

Nonlinear Model Predictive Control for pH Neutralization Process based on SOMA Algorithm

Hajer Degachi

Tunis El Manar University
National Engineering school of Tunis
LR11ES20, Analysis, Conception
and Control of Systems Laboratory

Wassila Chagra

Tunis El Manar University
El Manar Preparatory Institute
for Engineering Studies
LR11ES20, Analysis, Conception
and Control of Systems Laboratory

Moufida Ksouri

Tunis El Manar University
National Engineering school of Tunis
LR11ES20, Analysis, Conception
and Control of Systems Laboratory

Abstract—In this work, the pH neutralization process is described by a neural network Wiener (NNW) model. A nonlinear Model Predictive Control (NMPC) is established for the considered process. The main difficulty that can be encountered in NMPC is solving the optimization problem at each sampling time to determine an optimal solution in finite time. The aim of this paper is the use of global optimization method to solve the NMPC minimization problem. Therefore, we propose in this work, to use the Self Organizing Migrating Algorithm (SOMA) to solve the presented optimization problem. This algorithm proves its efficiency to determine the optimal control sequence with a lower computation time. Then the NMPC is compared to adaptive PID controller, where we propose to use the SOMA algorithm to formulate the PID in order to determine the optimal parameters of the PID. The performances of the two controllers based on the SOMA algorithm are tested on the pH neutralization process.

Keywords—Nonlinear model predictive control; optimization; SOMA algorithm; adaptive PID; pH neutralization process

I. INTRODUCTION

The pH neutralization process is characterized by a nonlinear behavior. The high nonlinear characteristic of this process makes the control of the pH a hard task. Since the necessity of maintaining the pH in a specific range value, many control strategies have been proposed. For this purpose, [1]–[3] designate a PID controller to control the pH value. An adaptive controller is developed in [4]. [5] proposed a model algorithmic control strategy. Other works [6]–[8] developed the Model predictive control (MPC) which is a more advanced control strategy. The MPC control strategy presents the major advantage to efficiently handle nonlinearity and constraints imposed on system input and output [10]. Indeed, for real processes, the control values are usually delimited by an upper and lower bound that should be respected.

The MPC is essentially based the choice of a suitable model and adequate optimization method to solve the minimization problem. Many structures was used to describe the nonlinear behavior of the pH neutralization process: NARX model [11], Fuzzy neural network model [12], Neural networks [13], Wiener model [8], [14]–[17].

Respecting the nonlinear nature of the model, the resulting NMPC minimization problem is nonlinear and nonconvex. The nonconvexity will complicate the implementation of the

NMPC. Added to that, solving a nonlinear optimization problem is a hard task in terms of computation time and burden. To overcome these difficulties a variety of solutions were proposed in literature to avoid solving a nonlinear optimization problem and ensure global convergence at each sampling time. In [18], [19], the minimization problem is converted into a linear one in the case of a simple polynomial description of the nonlinear block. In this case, the nonlinearity is removed by considering the inverse of the polynomial function. As a result, the input control will be linearly expressed in the predicted output leading to a quadratic optimization problem. Also, the nonlinear optimization problem is formulated in [8], [15], [16] as a linear problem by performing a linearization of the nonlinear model. To avoid solving the nonlinear optimization problem and reduce the implementation complexity of the NMPC, [20], proposed to describe the nonlinear process by a set of uncertain linear models instead of one nonlinear model. We can note that all these works avoid to solve the nonconvex optimization problem regarding the difficulty of implementation and the high computation burden necessary at each sampling time.

To obtain good control accuracy the NMPC optimization problem must be solved as nonlinear [21]. However, global optimization methods are generally not recommended since they are not able to ensure the real time feasibility of the NMPC. In fact, the sampling period of the process under consideration must be respected at each iteration when solving the NMPC problem. This constraint is difficult to be satisfied using global optimization method due to their slow convergence. Therefore, the main aim of this work is the use of an efficient algorithm to solve the NMPC optimization problem and ensure the real-time feasibility of the control algorithm. For this, we propose to use the Self Organizing Migrating algorithm (SOMA) in this work to solve the optimization problem. The SOMA is an evolutionary algorithm such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Artificial Bee Colony algorithm (ABC) and so on. The SOMA algorithm was successfully applied to solve a variety of engineering problems. The most interesting are: control issues [22], antenna design [23], system identification [24], [25], Aircraft wing design and Synthesis of robot control program [26]. This method presents the significant advantage of high convergence speed.

In This work, we will be interested to represent the pH neutralization process by a Wiener model. Wiener model

belongs to block oriented models that are defined by a dynamic linear block followed by a nonlinear static one. Due to the specific structure of the pH neutralization process, the Wiener model appears able to well reproduce the behavior of the process. Many works used the Wiener model to describe the nonlinear dynamic of the process.

The different representations vary in the way that linear and nonlinear blocks are described.

Many structures are used to describe the steady state block : Polynomial, [14], [27], support vector machine [16], neural network [8], [15], [19], cubic spline [18]. In this work, the nonlinear block is represented by a feed forward Neural Network (NN).

In this paper, the NMPC is integrated with SOMA algorithm to solve the optimization problem. We prove, in this work, the ability of the SOMA algorithm to ensure good control performances with a low computation time despite the large prediction and control horizons. In the sequel, The NMPC strategy integrated with SOMA algorithm is compared to adaptive PID controller. We propose, in this work, to adjust PID parameters using SOMA algorithm.

This paper is organized as follows: Section 2 describes the Wiener model. The identification of the considered process is detailed in Section 3. NMPC based on the Wiener model is presented in Section 4. The SOMA algorithm used to solve the minimization problem is described in Section 5. Simulation results are given in Section 6.

II. WIENER MODEL

Wiener model belongs to block oriented models. It is described by a linear dynamic block followed by a nonlinear static one as shown in Fig. 1.

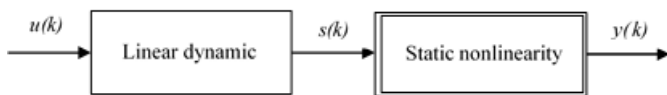


Fig. 1. Wiener model.

In this work, the linear dynamic block is described by a simple autoregressive model defined as:

$$A(q^{-1})s(k) = B(q^{-1})u(k) \quad (1)$$

Where polynomials A and B are given by:

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + \dots + a_{n_a}q^{-n_a} \quad (2)$$

$$B(q^{-1}) = b_1q^{-1} + b_2q^{-2} + \dots + b_{n_b}q^{-n_b} \quad (3)$$

n_a and n_b are respectively the orders of the two polynomials A and B.

The nonlinear block is defined as:

$$y(k) = f(s(k)) \quad (4)$$

$f(\cdot)$ is a nonlinear function.

Different forms are used to describe the nonlinear block for Wiener model starting from the simple polynomial form

[19], [28] to more complex description Neural network [19], [15], support vector machine [16]. In this work the nonlinear static block of the Wiener model is given by a feed-forward neural network as adopted in [15].

III. IDENTIFICATION OF THE WIENER MODEL

The aim of this paragraph is the determination of the parameters a_i and b_i of the linear block as well as the weights w_i of the network that present the parameters of the nonlinear block.

The structure of the Wiener model is depicted in Fig. 2.

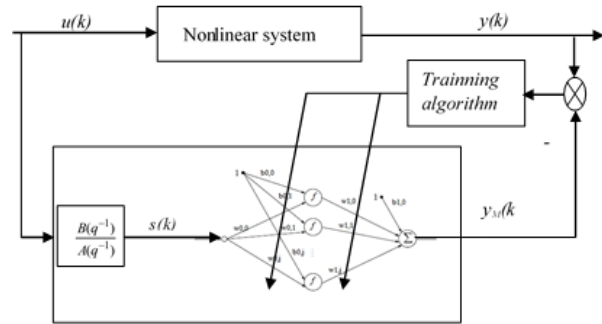


Fig. 2. Identification of the Wiener model.

A. Identification of the Linear Block

Firstly, a small input signal is applied to the system to ensure linear perturbation of the nonlinear system [29]. Then, the recursive least square algorithm is firstly used to identify the parameters of linear dynamic block.

The output of the linear block can be expressed as:

$$y(k) = \psi^T(t)\theta(t-1) \quad (5)$$

ψ is the data vector defined as:

$$\psi^T(t) = [-y(k-1), \dots, -y(k-n_a) u(k-1), \dots, u(k-n_b)] \quad (6)$$

θ is the parameter vector:

$$\theta^T = [a_1 a_2, \dots, a_{n_a} b_1 b_2, \dots, b_{n_b}] \quad (7)$$

The update of the parameter vector is ensured by the following equation:

$$\theta(t) = \theta(t-1) + P(t)\psi(t)\varepsilon(t) \quad (8)$$

P is weighting matrix given by:

$$P(t) = P(t-1) - \frac{P(t-1)\psi(t)\psi^T(t)P(t-1)}{1 + \psi^T(t)P(t-1)\psi(t)} \quad (9)$$

The initial parameter vector is $\theta(0) = 0$.

Once the parameters of the linear block are determined, the back-propagation algorithm is applied to train the feed-forward neural network such as in [29].

B. Identification of the Nonlinear Block

The structure of the nonlinear block is illustrated in Fig. 3. The output of the nonlinear block can be computed from Fig. 3 as:

$$y_M(k) = b_{1,0} + \sum_{i=1}^j w_{1,i} f(e(k)) \quad (10)$$

where j is the number of hidden nodes, f is a nonlinear activation function taken as the hyperbolic tangent function and $e(k)$ is the output of the hidden layer defined as:

$$e_i(k) = b_{0,i} + w_{0,i} s(k) \quad (11)$$

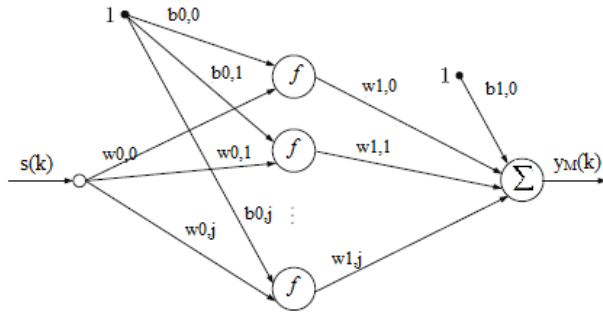


Fig. 3. Neural Network representation of the nonlinear Wiener block.

The weights $w_{i,j}$ of the neural network are updated by minimizing the following criterion:

$$J_{iden}(k) = \sum_{i=1}^N (y(k) - y_M(k))^2 \quad (12)$$

where $y(k)$ is the process output and $y_M(k)$ is the model output defined as:

$$y_M(k) = b_{1,0} + \sum_{i=1}^j w_{1,i} f \left(b_{0,i} + w_{0,i} \left(\sum_{i=1}^{n_a} -a_i s(k-i) + \sum_{i=1}^{n_b} b_i u(k-i) \right) \right) \quad (13)$$

Applying the back-propagation training algorithm the optimization is carried out in order to minimize the criterion (12) with respect to the weights $w_{i,j}$ of the network.

$$\frac{\partial J}{\partial w} = - (y(k) - y_M(k)) \frac{\partial y_M(k)}{\partial w} \quad (14)$$

The update equation of the different weights is defined as:

$$w(k+1) = w(k) + \Delta w(k) \quad (15)$$

where $\Delta w(k)$ is defined as: $\Delta w(k) = -\mu \frac{\partial J}{\partial w}$

$$w(k+1) = w(k) + \mu (y(k) - y_M(k)) \frac{\partial y_M(k)}{\partial w} \quad (16)$$

μ is the learning coefficient.

The Neural Network Wiener (NNW) model is used in this work to compute the j -step ahead predictions. Future prediction are used to define the cost function of the NMPC strategy.

IV. MODEL PREDICTIVE CONTROL DESIGN FOR WIENER MODEL

The basic idea of the MPC strategy is the use of the process model to compute the j -step ahead prediction over a prediction horizon N_p . At each sampling time, a control sequence $U = [u(k), u(k+1), \dots, u(k+N_u-1)]^T$ is computed, where N_u is the control horizon, by minimizing the cost function defined as the difference between the predicted output $\hat{y}(k+i|k)$ and the future set-point $y^{sp}(k+i)$ defined by:

$$J(k) = \sum_{i=1}^{N_p} (y^{sp}(k+i|k) - \hat{y}(k+i|k))^2 + \lambda \sum_{i=0}^{N_u-1} \Delta u(k+i|k)^2 \quad (17)$$

subject to

$$\Delta u^{\min} \leq \Delta u(k+i|k) \leq \Delta u^{\max}$$

$$u^{\min} \leq u(k+i|k) \leq u^{\max}$$

λ represents a positive weighting coefficient, $\Delta u(k+i|k)$ is defined as $\Delta u(k+i|k) = u(k+i|k) - u(k+i-1|k)$.

Based on the receding control horizon, only the first element of the control sequence U is applied to the system. Then, the whole procedure will be repeated at the next sampling time and the prediction horizon will be shifted one step forward.

In order to get efficient control results, predictions must be accurately computed. It is so important to take into account the effects of system and model mismatch coming from modeling errors and unmeasured disturbance, a correction term as in the dynamical matrix control, is added to model output defined as [30]:

$$d(k) = y(k) - y_M(k+i|k) \quad (18)$$

where $y_M(k+i|k)$ is the model output and $y(k)$ is the process output. The correction term is constant over the prediction horizon. Therefore, the model expression that will be used to compute predictions is defined as:

$$\hat{y}(k+i|k) = y_M(k+i|k) + d(k) \quad (19)$$

Due to the nonlinear nature of the NNW model the resulting optimization problem will be nonlinear and nonconvex. The convergence of this optimization problem is not guaranteed and the algorithm may be trapped in a local minimum that will lead necessarily to suboptimal control performances. Solving a nonlinear optimization problem is a high time demanding task. Added to that, real process requires generally a large prediction and control horizons that will raise the computation time. Moreover, the sampling period of the process presents an additional constraint that should be respected when solving the NMPC optimization problem.

Therefore, a fast convergence speed algorithm must be used to ensure the global convergence and the real-time feasibility of the control algorithm. Deterministic optimization methods are used in [9], [31] to solve the NMPC optimization problem. These methods are high time consuming also they can be

only applied to systems requiring small prediction and control horizons.

Many research papers propose to use stochastic optimization methods to solve the minimization problem. The Artificial Bee Colony (ABC) algorithm is combined with the NMPC in [32] to solve the minimization problem. This algorithm proves its simplicity of implementation and reduced computational complexity. The genetic algorithm (GE) is used in [33], [34] to determine the optimal control sequence. The particle swarm optimization (PSO) algorithm is integrated in [35] with NMPC to solve the resulting optimization problem. In [36], the neural network is used to determine the solution of the minimization problem. In [37], the Nelder Mead algorithm was applied which leads to global solution by using far initialization. The simulations gave optimal results with least computation time for SISO and MIMO models. However, it remains a local optimization method.

We propose in this work to use the SOMA algorithm to solve the presented optimization problem. SOMA presents an effective, robust and simple global optimization method.

V. SELF ORGANIZING MIGRATING ALGORITHM

SOMA is a stochastic optimization method proposed first by Zelinka [25]. SOMA is based on the social group of individual not on the philosophy of evolution. The classification of SOMA as evolutionary algorithm is explicated by the fact that the obtained results after a migration loop is the same as the result of one generation of evolutionary algorithm [26]. The principle of SOMA algorithm can be summarized as a migration loops during which the position of each individual is enhanced in order to reach the leader position (individual with the best fitness).

Each individual will be randomly initialized in the search space described by the upper and the lower bounds of the variables. At each migration loop, individuals are evaluated, the one that has the less fitness will be the leader, the rest individuals will cross a trajectory (*pathlength*) with step *t* in the direction of the leader.

Similar to other evolutionary algorithms, the operation of SOMA is ensured by some control parameters. The recommended ranges of these parameters are fixed based on great number of empirical tests and they are defined and given by [24], [26]:

- **Dim:** It defines the problem dimension (number of decision variable).
- **Population size (*Popsiz*e):** It defines the number of individuals in population, *recommended value* ≥ 10 .
- **Migrations:** It defines the maximum number of iterations (migration loops). It is the stopping criterion in SOMA recommended range [10, *upto user*].
- **Pathlength:** It fixes how far the individual will stop its movement from the leader. If the *pathlength*=1 the individual will stop at the leader position, if *pathlength*=2 the individual will surpass the leader position by the same distance from the initial position, recommended value 3.

- **step:** It defines the step that uses individuals to cross the path.
- **PRT:** The *PRT* is used to determine the *PRTvec*. Individuals are allowed to change their position based on the *PRT*, recommended value 0.4.

SOMA is a population based algorithm. The initial population *P* is generated randomly in the search space defined by the lower x_{min} and upper x_{max} bound of the manipulated variables. So, *P* is defined as: $P = \{X_1, X_2, \dots, X_{Popsiz$ e}\}. The *i*th individual X_i is defined by $X_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,Dim}\}$, $i = 1, \dots, Popsiz$ e, $j = 1, \dots, Dim$.

$$x_{i,j} = x_{min,j} + (x_{max,j} - x_{min,j}).rand \quad (20)$$

Like other evolutionary algorithm, SOMA performs a set of stochastic evolutionary operators. These operators are defined by mutation and crossover.

- **Mutation:** This operator defines in evolutionary algorithm the diversity in the population. In SOMA mutation is applied differently. It is performed by the *PRT* vector noted *PRTvec*. The *PRTvec* aims to perturb the path of individuals randomly in order to ensure diversification among them [38]. The perturbation in SOMA algorithm presents the mutation phase in the GA algorithm. Two values can be affected to this vector: 0 or 1 based on the SOMA *PRT* control parameter. Only individuals with *PRTvec* equal to 1 are allowed to change their positions. The *PRTvec* is created as follows:

$$\begin{aligned} & \text{if } rand < PRT \\ & \quad PRTvec_j = 1; \\ & \text{else } PRTvec_j = 0; \\ & \text{end} \end{aligned} \quad (21)$$

- **Crossover:** Crossover operator means the creation of new individual during the search. Since in SOMA algorithm no new individual is generated, the crossover is defined in this case by generating a new best position of individual across the search space to reach the leader. At each migration loop individuals explore a set of positions when mapping the path, memorize the best found one and move to this position at the end of the path. At the next migration loop, individuals start from this position. The movement of individuals to reach the position of the leader is given by the following equation:

$$x_{i,j}^{k+1} = x_{i,j}^k + (x_{L,j}^k - x_{i,j}^k).t.PRTvec_j \quad (22)$$

where $x_{i,j}^{k+1}$ is the new individual position., $x_{i,j}^k$ is the position of individual at iteration *k*, $x_{L,j}^k$ is the leader position and $t \in [0 : step : pathlength]$

The SOMA algorithm can be summarized as follows:

- 01** Choose the control parameters :PRT,step,pathlengthandmigration
- 02** Evaluate the initial population *P* using equation (20),
- 03** Evaluate individuals of the population
- 04** Sort individuals and select the leader
- 05** *iteration* = 1;

```

while (iteration < migrations)
for i = 1 : Popsiz
Generate PRTvec using equation(21)
Move each individual toward the leader using
equation (22)
Evaluate the individual at the new position

    new fitness < fitness(individual (i))
Move individual(i) to the new best position.
end if
end for
sort individuals and select the new leader
iteration = iteration + 1
end while
    
```

VI. SIMULATION RESULTS

The considered system is composed of the base stream q_1 , the acid stream q_2 and the buffer stream q_3 that are mixed in continuous stirred tank reactor. The dynamic model of the reactor is derived from the conservation equation and equilibrium relations [4]. The dynamic of the pH process is given by two differential equations and a nonlinear one given by (23) and (24).

$$\begin{aligned}
 \dot{W}_a(t) &= \frac{q_1(t)}{V}(W_{a1} - W_a(t)) + \frac{q_2(t)}{V}(W_{a2} - W_a(t)) \\
 &+ \frac{q_3(t)}{V}(W_{a3} - W_a(t)) \\
 \dot{W}_b(t) &= \frac{q_1(t)}{V}(W_{b1} - W_b(t)) + \frac{q_2(t)}{V}(W_{b2} - W_b(t)) \\
 &+ \frac{q_3(t)}{V}(W_{b3} - W_b(t))
 \end{aligned} \tag{23}$$

$$\begin{aligned}
 W_a(t) + 10^{pH(t)-14} - 10^{-pH(t)} + \\
 W_b(t) \frac{1 + 2 \times 10^{pH(t)-k_2}}{1 + 10^{k_1-pH(t)} + 10^{pH(t)-k_2}} = 0
 \end{aligned} \tag{24}$$

where W_a and W_b represent the charge balance coefficients. The base stream q_1 is manipulated to control the pH value, the acid stream q_2 , the buffer stream q_3 are maintained constant.

the different parameters of the reactor are listed in Table I.

In this section, we will consider the pH neutralization process.

TABLE I. PARAMETERS OF THE pH PROCESS

Parameter	value
W_{a1}	-3.05e-3
W_{a2}	-3e-2
W_{a3}	3e-3
W_{b1}	5e-5
W_{b2}	3e-2
W_{b3}	0
q_2	0.55
q_3	16.60
V	2900
k_1	6.35
k_2	10.25

A. Identification of the pH Neutralization Process

The pH neutralization process will be described by a NNW model, where the linear block is described by an auto-regressive model and the nonlinear block is given by a multi-layer feed-forward neural network with one hidden layer. The plant is excited using two different input signals to get the identification and the validation data as shown respectively in Fig. 4 and 5. The bounds on the input are 0 and 30.

A feed-forward neural network with one hidden layer and 3 nodes appears sufficient to describe the nonlinear block of the Wiener model [15].

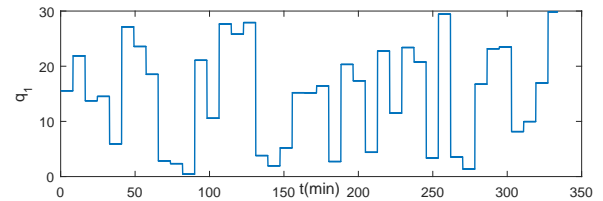


Fig. 4. Identification signal.

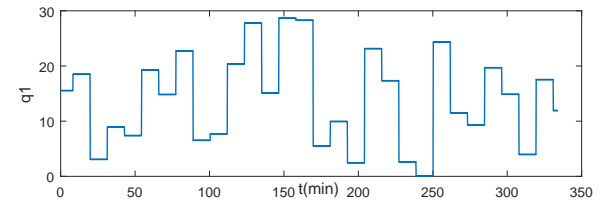


Fig. 5. Validation signal.

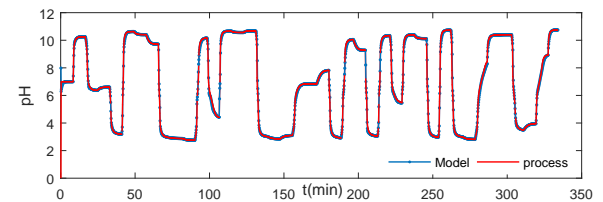


Fig. 6. Identification of the NNW model.

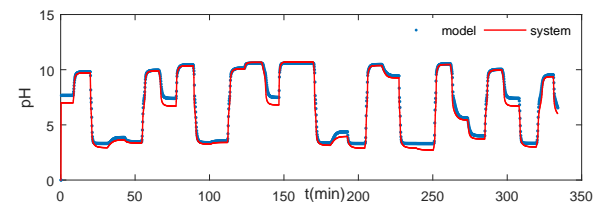


Fig. 7. Validation of the NNW model.

Fig. 6 and 7 illustrate the identification and the validation of the system. We can conclude that the neural Wiener model can reproduce the dynamic of the pH process with sufficient accuracy. This is well confirmed by the validation error depicted in Fig. 8 defined as the difference between the system and the model output $y(k)$ and $y_M(k)$, respectively.

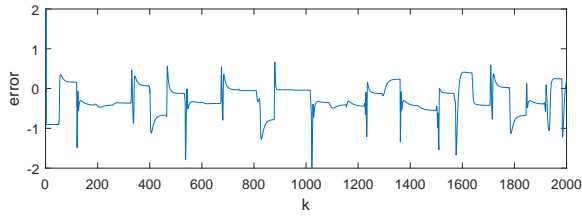


Fig. 8. Validation error.

B. NMPC Control of the pH Neutralization Process

The NMPC strategy aim to maintain the pH value at a desired value. For this the neural Wiener model is used to compute the j-step-ahead predictions for the NMPC. The parameters of the NMPC are fixed as $N_p = 10$, $N_u = 3$, $\lambda = 0.08$, $q^{\min} = 0$ and $q^{\max} = 30$.

The recommended values of the parameters of the SOMA algorithm are fixed as: $pathlength = 3$, $step = 0.31$, $PRT = 0.4$, $popsiz = 10$, the number of migration loops is 30.

We can note that the SOMA algorithm proves its efficiency to ensure good output tracking as depicted in Fig. 9. The performance offered by the SOMA algorithm outperforms those offered by the GBM method. This can be proved in terms of less overshoot and best control quality as illustrated in Fig. 9.

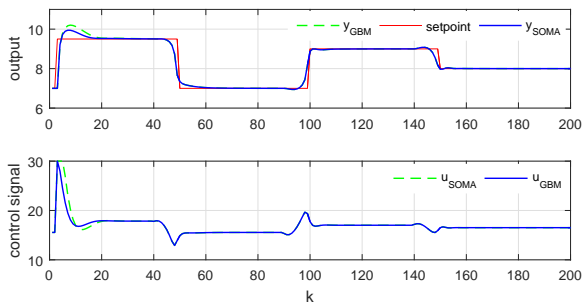


Fig. 9. Output tracking using the SOMA algorithm and the GBM.

In order to test the efficiency of the SOMA algorithm in presence of disturbance a constant v is added to the system output as:

$$v(k)=0.8 \quad 110 \leq k \leq 130$$

$$v(k)=0.6 \quad 160 \leq k \leq 180$$

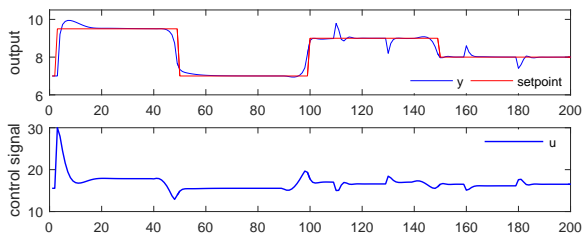


Fig. 10. Output tracking in the presence of disturbance.

We can note from Fig. 10 the good ability of the SOMA

algorithm to reject disturbance. This latter is rejected within 10 iterations that present a short time.

The NMPC strategy is compared to an adaptive PID controller. The Sum of square Error (SAE) defined by equation (25) is minimized in order to determine the proportional, integral and derivative coefficients of the PID controller: k_p , k_i and k_d , respectively:

$$SAE(k) = \sum_{i=1}^k |e(k)| \quad (25)$$

Respecting the high nonlinear considered process, the SOMA algorithm is adopted to derive the coefficients of the adaptive PID controller at each iteration.

The control parameters of the SOMA algorithm are fixed for the PID controller as those of the NMPC.

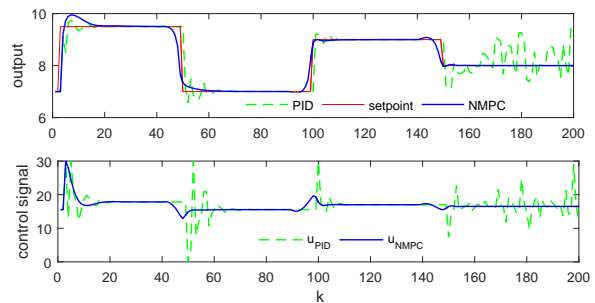


Fig. 11. Output tracking for NMPC and PID.

Simulation results for PID and NMPC are given in Fig. 11. The results show that both controllers ensure good output tracking in steady state for the first and second setpoint change. However, we can remark that the PID exhibits more overshoots as well as the deterioration of the control quality and consequently the output tracking in the last output change. This proves the superiority of the NMPC strategy to ensure good performances compared to PID.

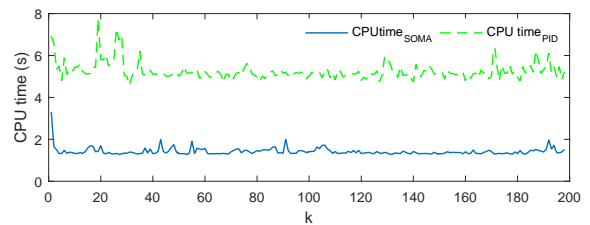


Fig. 12. Computation time of the NMPC and PID

Computation time present an important performance index in control problem. Fig. 12 shows the computation time for both NMPC and PID. The present results confirm the good ability of NMPC to give better performances with the low computation time. In fact, the determination of optimal PID parameters on line at each sampling time using the SOMA algorithm requires more migration loops this will directly affect the computation time of the implementation of the PID controller.

VII. CONCLUSION

In this paper, a nonlinear model predictive control is combined with the SOMA algorithm to solve the NMPC optimization problem. The control performances of the SOMA algorithm are tested on a high nonlinear process. Control results prove the efficiency of this algorithm to ensure good output tracking and control accuracy with a low computation time. NMPC based on SOMA algorithm is compared to adaptive PID controller. We can conclude from simulation results that the NMPC outperforms the PID in terms of less overshoot and computation time. NMPC offers the best control results and the less computation time compared to PID. The difficulty of the determination of PID parameters for high nonlinear process limits the performance of this controller.

REFERENCES

- [1] Rose, T.P., Devadhas, G.G., and Rex, S.R. "Invention of a suitable controller for a non linear Chemical process". In International Conference on Control, Instrumentation, Communication and Computational Technologies, pp. 1462-1467, 2014.
- [2] Ram, S.S., Kumar, D.D., Meenakshipriya, B., and Sundaravadiu, K. "Designing and comparison of controllers based on optimization techniques for pH neutralization process". In International Conference on Information Communication and Embedded Systems, pp. 1-5, 2016.
- [3] Puchalski, B., Rutkowski, T., Tarnawski, J., and Duzinkiewicz, K. "Comparison of tuning procedures based on evolutionary algorithm for multi-region fuzzy-logic PID controller for non-linear plant". In Methods and Models in 20th International Conference on Automation and Robotics (MMAR), pp. 897-902, 2015.
- [4] Henson, M.A., and Seborg, D.E. "Adaptive nonlinear control of a pH neutralization process". IEEE transactions on control systems technology, vol.2, pp. 169-182, 1994.
- [5] Zhiyun, Z.O.U., Meng, Y.U., Zhizhen, W.A.N.G., Xinghong, L.I.U., Yuqing, G.U.O., ZHANG, F., and Ning, G.U.O. "Nonlinear model algorithmic control of a pH neutralization process". Chinese Journal of Chemical Engineering, vol. 21, pp. 395-400, 2013.
- [6] Oblak, S., and Skrjanc, I. "Continuous-time Wiener-model predictive control of a pH process based on a PWL approximation". Chemical Engineering Science, vol. 65, pp. 1720-1728, 2010.
- [7] Waller, J.B., and Toivonen, H.T. "A neuro-fuzzy model predictive controller applied to a ph-neutralization process". IFAC Proceedings Volumes, vol. 35, pp. 495-500, 2002.
- [8] Lawrynczuk, M. "Computationally efficient nonlinear predictive control based on neural Wiener models". Neurocomputing, vol. 74, pp. 401-417, 2010.
- [9] Degachi, H., Chagra, W., Ksouri, M. (2015, March). Global optimization method for model predictive control based on Wiener model. In 12th International Multi-Conference on Systems, Signals & Devices (SSD), pp. 1-6, 2015.
- [10] Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scaekaert, P.O. "Constrained model predictive control: Stability and optimality". Automatica, vol. 36, pp. 789-814, 2000.
- [11] Bello, O., Hamam, Y., and Djouani, K. "Nonlinear model predictive control of a coagulation chemical dosing unit for water treatment plants". IFAC Proceedings Volumes, vol. 47, pp. 370-376, 2014.
- [12] Xue, A., Peng, D., and Guo, Y. "Modeling of pH neutralization process using fuzzy recurrent neural network and DNA based NSGA-II". Journal of the Franklin Institute, vol. 351, pp. 3847-3864, 2014.
- [13] Tharakan, L.G., Benny, A., Jaffar, N.E., and Jaleel, J.A. "Neural network based pH control of a weak acid Strong base system". In International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing, pp. 674-679, 2013.
- [14] Mahmoodi, S., Poshtan, J., Jahed-Motlagh, M.R., and Montazeri, A. "Nonlinear model predictive control of a pH neutralization process based on Wiener Laguerre model". Chemical Engineering Journal, vol. 146, pp. 328-337, 2009.
- [15] Lawrynczuk, M. "Practical nonlinear predictive control algorithms for neural Wiener models". Journal of Process Control, vol. 2, pp. 696-714, 2013.
- [16] Lawrynczuk, M. "Modelling and predictive control of a neutralisation reactor using sparse support vector machine Wiener models". Neurocomputing, vol. 205, pp. 311-328, 2016.
- [17] Gomez, J.C., Jutan, A., and Baeyens, E. "Wiener model identification and predictive control of a pH neutralisation process". IEE Proceedings-Control Theory and Applications, vol. 151, pp. 329-338, 2004.
- [18] Norquay, S.J., Palazoglu, A., and Romagnoli, J.A. "Application of Wiener model predictive control (WMPC) to a pH neutralization experiment". IEEE Transactions on Control Systems Technology, vol. 7, pp. 437-445, 1999.
- [19] Peng, J., Dubay, R., Hernandez, J.M., Abu-Ayyad, M.M. "A Wiener neural network-based identification and adaptive generalized predictive control for nonlinear SISO systems". Industrial & Engineering Chemistry Research, vol. 5, pp. 7388-739, 2011.
- [20] Khani, F., and Haeri, M. "Robust model predictive control of nonlinear processes represented by Wiener or Hammerstein models". Chemical Engineering Science, vol. 129, pp. 223-231, 2015.
- [21] Allgower, F., Findeisen, R., Nagy, Z.K. "Nonlinear model predictive control: From theory to application". J. Chin. Inst. Chem. Engrs, vol. 3, pp. 299-315, 2004.
- [22] David, N., and Lubomir, M. "Self-Organizing Migrating Algorithm used to control a semi-batch chemical reactor". In 13th International Conference on Control, Automation and Systems, pp. 1266-1269, 2013.
- [23] Kadlec, P., and Raida, Z. "Multi-objective self-organizing migrating algorithm applied to the design of electromagnetic components". IEEE Antennas and Propagation Magazine, vol. 55, pp. 50-68, 2013.
- [24] dos Santos Coelho, L., and Alotto, P. "Electromagnetic optimization using a cultural self-organizing migrating algorithm approach based on normative knowledge". IEEE Transactions on Magnetics, vol. 4, pp. 1446-1449, 2009.
- [25] Qi, H., Niu, C.Y., Jia, T., Wang, D.L., and Ruan, L.M. "Multiparameter estimation in nonhomogeneous participating slab by using self-organizing migrating algorithms". Journal of Quantitative Spectroscopy and Radiative Transfer, vol. 157, pp. 153-169, 2015.
- [26] Zelinka, I. "SOMA Self-organizing Migrating Algorithm". In Self-Organizing Migrating Algorithm Springer International Publishing, 2016.
- [27] Kazemi, M., and Arefi, M.M. "A fast iterative recursive least squares algorithm for Wiener model identification of highly nonlinear systems". ISA transactions, vol. 67, pp. 382-388, 2017.
- [28] Hagenblad, A., Ljung, L., Wills, A., "Maximum likelihood identification of Wiener models". Automatica, vol. 44, pp. 2697-2705, 2008.
- [29] Al-Duwaish, H., Karim, M. N., Chandrasekar, V. "Use of multilayer feedforward neural networks in identification and control of Wiener model". IEE Proceedings-Control Theory and Applications, vol. 143, pp. 255-258, 1996.
- [30] Tatjewski, P. "Advanced control of industrial processes: structures and algorithms". Springer Science & Business Media, 2007.
- [31] Kheriji, A., Bouani, F., and Ksouri, M. "A GGP approach to solve non convex min-max predictive controller for a class of constrained MIMO systems described by state-space models". International Journal of Control, Automation and Systems. vol. 9, pp. 452-460, 2011.
- [32] Sahed, O.A., Kara, K., and Benyoucef, A. "Artificial bee colony-based predictive control for non-linear systems". Transactions of the Institute of Measurement and Control, vol. 37, pp. 780-792, 2015.
- [33] Al-Duwaish, H., and Naem, W. "Nonlinear model predictive control of hammerstein and wiener models using genetic algorithms". In Proceedings of the 2001 IEEE International Conference On Control Applications, pp. 465-469, 2001.
- [34] Chen, W., Li, X., and Chen, M. "Suboptimal nonlinear model predictive control based on genetic algorithm". In Third International Symposium on Intelligent Information Technology Application Workshops, pp. 119-124, 2009.
- [35] Kaddah, S.S., Abo-Al-Ez, K.M., and Megahed, T.F. "Application of nonlinear model predictive control based on swarm optimization in power systems optimal operation with wind resources". Electric Power Systems Research. vol. 143, pp. 415-430, 2017.

- [36] Ramirez, D.R., Arahal, M.R., and Camacho, E.F. "Min-max predictive control of a heat exchanger using a neural network solver". *IEEE transactions on control systems technology*, vol. 12. pp. 776-786, 2004.
- [37] Chagra, W., Degachi, H., and Ksouri, M. "Nonlinear model predictive control based on Nelder Mead optimization method". *Nonlinear Dynamics*, pp. 1-12, 2017.
- [38] Agrawal, S., and Singh, D. "Modified Nelder-Mead self organizing migrating algorithm for function optimization and its application". *Applied Soft Computing*, vol. 51. pp. 341-350, 2017.