveness of our approach has been emphasized by comparing the obtained performances to other existing methods.

Keywords: Model Predictive Control, tuning parameters, Artificial Neural Network, FOPDT, PSO, OS-ELM.

1. INTRODUCTION

The Model Predictive Control (MPC) has proven to be an excellent candidate for controlling complexe systems. It is now widely implemented in industry since many year Qin and Badgwell (2003). Its increase of the productivity and its ability to meet the requirements of the process performance are attracting more interest into this controller.

In our work, we tend to use a state space model-based predictive control.

This optimal control is based on the minimization of cost function J and receding horizon principle. This advanced control technique has the ability to anticipate future output from past input/output.

Previous studies have proven that control horizon N_c, prediction horizon N_p, and weighting factor λ are the parameters dominating the behavior of MPC controllers. The values of these parameters significantly influence the closed loop behavior of the system especially in terms of stability, robustness, and accuracy between desired output and reference. These three parameters will be considered in this study.

In the literature there exists many strategies for tuning parameters :

- The experimental approach allows the computing of tuning parameters combined. Yet it requires a test bench obligatory. Boucher and Dumur (1996), Tohidi and Hajieghrary (2016)
- The heuristic approach in their turn, it requires a statistical calculation technique which does not allow the identification of the system robustness zone. Mamboundou and Langlois (2011), Gutirrez-Urqudez et al. (2015)
- The analytical approach generates optimal tuning parameters. Whereas it requires a huge computational time. Turki et al. (2017)

Our contribution consists of MPC tuning method using online learning algorithm (OS-ELM), in order to apply it to complex industrial systems.

Based on ANN, this approach is completely original and has never been published in the literature yet. Some approaches in the literature deal with predictive control with ANN, but no one has used the ANN with online learning algorithm to solve the problem of tuning parameters of the MPC. The majority of these existing approaches use ANN for modeling purposes, Karla and Bakker (1995), whereas in other applications the ANN has been dedicated for control purpose. Moreover, ANN has been also used for parameters calculation based on the offline learning

⋆ This work is financed by Segula Technologies and the National Association of research and Technologies (ANRT).

algorithm such as Back Propagation Qi-An Li and Shu-Qing Wang (2004).

This paper is organized as follows: section 2 reminds principle of MPC based on state space representation. Section 3 introduces the proposed ANN-based tuning approach. Section 4 deals with simulation results obtained from the given FOPDT system. The performance of the novel approach is set under test with two existing methods to emphasize the effectiveness of our approach. Section 5 concludes the paper with a summary and future research directions.

2. REMIND ON MODEL PREDICTIVE CONTROL

2.1 Augmented state-space model

In this paper, we consider the case of a single-input-single-output (SISO) system represented by the following discrete-time state-space model:

$$\begin{aligned} x_m(k+1) &= A_m x_m(k) + B_m u(k) \\ y(k) &= C_m x_m(k) \end{aligned} \tag{1}$$

Where $u \in R$ is the manipulated variable or input variable, $y \in R$ is the system output, and the row matrix x_m is the state-space vector of size n_{A_m}. k is the sampling instant. In (1), A_m is a $(n_{A_m} \times n_{A_m})$ matrix. Thus, in order to design predictive controller, the formulation of the augmented-state model is considered with embedded integrators whose advantages have been already discussed Wang (2009).
Let consider the new state space vector:

$$x(k) = \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} \tag{2}$$

Then:

$$\begin{aligned} \Delta x_m(k) &= x_m(k) - x_m(k-1) \\ \Delta u(k) &= u(k) - u(k-1) \end{aligned} \tag{3}$$

Where the augmented state space model is defined by:

$$\begin{aligned} \overbrace{\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix}}^{x(k+1)} &= \overbrace{\begin{bmatrix} A_m & 0_m^t \\ C_m A_m & 1 \end{bmatrix}}^{A} x(k) + \overbrace{\begin{bmatrix} B_m \\ C_m B_m \end{bmatrix}}^{B} \Delta u(k) \\ y(k) &= \overbrace{\begin{bmatrix} 0_m & 1 \end{bmatrix}}^{C} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} \end{aligned} \tag{4}$$

Where $0_m = [0\ 0 \cdots\ 0]$ is a row matrix of size n_{A_m}. The dimensions of the matrices A, B and C are $(n_A \times n_A)$, $(n_A \times 1)$ and $(1 \times n_A)$ respectively with $(n_A = n_{A_m} + 1)$.

2.2 MPC formulation

As an hypothesis, the system is supposed to be observable and controllable. The incremental control signal vector ΔU of dimension $(1 \times N_c)$ is defined by Wang (2009):

$$\Delta U = [\Delta u(k)\ \Delta u(k+1)\ \cdots\ \Delta u(k+N_c-1)]^T \tag{5}$$

where T indicates the matrix transpose. The set-point which is our desired output Y_{des} of size $(N_p \times 1)$ is:

$$Y_{des} = [y_{des}(k+1)\ y_{des}(k+2)\ \cdots\ y_{des}(k+N_p)]^T \tag{6}$$

Assuming that the predicted output vector $\hat{Y}$ is defined by:

$$\hat{Y} = Fx(k_i) + \Phi \Delta U \tag{7}$$

where:

$$F = \left[CA\ CA^2\ CA^3\ \cdots\ CA^{N_p}\right]^T, \tag{8}$$

$$\hat{Y} = [\hat{y}(k+1 \mid k)\ \hat{y}(k+2 \mid k)\ \cdots\ \hat{y}(k+N_p \mid k)]^T \tag{9}$$

and

$$\Phi = \begin{bmatrix} CB & 0 & 0 & ... & 0 \\ CAB & CB & 0 & ... & 0 \\ CA^2B & CAB & CB & ... & 0 \\ \vdots & & & & \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & ... & CA^{N_p-N_c}B \end{bmatrix} \tag{10}$$

Let consider the cost function to minimize:

$$J = (Y_{des} - \hat{Y})^T (Y_{des} - \hat{Y}) + \Delta U^T \bar{R} \Delta U \tag{11}$$

where $\bar{R}$ is a matrix defined by:

$$\bar{R} = \lambda I_{(N_c, N_c)} \tag{12}$$

And I is the identity matrix. The weighting factor is defined by:

$$\lambda = [\lambda_1\ \lambda_2\ \cdots\ \lambda_{N_c}] \tag{13}$$

The optimal predictive control is obtained from the partial derivative of J with respect to ΔU as in the following equation:

$$\frac{\partial J}{\partial \Delta U} = -2\Phi^T (Y_{des} - Fx(k)) + (\Phi^T \Phi + \bar{R}) \Delta U \tag{14}$$

After obtaining ΔU, we extract the first element $\Delta u(k)$ of the mentioned vector using the receding horizon principle to generate control which will be applied to the process:

$$u(k) = u(k-1) + \Delta u(k) \tag{15}$$

$$u(k) = u(k-1) + I_{(1 \times N_c)} (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (Y_{des} - Fx(k)) \tag{16}$$

Where the state space vector $x(k)$ forms the current process to estimate the optimal control signal. If there exists constraints on control input, state and also on the output signal the way to compute ΔU differs. We can use some toolbox on Matlab to solve the problem like (YALMIP, IPOPT,...), Jerry Mamboundou (2013), Turki et al. (2018). Whereas in our work we adopted linear inequality constraints as in (17), to be applied to the control signal:

$$U_{min} \leq U \leq U_{max}. \tag{17}$$

3. THE PROPOSED ARTIFICIAL-NEURAL-NETWORK BASED APPROACH

This section seeks to address the algorithm of calculating the MPC parameters (λ, N_c, N_p).
Consequently, the research leads us to elaborate a novel approach based on the ANN.

As a first step, the considered performances will be introduced and investigated to compare the existing approaches to ours. Furthermore, these performances are assigned as the inputs of our ANN. while in the output of the ANN, we will put the tuning parameters of the MPC in order to reach our aim.

Moreover, to build the data learning base of the ANN, we have adopted the Particle Swarm Optimization (PSO) which is a meta-heuristic tuning approach.

As a final step, we used a reliable algorithm, Online Sequential Extreme Learning Machine (OS-ELM), to compute the tuning parameters on an online basis. The following section shows in details the steps of our proposed approach:

3.1 The considered performances criteria

In order to establish a comparative study among our tuning approach and two existing methods in the literature, that uses the tuning parameters of the MPC for controlling delayed systems of parameters tuning of the predictive control, we have considered the following performance criteria which react to a given set point:

- Rise Time (RT) between 10% and 90%.
- OVershoot (OV) : this criterion presents the maximum overshoot on the output signal.
- The average of the tracking error (ATE) : this criterion is used to evaluate the accuracy of the system at each sampling step.
- Settling time (ST) within 2%.

3.2 The Building of the Data learning base

A methaheuristic approach is employed to build a data learning algorithm. In its turn, it facilitates the computation of the tuning parameters of the MPC. The latter is based on particle swarm optimisation (PSO). The PSO metaheuristic is an evolutionary computation method developed by Kennedy' and Eberhart (1995). This advanced technique is inspired by the swarming behaviour of biological populations as bird flocks and fish schools.

The PSO is considered as one of the most powerful methods for solving global optimization problem, it is known in its simple implementation of coding and theoretical concepts.

The (PSO) approach has been mentioned in several articles, in particular for the calculation of PID parameters Khaled et al. (2018)

3.3 The considered learning algorithm

After building the data learning base of the ANN, a problem has arisen which is the learning of the latter. In this section we will explain in details the different learning methods as well as the one we chose for our algorithm.

The learning phase of an ANN uses either the offline or online learning algorithm:

- Offline algorithm: in this algorithm, the adjustment of the network parameters is executed, after ensuring the availability of each and every sample.
- Online algorithm: the adjustment of the network parameters is executed when each new data sample is generated. The cost function to be minimized is therefore the total instantaneous energy error.

In this paper, we took advantage of the online learning algorithm which is also known as dynamic algorithm. It is beneficial in terms of less storage consumption and computing time. A remarkable feature in this online algorithm is shown in its ability to deal with gradually changing data over time.

One of the most used algorithm is the Online Sequential Extreme Learning Machine (OS-ELM). It is an algorithm developed for a single hidden layer, which is able to learn the data elements, one by one or block by block. OS-ELM is inspired by ELM concept of offline learning. Examples to learn can be presented sequentially with a variable or fixed size Huang et al. (2006).

The algorithm is translated into two steps:

- Initialization phase: learning is initialized with a small number of examples. The initial number of examples must be larger than the number of neurons in the hidden layer.
- Sequential learning phase: let consider a set of N sample, (data learning) which represent, respectively the input and the output of the ANN.

$$y_j = \sum_{\substack{i=0 \\ i\neq i_0}}^{M} \beta_i f(w_i x_j + b_i), \quad j \in [1, N] \tag{18}$$

where f is the activation function, w_i represents the weight between the input and the output layers, b_i is the value of the neural bias of the hidden layer, and β_i is the output weight between the hidden and the output layer. The previous equation can be presented in vector form:

$$H\beta = Y \tag{19}$$

Where H is the output matrix of the hidden layer defined by:

$$H = \begin{bmatrix} f(w_1x_1 + b_1) & \dots & f(w_Mx_1 + b_M) \\ f(w_1x_2 + b_1) & \dots & f(w_Mx_2 + b_M) \\ \vdots & & \\ \vdots & & \\ f(w_1x_N + b_1) & \dots & f(w_Mx_N + b_M) \end{bmatrix} \tag{20}$$

and:

$$\beta = [\beta_1 \ \beta_2 \ \cdots \beta_M]^T \tag{21}$$

The vector β is determined by analytically solving the quadratic error with the least square method at shown in the following equations:

$$S = || \ H\beta - Y \ ||^2 \tag{22}$$

$$(H\beta - Y)^T (H\beta - Y) = 0 \tag{23}$$

and we get:

$$\beta = (H^T H)^{-1} H \times Y = H^+ \times Y \tag{24}$$

where H^+ is pseudo the inverse of the matrix H.

The sequential learning phase consists of updating the output weight for $(k + 1)$ new elements. $N_k + 1$represents the new examples to take into account for the learning process, by computing the new output matrix. Thus, the output weight for $(k+1)$ elements can now be calculated using the new output matrix as shown in the equation below:

$$\beta^{k+1} = \beta^k + K_{k+1}^{-1} H_{k+1}^T (Y_{k+1} - H_{k+1}\beta^k) \quad (25)$$

With:

$$K_{k+1} = H_k + H_{k+1}^T H_{k+1} \quad (26)$$

3.4 The proposed control strategy

In the previous three parts, we have specified the considered performance criteria along with the needed steps to build the data base, as well an online algorithm that has been used for the learning of the ANN.

The different steps of the original approach we propose are:

- Step 1: Choose N_c and N_p very large, with $N_c > N_p$ (ideally N_c and N_p tend towards infinity).
- Step 2: Calculate the augmented state-space model of the system to obtain the matrices A, B, C and ϕ.
- Step 3: Chose the performances criteria which will be the input of the ANN.
- Step 4: Compute N_c, N_p, λ along with PSO for FOPDT system to build the data learning base, these tuning parameters will be the output of the ANN.
- Step 5: Use the ELM algorithm to learn the ANN.
- Step 6: Utilize ANN for calculating tuning parameters (N_c, N_p, λ) for any SISO system (not existing in the learning base).

Figure 3 shows the proposed control strategy, in which the MPC is applied to the system. From this figure it is possible to analyze the behavior of the real output and compare this latter to the desired one.

4. NUMERICAL APPLICATION

In this section our ANN based approach is applied in simulation with Matlab/Simulink to control a FOPDT system. The performance of the novel method is set under test with two existing methods to emphasize the effectiveness of our approach.

4.1 System description

Many processes in the literature can be approximated to a FOPDT system for control purposes. Therefore, the following method can be extended to other process examples Bordons and Camacho (1998). The considered FOPDT system is modelled by the following transfer function Schwarz et al. (2010):

$$G(s) = \frac{1}{1+0.5s} e^{-s} \quad (27)$$

In this paper, our method is compared to two heuristic tuning approaches applicable to delayed systems Iglesias and Sanjun (2006), and Shridhar and Cooper (1996).

As the first case, the control signal is subject to linear inequality constraints, taken as:

$$\begin{cases} U_{min} = 0 \\ U_{max} = 2.5 \end{cases} \quad (28)$$

4.2 Simulation results

The values of the MPC parameters used in simulations are presented in Table 1: Figure 1 and 2 show the simulation results. Table 2 shows the performance comparison results for the first case of constraints:

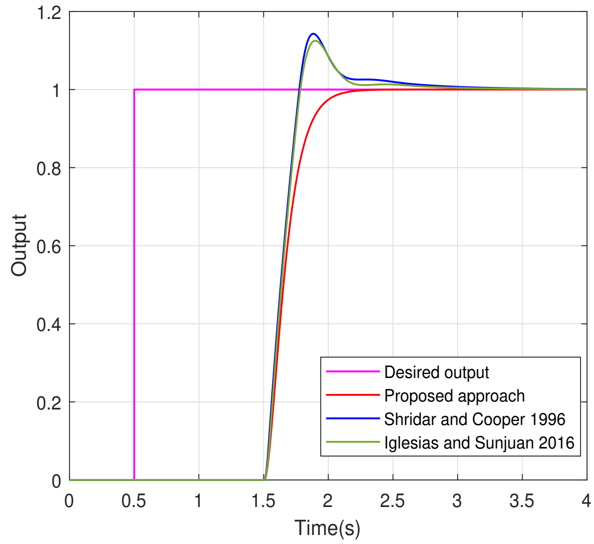

Fig. 1. Output signal vs.Time

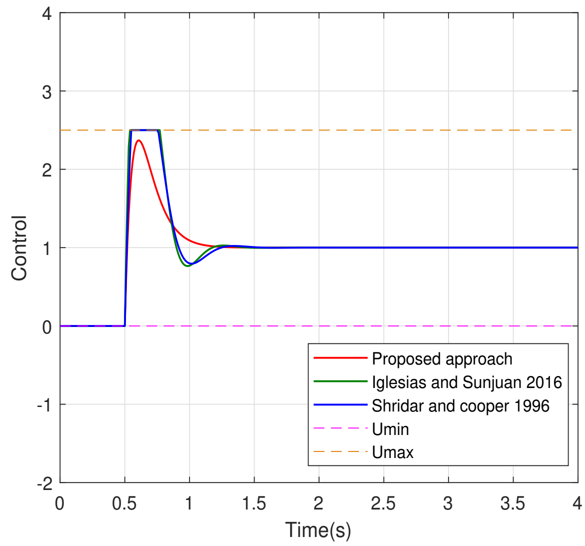

Fig. 2. Control signal vs.Time

The observed results inspection of Figure 1 and Table 2 have emphasized the effectiveness of our approach. Specially our approach has shown better performances

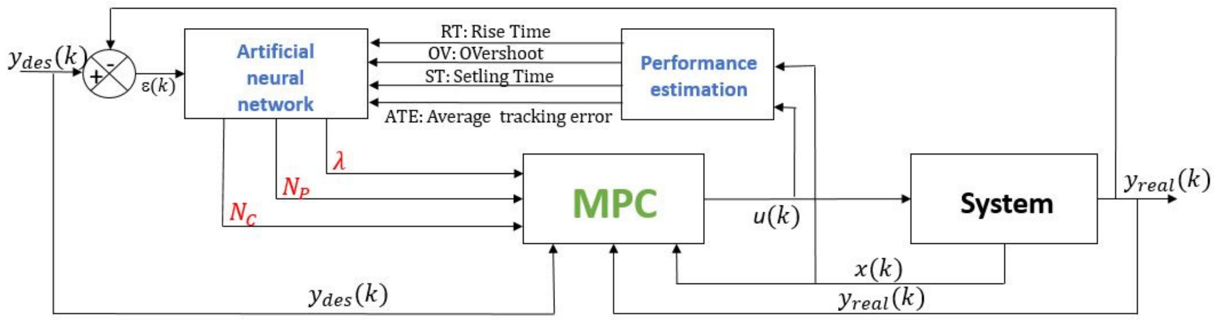

Fig. 3. Proposed Control Strategy

Table 1. MPC parameters

	N_c	N_p	λ
Schridar and Cooper 1996	21	61	1.134
Iglesias and Sanjuan 2016	21	61	2.166
Proposed approach	4	39	5

Table 2. Performances comparaison of MPC

	Schridar and cooper 1996	**Iglesias and Sunjuan 2016**	**Proposed approach**
OV(%)	11.79	10.55	0.5
RT(s)	0.127	0.152	0.294
ST(s)	0.69	0.61	0.47
ATE	0.58	0.47	0.29

compared to the existing methods in terms of overshoot which nearly null.

Furthermore, the fastest and the smoothest response is assigned to our approach as observed in Table 2. And also, Figure 2 shows that with our approach, there is any control saturation, despite the constraints presence.

It is important to note that the lowest tracking error is also obtained by our ANN based method. Moreover, the latter allows the MPC to reduce the energy consumption to reach the control objective.

Finally, the best trade-off among stability, accuracy and rapidity is achieved with our tuning method.

5. CONCLUSION

In this paper, an ANN based approach is proposed to tune parameters of constrained MPC using an Online Sequential Extreme Learning Machine (OS-ELM) and Particle Swarm Optimization (PSO). As an application, the proposed approach has been used to control a FOPDT controllable system. A results comparison with other existing methods has been led to show its effectiveness in terms of performance. Our future scope is aiming to propose an online updating approach for MPC tuning parameters over time and to extend our algorithm to match nonlinear multi-input multi-output systems.

REFERENCES

Bordons, C. and Camacho, E. (1998). A generalized predictive controller for a wide class of industrial processes. 6(3), 372–387.

Boucher, P. and Dumur, D. (1996). *La commande prdictive*. Technip. OCLC: 636661294.

Gutirrez-Urqudez, R.C., Valencia-Palomo, G., Rodrguez-Elias, O.M., and Trujillo, L. (2015). Systematic selection of tuning parameters for efficient predictive controllers using a multiobjective evolutionary algorithm. 31, 326–338.

Huang, G.B., Zhu, Q.Y., and Siew, C.K. (2006). Extreme learning machine: Theory and applications. 70(1), 489–501.

Iglesias, E.J. and Sanjun, M.E. (2006). Tuning equation ford dynamic matrix control in siso loops. 14.

Jerry Mamboundou, L.N. (2013). Commande prdictive base dtat adaptive indirecte supervise par la logique floue.

Karla, V. and Bakker, H. (1995). Neural-network-based model predictive control: a case study. In *Proceedings 1995 Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*, 355–358. IEEE Comput. Soc. Press.

Kennedy', J. and Eberhart, R. (1995). Particle swarm optimization. 7.

Khaled, B., Bouallgue, S., and Ayadi, M. (2018). Design and co-simulation of a fuzzy gain-scheduled PID controller based on particle swarm optimization algorithms for a quad tilt wing unmanned aerial vehicle. 40(14), 3933–3952.

Mamboundou, J. and Langlois, N. (2011). Indirect adaptive model predictive control supervised by fuzzy logic. In *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, 2979–2986. IEEE.

Qi-An Li and Shu-Qing Wang (2004). Fast algorithm for adaptive generalized predictive control based on BP neural networks. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, volume 2, 738–743. IEEE.

Qin, S. and Badgwell, T.A. (2003). A survey of industrial model predictive control technology. 11(7), 733–764.

Schwarz, M., Cox, C., and Brcsk, J. (2010). A filtered tuning method for a GPC controller. In *IET Irish Signals and Systems Conference (ISSC 2010)*, 180–185. IET.

Shridhar, R. and Cooper, D.J. (1996). A tuning strategy for unconstrained SISO model predictive control. 36(3), 729–746.

Tohidi, A. and Hajieghrary, H. (2016). Self-tuning adaptive multiple model predictive control with application to pH control process. In *2016 IEEE International*

Conference on Automation Science and Engineering (CASE), 1219–1224. IEEE.

Turki, M., Langlois, N., and Yassine, A. (2017). A tuning approach for constrained MPC: Nominal stability ensured and energy consumption optimized. In *2017 IEEE Conference on Systems, Process and Control (ICSPC)*, 1–6. IEEE.

Turki, M., Langlois, N., and Yassine, A. (2018). An analytical tuning approach for adaptive MPC parameters applied to LTV SISO systems. In *2018 Annual American Control Conference (ACC)*, 1534–1539. IEEE.

Wang, L. (2009). *Model predictive control system design and implementation using MATLAB*. Advances in industrial control. Springer. OCLC: 440654711.