Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 9497618, 8 pages
https://doi.org/10.1155/2018/9497618

Research Article

Filled Function Method for Nonlinear Model Predictive Control

Hajer Degachi ,[1] Bechir Naffeti,[2] Wassila Chagra,[1] and Moufida Ksouri[1]

[1] *Tunis El Manar University, National Engineering School of Tunis, Tunisia*
[2] *Carthage University, Faculty of Sciences of Bizerte, Tunisia*

Correspondence should be addressed to Hajer Degachi; degachihajer@gmail.com

Received 15 December 2017; Accepted 7 May 2018; Published 4 June 2018

Academic Editor: Guillermo Cabrera-Guerrero

A new method is used to solve the nonconvex optimization problem of the nonlinear model predictive control (NMPC) for Hammerstein model. Using nonlinear models in MPC leads to a nonlinear and nonconvex optimization problem. Since control performances depend essentially on the results of the optimization method, in this work, we propose to use the filled function as a global optimization method to solve the nonconvex optimization problem. Using this method, the control law can be obtained through two steps. The first step consists of determining a local minimum of the objective function. In the second step, a new function is constructed using the local minimum of the objective function found in the first step. The new function is called the filled function; the new constructed function allows us to obtain an initialization near the global minimum. Once this initialization is determined, we can use a local optimization method to determine the global control sequence. The efficiency of the proposed method is proved firstly through benchmark functions and then through the ball and beam system described by Hammerstein model. The results obtained by the presented method are compared with those of the genetic algorithm (GA) and the particle swarm optimization (PSO).

1. Introduction

Model predictive control (MPC) is a powerful control tool. This control strategy is formulated as the repeated solution of an open loop control problem [1]. At each sampling time, a control sequence is determined where only the first component is applied to the system and the optimization task is repeated at the next sampling instant.

The MPC algorithm presents the major advantage to efficiently handle constraints on input and output [2–4]. MPC is also able to control a wide variety of processes starting from systems that present a simple behavior like linear process [5] as well as those that exhibit more complex behavior like nonlinear [6, 7] and multivariable process [8].

The MPC theory was successfully applied in many areas of application such as chemical, petrochemical, pulp and paper, aerospace and defense, and food processing [9].

Firstly, MPC was well developed for linear models. Since almost real processes have a nonlinear behavior, this fact motivates the development of the NMPC strategy [4].

The NMPC is considered as a purely optimization-based algorithm. In fact, the minimization problem of the NMPC problem is nonlinear and nonconvex due to the nonlinear nature of the model. The determination of the control sequence, solution of the NMPC optimization problem, should be done using a global optimization algorithm to ensure good control performances. For this, a variety of solutions were proposed in literature to solve this kind of problem. Many works were focused on online linearization of the nonlinear model [10–12]. A second solution was proposed in some other works that consists of reversing the nonlinear block to remove nonlinearity. This solution is only addressed to block-oriented models [13] where the nonlinear block is described by a polynomial. The two mentioned solutions allow obtaining a quadratic cost function at each sampling time where the global solution can be easily determined.

Since solving a nonlinear optimization problem at each sampling time is a hard-computational task, [14] has proposed to describe the nonlinear process by a set of uncertain linear models to overcome the online computational complexity of the NMPC. Since the used model in the NMPC is nonlinear and since real process must operate under rigorous conditions, the nonlinear characteristic of the model should be kept without simplification during the online computation of the control sequence to get performances with sufficient accuracy.

For this, genetic algorithm (GA) [15], particle swarm optimization (PSO) [16], and neural network (NN) [17] were used as global optimization methods to solve the nonlinear optimization problem. Also, deterministic method such as Generalized Geometric Programming (GGP) was used as global optimization method for the NMPC [18, 19].

In this work, we will use the filled function as a global deterministic optimization method to solve the NMPC optimization problem. This method is based on the construction of a new function starting from the first local minimum found of the original function to be minimized and carry out the search of the global minimum using a local optimization method. This function allows determining the global minimum by finding an appropriate initialization.

In this work, the NMPC strategy is based on Hammerstein model. The Hammerstein model belongs to block-oriented models that is described by a nonlinear static block followed by a linear dynamic one. The special structure of the Hammerstein model was widely exploited to describe a variety of processes such as heat exchanger [17], SOFC [20], and pH neutralization process [21].

The rest of the paper is organized as follows. In Section 2, we describe the Hammerstein model. The NMPC based on Hammerstein model is presented in Section 3. In Section 4, we outline the optimization method used to solve the NMPC minimization problem. Simulation results on benchmark functions based on the global optimization method are given in Section 5. The efficiency of the proposed method for NMPC is illustrated in Section 6.

2. Hammerstein Model

The Hammerstein model is composed of a static nonlinear block followed by a linear dynamic one as depicted in Figure 1.

The outputs of the two blocks are given by the following equation:

$$S : \begin{cases} s(k) = f(u(k)) \\ \mathrm{A}\left(q^{-1}\right) y(k) = B\left(q^{-1}\right) s(k) \end{cases} \tag{1}$$

The polynomials $\mathrm{A}(q^{-1})$ and $B(q^{-1})$ are defined as

$$A\left(q^{-1}\right) = 1 + a_1 q^{-1} + a_2 q^{-2} + \ldots + a_{na} q^{-n_a} \tag{2}$$

$$B\left(q^{-1}\right) = b_1 q^{-1} + b_2 q^{-2} + \ldots + b_{nb} q^{-n_b} \tag{3}$$

where q^{-1} is the unit delay operator and n_a and n_b define the order of polynomials $\mathrm{A}(q^{-1})$ and $\mathrm{B}(q^{-1})$.

Various types of nonlinearity can be used to describe the nonlinear block of the Hammerstein model. In this work, the polynomial form is chosen, and the function f is defined as

$$f(u(k)) = c_1 u(k) + c_2 u(k)^2 + \cdots + c_{r_1} u(k)^{n_c} \tag{4}$$

where n_c is the order of the polynomial.

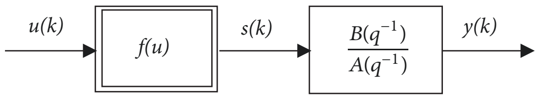

Figure 1: Hammerstein model.

3. Nonlinear Model Predictive Control Design

The aim of the NMPC strategy is to compute the control sequence$U = [u(k), u(k+1), \ldots, u(k+N_u-1)]$ by minimizing the criterion J defined as the difference between the desired trajectory $y^{sp}(k+j)$ and the model output $\hat{y}(k+j)$ over the prediction horizon N_p and penalizing the control increments over the control horizon N_u

$$\begin{aligned} J = {} & \sum_{i=1}^{N_p} \left(y^{sp}(k+i) - \hat{y}(k+i)\right)^2 \\ & + \lambda \sum_{i=0}^{N_u-1} \Delta u(k+i)^2 \end{aligned} \tag{5}$$

$$\begin{aligned} \Delta u^{\min} \le \Delta u(k+j) \le \Delta u^{\max}, \quad j = 0, \ldots, N_u - 1 \\ u^{\min} \le u(k+j) \le u^{\max}, \quad j = 0, \ldots, N_u - 1 \end{aligned} \tag{6}$$

where $\Delta u(k+i) = u(k+i) - u(k+i-1)$ and λ is a positive weighting coefficient.

Based on the receding horizon principle, only the first component will be applied to the system. Then the horizon will be shifted one step forward and the whole procedure will be repeated.

The predicted output of the Hammerstein model is defined as

$$\hat{y}(k+j) = -\sum_{i=1}^{n_a} a_i \hat{y}(k-i+j) + \sum_{i=1}^{n_b} b_i \hat{s}(k-i+j) \tag{7}$$

$\hat{s}(k+j)$ is the output of the nonlinear block defined as

$$\hat{s}(k+j) = \sum_{l=1}^{n_c} c_l u(k+j)^l \tag{8}$$

By using (7) and (8), the criterion expression (5) becomes

$$\begin{aligned} J(k) = {} & \sum_{j=1}^{N_p} \Bigg(y^{sp}(k+j) \\ & + \sum_{i=1}^{n_a} a_i \hat{y}(k-i+j) - \sum_{i=1}^{n_b} b_i \sum_{l=1}^{n_c} c_l u(k-i+j)^l \Bigg)^2 \\ & + \lambda \sum_{j=0}^{N_u-1} \Delta u(k+j \mid k)^2 \end{aligned} \tag{9}$$

Due to the nonlinear nature of Hammerstein model, the obtained objective function defined by (9) of the NMPC problem is nonconvex.

Solving the presented minimization problem using standard optimization methods will lead to suboptimal results in control. Deterministic optimization methods have the major benefit to offer the global solution. For this, we propose in this work to use the filled function method to solve the NMPC optimization problem. The present method was found to be easier to implement and offer an accurate solution but suffer from the high computation time.

4. Optimization Method

In this work, we will use the filled function method to solve the NMPC optimization problem.

The concept of this global optimization method consists of determining a good initialization to obtain the global minimum. Therefore, the key of this method is to construct a function "P" that allows moving from the local minimum of the objective function to obtain a more promising initialization point. This point presents a stationary point of the function P. The new initialization will allow finding easily the global minimum.

Thus, we consider a function $f(x)$ to be minimized. The principle of the filled function method can be summarized as follows [22].

Step 1. Use a local optimization method to find a local minimum x_1 of $f(x)$.

Step 2. Construct the filled function $P(x)$ at the local minimum x_1 of $f(x)$. Minimize the new function P using a local optimization method and so the minimum, t_1, of the filled function is found. The minimum $t_1$of the filled function is used as initialization to minimize again the function f, where a new better minimum will be found. Many filled functions are proposed in literature. Most of them are described using some parameters that can be difficult to be adjusted [23]. Other works minimize this number of parameters and reduce the filled function into a mono adjustable parameter function.

In this work, we will use the nonparametric filled function method proposed by [22] and defined as

$$P\left(\mathrm{t}, x_1\right) = \begin{cases} -\displaystyle\int_{x_1}^{x} \left(f\left(\mathrm{t}\right) - f\left(x_1\right)\right) dt & \left(x \geq x_1\right) \\ -\displaystyle\int_{x}^{x_1} \left(f\left(\mathrm{t}\right) - f\left(x_1\right)\right) dt & \left(x \leq x_1\right) \end{cases} \tag{10}$$

The nonparametric filled function P should satisfy the following properties.

Theorem 1. *If $x_1$is an isolated minimizer of $f : \Omega \subset R \rightarrow R$, then $x_1$must be a maximizer of $P(x, x_1)$.*

Theorem 2. *If $x_1$is an isolated minimizer of $f : \Omega \subset R \rightarrow R$, then $P(x, x_1)$ does not have any stationary point for $f(x) > f(x_1)$.*

Theorem 3. *If $x_1$is an isolated minimizer of $f : \Omega \subset R \rightarrow R$, then $P(x, x_1)$ must have a stationary point for $f(x) \leq f(x_1)$. The flowchart of the considered method is given in Figure 2 that details the different steps of the algorithm. The variable α in the algorithm presents a small step within the interval* $[0, 1]$.

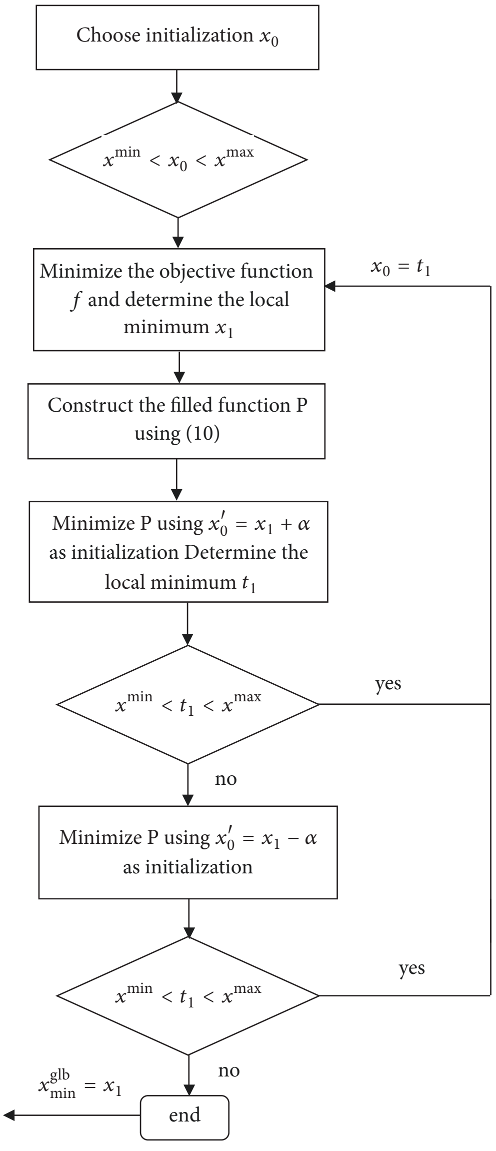
Choose initialization x_0
$x^{\min} < x_0 < x^{\max}$
Minimize the objective function f and determine the local minimum x_1
$x_0 = t_1$
Construct the filled function P using (10)
Minimize P using $x_0' = x_1 + \alpha$ as initialization Determine the local minimum t_1
$x^{\min} < t_1 < x^{\max}$
yes
no
Minimize P using $x_0' = x_1 - \alpha$ as initialization
$x^{\min} < t_1 < x^{\max}$
yes
no
$x^{\mathrm{glb}}_{\min} = x_1$
end

Figure 2: Flowchart of the filled function algorithm.

5. Simulation Results on Benchmark Function

The efficiency of the filled function is demonstrated through benchmark functions defined in Table 1. We consider 3 multimodal functions as depicted in Figures 3, 4, and 5. The function f_1 has a global minimum at $x = 17.0392$ and two local minima at $x = 5.3622$ and $x = 10.4535$. The function f_2 has a global minimum at $x = 5.1994$ and two local minima at $x = 3.4392$ and $x = 7.0678$. The function f_3 has a global minimum at x = -1 and a local one $x = 2$. The minimization is carried out using the filled function and the local method illustrated by the "fmincon" function of the MATLAB environment.

Consider the first function f_1.

Table 1: Benchmark functions.

Function	Min	Research interval
$f_1(x) = \sin(x) + \sin\left(\frac{2x}{3}\right)$	17.03	[0, 20]
$f_2(x) = \sin(x) + \sin\left(\frac{10x}{3}\right) + \ln(x) - 0.84x$	5.36	[2.7, 7.5]
$f_3(x) = x^4 - 3x^3 - 1.5x^2 + 10x$	-1	[-2, 3]

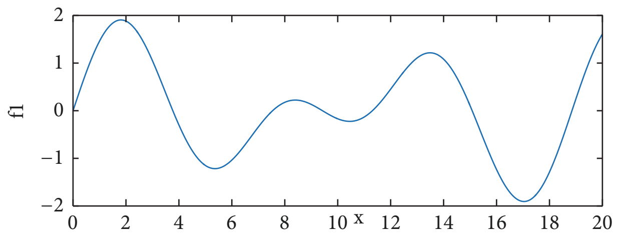

Figure 3: Evolution of the function $f1$.

Iteration 1.

Step 1. The first step in the filled function optimization method consists of using a local optimization method to determine a local minimum of f. We choose as initialization x_0=2.

In this work, we use the gradient descent method to find a local minimum x_1 of f. Using x_0=2 as initialization and using the gradient we can get a local minimum, x_1 =5.3622, as indicated in Figure 6

Step 2. Using x_1 we construct the filled function $P(t, x_1)$ using (10) as illustrated in Figure 6.

Step 3. Once we have P, we apply a local optimization method to determine a local minimum of P, and we set initialization as $t_0 = x_1 + \alpha$, where α is a small step real value. The obtained minimum of P is t_1

Iteration 2. If t_1 belongs to the interval of variation of the manipulated variable, it will be used in the second iteration as initialization for f.

Step 1. Since this condition is true in our case, t_1 presents a new initialization for f as depicted in Figure 6.

The new initialization presents a promising point to get a best minimum than that found in the first step.

The new minimum, x_1, of f found with initialization t_1 is shown in Figure 7.

Step 2. Using the new found minimum x_1 indicated in Figure 7 we construct the filled function $P(t, x_1)$ using (10) as illustrated in Figure 7. We can conclude from Figure 7 that P does not have a stationary point in the considered interval. So, the global minimum of f_1 is x_1=17.0328.

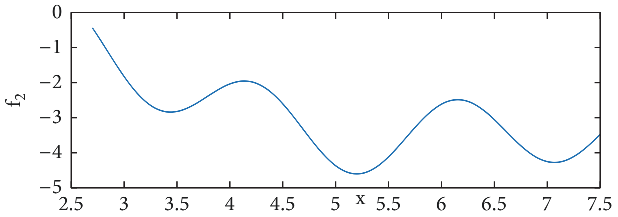

Figure 4: Evolution of the function f_2.

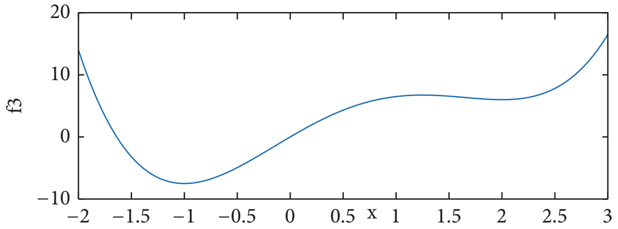

Figure 5: Evolution of the function f_3.

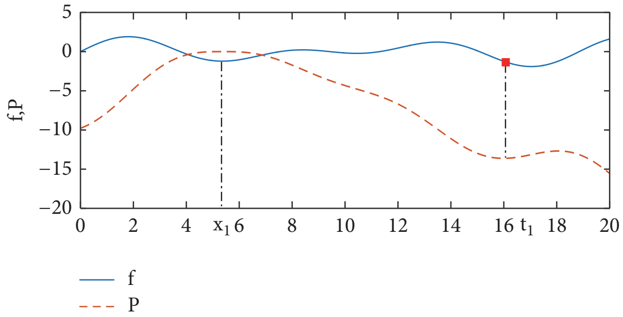

Figure 6: Evolution of $f1$ and P in the first iteration.

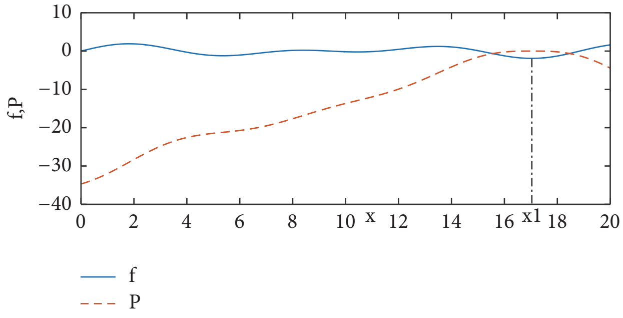

Figure 7: Evolution off_1 and P in the second iteration.

The minimization results for the considered function, f_1, f_2, and f_3, are illustrated, respectively, in Tables 1, 2, and 3

Based on results given by Tables 2, 3, and 4, we can note that, for different initialization values, the convergence to the global minimum using the local optimization method is not always guaranteed. Contrariwise, we can conclude that whatever the value of the initialization the minimization of the considered function using the proposed method is always guaranteed.

Table 2: Minimization of $f_1$using global and local optimization method.

Initialization	Reached min (proposed method)	Reached min (local method)
3	17.0328	5.3622
7	17.0328	5.3622
9	17.0328	10.4535
14	17.0328	17.0328

Table 3: Minimization of $f_2$using global and local optimization method.

Initialization	Reached min (proposed method)	Reached min (local method)
4.5	5.1995	3.4392
6	5.2006	7.0678
6.5	5.1994	5.1998
7	5.1994	7.0697

Table 4: Minimization of $f_3$using global and local optimization method.

Initialization	Reached min (proposed method)	Reached min (local method)
2.5	-1	-1
1.5	-1	2
-1.5	-1	-1

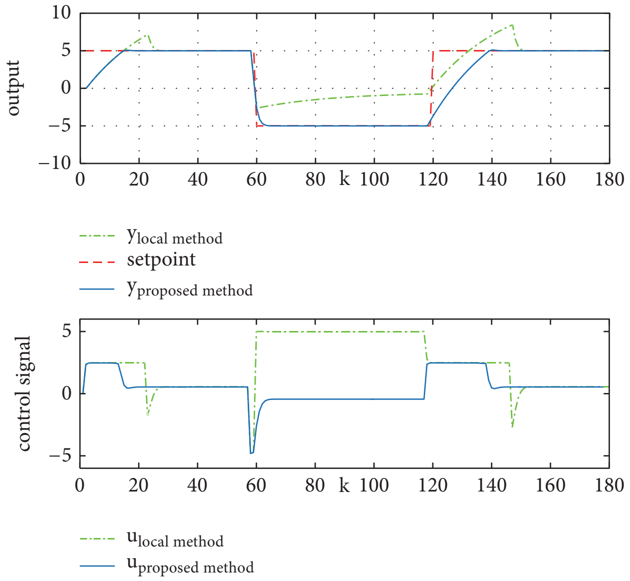

Figure 8: Output tracking using the proposed method and a local method for $\lambda = 0.1$.

6. Simulation Results

The system under consideration is a ball and beam which is composed of ball, ball position sensor, a center pivoted beam on which the ball rolls on, beam angle sensor, and the servomotor. The control objective is to fix the ball on a desired position. The ball and beam system is described by a Hammerstein model defined as [24]

$$S : \begin{cases} y(k) = -a_1 y(k-1) + b_1 s(k-1) \\ s(k) = c_1 u(k) + c_2 u(k)^2 \end{cases} \tag{11}$$

The different parameters of the linear and nonlinear blocks are given by $a_1 = -0.62$, $b_1 = 1$, $c_1 = 0.396$, and $c_2 = -0.036$.

The prediction and the control horizons are fixed to $N_p = 2$, $N_u = 1$ and constraints on the manipulated variable u are fixed as $-5 \leq u(k) \leq 5$.

First, the NMPC algorithm for the ball and beam system is implemented using the filled function method to solve the minimization problem. The desired trajectory is fixed as -5 and 5 represented by three step changes. The NMPC algorithm based on the proposed method is compared to a local optimization method illustrated by "fmincon" function of the MATLAB environment. The simulation results are depicted in Figure 8.

We can note from Figure 8 that the NMPC performances based on the filled function method are significantly better than those of the NMPC based on a local optimization method. The good ability of the filled function method to solve the optimization problem and offer the optimal control quality is justified by the fact that the NMPC based on the proposed method ensures the setpoint tracking with zero steady state error. Also, we can note that the response of the system based on the local optimization method is characterized by a high overshoot compared to the results given by the response of the system based on the filled function method.

In the sequel, the NMPC based on the proposed method is compared with the NMPC based on GA and PSO algorithms.

To compare the performances of the different algorithms we consider two performance indexes: the SSE and the SCV.

The SSE is defined as

$$SSE = \frac{1}{N} \sum_{i=1}^{N} \left(y^{sp}(k) - y(k) \right)^2 \tag{12}$$

Since our goal in control is to reach the desired trajectory and ensure the good output tracking by determining the optimal control sequence that gives the minimum dissipated energy, we will consider the second performance index SCV defined as

$$SCV = \frac{1}{N} \sum_{k=1}^{N} u(k)^2 \tag{13}$$

We can note from Figure 9 that the three algorithms are able to reach the desired trajectory and ensure good output tracking as proved by the SSE value. The significant difference lays in the control quality. Based on Figure 9 we can note than each algorithm reaches the desired trajectory with a different control sequence. Table 5 shows that the best

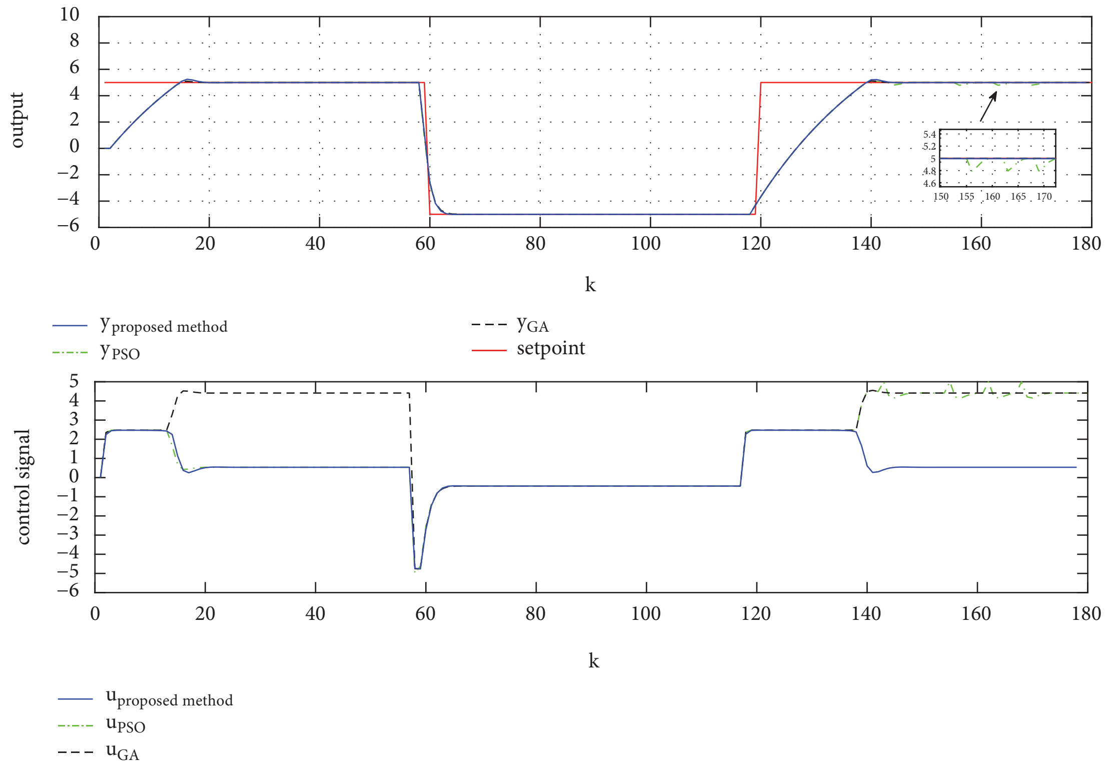

Figure 9: Output tracking using the GA, PSO, and the proposed method for $\lambda = 0.1$.

Table 5: The performance of the NMPC based on the three algorithms for $\lambda = 0.1$.

Method	Overshoot	*SCV*	*SSE*
Proposed method	0.0142	1.635	3.114
GA	0.0178	10.729	3.120
PSO	0.0178	5.990	3.112

Table 6: The performance of the NMPC based on the three algorithms for $\lambda = 0.2$.

Method	Overshoot	*SCV*	*SSE*
proposed method	0.0494	1.646	3.114
GA	0.0594	5.994	3.123
PSO	0.0594	10.681	3.124

control quality is given by the proposed method since it offers the small SCV. Also, we can remark that the filled function method offers the less overshoot as illustrated by Table 5. We can conclude that for the proposed method the setpoint is reached with the best control quality and the less overshoot. Although the results of setpoint tracking obtained by the GA, PSO, and the proposed method are quite similar, it is very clear that the control quality of GA and PSO is inferior to the filled function method. In fact, setting parameters of evolutionary algorithms as coding of individuals, population size, crossover and mutation rates, and stopping criterion has an important role in the determination of an accurate solution [25].

To show the effectiveness of our proposed method with different values of λ we suggest varying λ and carry out simulation for $\lambda = 0.2$.

We can notice from Figure 10 and Tables 5 and 6 that the increase of the weighting factor λ results in a high overshoot value and slow system response. Based on Table 4 we can conclude that the SCV offered by the filled function method is better than that given by GA and PSO. This confirms the good ability of the filled function method to give the best control quality.

Tables 5 and 6 give the performance of the different algorithms for $\lambda = 0.1$ and $\lambda = 0.2$. We can note that for different values of λ the less overshoot and SCV are given by the filled function method.

7. Conclusion

In this work, a nonlinear model predictive control for Hammerstein model is presented. Using nonlinear model in the MPC strategy complicates the solution of the optimization problem and leads to computational difficulties. Consequently, an efficient optimization algorithm should be used to overcome these difficulties and ensure good control results. The filled function, as a global optimization method, is used in this work to solve the nonconvex minimization problem of the NMPC. The proposed method proves its efficiency to give good control performances through nonlinear process

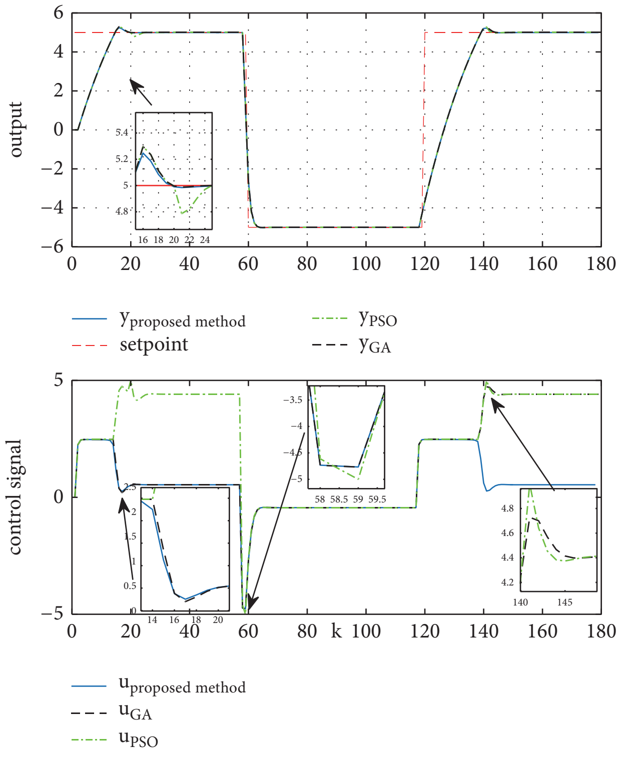

Figure 10: Output tracking using the GA, PSO, and the proposed method for $\lambda = 0.2$.

described by a ball and beam system. It offers a good output tracking and a less overshoot. The filled function method is then compared with two heuristic global optimization algorithms: GA and PSO. We can remark that the GA and the PSO algorithms are quite similar. We can also conclude that the NMPC with the proposed method is much better than the GA and PSO since it offers less overshoot and gives the best control sequence at each sampling time.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] F. Allgöwer, R. Findeisen, and Z. K. Nagy, "Nonlinear model predictive control: From theory to application," *J. Chin. Inst. Chem. Engrs*, vol. 35, no. 3, pp. 299–315, 2004.

[2] P. Tatjewski, *Advanced control of industrial processes: structures and algorithms*, Springer Science & Business Media.

[3] D. Knuth, *The Art of Programming*, 2007, pp. 82-86.

[4] B. Kouvaritakis and M. Cannon, Eds., *Non-linear Predictive Control: theory and practice*, vol. 61, 2001.

[5] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[6] E. Hellström, M. Ivarsson, J. Åslund, and L. Nielsen, "Look-ahead control for heavy trucks to minimize trip time and fuel consumption," *Control Engineering Practice*, vol. 17, no. 2, pp. 245–254, 2009.

[7] P. Ortner, R. Bergmann, H. J. Ferreau, and L. Del Re, "Nonlinear model predictive control of a diesel engine airpath," *IFAC Proceedings Volumes*, vol. 42, no. 2, pp. 91–96, 2009.

[8] A. Kheriji, F. Bouani, and M. Ksouri, "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, no. 3, pp. 452–460, 2011.

[9] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.

[10] M. Lawryńczuk, "Nonlinear predictive control of dynamic systems represented by Wiener-Hammerstein models," *Nonlinear Dynamics*, vol. 86, no. 2, pp. 1193–1214, 2016.

[11] M. Ławryńczuk, "Computationally efficient nonlinear predictive control based on neural Wiener models," *Neurocomputing*, vol. 74, no. 1-3, pp. 401–417, 2010.

[12] M. Ławryńczuk, "Practical nonlinear predictive control algorithms for neural Wiener models," *Journal of Process Control*, vol. 23, no. 5, pp. 696–714, 2013.

[13] S. J. Norquay, A. Palazoglu, and J. A. Romagnoli, "Application of Wiener model predictive control (WMPC) to a pH neutralization experiment," *IEEE Transactions on Control Systems Technology*, vol. 7, no. 4, pp. 437–445, 1999.

[14] F. Khani and M. Haeri, "Robust model predictive control of nonlinear processes represented by Wiener or Hammerstein models," *Chemical Engineering Science*, vol. 129, pp. 223–231, 2015.

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

[16] S. S. Kaddah, K. M. Abo-Al-Ez, and T. F. Megahed, "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.

[17] D. R. Ramírez, M. R. Arahal, and E. F. Camacho, "Min-max predictive control of a heat exchanger using a neural network solver," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 5, pp. 776–786, 2004.

[18] D. Hajer, C. Wassila, and K. Moufida, "Global optimization method for Min-Max MPC based on Wiener and Hammerstein model," in *Proceedings of the 2015 7th International Conference on Modelling, Identification and Control (ICMIC)*, pp. 1–6, Sousse, Tunisia, December 2015.

[19] W. Chagra, H. Degachi, and M. Ksouri, "Nonlinear model predictive control based on Nelder Mead optimization method," *Nonlinear Dynamics*, pp. 1–12, 2017.

[20] H.-B. Huo, X.-J. Zhu, W.-Q. Hu, H.-Y. Tu, J. Li, and J. Yang, "Nonlinear model predictive control of SOFC based on a Hammerstein model," *Journal of Power Sources*, vol. 185, no. 1, pp. 338–344, 2008.

[21] E. F. Camacho and C. B. Alba, *Model predictive control*, Springer Science & Business Media, 2007.

[22] B. M. Ismail, K. W. Goh, and B. D. Yosza, "Solving global optimization using non-parametric filled function," *Jurnal Mat Stat*, vol. 11, no. 2, 2011.

[23] W.-B. Du, Y. Gao, C. Liu, Z. Zheng, and Z. Wang, "Adequate is better: particle swarm optimization with limited-information," *Applied Mathematics and Computation*, vol. 268, pp. 832–838, 2015.

[24] A. Raees and M. B. Kadri, "Fuzzy Hammerstein model based generalized predictive control for ball and beam system," in *Proceedings of the 2012 International Conference on Emerging Technologies (ICET)*, pp. 1–6, Islamabad, Pakistan, October 2012.

[25] F. J. Lobo, C. F. Lima, and Z. Michalewicz, Eds., *Parameter setting in evolutionary algorithms*, vol. 54, Springer Science & Business Media, 54, 2007.

Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
The Scientific World Journal
Journal of Probability and Statistics
International Journal of Mathematics and Mathematical Sciences
Hindawi
Submit your manuscripts at
www.hindawi.com
Journal of Optimization
International Journal of Engineering Mathematics
International Journal of Analysis
Journal of Complex Analysis
Advances in Numerical Analysis
Mathematical Problems in Engineering
International Journal of Differential Equations
Discrete Dynamics in Nature and Society
International Journal of Stochastic Analysis
Journal of Mathematics
Journal of Function Spaces
Abstract and Applied Analysis
Advances in Mathematical Physics