Nonlinear Offset-Free Model Predictive Control based on Dynamic PLS Framework

Qiang Zhao¹, Xin Jin¹, Huapeng Yu¹ and Shan Lu²,*

¹ School of Information and Control Engineering, Liaoning Petrochemical University, Fushun 113001, China; lnhzhq@126.com (Q.Z.); jinxin@lnpu.edu.cn (X.J.); huay3964@gmail.com (H.Y.)
² Institute of Intelligence Science and Engineering, Shenzhen Polytechnic, Shenzhen 518055, China
* Correspondence: lushan@szpt.edu.cn

Abstract: A nonlinear offset-free model predictive control based on a dynamic partial least square (PLS) framework is proposed in this paper. A multi-output multi-input system is projected into latent variable space by a PLS outer model. For each latent variable model, the T–S fuzzy model is used to describe the nonlinear characteristics of the system; while the state-space model is used in T–S fuzzy model consequent parameters to describe the dynamic characteristics. A disturbance model is introduced in the state-space model. For model state variables, a state observer is used to compensate for the mismatch of the model. The case study results for the pH neutralization process show that the MPC controller based on this method can guarantee the tracking performance of the nonlinear system without static error.

Keywords: partial least square; model predictive control; nonlinear system; offset-free control

1. Introduction

Nonlinearity is a common characteristic in industrial processes. Many control algorithms are proposed for nonlinear systems [1]. A nonlinear model predictive control (MPC) is one of its significant branches [2]. A nonlinear MPC can be used to solve the control problem of nonlinear process. However, a mismatch between model and actual processes exists [3,4], which is intensified by unmeasurable disturbance. This leads to a decline in MPC control performance. Conventional nonlinear MPC cannot guarantee a zero-static error performance [5].

Many researchers have proposed a variety of solutions to the offset-free control problem for nonlinear systems. Jap et al. [6] presented a stochastic-tube MPC strategy that systematically handles plant–model mismatch, guarantees stability in the presence of changing operating conditions, and ensures offset-free tracking for all reachable operating conditions. Wang et al. [7] proposed an offset-free robust model predictive control for the current control of a three-phase induction motor where an incremental model is adopted to improve the parameter robustness and to realize offset-free prediction, and an improved current observer is designed to realize time-delay compensation in control variable optimization. Awha et al. [8] proposed an offset-free control scheme based on a multiple model predictive controller (MMPC) and an adaptive integral action controller for nonlinear processes; the multiple model is used to capture the essence of the nonlinear process, a min–max approach is utilized to counter the effect of parametric uncertainty between the linear models and the nonlinear process, and an adaptive integral action controller is run in parallel to the proposed controller.

The neural network and fuzzy models are two nonlinear models that are widely used in nonlinear modeling and control [9]. Benrabah et al. [10] proposed a feed-forward multi-layer neural network to approximate nonlinear mapping, and a teaching–learning-based optimization strategy is used to resolve the optimization problem in MPC. Wu et al. [11]
introduced a hybrid modeling method to integrate the first-principles model and the recurrent neural network. With the obtained model, a nonlinear MPC is designed and applied to a chemical process. L. Kong and J. Yuan [12] proposed a fuzzy model predictive controller for a boiler-turbine system, which guarantees asymptotic stability and satisfies the input constraints. Escano et al. [13] proposed a complexity reduction Takagi–Sugeno fuzzy model that is obtained by finding a function basis via a functional principal component analysis and used for MPC. It is applied in a low-cost microcontroller system. Han et al. [14] used an adaptive fuzzy neural network identifier to match the nonlinear behaviors of wastewater treatment processes with process data and proposed a multi-objective control strategy that had low computational cost.

Partial least square (PLS) proved to be rewarding in the data-driven modeling area and has been applied to many areas such as quality prediction, process monitoring, and chemometrics [15]. The principle of PLS is to maximize the covariance between the input and output data sets in a reduced space. Its main advantages include: (1) original data dimensionality reduction; (2) multivariate auto-pairing; (3) modeling of non-square system. In order to handle the dynamic modeling problem, a variety of methods combing the dynamic model with PLS have been proposed in the past decades. Yining Dong and S. Joe Qin [16] proposed a dynamic inner PLS model where the explicit dynamic inner model was given, and the inner and outer models were made consistent at the same time. A method that combined the autoregressive exogenous structure and PLS was proposed by Kaspar and Ray [17]. Qinghua Chi et al. [18] extended this method and discussed the relevant identification method in the inner PLS. Junghui Chen et al. [19] proposed another dynamic PLS framework with the ARX model. Besides these modeling methods, many attempts have been made to put forward new control strategies that compromise the merits of PLS. Junghui Chen et al. [19] proposed a novel decoupling PID strategy with PLS. Lü and Liang [20] proposed a multi-loop constrained MPC scheme. Jianhua Zhang [21] proposed a robust control based on PLS. The studies on nonlinear MPC are relatively fewer. There are two main difficulties. One is that PLS is a linear regression method. It is unable to deal with nonlinear problems. The other is that conventional PLS could not describe the dynamic characteristics of the system. In order to address the first shortcoming, some researchers proposed some nonlinear PLS methods by introducing a nonlinear modeling method [22–25]. Some researchers extended these methods to nonlinear control problems and proposed nonlinear PLS control methods. Zhao et al. [26] used neural networks with an ARX structure as internal models to describe the nonlinear characteristics of the system and designed an internal model controller. Chi et al. [27] proposed a multi-model predictive control strategy, which divided the process into several blocks according to the working area and used the area models to describe the system and design the controller.

This paper attempts to solve the nonlinear dynamic PLS modeling problem. The main work of this paper is as follows: a brief review of the T–S fuzzy model and the fuzzy model used in this paper is presented in Section 2. In Section 3, a T–S fuzzy model is used in the inner model of a dynamic PLS framework to describe the nonlinear character of the system. Based on the model in Section 3, a nonlinear MPC in a dynamic PLS framework is given in Section 4, and an offset-free control method based on it is discussed. In Section 5, a case study based on the PH process is described to demonstrate the merit of the proposed method. Finally, conclusions are drawn in Section 6.

2. T–S-Fuzzy Model-Overview

The fuzzy model is a modeling method based on the fuzzy combination rule. The fuzzy rule is expressed in the form of “IF...THEN...”, where “IF...” is called the fuzzy rule premise and “THEN...” is called the fuzzy rule consequent. Fuzzy models can be divided into three categories according to the different consequents: (1) Mamdani type (fuzzy language model); (2) Fuzzy relational type; (3) Takagi–Sugeno (T–S) type.
The T–S model expresses the global nonlinear system with a local linearized model under each fuzzy rule. Such a model form is advantageous for the design of the corresponding controller of each fuzzy rule and for the use of the principle of parallel distributed compensation (PDC) to obtain the overall controller output. In this section, the T–S model is used as the inner model of PLS to describe the dynamic nonlinear characteristic of the system and to design the corresponding controller. The inner model of PLS is a series of SISO sub-models; however, only the SISO T–S model is introduced here.

In the T–S model [28], the form of the fuzzy rule describing the system dynamics is shown in Equation (1):

\[ L_i : \text{IF } z(k) \text{ is } \Omega_i, \text{ THEN } y_i(k) = A_i y(k-1) + B_i x(k), \quad i = 1, \ldots, K. \]  

where \( L_i \) is the \( i \)-th fuzzy rule; \( z(k) \) is the premise input vector composed of input and output data. \( \Omega_i \) is the fuzzy set. Here, the ARX model is used as the consequent parameter of the fuzzy rule. \( A_i \) and \( B_i \) are polynomials, and \( K \) is the number of fuzzy rule. The fuzzy set \( \Omega_i \) is defined as:

\[ \mu_i(z(k)) : R \to [0, 1]. \]

where \( \mu_i(z(k)) \) is the membership function of the fuzzy set \( \Omega_i \), which characterizes the degree that \( z(k) \) belongs to the fuzzy rule \( L_i \).

With Equations (1) and (2), the total output of the T–S model is obtained as follows:

\[ y(k) = \sum_{i=1}^{K} \frac{\mu_i(z(k)) y_i(k)}{\sum_{i=1}^{K} \mu_i(z(k))}. \]  

The fuzzy model can be derived by using the first principle model of the nonlinear system. But this method is not universal. A more common method of establishing a fuzzy model is by identifying the input and output data. The following is an introduction to the identification of the T–S model premise (the membership function) and the T–S model consequent (the linear equation parameters).

The membership function is obtained by clustering the algorithm. Commonly used clustering algorithms include K-means clustering, fuzzy, C-means (FCM) clustering, and Gustafson–Kessel (GK) clustering.

The K-means theory was first seen in the E. W. Forgy article [29], which is the most widely used clustering algorithm because of its simplicity and efficiency. In the K-means algorithm, the distance between the sample and the cluster center is determined by Euclidean distance. The goal is to minimize the distance in the same cluster and to maximize the distance between different clusters by minimizing the objective function, as shown in Equation (4):

\[ J = \sum_{i=1}^{K} \sum_{k=1}^{l} Z_i^k ||z(k) - v_i||^2. \]

where \( l \) denotes the number of samples, \( v_i \) represents the cluster center, \( Z_i^k \) represents the membership function and satisfies \( Z_i^k = \begin{cases} 1, & z(k) \in \text{cluster } i \\ 0, & \text{else} \end{cases} \).

The disadvantage of the K-means clustering is that it is not global optimum but local optimum, and is only suitable for clustering numerical data. A direct fuzzy K-means clustering method of classifying the data is by a flexible partitioning method where the membership function is not in the form of 0 or 1, but a value between 0 and 1. Bezdek [30] introduced the fuzzy index \( m \) into the objective function Equation (4) and rewrote it as follows:

\[ J(U, V) = \sum_{i=1}^{K} \sum_{k=1}^{l} (\mu_i(k))^m ||z(k) - v_i||^2. \]
where \( \mathbf{U} = [\mu_1(k), \cdots, \mu_K(k)] \) is the fuzzy partition of \( z \), \( \mu_i(k) \) is the membership degree of \( z(k) \) corresponding to cluster center \( v_i \), which satisfies \( \mu_i(k) \in [0, 1] \) and \( \sum_{i=1}^{K} \mu_i(k) = 1 \), \( \mathbf{V} = [v_1, \cdots, v_K] \) is the cluster center vector, and \( m \) is the fuzzy index.

A fuzzy K-means clustering method based on the objective function Equation (5) is called fuzzy C-means clustering. The solution of Equation (5) based on the Lagrangian multiplier method is:

\[
\mu_i(k) = \frac{1}{\sum_{j=1}^{K} \left( \frac{d_{ik}}{d_{jk}} \right)^{2m-1}}, \text{ where } d_{ik}^2 = ||z(k) - v_i||^2.
\]

The equation of the cluster center is

\[
v_i = \frac{\sum_{k=1}^{l} \mu_i^m(k)z(k)}{\sum_{k=1}^{l} \mu_i^m(k)}
\]

The GK clustering method [31] introduces an additional symmetric positive definite parameter matrix in each cluster, which is similar to the multivariate normal distribution covariance matrix, so that it can be used to deal with elliptic clustering problems. This means that the solution to this positive parameter matrix must be included in the optimization step of Equation (5) to solve the cluster center. The objective function of GK clustering is defined as follows:

\[
J(\mathbf{U}, \mathbf{V}, \mathbf{A}) = \sum_{i=1}^{K} \sum_{k=1}^{l} (\mu_i(k))^m d_{ik}^2 A_i
\]

The Euclidean distance in K-means clustering and fuzzy C-means clustering is replaced by the Mahalanobis distance in GK clustering, which is defined as:

\[
d_{ik}^2 \mathbf{A_i} = (z(k) - v_i)^T A_i (z(k) - v_i)
\]

where \( A_i = [\rho_i \det(F_i)]^{1/p} F_i^{-1}, \rho_i > 0 \forall i, \)

\[
F_i = \frac{\sum_{k=1}^{l} \mu_i^m(k) (z(k) - v_i)^T (z(k) - v_i)}{\sum_{k=1}^{l} \mu_i^m(k)}
\]

\( \mathbf{F} = [F_1, \cdots, F_K] \) is the fuzzy covariance matrix of GK clustering.

In this paper, GK clustering is used to obtain the membership function. The steps of the solution are as follows:

Step 1: Parameter initialization. Select the number of clusters \( K \), the fuzzy index \( m \), and the convergence error \( \varepsilon > 0 \). The fuzzy partition matrix \( \mathbf{U} \) is then randomly initialized;
Step 2: Calculate the cluster center \( \mathbf{V} \) according to Equation (7);
Step 3: Calculate the covariance matrix \( \mathbf{F} \) according to Equation (10);
Step 4: Calculate the Mahalanobis distance \( d_{ik}^2 \mathbf{A_i} \) according to Equation (9);
Step 5: Update the fuzzy partition according to Equation (11);

\[
\begin{align*}
\mu_i(k) &= \frac{1}{\sum_{j=1}^{K} \left( \frac{d_{ik}}{d_{jk}} \right)^{2m-1}}, d_{ik} > 0 \\
\mu_i(k) &= 1, \quad d_{ik} = 0
\end{align*}
\]
Step 6: Repeat Step 2 up to Step 5 until the sum of the squared error of two iterations of the fuzzy partition matrices is less than the given convergence error.

The identification of the consequent parameters can be performed after \( U, V \), and \( F \) are obtained. The ARX model shown in Equation (1) is the consequent, and the least squares method can be used for identification.

For the subsequent controller design, the ARX structure consequent is rewritten as a state space based form. Rewrite Equation (1) as:

\[
L_i : \text{IF } z(k) \text{ is } \Omega_i, \text{ THEN } \left\{ \begin{array}{l}
x_i(k+1) = A_i x_i(k) + B_i u(k) \\
y_i(k) = C_i x_i(k)
\end{array} \right., \quad i = 1, \ldots, K
\]  

(12)

where \( A_i, B_i \), and \( C_i \) are state equation parameters and are satisfied:

\[
A_i = \begin{bmatrix}
-a_{i,1} & 1 & 0 & \cdots & 0 \\
-a_{i,2} & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
-a_{i,n_i-1} & 0 & 0 & \cdots & 1 \\
-a_{i,n_i} & 0 & 0 & \cdots & 0
\end{bmatrix}, \quad B_i = \begin{bmatrix}
b_{i,1} \\
\vdots \\
b_{i,n_i}
\end{bmatrix}, \quad C_i = \begin{bmatrix}
1 \\
0 \\
\vdots \\
0
\end{bmatrix}
\]

converted from the ARX model of Equation (1).

According to the fuzzy inference method [32], the fuzzy inference model and the single-point fuzzification theory generated by the central average defuzzification can be expressed as the global fuzzy model (shown in Equation (13)).

\[
\left\{ \begin{array}{l}
x(k+1) = Ax(k) + Bu(k) \\
y(k) = Cx(k)
\end{array} \right.
\]  

(13)

where \( A = \sum_{i=1}^{K} \mu_i(k) A_i, B = \sum_{i=1}^{K} \mu_i(k) B_i, C = \sum_{i=1}^{K} \mu_i(k) C_i. \)

Equation (13) replacing Equation (12) is also known as the dynamic fuzzy global state space model.

3. Nonlinear Dynamic PLS Model

3.1. Dynamic PLS Model

PLS was first proposed by Herman Wold’s original nonlinear iterative partial least square (NIPALS) algorithm [33]. It requires two data matrices, input data matrix \( X = (x_{ij})_{l \times m} \) and output data matrix \( Y = (y_{ij})_{l \times n} \), where \( l \) denotes the number of sample, while \( m \) and \( n \) denote the number of input variables and output variables, respectively. The raw data \( X \) and \( Y \) need to be scaled. In this paper, we suppose that \( X \) and \( Y \) are scaled by matrix \( W_x \) and \( W_y \), respectively. In the NIPALS method, \( X \) and \( Y \) are projected into latent variable space as a sum of a series of vectors, as shown in Equation (14).

This is called the outer model:

\[
\begin{align}
X &= \sum_{r=1}^{R} l_r p_r^T + E^* = TP^T + E^* \\
Y &= \sum_{r=1}^{R} u_r q_r^T + F^* = UQ^T + F^*
\end{align}
\]  

(14)

where \( T = [t_1, t_2, \ldots, t_R] \) and \( U = [u_1, u_2, \ldots, u_R] \) are the score matrices of \( X \) and \( Y \), respectively. \( P = [p_1, p_2, \ldots, p_R]^T \) and \( Q = [q_1, q_2, \ldots, q_R]^T \) are the loading matrices of \( X \) and \( Y \), respectively. \( E^* \) and \( F^* \) are residual matrices of \( X \) and \( Y \), respectively. \( R \) is the number of latent variables.

The relationship between \( X \) and \( Y \) is constructed as the relationship between \( t_r \) and \( u_r \) in latent variable space, which is called the inner model:

\[
u_r = b_r t_r
\]  

(15)

where \( b_r = \frac{u_r}{t_r} \), \( B = (T^T T)^{-1} T^T U \) with \( U = TB \).
Combing Equation (14) with Equation (15), the total regression model explained in $Y$ can be written as follows:

$$Y = T B Q^T + F^* = \sum_{r=1}^{R} b_r t_r q_r^T + F^*$$  \hspace{1cm} (16)

3.2. Nonlinear Dynamic Fuzzy PLS Model

The commonly used nonlinear PLS modeling methods usually have two forms. One is to modify the outer and inner models of PLS at the same time in order to fit the nonlinear characteristics. Such a method is adopted by Malthouse et al. [34], projection pursuit regression is used in the outer model to extract the principal component of input and output data, and the feedforward neural network is used in the inner model to regress the score matrix $T$ and $U$. The other is to add nonlinearity in the inner model while keeping the outer model unchanged. There has been a lot of research on this method. Qin and McAvoy [35] used the neural network in the inner model. Bang et al. [22] proposed the fuzzy PLS method, which uses the fuzzy model as the inner model of PLS. Abdel-Rahman and Lim improved the fuzzy PLS model and proposed to use the type-two fuzzy model as the inner model of PLS, which is called QFPLS. These methods, no matter what form is used to improve the original PLS method, are all for static data modeling, just as the linear PLS method. In this paper, the T-S model introduced in Section 2 is used as the inner model of PLS to build a nonlinear dynamic PLS model. In the latent variable space, each sub-model is a SISO model. Therefore, the fuzzy model Equation (13) is used as the sub-model. The input and output of the model are the corresponding $t_r$ and $u_r$ scores.

The $r$-th inner fuzzy model in latent variable space is as follows:

$$L_i : \text{IF } z(k) \in \Omega_i, \text{ THEN } \left\{ \begin{array}{l}
x_r(k+1) = A_r x_r(k) + B_r t_r(k) \\
u_r(k) = C_r x_r(k)
\end{array} \right., \quad i = 1, \ldots, K$$  \hspace{1cm} (17)

where $z(k) = [u_r(k-1), \ldots, u_r(k-n_b), t_r(k-1), \ldots, t_r(k-n_b)]$.

The global form of Equation (17) is the following:

$$\left\{ \begin{array}{l}
x_r(k+1) = \sum_{i=1}^{K} \mu_i(k) (A_{r,i} x_r(k) + B_{r,i} t_r(k)) \\
u_r(k) = \sum_{i=1}^{K} \mu_i(k) C_{r,i} x_r(k)
\end{array} \right. \quad \text{or} \quad \left\{ \begin{array}{l}
x_r(k+1) = A_r x_r(k) + B_r t_r(k) \\
u_r(k) = C_r x_r(k)
\end{array} \right.$$  \hspace{1cm} (18)

where $A_r = \sum_{i=1}^{K} \mu_i(k) A_{r,i}$, $B_r = \sum_{i=1}^{K} \mu_i(k) B_{r,i}$, $C_r = \sum_{i=1}^{K} \mu_i(k) C_{r,i}$.

The modeling process of dynamic fuzzy PLS can be summarized in the following six steps:

Step 1: The data matrix $X$ and $Y$ are normalized to $X_0$ and $Y_0$, and the normalized coefficient matrix $W_x$ and $W_y$ are obtained; $r = 1$.

Step 2: Let the first row of $Y_0$ be $u_r$; obtaining score vectors $t_r$ and $u_r$ by iterative solution using Equation (19) up to Equation (22) with the NIPALS method;

$$w_r^T = \frac{u_r^T X_{r-1}}{(u_r^T u_r)}$$  \hspace{1cm} (19)

$$t_r = \frac{X_{r-1} w_r}{\|X_{r-1} w_r\|}$$  \hspace{1cm} (20)

$$q_r = \frac{Y_{r-1}^T t_r}{\|Y_{r-1}^T t_r\|}$$  \hspace{1cm} (21)

$$u_r = Y_{r-1} q_r$$  \hspace{1cm} (22)
Step 3: \( t_r \) and \( u_r \) are used as input and output data of the T–S fuzzy model. Identify the fuzzy model premise by using Equation (8) up to Equation (10) and the GK clustering steps, and obtain the fuzzy partition \( U \), cluster center \( V \), and covariance matrix \( F \).

Step 4: Obtain the fuzzy model consequent of Equation (12) or Equation (13) by identifying consequent parameters with the least square. Then, obtain the latent variable output \( u^*_r \) with it.

Step 5: Calculate the loading vectors of \( X \) and \( Y \) with Equations (23) and (24):

\[
p = \frac{X^T_{r-1} t_r}{(t^T_r t_r)}
\]

\[
q^*_r = \frac{Y^T_{r-1} u^*_r}{\|Y^T_{r-1} u^*_r\|}
\]

Step 6: Let \( X_r = X_{r-1} - t_r p^T_r \) and \( Y_r = Y_{r-1} - u_r q^T_r \). \( X_r \) and \( Y_r \) are used as new data matrices \( X_0 \) and \( Y_0 \). Return to Step1 until the PLS model meets the accuracy requirements.

4. Offset-Free Model Predictive Control Based on a Nonlinear Dynamic PLS Framework

4.1. Dynamic PLS Control Framework

After the dynamic PLS is performed, the MIMO system is described as a series of SISO sub-models in latent variable space. Due to the orthogonality of the score vector pairs \( t_r \) and \( v_r \), these sub-models are completely decoupled. Hence, many conventional SISO control strategies can be applied to it. This method was proposed by Kaspar and Ray [17], which is called the dynamic PLS control framework. This framework is shown in Figure 1, where \( Q^* \) is the appropriate inverse of \( Q \). Setpoint \( Y_{set} \) and system output \( Y \) are scaled by scaling matrix \( W^{-1}_Y \), and projected into latent variable space by inverse loading matrix \( Q^+ \). The resulting errors are used as the inputs of the controllers. The control law \( T \) is back projected into the original space by loading matrix \( P \) and anti-scaled by matrix \( W_X \), then they are used as the input of actual system.

4.2. Offset-Free Fuzzy Model Predictive Control Based on Dynamic PLS Framework

Based on the dynamic inner model Equation (18), the model predictive controller can be designed. At time \( k \), the fuzzy model premise vector \( z(k) \) is known. The fuzzy partition \( \mu_i(k) \) at \( k \) can then be calculated. Following Equation (18), the \( r \)-th sub-model output \( u_r(k) \) at time \( k \) can be obtained. It should be noted that \( \mu_i(k) \) is a time-varying variable. It needs to be calculated at every new moment. Hence, although \( A_i, B_i \), and \( C_i \) in Equation (18) are time invariant models, \( A_r \), \( B_r \), and \( C_r \) need to be calculated at every new moment. When new models are available, the control law could be solved.

Figure 1. Dynamic PLS control framework proposed by Kaspar and Ray.
Similar to linear systems, the feedback of control is the model state $x_r$. When the state space model in the dynamic PLS framework matches the plant model accurately, $x_r$ can explain the system output $Y$ well. Due to the decoupling scheme of PLS, the mismatch error $F^*(k)$ is unavoidably created. This leads to poor control performance or a steady-state error. In order to solve this problem, the output feedback is introduced into the controller through the observer. Define the error between the actual system output $y(k)$ in latent variable space and the inner model output $u_r(k|k - 1)$ with the following formula:

$$d_r(k) = u_y(k) - u_r(k)$$  

(25)

Put the error into Equation (18) as:

$$\begin{cases} x_r(k + 1) = Ax_r(k) + Bu_r(k) + E_r d_r(k) \\ u_r(k) = C_x x_r(k) + d_r(k) \end{cases}$$  

(26)

where $E_r = \sum_{i=1}^{K} \mu_i(k)E_{r,i}E_{r,i} = \begin{bmatrix} -a_{i,1} & -a_{i,2} & \cdots & -a_{i,n_a} \end{bmatrix}^T$.

Establish the fuzzy observer shown in Equation (27).

$$\begin{cases} \dot{x}_r(k+1) = A_r \dot{x}_r(k) + B_r u_r(k) + E_r \tilde{d}_r(k) + F_{r,i}(u_r(k) - \dot{a}_r(k)) \\ \dot{a}_r(k) = C_r \dot{x}_r(k) + d_r(k) \end{cases}, \quad i = 1, \ldots, K$$  

(27)

where $F_{r,i}$ is the observer gain matrix, $\dot{x}_r(k)$ and $\dot{y}_r(k)$ are the estimation of state and output, respectively. Equation (27) can be augmented and rewritten in a global fuzzy model form as follows:

$$\begin{cases} \dot{x}_r(k+1) = \left( \begin{array}{c} A_r \ E_r \\ 0 \ 1 \end{array} \right) \dot{x}_r(k) + \left( \begin{array}{c} B_r \\ 0 \end{array} \right) u_r(k) + \left( \begin{array}{c} F_r \\ 0 \end{array} \right) \left( u_r(k) - \dot{a}_r(k) \right) \\ \dot{a}_r(k) = C_r \dot{x}_r(k) + d_r(k) \end{cases}$$  

(28)

Bring the output equation in Equations (26) and (28) to the state equation in Equation (28) as:

$$\begin{cases} \dot{x}_r(k+1) = \sum_{i=1}^{K} \sum_{j=1}^{K} \mu_i(k) \mu_j(k) \left( \begin{array}{c} A_{r,j} \\ 0 \end{array} \right) \dot{x}_r(k) + \left( \begin{array}{c} B_{r,j} \\ 0 \end{array} \right) u_r(k) + \left( \begin{array}{c} F_{r,j} \\ 0 \end{array} \right) \left( u_r(k) - \dot{a}_r(k) \right) \\ + \left( \begin{array}{c} F_{r,j} C_r \\ 0 \end{array} \right) \left( x_r(k) - \dot{x}_r(k) + d_r(k) - \dot{d}_r(k) \right) \end{cases}$$  

(29)

Equation (26) is subtracted from Equation (29). The error state of observer is as follows:

$$\begin{cases} \dot{x}_r(k+1) = \sum_{i=1}^{K} \sum_{j=1}^{K} \mu_i(k) \mu_j(k) \left( \begin{array}{c} A_{r,j} \\ 0 \end{array} \right) \dot{x}_r(k) + \left( \begin{array}{c} F_{r,j} C_r \\ 0 \end{array} \right) \left( x_r(k) - \dot{x}_r(k) + d_r(k) - \dot{d}_r(k) \right) \end{cases}$$  

(30)

where $\dot{x}_r(k) = x_r(k) - \dot{x}_r(k), \dot{d}_r(k) = d_r(k) - \dot{d}_r(k)$.

Thus, in the PLS framework, $r$-th MPC is based on the minimization of a quadratic cost function as:

$$J_r = \sum_{j=1}^{N_{tr}} \delta_{r,j} \| u_{set,r}(k+j) - \dot{a}_r(k+j) \|^2_2 + \sum_{j=1}^{N_{tr}} \lambda_r(j) \| \Delta l_r(k+j-1|k) \|^2_2$$  

(31)

where $N_{tr}$ and $N_{cr}$ denote prediction horizon and control horizon, respectively.$u_{set,r}$ is the setpoint in the latent space, which is transformed from the setpoint in original space.

### 4.3. Getting a Proper Observer Gain $F_{r,i}$

Wang et al. [36] pointed out that the sub-model (Equation (27)) stability of each fuzzy rule in the fuzzy model stable cannot guarantee a global model (Equation (28)) stability. Therefore, the stability of the observer should be discussed before solving the observer gain $F_{r,i}$. The stability of the T–S
fuzzy model is mainly analyzed by the Lyapunov stability theory [37]. The global Lyapunov function is used for stability analysis in this paper.

When \( r_t(k) = 0 \), one can determine the Lyapunov function for the discrete fuzzy system of Equation (26) discussed in this paper as follows:

\[
V(x) = x^T P^{-1} x
\]  
(32)

where \( P \) is the positive definite symmetric matrix, \( x \) is the state variable. Similar conclusions to Theorem 1 can be obtained from Ref. [38].

**Theorem 1.** If a symmetric positive definite matrix \( P \) exists, it satisfies the linear matrix inequality (LMI) Equation (33). The discrete fuzzy system described in Equation (26) is globally exponentially stable.

\[
A_i^T P^{-1} A_i - P^{-1} < 0, \quad i = 1, \ldots, K
\]
(33)

Using the Schur complements [39], Equation (33) is equivalent to the following formula:

\[
\begin{bmatrix}
-P & A_i P \\
P A_i^T & -P
\end{bmatrix} < 0, \quad i = 1, \ldots, K
\]
(34)

For the fuzzy model of Equation (30), one can obtain the conclusion of Theorem 2.

**Theorem 2.** If there are positive definite symmetric matrices \( P \) and a set of matrices \( Q_j \) \( (i = 1, \ldots, K) \) satisfying LMI such as Equations (35) and (36), then the observer error state equation of Equation (30) is globally exponentially stable. Moreover, observer gain is \( \hat{F}_{ij} = P^{-1} Q_j, \quad i = 1, \ldots, K \).

\[
\begin{bmatrix}
-A_{i,j} P - C_{i,j} Q_j^T & -P
\end{bmatrix} < 0
\]
(35)

where \( i < j \leq K \)

**Proof of Theorem 2.** Equation (30) can be rewritten as follows:

\[
\begin{aligned}
\hat{x}_r(k + 1) &= \sum_{i=1}^{K} h_i(k) \mu_i(k) \left( \hat{A}_{r,i} - \hat{F}_{r,i} \hat{C}_{r,i} \right) G_{ij} \hat{x}_r(k) \\
&\quad + 2 \sum_{i=1}^{K} \sum_{j>i} h_i(k) \mu_i(k) \left( \frac{\hat{A}_{r,i} - \hat{F}_{r,i} \hat{C}_{r,i}}{2} \right) \hat{x}_r(k)
\end{aligned}
\]
(37)

where

\[
\begin{bmatrix}
\hat{A}_{r,i} & 0 \\
\hat{F}_{r,i} & \hat{C}_{r,i}
\end{bmatrix} = \begin{bmatrix}
A_{r,i} & 0 \\
F_{r,i} & C_{r,i}
\end{bmatrix}, \quad \begin{bmatrix}
\hat{A}_{r,j} & \hat{F}_{r,j} \\
\hat{C}_{r,j}
\end{bmatrix} = \begin{bmatrix}
A_{r,j} & E_{r,j} \\
0 & 1
\end{bmatrix}, \quad \begin{bmatrix}
\hat{A}_{r,j} & \hat{F}_{r,j} \\
\hat{C}_{r,j}
\end{bmatrix} = \begin{bmatrix}
E_{r,j} & 0 \\
F_{r,j} & 0
\end{bmatrix}
\]

According to Theorem 1, one can get the conclusion that the discrete fuzzy system described in Equation (37) is exponentially stable if there exists a symmetric positive definite matrix \( P \) satisfying the LMI Equations (38) and (39).

\[
\left( \hat{A}_{r,i} - \hat{F}_{r,i} \hat{C}_{r,i} \right)^T P^{-1} \left( \hat{A}_{r,i} - \hat{F}_{r,i} \hat{C}_{r,i} \right) - P^{-1} < 0
\]
(38)

\[
\left( \frac{\hat{A}_{r,i} - \hat{F}_{r,i} \hat{C}_{r,i}}{2} + \hat{A}_{r,j} - \hat{F}_{r,j} \hat{C}_{r,j} \right)^T P^{-1} \left( \frac{\hat{A}_{r,i} - \hat{F}_{r,i} \hat{C}_{r,i}}{2} + \hat{A}_{r,j} - \hat{F}_{r,j} \hat{C}_{r,j} \right) - P^{-1} < 0
\]
(39)

where \( i < j \leq K \).
The parameters of the process.

Table 1.

| Parameter            | Value                   |
|----------------------|-------------------------|
| A                    | 207 cm²                 |
| C₀                   | 8.75 mL·cm⁻¹·s⁻¹        |
| pK₁                  | 6.35                    |
| pK₂                  | 10.25                   |
| Wₐ₁                  | 3 x 10⁻³ M              |
| Wₐ₂                  | -3 x 10⁻² M             |
| Wₐ₃                  | -3.05 x 10⁻³ M          |
| W₉₃                  | -5.05 x 10⁻⁵ M          |
| q₁                   | 16.6 mL·s⁻¹             |
| q₂                   | 0.55 mL·s⁻¹             |
| q₃                   | 15.6 mL·s⁻¹             |
| [acid]               | 0.003 M HNO₃            |
| [buffer]             | 0.03 M NaHCO₃           |
| [base]               | 0.003 M NaOH            |

Equation (38) can be rewritten as follows:

\[
\begin{bmatrix} A_j - \hat{F}jC_j \\ P \end{bmatrix}^T P^{-3} P \begin{bmatrix} A_j - \hat{F}jC_j \\ P \end{bmatrix} - P^{-1} < 0 \quad (40)
\]

\[
\begin{bmatrix} PA_j - \hat{F}jC_j \\ P \end{bmatrix}^T P^{-3} \left( PA_j - \hat{F}jC_j \right) - P^{-1} < 0 \quad (41)
\]

Use the Schur complements [39] to satisfy the following LMI establishment as:

\[
\begin{bmatrix} -P & \hat{F}jC_j \\ A_jP - C_jQ_j & -P \end{bmatrix} \begin{bmatrix} \hat{A}_j - Q_jC_j \\ \hat{P} \end{bmatrix} < 0 \quad (42)
\]

Let \( Q_i = F_iC_i \), then Equation (42) is rewritten as:

\[
\begin{bmatrix} -P & \hat{F}jC_j \\ A_jP - C_jQ_j & -P \end{bmatrix} \begin{bmatrix} \hat{A}_j - Q_jC_j \\ \hat{P} \end{bmatrix} < 0, \quad i = 1, \ldots, K. \quad (43)
\]

Empathy Equation (39) is rewritten as follows:

\[
\begin{bmatrix} -P & \hat{A}_j - Q_jC_j \\ A_jP - C_jQ_j & -P \end{bmatrix} \begin{bmatrix} \hat{A}_j - Q_jC_j \\ \hat{P} \end{bmatrix} < 0, \quad i < j \leq K \quad (44)
\]

By solving Equations (43) and (44), one can get the feasible \( P \) and \( Q_i \). Additionally, observer gain is \( F_i = P^{-1}Q_i, \quad i = 1, \ldots, K \). □

Based on Theorem 2, Equation (30) of global stability is obtained; it can then further ensure the stability of the proposed method.

5. Case Study

A typical nonlinear process, the PH neutralization titration process, is used as a case study to verify the effectiveness of the proposed method. The schematic illustration is shown in Figure 2. The process has three inputs: \( q_1 \) (HNO₃), \( q_2 \) (NaHCO₃), and \( q_3 \) (NaOH). The three enter the reactor and are fully mixed and neutralized after mixing. In this case study, \( q_1 \) and \( q_2 \) are selected as the manipulation variables, and \( q_3 \) is a constant value. The outputs are the PH value and the level of reactor \( h \) (cm). The model of the process consists of three nonlinear ordinary differential equations and a nonlinear equation, which is shown in Equation (45) [40]. The parameters of the model are shown in Table 1.

\[
\frac{dh}{dt} = \frac{1}{4}(q_1 + q_2 + q_3 - C_vh^{0.5})
\]

\[
\begin{align*}
\frac{dW_{a1}}{dt} &= \frac{1}{2}[W_{a1} - W_{a4}]q_1 + (W_{a2} - W_{a4})q_2 + (W_{a3} - W_{a4})q_3 \\
\frac{dW_{a2}}{dt} &= \frac{1}{2}[W_{a3} - W_{a4}]q_1 + (W_{a2} - W_{a4})q_2 + (W_{a3} - W_{a4})q_3 \\
\frac{dW_{a4}}{dt} &= W_{a4} + 10^{14-pH} + W_{b4} \frac{1+2\times10^{-pH}}{1+10^{10^{-pH}-pH}} - 10^{-pH} = 0 \\
W_{a4} + 10^{14-pH} + W_{b4} \frac{1+2\times10^{-pH}}{1+10^{10^{-pH}-pH}} - 10^{-pH} = 0
\end{align*}
\]
In order to build a dynamic PLS model, a set of pseudo random signals is generated to excite the system and to generate identification data. This set of signals should include the range of manipulating variables as far as possible. The data set (1500 samples) for identification is shown in Figure 3. The sampling-time interval for this case study is 15 s.

In this case study, the number of latent variables in the proposed model $R$ is 2. The fuzzy rule of each sub-latent variable model is 2. The model order of consequent is $n_a = 1$ and $n_b = 1$, and the fuzzy exponent is 2. The other data set (1500 samples) generated is used for cross validation; the validation results are shown in Figure 4. Here, a new norm shown in Equation (46) is used to measure the accuracy of the model.

$$
\eta = \frac{\sum_{n=1}^{\text{sample number}} |\text{Prediction output}_n - \text{Actual output}_n|}{\text{Actual output}_n} \quad (46)
$$

The $\eta$ for the cross-validation results of the proposed method shown in Figure 4 are 68.99 and 124.70 for $h$ and PH, respectively. From the results, one can conclude that the model of level $h$ is more accurate. There is an error in the PH model, which is because the nonlinearity of PH is stronger.

A control simulation based on the proposed method with the model obtained (Equation (28)) and the conventional MPC with model Equation (18) is made; the results are shown in Figure 5. OMPC denotes the proposed method. The control horizon is 2 and the predictive horizon is 9 for both methods. One can conclude that although the conventional MPC method guarantees stability, there is static error. The proposed method can track the setpoints without static error. Shown in Figure 6 is the control results of the conventional method in latent variable space. The solid line represents the output of the dynamic PLS model in latent variable space. The dashed line represents the projection of the actual system output in latent variable space. Compared with the results of conventional MPC in original space (Figure 5), one can conclude that conventional MPC can make the sub-model in latent variable space track the projection of setpoints in this space. However, due to the model mismatch, the control law projected back into original space cannot make the system output track the setpoints without static error.

Figure 2. A schematic illustration of the process.
Figure 3. The data set for identification.

Figure 4. Cross-validation results of the proposed method.
To verify the control effect when the parameters of the actual system change or the model parameters are uncertain, the method proposed in this paper is compared with the predictive function control based on the adaptive dynamic fuzzy PLS model introduced in reference [41] (denoted as PFC). In the following simulation, the model parameter $C_v$ is modified to simulate the actual system change. A 500-step simulation is performed. At the 100th step, $C_v$ is changed from 8.79 to 9.79. The results are shown in Figure 7.
In the method used in reference [41], the parameters of the PLS model are identified and modified adaptively. This method will increase the computational burden and cause a large fluctuation of the system because it involves updating the internal and external models of the dynamic PLS model at the sample time of the parameter change and some subsequent sampling moments. The method proposed in this paper can still make the output of the system track the setpoints through its feedback when the model parameters of the system change and the fluctuation is small. For the nonlinear PH variable, there are static errors in both methods. The static error of the proposed method is 0.027, less than that of the method in reference [41], which is 0.055. This is because the proposed model is not reidentified when the model parameters changed.

6. Conclusions

In order to solve the problem of offset-free control for nonlinear systems, a nonlinear offset-free MPC based on a dynamic fuzzy PLS model is proposed. The T–S fuzzy model is used as the inner model of PLS to describe the dynamic nonlinear characteristics of the MIMO system. Moreover, the disturbance model and observer are used to compensate the model mismatch during the MPC controller design. The simulation results of the PH neutralization titration show that the proposed method can achieve the offset-free tracking to the setpoint. It also achieves satisfactory control performance when external disturbances and model parameters are changed in the system.

Author Contributions: Conceptualization, X.J. and S.L.; methodology, Q.Z.; software, H.Y.; validation, Q.Z., X.J. and S.L.; formal analysis, Q.Z. and X.J.; investigation, H.Y.; writing—original draft preparation, Q.Z.; writing—review and editing, X.J. and H.Y.; visualization, H.Y.; supervision, S.L.; project administration, X.J. and S.L.; funding acquisition, Q.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant number 62073158.

Conflicts of Interest: The authors declare no conflict of interest.
34. Malthouse, E.C.; Tamhane, A.C.; Mah, R. Nonlinear partial least squares. *Comput. Chem. Eng.* 1997, 21, 875–890. [CrossRef]
35. Qin, S.J.; Mcavoy, T.J. Nonlinear PLS modeling using neural networks. *Comput. Chem. Eng.* 1992, 16, 379–391. [CrossRef]
36. Wang, H.O.; Tanaka, K.; Griffin, M.F. An approach to fuzzy control of nonlinear systems: Stability and design issues. *IEEE Trans. Fuzzy Syst.* 2002, 4, 14–23. [CrossRef]
37. Zhang, T.; Gang, F.; Zeng, X.J. Output tracking of constrained nonlinear processes with offset-free input-to-state stable fuzzy predictive control. *Automatica* 2009, 45, 900–909. [CrossRef]
38. Tanaka, K.; Ikeda, T.; Wang, H.O. Fuzzy regulators and fuzzy observers: Relaxed stability conditions and LMI-based designs. *IEEE Trans. Fuzzy Syst.* 2002, 6, 250–265. [CrossRef]
39. Su, H. *Basic Theory of Robust Control*; Science Press: Beijing, China, 2010.
40. Hall, R.C.; Seborg, D.E. Modelling and Self-Tuning Control of a Multivariable pH Neutralization Process Part I: Modelling and Multiloop Control. In Proceedings of the American Control Conference, Pittsburgh, PA, USA, 21–23 June 1989.
41. Lü, Y. *Modeling and Predictive Controller Design Based on the Dynamic PLS Method*; Zhejiang University: Hangzhou, China, 2013.