Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 9402684, 15 pages
https://doi.org/10.1155/2017/9402684

Research Article

Data-Driven Optimization Framework for Nonlinear Model Predictive Control

Shiliang Zhang,[1] Hui Cao,[1] Yanbin Zhang,[1] Lixin Jia,[1] Zonglin Ye,[1] and Xiali Hei[2]

[1] *State Key Laboratory of Smart Grid, State Key Laboratory of Electrical Insulation and Power Equipment, School of Electrical Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China*
[2] *Department of Computer and Information Sciences, Delaware State University, Dover, DE 19901, USA*

Correspondence should be addressed to Lixin Jia; lxjia@mail.xjtu.edu.cn

Received 5 June 2017; Accepted 24 July 2017; Published 28 August 2017

Academic Editor: Asier Ibeas

The structure of the optimization procedure may affect the control quality of nonlinear model predictive control (MPC). In this paper, a data-driven optimization framework for nonlinear MPC is proposed, where the linguistic model is employed as the prediction model. The linguistic model consists of a series of fuzzy rules, whose antecedents are the membership functions of the input variables and the consequents are the predicted output represented by linear combinations of the input variables. The linear properties of the consequents lead to a quadratic optimization framework without online linearisation, which has analytical solution in the calculation of control sequence. Both the parameters in the antecedents and the consequents are calculated by a hybrid-learning algorithm based on plant data, and the data-driven determination of the parameters leads to an optimization framework with optimized controller parameters, which could provide higher control accuracy. Experiments are conducted in the process control of biochemical continuous sterilization, and the performance of the proposed method is compared with those of the methods of MPC based on linear model, the nonlinear MPC with neural network approximator, and MPC nonlinear with successive linearisations. The experimental results verify that the proposed framework could achieve higher control accuracy.

1. Introduction

Model predictive control (MPC) has been recognised as an efficient means and applied in many industrial applications successfully [1–5]. MPC refers to a class of computer algorithms that use a dynamic model to predict future behavior of the process [6–8]. In the family of MPC, the MPC based on linear model is more applicable as it brings about a quadratic programming optimization problem in online control [9], where analytical solutions exist. Since properties of many technological processes are nonlinear, the linear MPC algorithms are not likely to work properly all the time [10]. In the case of severe nonlinear process, linear MPC algorithms may result in poor control performance and even instability. Thus, nonlinear MPC is considered necessary.

Nonlinear MPC schemes based on the direct use of nonlinear models have been proposed, like the nonlinear quadratic dynamic matrix control [11] and the modified nonlinear internal model control [12] as well as their extensions. Both the two schemes require a fundamental model. The derivation of fundamental models can be very time consuming and even elusive if the process is not well understood, which limits its application in practice. With the plant data, a nonlinear empirical model can be identified with less computational intensity. Some MPC algorithms employ the data-driven method for more effective implementation of control schemes. A nonlinear MPC scheme based on the second-order Volterra series model was proposed, where a successive substitution algorithm is employed [13]. Neural networks provide an alternative nonlinear modeling approach for MPC [14, 15]. Nonlinear MPC algorithms based on neural network [16, 17] imply the minimization of a cost function using computational methods to obtain the optimal control actions [18]. In the control procedure of MPC based on neural network model, gradients of the cost function are approximated numerically and the nonlinear optimization problem is solved online [19, 20]. The gradient-based optimization techniques may terminate in local minima, while global

minima substantially increase the computational burden, yet they still give no guarantee that the global optimal solution is found [21].

An efficient optimization framework is the key to successful implementation of nonlinear MPC. Some MPC algorithms with improved optimization structures have been proposed. Model predictive control with neural network approximator [22–24] (MPC-NNA) and its extension [25] are proposed to approximate the solution of the optimization procedure in nonlinear MPC. With an approximator trained by offline calculations [26, 27], the whole MPC algorithm in online control is replaced by the neural network approximator [28]. Though this method provides computational convenience in online control, the training process of the neural network approximator still cannot get rid of the nonconvex nonlinear optimization problem. Linearisation technologies could provide quadratic optimization framework in nonlinear MPC, and analytical solution exists in such case. In most practical applications, the accuracy of the linearisation method is sufficient compared with full nonlinear optimization in model predictive control [29, 30]. Nonlinear MPC based on multiple piecewise linear models [31] is proposed to calculate the manipulated variable based on a series of local linear MPC controllers, where quadratic optimization problem exists and analytical solution could be obtained. In this approach, the recurrent neurofuzzy [32] model is employed to represent the process, which is partitioned into several fuzzy operating regions. Within each region, a local linear model is used to represent the process. Nevertheless, the coefficients of the membership function in the local linear models are determined subjectively, which may result in undesirable control accuracy. MPC nonlinear with successive linearisation (MPC-NSL) has been described [33, 34], where online linearisation is performed at each sampling instant. With the linearisation procedure, a linear approximation model is obtained and a quadratic programming optimization can be derived. Some extensions have been reported like the MPC with nonlinear prediction and linearisation [35–37], which considers the free response of the nonlinear model. Though simple in implementation, the online linearisation approach may be susceptible to system disturbances and lower the control accuracy. A linguistic modeling method [38, 39] is reported where a series of linear models are used to represent the nonlinear model. A quadratic optimization problem without online linearisation could be obtained in the control procedure of MPC for the sake of the linear property of the consequents of the rules in the linguistic model [40]. The coefficients of the linguistic model are identified based on the plant data without subjective factors, resulting in more effective optimization procedure in MPC.

In this paper, a data-driven optimization framework for nonlinear MPC is proposed, which employs the linguistic model as the prediction model. The linguistic model consists of a series of fuzzy rules, whose antecedents and consequents are the membership functions of the input variables and the predicted output calculated based on linear combinations of the input variables, respectively. Because of the linear properties of the consequent, a quadratic optimization framework can be derived where analytical solution exists. With the resulting quadratic optimization, the nonconvex nonlinear optimization or the susceptible-to-disturbance online linearisation procedure is avoided in nonlinear MPC. Based on the plant data, the antecedent parameters are initialized by the Gaussian kernel fuzzy clustering algorithm, and a hybrid-learning algorithm is employed to tune the coefficients of both the antecedents and the consequents of the model. The data-driven determination of the antecedent and consequent in the linguistic model provides optimized parameters free of subjective factors, which could bring better control accuracy. The process control of continuous sterilization is employed in the experiments to evaluate the effectiveness of the proposed framework, and the performance of the proposed framework is compared with those of the MPC based on linear model, MPC-NNA, and the MPC-NSL.

The article is organized as follows: the MPC problem formulation is briefly shown in Section 2; the proposed optimization framework is explained in detail in Section 3; Section 4 shows the experiments results of the process control of biochemical continuous sterilization; conclusions are given in Section 5.

2. MPC Problem Formulation

MPC algorithms aim to achieve accurate and fast control. The most important task is to minimize the difference between the controlled output variable y and its desired reference trajectory y^{ref}. Also, the changes of the value of the manipulated variable are expected not to be very big [24]. A set of control increments is calculated at each sampling instant t by the MPC algorithms as

$$\begin{aligned}\Delta\mathbf{u}(t) = \big[&\Delta u(t+1 \mid t) \\ &\cdot \Delta u(t+2 \mid t) \cdots \Delta u\big(t+H_c-1 \mid t\big)\big]^T,\end{aligned} \tag{1}$$

where H_c is the control horizon, and the increment in the equation is defined as

$$\begin{aligned}\Delta u(t+k \mid t) = u(t+k \mid t) - u(t+k-1 \mid t), \\ k \in \{1, 2, \ldots, H_c - 1\}.\end{aligned} \tag{2}$$

To minimize the difference between the predicted output $\hat{y}(t+k \mid t)$ and the reference trajectory $y^{\text{ref}}(t+k \mid t)$ over the prediction horizon N and to penalize excessive control increments, the following quadratic cost function is usually used in MPC algorithms:

$$\begin{aligned}C(t) = &\sum_{k=1}^{N}\left(\hat{y}(t+k \mid t) - y^{\text{ref}}(t+k \mid t)\right)^2 \\ &+ \sum_{k=0}^{H_c}\lambda\Delta u(t+k \mid t)^2,\end{aligned} \tag{3}$$

where λ is the weighting coefficient. The control increments are calculated by minimizing the cost function online. Only the first value of the obtained control increments is applied to the control process, and the whole procedure is repeated at the next sampling instant.

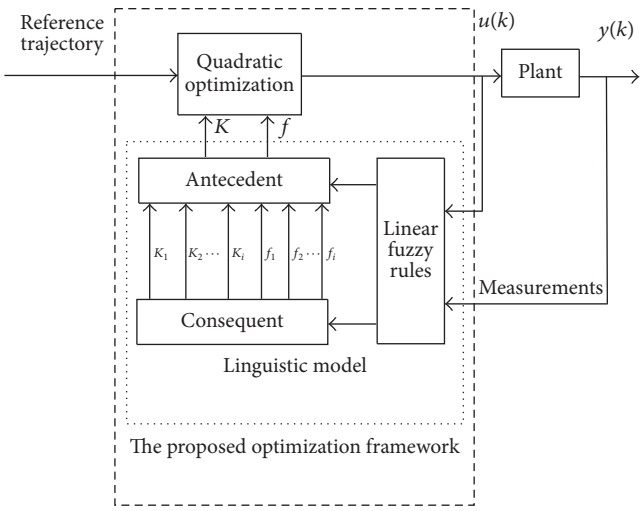


Figure 1: Data-driven quadratic optimization framework for nonlinear MPC.

An explicit model is used in MPC algorithms to predict the value of future outputs of the process, based on which the MPC algorithms are derived. The selection of a suitable model is a critical task in constructing an effective MPC algorithm, as different models lead to different MPC algorithms and different optimization framework in minimizing the cost function illustrated in the Introduction. In this paper, the linguistic model is employed to improve control performance in MPC.

3. The Proposed Framework

Figure 1 shows the block diagram of the proposed framework. The linguistic model is employed as the prediction model to make predictions on system output. The antecedent of the linear fuzzy rules in the model is the membership functions of the input variables, and the consequent is the predicted output represented by system input variables. K_i and f_i mean the forced response parameters and the free response parameters corresponding to the ith rule, respectively. Both of K_i and f_i are calculated by the consequent. The final value of the forced response K and the final value of the free response f are calculated by the antecedent considering all the rules. Given the reference trajectory of the controlled variable, a quadratic optimization problem could be derived where the manipulated values within the control horizon can be calculated. Only the first value of the manipulated values is employed by the plant, and the whole optimization procedure is repeated in the following control process.

3.1. Linguistic Modeling Based on Plant Data. In the proposed framework, predictions on the system dynamic are made based on the linguistic model. The parameters of the linguistic model are identified based on the plant data. The linguistic model consists of a series of Tagaki-Sugeno- (TS-) type fuzzy rules, where both the antecedent parameters and the consequent parameters of the fuzzy rules are calculated in an optimized way.

3.1.1. Initialization of the Fuzzy Rules. Given that there exist $x_1, x_2, \ldots, x_n$ as the input variables and y as the output variable, the centers of the clusters of the process input-output dataset can be obtained by the Gaussian kernel fuzzy clustering algorithm. The centers of the clusters are used to generate the initial membership functions and a set of rules, which are used to calculate the weighted parameters of the linear models. Gaussian kernel fuzzy clustering algorithm is based on FCM clustering algorithm, and it replaces the Euclidean distance with a Gaussian kernel-induced distance [41, 42]. Assuming that there are C_n clusters obtained, then the number of linguistic terms and the number of the total TS-type rules are both C_n. The Gaussian function is employed as the membership function and the ith rule's firing strength μ_i is calculated by

$$\mu_i = \prod_{j=1}^{n} e^{(x_j - c_{ij})^2 / 2\sigma_{ij}^2}, \quad j \in \{1, 2, \ldots, n\}, \; i \in \{1, 2, \ldots, C_n\}, \tag{4}$$

where c_{ij} is the jth dimension value of the ith cluster's center and σ_{ij} is the width of the membership function, which is usually set as $[\max(x_j) - \min(x_j)]/\sqrt{8}$. With the initial parameters of the antecedent in the fuzzy obtained rules, the initial parameters of the consequent in linear models are estimated as

$$l_{ij} = \frac{y}{x_j} \cdot \frac{\sum_{i=1}^{C_n} \mu_i}{\mu_i}, \tag{5}$$

where l_{ij} is the jth input variable's coefficient corresponding to the ith rule.

With both the initial antecedent and the consequent parameters of the fuzzy rules obtained, the predicted system output $\hat{y}$ could be calculated by a set of n-dimensional inputs:

$$\hat{y} = \frac{\sum_{i=1}^{C_n} \mu_i y_i^*}{\mu_i}, \tag{6}$$

where y_i^* is the predicted output variable under the ith fuzzy rule and $y_i^* = l_{i1}x_1 + l_{i2}x_2 + \cdots + l_{in}x_n$.

3.1.2. Optimization of the Parameters in the Fuzzy Rules. For better prediction of the system output, this paper employs an iterative hybrid-learning algorithm to adjust the initial parameters of the fuzzy rules.

Define the matrices P and S as

$$P = \left[l_{11}, \ldots, l_{1n}, l_{21}, \ldots, l_{2n}, \ldots, l_{C_n 1}, \ldots, l_{Cn}\right]^T,$$
$$S = \begin{bmatrix} m & 0 & 0 & \cdots & 0 \\ 0 & m & 0 & \cdots & 0 \\ 0 & 0 & m & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & m \end{bmatrix}_{N \times N}, \tag{7}$$

where $N = C_n \cdot n$ and m represents a positive infinite value, which is often assigned as 1e6. For the kth entry of the training data, $\{x_1(k), x_2(k), \ldots, x_n(k), y(k)\}$, where $k \in 1, 2, \ldots, N_t$ and N_t is the total number of the training data. Assuming that

$$\begin{aligned}\alpha(k) = \big[&\mu_1(k)\, x_1(k), \mu_1(k)\, x_2(k), \ldots, \mu_1(k) \\ &\cdot x_n(k), \mu_2(k)\, x_1(k), \mu_2(k)\, x_2(k), \ldots, \mu_2(k) \\ &\cdot x_n(k), \ldots, \mu_{C_n}(k)\, x_1(k), \mu_{C_n}(k)\, x_2(k), \ldots, \mu_{C_n}(k) \\ &\cdot x_n(k)\big]^T,\end{aligned} \tag{8}$$

where $\mu_i(k)$, $i \in \{1, 2, \ldots, C_n\}$ is the degree of membership of the ith rule corresponding to the kth entry of the training data, P and S could be approximated as

$$\begin{aligned}P(k+1) &= P(k) + S(k) \\ &\quad \cdot \left[\alpha(k) \cdot \left(y(k) - (\alpha(k))^T P(k)\right)\right], \\ S(k+1) &= S(k) - \frac{(S(k) \cdot \alpha(k))\left((\alpha(k))^T \cdot S(k)\right)}{1 + (\alpha(k))^T \cdot (S(k) \cdot \alpha(k))}.\end{aligned} \tag{9}$$

The initial value of S is defined in (7). Let the initial P be a zero matrix, and the new values of l_{ij} can be obtained in the iterative calculation. The iteration will not stop until the RMSE is no larger than TH, which is the user-defined threshold. The update of the antecedent parameters of the fuzzy rules, c_{ij} and σ_{ij}, in the Gaussian membership functions is shown as follows. For the training data $\{x_1(k), x_2(k), \ldots, x_n(k), y(k)\}$, $k \in \{1, 2, \ldots, N_t\}$, de_k can be calculated as

$$de_k = (-n \cdot C_n) \sum_{i=1}^{C_n} \frac{-2\mu_i(k)\, y_i^*(k) \cdot \left(y(k) - \hat{y}(k)\right)}{\sum_{i=1}^{C_n} \mu_i(k)}, \tag{10}$$

where $\hat{y}(k)$ is the predicted value of the linguistic model corresponding to the input $\{x_1(k), x_2(k), \ldots, x_n(k), y(k)\}$ and $y_i^*(k)$ means the predicted output under the ith fuzzy rule. Assuming that T^c and T^σ are both $C_n \times n$ matrices, the jth column elements of the ith row of T^c and T^σ in the $(k+1)$th iterative calculation are

$$\begin{aligned}T^c_{(i,j)}(k+1) &= T^c_{(i,j)}(k) + \sum_{l=1}^{N_t} de_l \cdot \frac{x_j(l) - c_{ij}(k)}{\sigma_{ij}(k)} \\ &\quad \cdot e^{(x_j(l) - c_{ij}(k))^2 / 2\sigma_{ij}^2(k)}, \\ T^\sigma_{(i,j)}(k+1) &= T^\sigma_{(i,j)}(k) + \sum_{l=1}^{N_t} de_l \cdot \frac{x_j(l) - c_{ij}(k)}{\sigma_{ij}(k)} \\ &\quad \cdot e^{(x_j(l) - c_{ij}(k))^2 / 2\sigma_{ij}^2(k)},\end{aligned} \tag{11}$$

where the initial T^c and T^σ are zero matrices. The values of c_{ij} and σ_{ij} in the $(k+1)$th iterative calculation can be obtained by

$$\begin{aligned}c_{ij}(k+1) &= c_{ij}(k) \\ &\quad - \frac{\mathrm{ts} \cdot T^c_{(i,j)}(k)}{\sqrt{\sum_{i=1}^{C_n} \sum_{j=1}^{n} \left[\left(T^c_{(i,j)}(k)\right)^2 + \left(T^\sigma_{(i,j)}(k)\right)^2\right]}}, \\ \sigma_{ij}(k+1) &= \sigma_{ij}(k) \\ &\quad - \frac{\mathrm{ts} \cdot T^c_{(i,j)}(k)}{\sqrt{\sum_{i=1}^{C_n} \sum_{j=1}^{n} \left[\left(T^c_{(i,j)}(k)\right)^2 + \left(T^\sigma_{(i,j)}(k)\right)^2\right]}},\end{aligned} \tag{12}$$

where ts is the training step.

The iterative calculation goes on until the value of RMSE is less than the defined threshold TH or the number of iterations reaches the defined value. When the iteration stops, all the optimized parameters are obtained and the linguistic model is established. As a result, the system dynamic is presented by the nonlinear linguistic model consisting of a series of weighted linear models. The linguistic model will be employed in nonlinear MPC for a quadratic optimization framework, which will be shown in detail in the next section.

3.2. The Quadratic Programming Optimization Framework Based on Linguistic Model for Nonlinear Model Predictive Control. Assuming that $y(t)$ is the system output and $u(t)$ is the manipulated input variable at the sampling instant t, the system dynamic can be represented by the linguistic model as follows:

$$y(t) = \frac{\sum_{i=1}^{C_n} \mu_i \cdot y_i^*}{\sum_{i=1}^{C_n}}, \tag{13}$$

where

$$\mu_i = \prod_{p=1}^{P} e^{(y(t-p) - cy_{ip})^2 / 2\sigma y_{ip}^2} \cdot \prod_{q=0}^{Q} e^{(u(t-q-1) - cu_{iq})^2 / 2\sigma u_{iq}^2} \tag{14}$$

and the integrated Controller Autoregressive Moving-Average (CARIMA) model [43] is used to describe the consequent of the linguistic model as follows:

$$y_i^* = \sum_{p=1}^{P} a_{ip} \cdot y(t-p) + \sum_{q=0}^{Q} b_{iq} u(t-q-1) + \frac{e_i(t)}{\Delta}, \tag{15}$$

where cy_{ip} means the center of the cluster of $y(t-p)$ under the ith fuzzy rule, cu_{iq} means the center of the cluster of $u(t-q)$ under the ith rule, σy_{ip} means the membership function width of $y(t-p)$ under the ith rule, σu_{iq} means the membership function width of $u(t-q-1)$ under

the ith rule, and $\Delta = 1 - q^{-1}$ means the differencing operator, while q^{-1} means the backward shift operator. All the centers of the cluster, the membership functions width, and the linear model parameters a_{ip}, $p \in \{1, 2, \ldots, P\}$ and b_{iq}, $q \in \{1, 2, \ldots, Q\}$ are determined in the linguistic modeling process. Equation (15) can be rewritten as

$$A\left(q^{-1}\right) y_i^*(t) = B_i\left(q^{-1}\right) u(t-1) + \frac{e_i(t)}{\Delta} \tag{16}$$

with

$$\begin{aligned} A_i\left(q^{-1}\right) &= 1 - a_{i1}q^{-1} - a_{i2}q^{-2} - \cdots - a_{iP}q^{-P}, \\ B_i\left(q^{-1}\right) &= b_{i0} + b_{i1}q^{-1} + b_{i2}q^{-2} + \cdots + b_{iQ}q^{-Q}. \end{aligned} \tag{17}$$

Consider the Diophantine equation:

$$\begin{aligned} 1 = E_{ij}\left(q^{-1}\right)\widetilde{A}_i\left(q^{-1}\right) + q^{-j}F_{ij}\left(q^{-1}\right) \\ \text{with } \widetilde{A}_i\left(q^{-1}\right) = \Delta A_i\left(q^{-1}\right). \end{aligned} \tag{18}$$

The degree of the polynomials E_{ij} and F_{ij} is defined as $j-1$ and $P+Q$, respectively, and both the two polynomials could be recursively calculated [43]. Multiply (16) by $\Delta E_{ij}q^j$; it can be obtained that

$$\begin{aligned} &\widetilde{A}_i E_{ij}\left(q^{-1}\right) y_i^*(t+j) \\ &\quad = E_{ij}\left(q^{-1}\right) B_i\left(q^{-1}\right) \Delta u(t+j-1) \\ &\qquad + E_{ij}\left(q^{-1}\right) e_i(t+j). \end{aligned} \tag{19}$$

Given (18) and (19), it can be obtained that

$$\begin{aligned} &\left(1 - q^{-j}F_{ij}\left(q^{-1}\right)\right) y_i^*(t+j) \\ &\quad = E_{ij}\left(q^{-1}\right) B_i\left(q^{-1}\right) \Delta u(t+j-1) \\ &\qquad + E_{ij}\left(q^{-1}\right) e_i(t+j) \end{aligned} \tag{20}$$

which could be rewritten as

$$\begin{aligned} y_i^*(t+j) &= F_{ij}\left(q^{-1}\right) y(t) \\ &\quad + E_{ij}B_i\left(q^{-1}\right) \Delta u(t+j-1) \\ &\quad + E_{ij}\left(q^{-1}\right) e_i(t+j). \end{aligned} \tag{21}$$

Because the degree of $E_{ij}(q^{-1})$ is $j-1$, the noise terms are all in the future in (20). Thus, the best prediction of $y_i^*(t+j)$ is calculated by

$$y_i^*(t+j \mid t) = K_{ij}\Delta u(t+j-1) + F_{ij}\left(q^{-1}\right) y(t), \tag{22}$$

where $K_{ij}(q^{-1}) = E_{ij}(q^{-1})B_i(q^{-1})$. For the degree of $B_i(q^{-1})$ is Q, it can be gained that the degree of $K_{ij}(q^{-1})$ is $j+Q-1$. So it could be obtained that

$$\begin{aligned} K_{ij}\left(q^{-1}\right) &= k_{i,0} + k_{i,1}q^{-1} + k_{i,2}q^{-2} + \cdots \\ &\quad + k_{i,j+Q-1}q^{j+Q-1}. \end{aligned} \tag{23}$$

For the ith rule of the linguistic model, the N ahead optimal predictions of the system output can be calculated as

$$\begin{aligned} \hat{y}_i^*(t+1 \mid t) &= K_{i1}\left(q^{-1}\right)\Delta u(t) + F_{i1}\left(q^{-1}\right) y(t), \\ \hat{y}_i^*(t+2 \mid t) &= K_{i2}\left(q^{-1}\right)\Delta u(t+1) + F_{i2}\left(q^{-1}\right) y(t), \\ &\vdots \\ \hat{y}_i^*(t+N \mid t) &= K_{iN}\left(q^{-1}\right)\Delta u(t+N-1) \\ &\quad + F_{iN}\left(q^{-1}\right) y(t) \end{aligned} \tag{24}$$

which can be rewritten as

$$\hat{\mathbf{y}}_i^* = \mathbf{K}_i\mathbf{u} + \mathbf{F}_i\left(q^{-1}\right) y(t) + \mathbf{K}_i'\Delta u(t-1), \tag{25}$$

where

$$\begin{aligned} \hat{\mathbf{y}}_i^* &= \begin{bmatrix} \hat{y}_i^*(t+1 \mid t) \\ \hat{y}_i^*(t+2 \mid t) \\ \vdots \\ \hat{y}_i^*(t+N \mid t) \end{bmatrix}, \\ \mathbf{u} &= \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+N-1) \end{bmatrix}, \\ \mathbf{K}_i &= \begin{bmatrix} k_{i,0} & 0 & 0 & \cdots & 0 \\ k_{i,1} & k_{i,0} & 0 & \cdots & 0 \\ k_{i,2} & k_{i,1} & k_{i,0} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k_{i,N-1} & k_{i,N-2} & k_{i,N-3} & \cdots & k_{i,0} \end{bmatrix}, \\ &\mathbf{K}_i'\left(q^{-1}\right) \\ &= \begin{bmatrix} \left(K_{i1}\left(q^{-1}\right) - k_{i,0}\right) q \\ \left(K_{i2}\left(q^{-1}\right) - k_{i,0} - k_{i,1}q^{-1}\right) q^2 \\ \vdots \\ \left(K_{iN}\left(q^{-1}\right) - k_{i,0} - k_{i,1}q^{-1} - \cdots - k_{i,N-1}q^{N-1}\right) q^N \end{bmatrix}, \\ \mathbf{F}_i\left(q^{-1}\right) &= \begin{bmatrix} F_{i1}\left(q^{-1}\right) \\ F_{i2}\left(q^{-1}\right) \\ \vdots \\ F_{iN}\left(q^{-1}\right) \end{bmatrix}. \end{aligned} \tag{26}$$

The last two terms in (26) depend only on the variables of the past instants, assuming that

$$\mathbf{f}_i\left(q^{-1}\right) = \begin{bmatrix} f_{i,1} \\ f_{i,2} \\ \vdots \\ f_{i,N} \end{bmatrix} = \mathbf{F}_i\left(q^{-1}\right) y(t) + \mathbf{K}_i' \Delta u(t-1) \tag{27}$$

and (25) could be rewritten as

$$\hat{\mathbf{y}}_i^* = \mathbf{K}_i \mathbf{u} + \mathbf{f}_i. \tag{28}$$

Considering all C_n rules of the linguistic model, the final optimal predictions N ahead could be calculated as follows:

$$\hat{\mathbf{y}} = \frac{\sum_{i=1}^{C_n} \mu_i \cdot \hat{\mathbf{y}}_i^*}{\sum_{i=1}^{C_n} \mu_i} \tag{29}$$

which could be represented by

$$\hat{\mathbf{y}} = \mathbf{K}\mathbf{u} + \mathbf{f}, \tag{30}$$

where

$$\mathbf{K} = \begin{bmatrix} k_0 & 0 & 0 & \cdots & 0 \\ k_1 & k_0 & 0 & \cdots & 0 \\ k_2 & k_1 & k_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k_{N-1} & k_{N-2} & k_{N-3} & \cdots & k_0 \end{bmatrix},$$

$$k_j = \frac{\sum_{i=1}^{C_n} \mu_i k_{i,j}}{\sum_{i=1}^{C_n} \mu_i}, \tag{31}$$

$$\mathbf{f}\left(q^{-1}\right) = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix},$$

$$f_j = \frac{\sum_{i=1}^{C_n} \mu_i f_{i,j}}{\sum_{i=1}^{C_n} \mu_i}.$$

Based on the predictions, the optimal control sequence could be calculated by minimizing a cost function. Considering the constraint on the control effort, the cost function could be presented as

$$C = \sum_{l=1}^{N} \left[\hat{y}(t+l \mid t) - w(t+l)\right]^2 + \sum_{l=1}^{N} \lambda \left[\Delta u(t+l-1)\right]^2 \tag{32}$$

which could be rewritten as

$$C = (\mathbf{K}\mathbf{u} - \mathbf{f} - \mathbf{w})^T (\mathbf{K}\mathbf{u} - \mathbf{f} - \mathbf{w}) + \lambda \mathbf{u}^T \mathbf{u}, \tag{33}$$

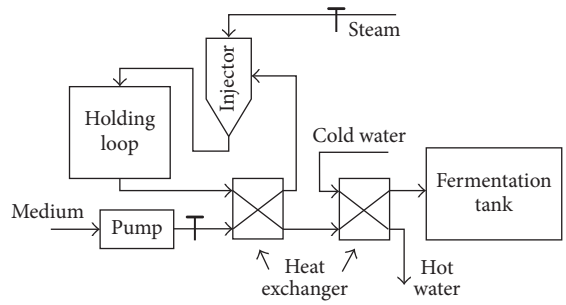


Figure 2: The continuous sterilization process.

where

$$\mathbf{w} = \left[w(t+1), w(t+2), \ldots, w(t+N)\right]^T \tag{34}$$

and $w(t+i)$, $i \in \{1, 2, \ldots, N\}$ means the reference trajectory of the controlled output variable at the instant $t+i$, and λ means the penalty coefficient on the control effort. Assuming that no more constraint is considered, the minimum of the cost function is calculated by making the gradient of C equal 0, resulting in

$$\mathbf{u} = \left(\mathbf{K}^T\mathbf{K} + \lambda\mathbf{I}\right)^{-1} \mathbf{K}^T (\mathbf{w} - \mathbf{f}). \tag{35}$$

Only the first value in $\mathbf{u}$ is passed to the control process. In the following sampling instants, the procedure will be repeated.

Now the data-driven optimization framework for nonlinear MPC has been described. Experiments are conducted in the next section to verify the effectiveness of the proposed framework.

4. Experiment Implementation

The process control of continuous sterilization is employed in the experiments, and the performance of the proposed framework is compared with those of the MPC based on linear model (MPC-L), the MPC based on neural network approximator (MPC-NNA), and the MPC nonlinear with successive linearisation (MPC-NSL). The continuous sterilization system is described briefly, and the modeling and the controlling on the continuous sterilization system are conducted based on plant data.

4.1. The Continuous Sterilization Process. Continuous sterilization is conducted on the medium before fermentation in biochemical engineering, and the nonsterilized medium is heated to the desirable temperature by steam in the sterilization process, so the undesired microorganisms are removed in the medium. After the continuous sterilization, necessary pure culture could be provided for the objective microorganism. Figure 2 illustrates the diagrammatic sketch of the continuous sterilization system.

In the continuous sterilization system, the quantity of steam used to heat the unsterilized medium is changed by the opening degree of the valve on the steam tube (O_v), while $\Delta O_v(i) = O_v(i) - O_v(i-1)$ is the manipulated variable. The controlled output variable is the temperature of the heated

Table 1: Parameters involved in modeling corresponding to different control methods.

Modeling method	H_r	r	N_n
Linear model	√	√	×
Linguistic model	√	√	×
Neural network	√	×	√
ANFIS-OL	√	√	×

Table 2: Parameters involved in modeling corresponding to different control methods.

Modeling method	H_r	r	N_n
Linear model	3	×	×
Linguistic model	5	0.5	×
Neural network	4	×	6
ANFIS-OL	4	0.6	×

Table 3: Performances of different modeling method on both training and test data.

Modeling method	Final MSE	
	Training data	Test data
Linguistic model	0.1524	0.4430
Linear model	0.2266	0.5026
Neural network	0.0823	0.5584
ANFIS-OL	11.5272	23.0631

medium at the outlet of the steam injector (T_h). There exist measurable disturbances as the flow rate of medium (F_m), the temperature of unsterilized medium (T_m), and the temperature of steam (T_s). The control procedure aims to provide suitable heating on the medium, so that the temperature of the heated medium can be kept around the set value.

4.2. Experiment Settings. To evaluate the effectiveness of the proposed framework in nonlinear MPC, the plant data in a real workshop of continuous sterilization are collected and experiments are conducted based on the obtained data. The plant data includes the input variable O_v, which adjusts the quantity of steam for the sterilization, while T_m, F_m, and T_s are the measurable system disturbances, and T_h is the output variable. The value ranges of O_v, T_m, F_m, T_s, and T_h are 0–100%, 0–70°C, 0–25 ton/h, 150–210°C, and 90–140°C, respectively.

Modeling and controlling are conducted based on the plant data, and MPC based on different models are employed, including the linear model, the neural network, the adaptive neurofuzzy inference system (ANFIS) [44–48] with online linearisation (ANFIS-OL), and the linguistic model. With those models, four MPC algorithms are derived: MPC-L, MPC-NNA, MPC-NSL, and the proposed optimization framework for nonlinear MPC based on linguistic model, respectively. The proposed method is expressed as MPC-LOF for short.

4.3. Experiments on Modeling. In the modeling process, 788 samples are treated as training data, and 394 samples are treated as test data. The performances of the linear model, the neural network, the ANFIS with online linearisation, and the linguistic model are compared. For there are different parameters contained in different models, the parameters are selected by scanning for the best result based on the training data. All the involved parameters in the modeling are listed in Table 1.

In Table 1, the receding horizon H_r is the sampling length of the variables employed in modeling of the prediction model. For the linguistic model in the proposed framework, the radius value r affects the result of the Gaussian kernel clustering algorithm and the number of rules obtained. For the MPC-NNA, the three-layer back-propagation neural network is employed, and N_n means the number of nodes in the hidden layers. For the MPC-NSL, the ANFIS model is adopted. For the ANFIS, the subtractive clustering algorithm constructs the initial fuzzy model, and the radius r would affect the number of clusters obtained. In the following experiments, the parameters involved in the linguistic model are discussed, and the performances of different modeling methods are compared.

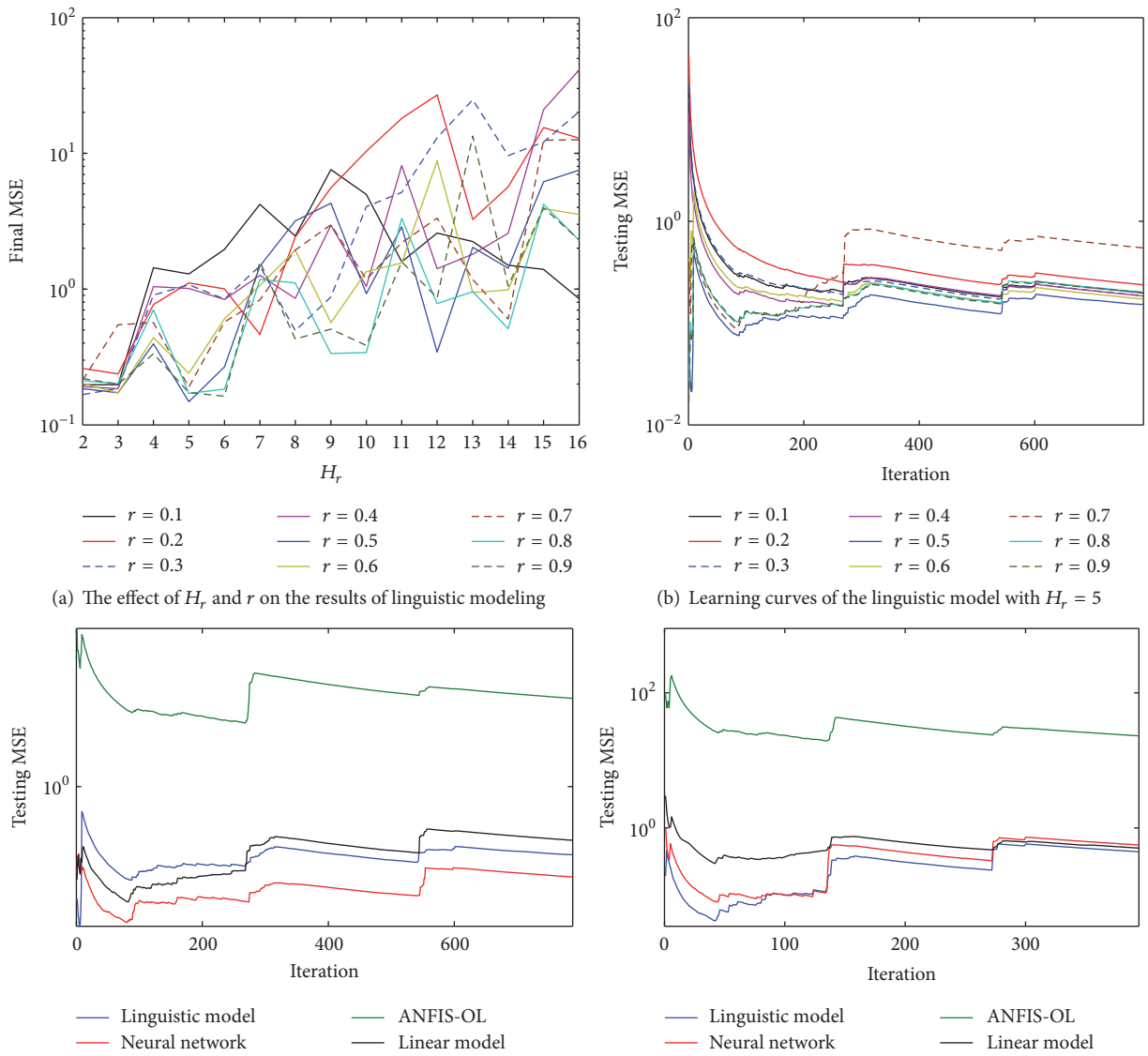
The first experiment shows the effect of H_r and r on the modeling of the linguistic model. With the training data, experiment is conducted and the modeling results under different couples of H_r and r are shown in Figure 3(a). From Figure 3(a), it can be seen that $H_r = 5$ tends to bring about lower mean square error (MSE) compared with the other values, and the best result is obtained when $H_r = 5$ and $r = 0.5$.

The second experiment is conducted with fixed $H_r = 5$ to observe in detail the effect of r on the modeling result of the linguistic model. The result is shown is in Figure 3(b), where r in the middle of its range tends to bring about higher accuracy.

To make comparison between different model methods, the parameters of the linear model, the ANFIS with online linearisation, and the neural network are selected by scanning for the best results based on the training data. The selected parameters are shown in Table 2.

The performances of different methods with the selected parameters are shown in Figures 3(c) and 3(d). From Figure 3(c), it can be seen that the neural network achieves the best performance than the linear model, the ANFIS with online linearisation (ANFIS-OL), and the linguistic model in the training process. The ANFIS-OL shows the worst accuracy, possibly because of low noise rejection. In Figure 3(d), the linguistic model provides the best modeling accuracy in the test data, while the neural network shows slightly lower accuracy than the linguistic model and the linear model, which shows lower generalization than the latter two methods. Yet, the ANFIS-OL method still shows the worst performance. Table 3 shows the detailed comparison between different modeling methods.

4.4. Experiments on Controlling. To evaluate the effectiveness of the proposed control method, the control methods of MPC-L, MPC-NNA, MPC-NSL, and MPC-LOF are conducted to make comparison based on the plant data, which



(a) The effect of H_r and r on the results of linguistic modeling

(b) Learning curves of the linguistic model with $H_r = 5$

(c) Modeling result on training data based on different modeling methods

(d) Modeling result on testing data based on different modeling methods

Figure 3: Results on the modeling.

contain 1187 samples as training data and anther 1097 samples as test data. The process model used in those MPC methods is employed as the same neural network model generated by the plant data with $H_r = 7$. In the experiments, the performance indicator is the control accuracy, which represents the approximation of the controlled variable to the expected reference trajectory. The control accuracy is also the most important indicator in a practical continuous sterilization. The indicator of the control accuracy I_c is calculated as

$$I_c = \frac{1}{N_v}\sum_{t=1}^{N_v}\left(T_h(t) - \mathrm{Re}(t)\right)^2, \tag{36}$$

where N_v means the number of data samples, while $\mathrm{Re}(t)$ means the reference trajectory at the time instant t. $\mathrm{Re}(t)$ is set to be 123°C in the control process. Different MPC methods are conducted, with the involved parameters discussed. All the involved parameters in the controlling experiments are listed in Table 4, where λ is the penalty coefficients on the control effort and H_c is the control horizon.

In the following experiments, the parameters involved in the MPC-LOF are discussed, and the performances of MPC-L, MPC-NNA, MPC-NSL, and the MPC-LOF are compared.

The first experiment investigates the effects of parameters in MPC-LOF on the controlling performance, and the best combination of parameters in MPC-LOF is selected by scanning for the best result. With the training data, the optimal parameters combination is selected as $H_r = 5$, $H_c = 5$, $r = 0.5$, and $\lambda = 3.78$. The parameter selection in MPC-LOF is shown in Figures 4–7 for illustration. The effect of H_r and r on the performance of MPC-LOF is shown for illustration in Figure 4, where the indicator of *Testing* I_c is

Table 4: Parameters involved in MPC methods.

Control method	Controller parameters				
	λ	H_c	H_r	r	N_n
MPC-L	√	√	√	×	×
MPC-NNA	√	×	√	×	√
MPC-NSL	√	√	√	√	×
MPC-LOF	√	√	√	√	×

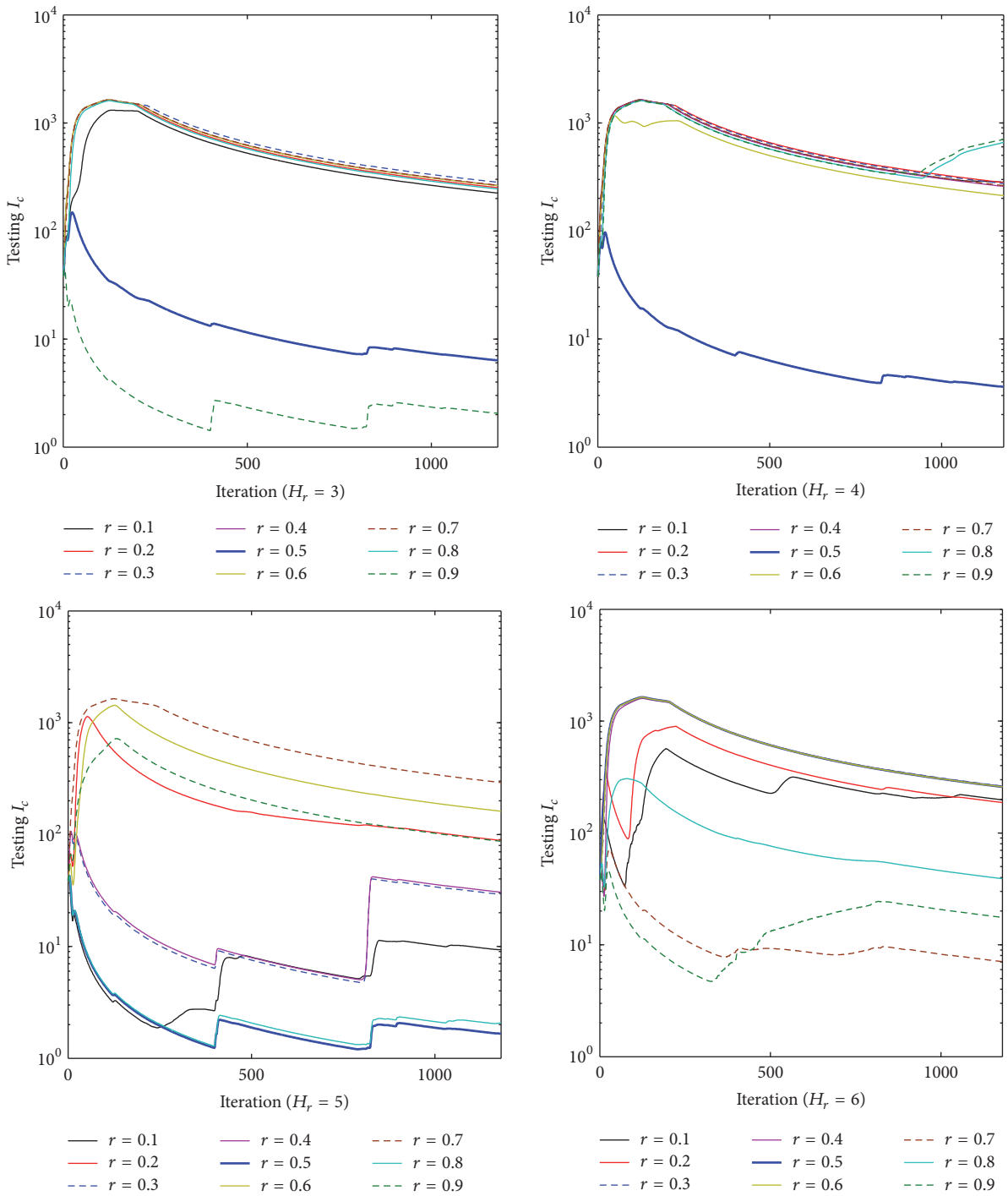


Figure 4: The effect of H_r and r on performance of MPC-LOF when $H_c = 5$ and $\lambda = 3.78$.

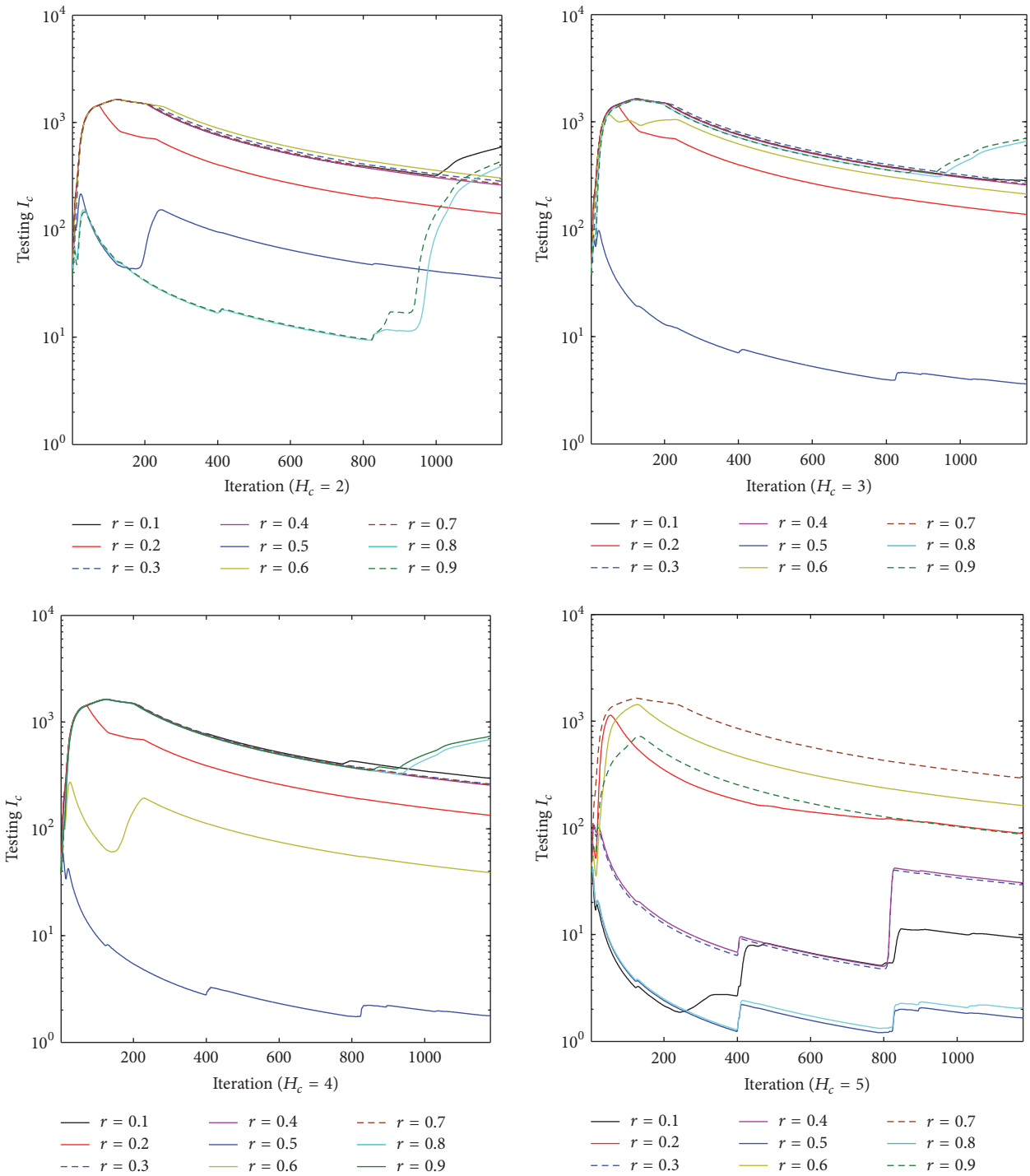


Figure 5: The effect of H_c and r on performance of MPC-LOF when $H_r = 5$ and $\lambda = 3.78$, part 1.

employed to show the control performance in detail. The ith *Testing* I_c is calculated as $\sum_{t=1}^{i} \|T_h(t) - \mathrm{Re}(t)\|^2/i$. It can be seen from Figure 4 that, when $H_r = 5$ and $\lambda = 3.78$, desirable result could be found when $H_r = 4$ or 5, and the best result is obtained when $H_r = 5$ and $r = 0.5$. The final I_c corresponding to different combinations of H_r and r with fixed $H_c = 5$ and $\lambda = 3.78$ is detailed in Table 5.

When there is H_r and $\lambda = 3.78$, the best performance is found when $H_c = 5$ and $r = 0.5$, which performs better than the other couples of parameters as shown in Figures 5-6 for illustration. The details of the final I_c with different couples of H_c and r with fixed $H_r = 5$ and $\lambda = 3.78$ are shown in Table 6.

Figure 7 shows the effect of λ and r on the MPC-LOF, and the final I_c under different couples of λ and r is shown

Table 5: The final I_c in MPC-LOF corresponding to different couples of H_r and r when $H_c = 5$ and $\lambda = 3.78$.

I_c		r								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
H_r	2	107.9560	265.6859	257.2303	252.4744	246.9592	245.1016	274.8260	254.3659	164.3792
	3	224.1424	251.8802	282.2562	263.9272	6.3428	262.2148	265.0530	244.6227	2.0565
	4	282.3905	280.4251	274.4064	258.4806	3.6191	212.4163	262.9407	662.4718	706.5165
	5	9.3565	88.5872	29.2493	30.5592	1.6593	161.6644	291.7744	2.0702	87.0250
	6	197.6085	187.1392	262.6352	259.0817	258.1882	258.5736	7.0774	39.0084	17.5822

Table 6: The final I_c in MPC-LOF corresponding to different couples of H_c and r when $H_r = 5$ and $\lambda = 3.78$.

I_c		r								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
H_c	2	593.6593	140.8252	283.2953	261.1969	35.1593	303.4396	267.9231	396.3824	439.1347
	3	283.0946	137.2669	274.4064	258.3611	3.6191	212.4163	262.9395	662.4733	706.5170
	4	297.0199	133.8319	263.1655	255.9483	1.7700	38.8709	259.8053	692.2658	738.5095
	5	9.3565	88.5872	29.2493	30.5592	1.6593	161.6644	291.7744	2.0702	87.0250
	6	338.4374	129.4590	262.1812	250.5345	31.3147	31.4507	250.3436	737.1146	766.6540
	7	184.7563	125.0401	261.9619	248.3564	32.4505	33.1497	245.2064	753.7656	771.5174
	8	187.6874	55.7913	261.8151	246.7770	32.7710	35.2300	241.9463	378.1504	733.8148
	9	203.9504	7.2457	261.5909	245.5387	32.6305	35.8943	240.0754	373.3230	561.3975
	10	186.2029	4.7204	261.3996	244.4238	32.9118	33.3927	238.0571	371.1604	570.3603
	11	162.1932	3.9902	261.0869	243.2266	33.9692	34.4692	235.5092	334.7633	587.1572
	12	142.0463	3.8393	260.5998	241.9458	34.0403	37.4656	231.6528	311.7689	548.6458
	13	120.2090	3.6480	259.9733	240.5188	35.1123	37.9268	183.0111	174.5661	556.9916
	14	80.8538	3.3041	258.9066	238.9687	35.6674	38.3702	169.9955	154.8572	541.8535
	15	56.8366	2.8585	256.4588	237.1937	35.9254	38.7456	22.6742	150.4693	510.2067
	16	61.1071	2.4136	254.7232	235.3396	36.1617	39.2147	9.7830	152.9248	354.7270

Table 7: Optimal parameters for four MPC methods.

Control method	Controller parameters				
	λ	H_c	H_r	r	N_n
MPC-L	3.16	5	3	×	×
MPC-NNA	4.26	×	3	×	6
MPC-NSL	0.50	5	4	0.2	×
MPC-LOF	3.78	5	5	0.5	×

for comparison. When $H_c = 5$ and $H_r = 5$, the best result is found when $\lambda = 3.78$ and $r = 0.5$.

The second experiment is conducted to show the performances comparison between the MPC-L, MPC-NNA, MPC-NSL, and the proposed MPC-LOF. For there are different parameters contained in different MPC methods, the parameters are selected by scanning for the best result based on the training data, whose selection results are shown in Table 7.

The plot of the manipulated variable (ΔO_v) and controlled variable (T_h) corresponding to the controller parameters in Table 7 are shown in Figure 8. Given the reference trajectory of 123°C, the controlled variable T_h can be kept within the interval [122, 124.5] under the MPC-NNA method, and there exists deviation from the reference trajectory in the controlled variable; for example, H_h is kept around 122°C from the 90th to the 140th iteration and rises to about 123.5°C in the following 200 iterations. There are no significant fluctuations in the trajectory of the controlled variable in MPC-NNA, but the controlled variable could not be held around the reference trajectory stably. As for the MPC-NSL method, the controlled variable shows better approximation to the reference compared with MPC-NNA, and H_h is kept within [121, 125] generally. Yet there exists fluctuation during the control process; for example, the controlled variable fluctuation appears from the 700th to the 860th iteration, whose corresponding manipulated variable is also in severe fluctuation as shown in Figure 8. The fluctuations show low resistance of the online linearisation approach in MPC-NSL to the disturbances. Besides, there are deviations in the control process of MPC-NSL; for example, the controlled variable rises from 120.5°C to 127°C at the 20th iteration and rises from 123°C to 130°C at the 690th iteration. In MPC-L, the controlled output variable is generally kept within [120.5, 125], and both severe deviation and fluctuation exist in the control process. For example, fluctuations appear from the 200th to the 320th iteration, and deviations exist at the

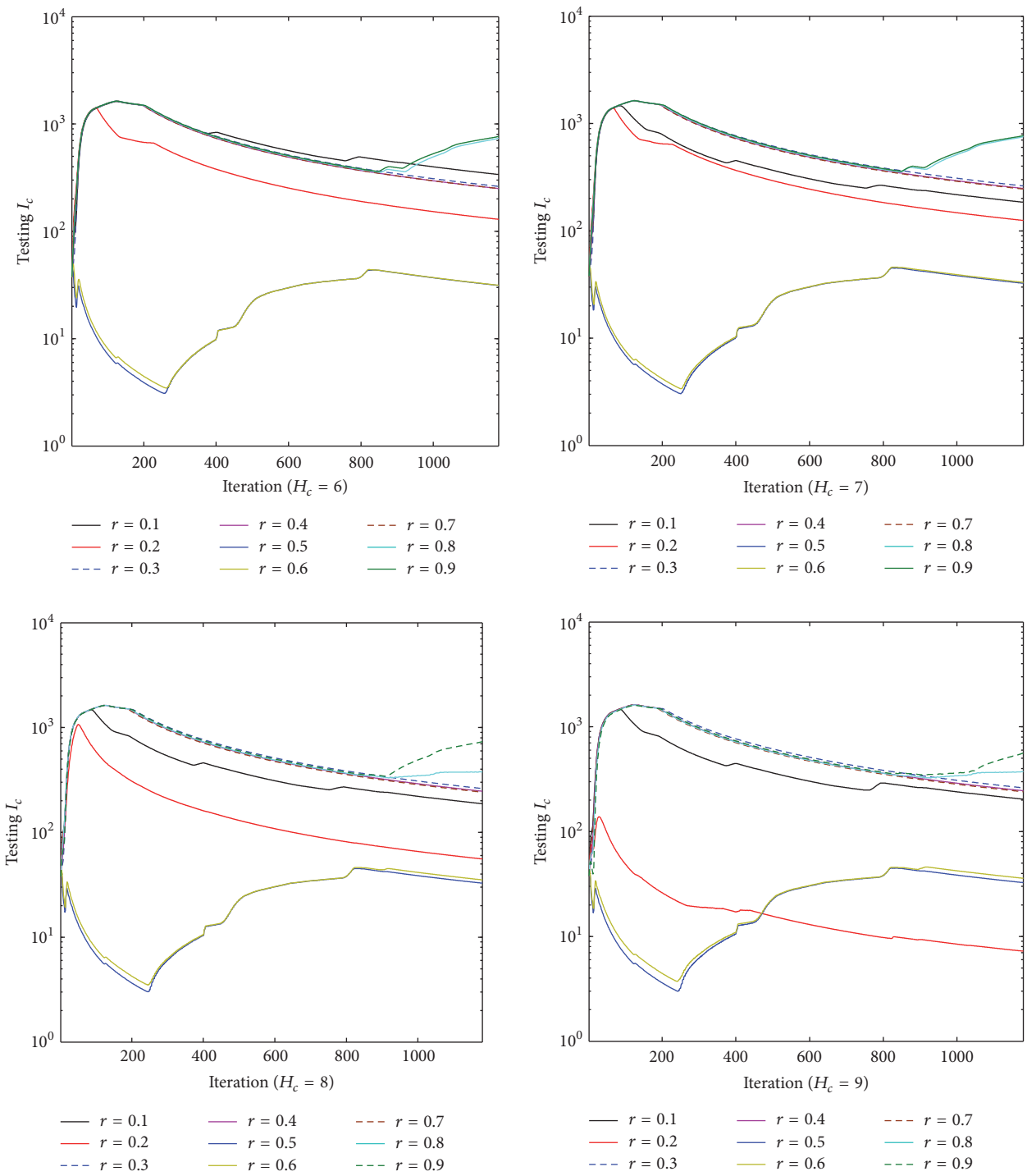


Figure 6: The effect of H_c and r on performance of MPC-LOF when $H_r = 5$ and $\lambda = 3.78$, part 2.

825th iteration with a decrease from 130°C to 117°C and the increase from 123°C to 128°C at the 690th iteration. Though the modeling accuracy of linear model is not the worst in the experiments on modeling, the bad control performance compared with the other methods shows the limit of the pure linear optimization framework in nonlinear system. As for the performance of the proposed MPC-LOF, T_h is kept within [122.5, 124]. There exist fluctuations in the controlled variable, yet they are much less than those in MPC-NSL and MPC-L, and the value of the controlled variable is kept closely around the reference compared with MPC-NNA. The value of indicator I_c of the MPC-LOF is 0.3829 and is less than that of the MPC-NNA (0.6452), MPC-NSL (1.1689), and MPC-L (2.2194), indicating that the proposed method could provide higher control accuracy than MPC-NNA, MPC-NSL, and MPC-L.

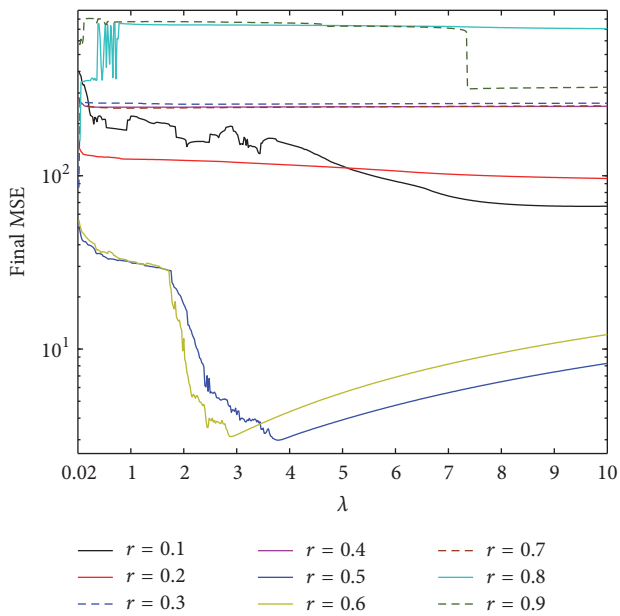


Figure 7: The effect of λ and r on performance of MPC-LOF when $H_c = 5$ and $H_r = 5$.

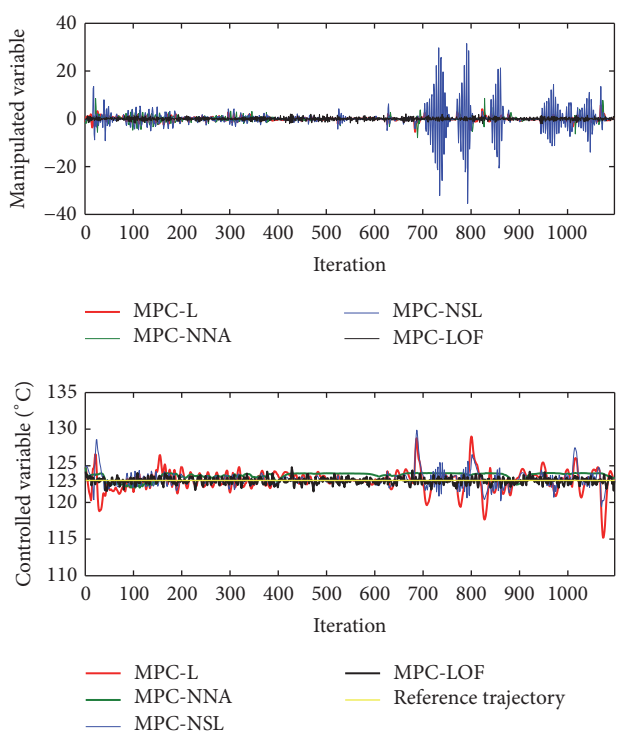


Figure 8: Manipulated variable and controlled variable under four control methods.

5. Conclusions

In this paper, the data-driven optimization framework for nonlinear MPC is proposed, which employs the linguistic model as the prediction model. The proposed framework has some advantages as follows. First, the proposed framework solves a quadratic optimization problem in online control without the calculation of piecewise linearisation or online linearisation procedure for local linear models. Second, the controller parameters are tuned based on the plant data, which is free of subjective factors, and the optimal parameters result in effective optimization framework that could bring about high control accuracy. Third, the data-driven determination of controller parameters makes the proposed method applicable in practice and easy to implement. The effectiveness is evaluated in comparison with MPC-L, MPC-NNA, and MPC-NSL in the control of continuous sterilization, and the experimental results show that the proposed framework has higher control accuracy. In the further research, we intend to apply the proposed method into other process control practices like the fermentation process in biochemical industry.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by National Natural Science Foundation of China (61375055), Program for New Century Excellent Talents in University (NCET-12-0447), Natural Science Foundation of Shanxi Province of China (2014JQ8365), State Key Laboratory of Electrical Insulation and Power Equipment (EIPE16313), and Fundamental Research Funds for the Central University.

References

[1] Y. G. Xi, D. W. Li, and S. Lin, "Model predictive control—current status and challenges," *Acta Automatica Sinica. Zidonghua Xuebao*, vol. 39, no. 3, pp. 222–236, 2013.

[2] A. Afram and F. Janabi-Sharifi, "Theory and applications of HVAC control systems—a review of model predictive control (MPC)," *Building and Environment*, vol. 72, pp. 343–355, 2014.

[3] X. Cai, P. Sun, J. Chen, and L. Xie, "ILC strategy for progress improvement of economic performance in industrial model predictive control systems," *Journal of Process Control*, vol. 24, no. 12, pp. 107–118, 2014.

[4] D. Laurí, B. Lennox, and J. Camacho, "Model predictive control for batch processes: Ensuring validity of predictions," *Journal of Process Control*, vol. 24, no. 1, pp. 239–249, 2014.

[5] K. H. Chan, E. J. Dozal-Mejorada, X. Cheng, R. Kephart, and B. E. Ydstie, "Predictive control with adaptive model maintenance: Application to power plants," *Computers and Chemical Engineering*, vol. 70, pp. 91–103, 2014.

[6] J. M. Maciejowksi, *Predictive Control: with Constraints*, Prentice-Hall, 2002.

[7] J. A. Rossiter, "Model-based predictive control," *Proceedings of the Institution of Mechanical Engineers Part A Journal of Power and Energy*, vol. 215, no. 5, pp. 623–638, 2003.

[8] P. Tatjewski, *Advanced Control of Industrial Processes. Structures and Algorithms*, Springer Ebooks, 2007.

[9] M. Ławryńczuk, "On improving accuracy of computationally efficient nonlinear predictive control based on neural models,"

Chemical Engineering Science, vol. 66, no. 21, pp. 5253–5267, 2011.

[10] M. Ławryńczuk, "Accuracy and computational efficiency of suboptimal nonlinear predictive control based on neural models," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 2202–2215, 2011.

[11] J. H. Lee and N. L. Ricker, "Extended kalman filter based nonlinear model predictive control," *Industrial & Engineering Chemistry Research*, vol. 33, no. 6, pp. 1530–1541, 1994.

[12] L. T. Biegler, "Multistep, newton-type control strategies for connstrained, nonlinear processes," *Chemical Engineering Research and Design*, no. 6, pp. 562–577, 1989.

[13] B. R. Maner, I. Doyle, B. A. Ogunnaike, and R. K. Pearson, "Nonlinear model predictive control of a simulated multivariable polymerization reactor using second-order Volterra models," *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 32, no. 9, pp. 1285–1301, 1996.

[14] E. E. Ekpo and I. M. Mujtaba, "Optimal control trajectories for a batch polymerisation reactor," *International Journal of Chemical Reactor Engineering*, vol. 5, article 80, no. 1, 2007.

[15] M. J. Willis, G. A. Montague, C. Di Massimo, M. T. Tham, and A. J. Morris, "Artificial neural networks in process estimation and control," *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 28, no. 6, pp. 1181–1187, 1992.

[16] E. E. Ekpo and I. M. Mujtaba, "Evaluation of neural networks-based controllers in batch polymerisation of methyl methacrylate," *Neurocomputing*, vol. 71, no. 7–9, pp. 1401–1412, 2008.

[17] H. Yu and Z. Zhang, "Predictive control based on neural networks of the chemical process," in *Proceedings of the 2006 Chinese Control Conference*, pp. 1143–1147, Harbin, China, August 2006.

[18] J. Ou and R. R. Rhinehart, "Grouped neural network model-predictive control," *Control Engineering Practice*, vol. 11, no. 7, pp. 723–732, 2003.

[19] M. A. Hussain, "Review of the applications of neural networks in chemical process control - simulation and online implementation," *Artificial Intelligence in Engineering*, vol. 13, no. 1, pp. 55–68, 1999.

[20] D. L. Yu and J. B. Gomm, "Implementation of neural network predictive control to a multivariable chemical reactor," *Control Engineering Practice*, vol. 11, no. 11, pp. 1315–1323, 2003.

[21] M. Mahfouf and D. A. Linkens, "Non-linear generalized predictive control (NLGPC) applied to muscle relaxant anaesthesia," *International Journal of Control*, vol. 71, no. 2, pp. 239–257, 1998.

[22] B. M. Åkesson, H. T. Toivonen, J. B. Waller, and R. H. Nyström, "Neural network approximation of a nonlinear model predictive controller applied to a pH neutralization process," *Computers and Chemical Engineering*, vol. 29, no. 2, pp. 323–335, 2005.

[23] L. Cavagnari, L. Magni, and R. Scattolini, "Neural network implementation of nonlinear receding-horizon control," *Neural Computing & Applications*, vol. 8, no. 1, pp. 86–92, 1999.

[24] M. Ławryńczuk, "Explicit nonlinear predictive control algorithms with neural approximation," *Neurocomputing*, vol. 129, pp. 570–584, 2014.

[25] S. Piche, B. Sayyar-Rodsari, D. Johnson, and M. Gerules, "Nonlinear Model Predictive Control Using Neural Networks," *IEEE Control Systems*, vol. 20, no. 3, pp. 53–62, 2000.

[26] J. Gómez Ortega and E. F. Camacho, "Mobile robot navigation in a partially structured static environment, using neural predictive control," *Control Engineering Practice*, vol. 4, no. 12, pp. 1669–1679, 1996.

[27] T. Parisini and R. Zoppoli, "A receding-horizon regulator for nonlinear systems and a neural approximation," *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 31, no. 10, pp. 1443–1451, 1995.

[28] B. M. Åkesson and H. T. Toivonen, "A neural network model predictive controller," *Journal of Process Control*, vol. 16, no. 9, pp. 937–946, 2006.

[29] M. Lawrynczuk and P. Tatjewski, "An efficient nonlinear predictive control algorithm with neural models and its application to a high-purity distillation process," in *Proceedings of the 8th international conference on Artificial Intelligence and Soft Computing*, pp. 76–85, 2006.

[30] P. Tatjewski, *Advanced Control of Industrial Processes*, Springer, London, UK, 2007.

[31] J. Zhang and J. Morris, "Nonlinear model predictive control based on multiple local linear models," in *Proceedings of the 2001 American Control Conference*, pp. 3503–3508, June 2001.

[32] J. Zhang and A. J. Morris, "Recurrent neuro-fuzzy networks for nonlinear process modeling," *IEEE Transactions on Neural Networks*, vol. 10, no. 2, pp. 313–326, 1999.

[33] R. Babuska, J. M. Sousa, and H. B. Verbruggen, "Predictive control of nonlinear systems based on fuzzy and neural models," in *Proceedings of the European Control Conference (ECC'99)*, pp. 3868–3873, September 1999.

[34] D. Megias, J. Serrano, and M. Y. El Ghoumari, "Extended linearised predictive control: Practical control algorithms for non-linear systems," in *Proceedings of the European Control Conference (ECC'99)*, pp. 2707–2712, September 1999.

[35] M. Ławryńczuk, "Modelling and nonlinear predictive control of a yeast fermentation biochemical reactor using neural networks," *Chemical Engineering Journal*, vol. 145, no. 2, pp. 290–307, 2008.

[36] M. Ławryńczuk, "A family of model predictive control algorithms with artificial neural networks," *International Journal of Applied Mathematics and Computer Science*, vol. 17, no. 2, pp. 217–232, 2007.

[37] P. Tatjewski and M. Lawrynczuk, "Soft computing in model-based predictive control," *International Journal of Applied Mathematics and Computer Science*, vol. 16, no. 1, pp. 7–26, 2006.

[38] H. Cao, Y. Wang, L. Jia, G. Si, and Y. Zhang, "Generalized tagaki-sugeno fuzzy rules based prediction model with application to power plant pulverizing system," in *Proceedings of the 52nd IEEE Conference on Decision and Control, CDC 2013*, pp. 7409–7414, ita, December 2013.

[39] Y. Wang, H. Cao, Y. Zhou, and Y. Zhang, "Characterization of the flue gas of a coal-fired power plant by ultraviolet-visible spectroscopy and a tagaki-sugeno model," *Instrumentation Science and Technology*, vol. 42, no. 5, pp. 576–585, 2014.

[40] S. Zhang, H. Cao, Y. Zhang, Y. Yuan, and X. Ma, "A nonlinear model predictive controller based on the linguistic model for biochemical continuous sterilization," in *Proceedings of the 35th Chinese Control Conference, CCC 2016*, pp. 4354–4359, chn, July 2016.

[41] X. Yang, G. Zhang, J. Lu, and J. Ma, "A kernel fuzzy c-Means clustering-based fuzzy support vector machine algorithm for classification problems with outliers or noises," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 1, pp. 105–115, 2011.

[42] S. Ramathilagam and Y.-M. Huang, "Extended Gaussian kernel version of fuzzy c-means in the problem of data analyzing," *Expert Systems with Applications*, vol. 38, no. 4, pp. 3793–3805, 2011.

[43] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control-Part I. the basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137–148, 1987.

[44] J. S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.

[45] A. F. Cabalar, A. Cevik, and C. Gokceoglu, "Some applications of Adaptive Neuro-Fuzzy Inference System (ANFIS) in geotechnical engineering," *Computers and Geotechnics*, vol. 40, pp. 14–33, 2012.

[46] P. Melin, J. Soto, O. Castillo, and J. Soria, "A new approach for time series prediction using ensembles of ANFIS models," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3494–3506, 2012.

[47] S. Palani, U. Natarajan, and M. Chellamalai, "On-line prediction of micro-turning multi-response variables by machine vision system using adaptive neuro-fuzzy inference system (ANFIS)," *Machine Vision and Applications*, vol. 24, no. 1, pp. 19–32, 2013.

[48] M. Singh and A. Chandra, "Real-time implementation of ANFIS control for renewable interfacing inverter in 3P4W distribution network," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 1, pp. 121–128, 2013.