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 Hei2

1State 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
2Department 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

Copyright © 2017 Shiliang Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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 non-convex nonlinear optimization problem. Linearisation techniques 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. A linguistic modeling method [38, 39] is reported where a series of linear models are used in MPC algorithms:

\[
\Delta u(t) = \left[ \Delta u(t+1|t) \right]_T,
\]

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

\[
\Delta u(t+k|t) = u(t+k|t) - u(t+k-1|t),
\]

\[ k \in \{1, 2, \ldots, H_c - 1\} \].

To minimize the difference between the controlled output variable \( y \) and its desired reference trajectory \( y_{\text{ref}} \). 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

\[
\Delta u(t) = \left[ \Delta u(t+1|t) \right]_T
\]

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 \( y_{\text{ref}} \). 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

\[
\Delta u(t) = \left[ \Delta u(t+1|t) \right]_T
\]

where

\[
\Delta u(t+k|t) = u(t+k|t) - u(t+k-1|t),
\]

\[ k \in \{1, 2, \ldots, H_c - 1\} \].

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

\[
C(t) = \sum_{k=1}^{N} \left( \hat{y}(t+k|t) - y_{\text{ref}}(t+k|t) \right)^2 + \sum_{k=0}^{H_c} \lambda \Delta u(t+k|t)^2,
\]

where \( \lambda \) 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.
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 \( i \)th 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 \( i \)th rule's firing strength \( \mu_i \) is calculated by

\[
\mu_i = \prod_{j=1}^{n} \left( \frac{e^{(x_j-c_{ij})^2/2\sigma_{ij}^2}}{\sigma_{ij}} \right),
\]

\[ j \in \{1, 2, \ldots, n\}, \ i \in \{1, 2, \ldots, C_n\}, \]

where \( c_{ij} \) is the \( j \)th dimension value of the \( i \)th cluster’s center and \( \sigma_{ij} \) 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

\[
l_i = \frac{y}{x_i} \cdot \frac{\sum_{i=1}^{C_n} \mu_i y_i^*}{\mu_i},
\]

where \( l_i \) is the \( j \)th input variable’s coefficient corresponding to the \( i \)th 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:

\[
\hat{y} = \sum_{i=1}^{C_n} \frac{\mu_i y_i^*}{\mu_i},
\]

where \( y_i^* \) is the predicted output variable under the \( i \)th 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

\[
P = \begin{bmatrix}
1_{11} & 1_{12} & \cdots & 1_{1n} & 1_{21} & 1_{22} & \cdots & 1_{2n} & \cdots & 1_{Cn,1} & \cdots & 1_{Cn,n}
\end{bmatrix}^T,
\]

\[
S = \begin{bmatrix}
m & 0 & 0 & \cdots & 0 \\
0 & m & 0 & \cdots & 0 \\
0 & 0 & m & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & m
\end{bmatrix}_{N \times N}
\]
where \( N = C_n \cdot n \) and \( m \) represents a positive infinite value, which is often assigned as 16. For the \( k \)th 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

\[
\alpha(k) = \left[ \mu_1(k) x_1(k), \mu_2(k) x_2(k), \ldots, \mu_t(k) \cdot x_t(k) \right] T,
\]

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

\[
P(k + 1) = P(k) + S(k),
\]

\[
S(k + 1) = S(k) - \left( \sum_{i=1}^{C_n} \mu_i(k) \right) P(k).
\]

The initial value of \( S \) is defined in (7). Let the initial \( P \) be a zero matrix, and the new values of \( \mu_i \) 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, \( c_i \) and \( \sigma_i \), in the Gaussian membership functions is shown as follows. For the training data \( \{ x_1(k), x_2(k), \ldots, x_n(k), y(k) \} \), \( k \in 1, 2, \ldots, N_t \), \( \delta \) can be calculated as

\[
\delta = \frac{-2\mu_i(k) y_i^*(k)}{\sum_{i=1}^{C_n} \mu_i(k)},
\]

where \( \bar{y}(k) \) 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 \( i \)th fuzzy rule. Assuming that \( T_r \) and \( T_c \) are both \( C_n \times n \) matrices, the \( j \)th column elements of the \( i \)th row of \( T_r \) and \( T_c \) in the \( (k + 1) \)th iterative calculation are

\[
T_r(i,j, k + 1) = T_r(i,j, k) + \sum_{l=1}^{N_t} \delta_l \cdot e^{[x_j(l) - c_j(k)]^2/2\sigma_j^2(k)},
\]

\[
T_c(i,j, k + 1) = T_c(i,j, k) + \sum_{l=1}^{N_t} \delta_l \cdot e^{[x_j(l) - c_j(k)]^2/2\sigma_j^2(k)},
\]

where the initial \( T_r \) and \( T_c \) are zero matrices. The values of \( c_{ij} \) and \( \sigma_{ij} \) in the \( (k + 1) \)th iterative calculation can be obtained by

\[
c_{ij}(k + 1) = c_{ij}(k) - \frac{ts \cdot T_r(i,j, k)}{\sum_{j=1}^{C_n} \left( T_r(i,j, k)^2 + T_c(i,j, k)^2 \right)},
\]

\[
\sigma_{ij}(k + 1) = \sigma_{ij}(k) - \frac{ts \cdot T_c(i,j, k)}{\sum_{j=1}^{C_n} \left( T_r(i,j, k)^2 + T_c(i,j, k)^2 \right)}.
\]

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:

\[
y(t) = \frac{\sum_{i=1}^{C_n} \mu_i(k) \cdot y_i^*}{\sum_{i=1}^{C_n} \mu_i(k)},
\]

where

\[
\mu_i = \prod_{p=1}^{P} e^{(p(t-\theta_p)c_y^2)^2/2u^2} \cdot \prod_{q=0}^{Q} e^{(q(t-\Delta - q) - cu_u^2)^2/2u^2},
\]

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

\[
y_i^* = \sum_{p=1}^{P} a_{yp} \cdot y(t - p) + \sum_{q=0}^{Q} b_{yu} u(t - q - 1) + \frac{c_i(t)}{\Delta},
\]

where \( cy_{yp} \) means the center of the cluster of \( y(t - p) \) under the \( i \)th fuzzy rule, \( cu_u \) means the center of the cluster of \( u(t - q) \) under the \( i \)th rule, \( \sigma_y \) means the membership function width of \( y(t - p) \) under the \( i \)th rule, and \( \sigma_u \) means the membership function width of \( u(t - q - 1) \) under...
the $i$th 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 $b_{ip}, q \in \{1, 2, \ldots, Q\}$ are determined in the linguistic modeling process. Equation (15) can be rewritten as

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

(16)

with

$$A_i(q^{-1}) = 1 - a_{i1}q^{-1} - a_{i2}q^{-2} - \cdots - a_{iP}q^{-P},$$

$$B_i(q^{-1}) = b_{i0} + b_{i1}q^{-1} + b_{i2}q^{-2} + \cdots + b_{iQ}q^{-Q}.$$  

(17)

Consider the Diophantine equation:

$$1 = E_{ij}(q^{-1}) \overline{A}_i(q^{-1}) + q^{-j}F_{ij}(q^{-1})$$

(18)

with $\overline{A}_i(q^{-1}) = \Delta A_i(q^{-1})$.

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

$$\overline{A}_i E_{ij}(q^{-1}) y_i^*(t + j) = E_{ij}(q^{-1}) B_i(q^{-1}) \Delta u(t + j - 1) + E_{ij}(q^{-1}) e_i(t + j).$$

(19)

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

$$\left(1 - q^{-j}F_{ij}(q^{-1})\right) y_i^*(t + j) = E_{ij}(q^{-1}) B_i(q^{-1}) \Delta u(t + j - 1) + E_{ij}(q^{-1}) e_i(t + j).$$

(20)

which could be rewritten as

$$y_i^*(t + j) = E_{ij}(q^{-1}) y(t)$$

$$+ E_{ij} B_i(q^{-1}) \Delta u(t + j - 1) + E_{ij}(q^{-1}) e_i(t + j).$$

(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

$$y_i^*(t + j | t) = K_{ij} \Delta u(t + j - 1) + F_{ij}(q^{-1}) y(t),$$

(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

$$K_{ij}(q^{-1}) = k_{i0} + k_{i1}q^{-1} + k_{i2}q^{-2} + \cdots + k_{ij+Q-1}q^{i+j-Q-1}.$$  

(23)

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

$$y_i^*(t + 1 | t) = K_{i1}(q^{-1}) \Delta u(t) + F_{i1}(q^{-1}) y(t),$$

$$y_i^*(t + 2 | t) = K_{i2}(q^{-1}) \Delta u(t + 1) + F_{i2}(q^{-1}) y(t),$$

$$\vdots$$

$$y_i^*(t + N | t) = K_{iN}(q^{-1}) \Delta u(t + N - 1) + F_{iN}(q^{-1}) y(t)$$

which can be rewritten as

$$\hat{y}_i^* = K_i u + F_i(q^{-1}) y(t) + K'_i \Delta u(t - 1),$$

(25)

where

$$K_i = \begin{bmatrix} k_{i0} & 0 & 0 & \cdots & 0 \\ k_{i1} & k_{i0} & 0 & \cdots & 0 \\ k_{i2} & k_{i1} & k_{i0} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k_{iN-1} & k_{iN-2} & k_{iN-3} & \cdots & k_{i0} \end{bmatrix},$$

$$K'_i(q^{-1}) = \begin{bmatrix} (K_{i1}(q^{-1}) - k_{i0}) q \\ (K_{i2}(q^{-1}) - k_{i0} - k_{i1}q^{-1}) q^2 \\ \vdots \\ (K_{iN}(q^{-1}) - k_{i0} - k_{i1}q^{-1} - \cdots - k_{iN-1}q^{N-1}) q^N \\ F_{i0}(q^{-1}) \\ F_{i1}(q^{-1}) \\ \vdots \\ F_{iN}(q^{-1}) \end{bmatrix}.$$
The last two terms in (26) depend only on the variables of the past instants, assuming that
\[
\mathbf{f}_i(q^{-1}) = \begin{bmatrix} \mathbf{f}_{i,1} \\ \mathbf{f}_{i,2} \\ \vdots \\ \mathbf{f}_{i,N} \end{bmatrix} = \mathbf{F}_i(q^{-1}) \mathbf{y}(t) + \mathbf{K}' \Delta \mathbf{u}(t-1) \tag{27}
\]
and (25) could be rewritten as
\[
\hat{\mathbf{y}}_i^* = \mathbf{K} \mathbf{u} + \mathbf{f}_i. \tag{28}
\]
Considering all \( C_n \) rules of the linguistic model, the final optimal predictions \( N \) ahead could be calculated as follows:
\[
\hat{\mathbf{y}} = \sum_{i=1}^{C_n} \mu_i \hat{\mathbf{y}}_i = \sum_{i=1}^{C_n} \mu_i \sum_{j=1}^{N} \mathbf{K}_{ij} \mathbf{u} + \mathbf{f}_i, \tag{29}
\]
which could be represented by
\[
\hat{\mathbf{y}} = \mathbf{K} \mathbf{u} + \mathbf{f}, \tag{30}
\]
where
\[
\mathbf{K} = \begin{bmatrix} k_0 & 0 & 0 & \cdots & 0 \\ k_1 & k_0 & 0 & \cdots & 0 \\ k_2 & k_1 & k_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k_{N-1} & k_{N-2} & k_{N-3} & \cdots & k_0 \end{bmatrix},
\]
\[
k_j = \frac{\sum_{i=1}^{C_n} \mu_i k_{ij}}{\sum_{i=1}^{C_n} \mu_i}, \tag{31}
\]
\[
\mathbf{f}(q^{-1}) = \begin{bmatrix} \mathbf{f}_{1} \\ \mathbf{f}_{2} \\ \vdots \\ \mathbf{f}_{N} \end{bmatrix},
\]
\[
f_j = \frac{\sum_{i=1}^{C_n} \mu_i \mathbf{f}_{ij}}{\sum_{i=1}^{C_n} \mu_i}.
\]

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
\[
C = \sum_{l=1}^{N} [\hat{\mathbf{y}}(t+l | t) - \mathbf{w}(t+l)]^2 + \sum_{l=1}^{N} \lambda [\Delta \mathbf{u}(t+l-1)]^2 \tag{32}
\]
which could be rewritten as
\[
C = (\mathbf{K} \mathbf{u} - \mathbf{f} - \mathbf{w})^T (\mathbf{K} \mathbf{u} - \mathbf{f} - \mathbf{w}) + \lambda \mathbf{u}^T \mathbf{u}, \tag{33}
\]
where
\[
\mathbf{w} = [\mathbf{w}(t+1), \mathbf{w}(t+2), \ldots, \mathbf{w}(t+N)]^T \tag{34}
\]
and \( \mathbf{w}(t+i) \), \( i \in \{1, 2, \ldots, N\} \) means the reference trajectory of the controlled output variable at the instant \( t+i \), and \( \lambda \) 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
\[
\mathbf{u} = (\mathbf{K}^T \mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{K}^T (\mathbf{w} - \mathbf{f}). \tag{35}
\]
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.


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...
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 \( O_s \), 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 \( O_s \), \( T_m \), \( F_m \), \( T_s \), 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, 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 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
contain 1187 samples as training data and another 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

$$ I_c = \frac{1}{N_s} \sum_{t=1}^{N_s} (T_h(t) - \text{Re}(t))^2, $$(36)

where $N_s$ means the number of data samples, while $\text{Re}(t)$ means the reference trajectory at the time instant $t$. $\text{Re}(t)$ 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 $\lambda$ 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

![Graph](image-url)
Table 4: Parameters involved in MPC methods.

<table>
<thead>
<tr>
<th>Control method</th>
<th>$\lambda$</th>
<th>$H_{c}$</th>
<th>$H_{r}$</th>
<th>$r$</th>
<th>$N_{n}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC-L</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>MPC-NNA</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>MPC-NSL</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>MPC-LOF</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
</tbody>
</table>

![Graphs showing the effect of $H_{c}$ and $r$ on performance of MPC-LOF when $H_{c} = 5$ and $\lambda = 3.78$.](image-url)
employed to show the control performance in detail. The $i$th Testing $I_i$ is calculated as $\sum_{i=1}^{I} \|T_h(t) - Re(t)\|^2/i$. It can be seen from Figure 4 that, when $H_r = 5$ and $\lambda = 3.78$, desirable result could be found when $H_\lambda = 4$ or 5, and the best result is obtained when $H_\lambda = 5$ and $r = 0.5$. The final $I_\lambda$ corresponding to different combinations of $H_\lambda$ and $r$ with fixed $H_r = 5$ and $\lambda = 3.78$ is detailed in Table 5.

When there is $H_r$ and $\lambda = 3.78$, the best performance is found when $H_\lambda = 5$ 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 $I_\lambda$ with different couples of $H_\lambda$ and $r$ with fixed $H_r = 5$ and $\lambda = 3.78$ are shown in Table 6.

Figure 7 shows the effect of $\lambda$ and $r$ on the MPC-LOF, and the final $I_\lambda$ under different couples of $\lambda$ 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 $\lambda = 3.78$.

<table>
<thead>
<tr>
<th>$I_c$</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>107.9560</td>
<td>265.6859</td>
<td>257.2303</td>
<td>252.4744</td>
<td>246.9592</td>
<td>245.1016</td>
<td>274.8260</td>
<td>254.3659</td>
<td>164.3792</td>
</tr>
<tr>
<td>3</td>
<td>224.1424</td>
<td>251.8802</td>
<td>282.2562</td>
<td>265.9592</td>
<td>262.1424</td>
<td>260.5345</td>
<td>297.6859</td>
<td>278.3659</td>
<td>169.3792</td>
</tr>
<tr>
<td>4</td>
<td>282.3905</td>
<td>280.4251</td>
<td>274.4064</td>
<td>258.4744</td>
<td>256.9592</td>
<td>255.1016</td>
<td>284.8260</td>
<td>264.3659</td>
<td>174.3792</td>
</tr>
<tr>
<td>5</td>
<td>9.3565</td>
<td>88.5872</td>
<td>29.2493</td>
<td>30.5592</td>
<td>1.6593</td>
<td>161.6644</td>
<td>291.7744</td>
<td>2.0702</td>
<td>87.0250</td>
</tr>
<tr>
<td>6</td>
<td>197.6085</td>
<td>187.1392</td>
<td>262.6352</td>
<td>259.0817</td>
<td>258.1882</td>
<td>258.5736</td>
<td>7.0774</td>
<td>39.0084</td>
<td>17.5822</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$I_c$</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>593.6593</td>
<td>140.8252</td>
<td>283.2953</td>
<td>261.1969</td>
<td>35.1593</td>
<td>303.4396</td>
<td>267.9231</td>
<td>396.3824</td>
<td>439.1347</td>
</tr>
<tr>
<td>3</td>
<td>283.0946</td>
<td>137.2669</td>
<td>274.4064</td>
<td>258.3611</td>
<td>3.6191</td>
<td>212.4163</td>
<td>262.9395</td>
<td>662.4733</td>
<td>706.5170</td>
</tr>
<tr>
<td>4</td>
<td>297.0199</td>
<td>133.8319</td>
<td>263.1655</td>
<td>255.9483</td>
<td>1.7700</td>
<td>38.8709</td>
<td>259.8053</td>
<td>692.2658</td>
<td>738.5095</td>
</tr>
<tr>
<td>5</td>
<td>9.3565</td>
<td>88.5872</td>
<td>29.2493</td>
<td>30.5592</td>
<td>1.6593</td>
<td>161.6644</td>
<td>291.7744</td>
<td>2.0702</td>
<td>87.0250</td>
</tr>
<tr>
<td>6</td>
<td>338.4374</td>
<td>129.4590</td>
<td>262.1812</td>
<td>250.5345</td>
<td>31.3147</td>
<td>31.4507</td>
<td>250.3436</td>
<td>737.1146</td>
<td>766.6540</td>
</tr>
<tr>
<td>7</td>
<td>184.7563</td>
<td>125.0401</td>
<td>261.9619</td>
<td>248.3564</td>
<td>32.4505</td>
<td>33.1497</td>
<td>245.2064</td>
<td>753.7656</td>
<td>771.5174</td>
</tr>
<tr>
<td>8</td>
<td>187.6874</td>
<td>55.7913</td>
<td>261.8151</td>
<td>246.7770</td>
<td>32.7710</td>
<td>35.2300</td>
<td>241.9463</td>
<td>378.1504</td>
<td>733.8148</td>
</tr>
<tr>
<td>9</td>
<td>203.9504</td>
<td>7.2457</td>
<td>261.5909</td>
<td>245.5345</td>
<td>32.6305</td>
<td>35.8943</td>
<td>240.0754</td>
<td>371.1604</td>
<td>712.1347</td>
</tr>
<tr>
<td>10</td>
<td>186.2029</td>
<td>4.7204</td>
<td>261.3996</td>
<td>243.8770</td>
<td>32.9118</td>
<td>33.3927</td>
<td>238.0571</td>
<td>371.1604</td>
<td>570.3603</td>
</tr>
<tr>
<td>11</td>
<td>162.1932</td>
<td>3.9902</td>
<td>260.8694</td>
<td>242.2666</td>
<td>33.9692</td>
<td>34.6927</td>
<td>235.5094</td>
<td>371.1604</td>
<td>587.1572</td>
</tr>
<tr>
<td>12</td>
<td>142.0463</td>
<td>3.8393</td>
<td>260.5998</td>
<td>241.9458</td>
<td>34.0403</td>
<td>34.7656</td>
<td>232.6528</td>
<td>311.7689</td>
<td>548.6458</td>
</tr>
<tr>
<td>13</td>
<td>120.2090</td>
<td>3.6480</td>
<td>259.9733</td>
<td>240.5188</td>
<td>35.1123</td>
<td>35.8943</td>
<td>229.0571</td>
<td>371.1604</td>
<td>556.9916</td>
</tr>
<tr>
<td>14</td>
<td>80.8538</td>
<td>3.3041</td>
<td>258.9066</td>
<td>238.9687</td>
<td>35.6674</td>
<td>38.3702</td>
<td>189.9955</td>
<td>518.5753</td>
<td>541.8535</td>
</tr>
<tr>
<td>15</td>
<td>56.8366</td>
<td>2.8585</td>
<td>256.4588</td>
<td>237.1937</td>
<td>35.9254</td>
<td>38.7456</td>
<td>22.6742</td>
<td>150.4693</td>
<td>510.2067</td>
</tr>
<tr>
<td>16</td>
<td>61.1071</td>
<td>2.4136</td>
<td>254.7232</td>
<td>235.3496</td>
<td>36.1617</td>
<td>39.2147</td>
<td>9.7830</td>
<td>152.9248</td>
<td>354.7270</td>
</tr>
</tbody>
</table>

Table 7: Optimal parameters for four MPC methods.

<table>
<thead>
<tr>
<th>Control method</th>
<th>$\lambda$</th>
<th>$H_c$</th>
<th>$H_r$</th>
<th>$r$</th>
<th>$N_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC-L</td>
<td>3.16</td>
<td>5</td>
<td>$\times$</td>
<td>6</td>
<td>$\times$</td>
</tr>
<tr>
<td>MPC-NNA</td>
<td>4.26</td>
<td>$\times$</td>
<td>3</td>
<td>$\times$</td>
<td>6</td>
</tr>
<tr>
<td>MPC-NSL</td>
<td>0.50</td>
<td>5</td>
<td>4</td>
<td>0.2</td>
<td>$\times$</td>
</tr>
<tr>
<td>MPC-LOF</td>
<td>3.78</td>
<td>5</td>
<td>5</td>
<td>0.5</td>
<td>$\times$</td>
</tr>
</tbody>
</table>

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 performance 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 ($\Delta O_v$) 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°C] 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 69th 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 $H_l = 5$ 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 $\lambda$ 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 (EIPEI6313), and Fundamental Research Funds for the Central University.

References


