frontiers in Control Engineering

ORIGINAL RESEARCH
published: 30 November 2021
doi: 10.3389/fcteg.2021.788492

Broad Learning Aided Model Predictive Control With Application to Continuous Stirred Tank Heater

Mo Tao[1,2]*, Tianyi Gao*[3]*, Xianling Li*[1]** and Kuan Li*[3]

[1]Science and Technology on Thermal Energy and Power Laboratory, Wuhan, China, [2]School of Automation Science and Electrical Engineering, Beihang University, Beijing, China, [3]Department of Control Science and Engineering, Harbin Institute of Technology, Harbin, China

This paper presents a data-driven predictive controller based on the broad learning algorithm without any prior knowledge of the system model. The predictive controller is realized by regressing the predictive model using online process data and the incremental broad learning algorithm. The proposed model predictive control (MPC) approach requires less online computational load compared to other neural network based MPC approaches. More importantly, the precision of the predictive model is enhanced with reduced computational load by operating an appropriate approximation of the predictive model. The approximation is proved to have no influence on the convergence of the predictive control algorithm. Compared with the partial form dynamic linearization aided model free control (PFDL-MFC), the control performance of the proposed predictive controller is illustrated through the continuous stirred tank heater (CSTH) benchmark.

Keywords: data driven, broad learning, tracking control, model predictive control, continuous stirred tank heater

OPEN ACCESS

Edited by:
Zehui Mao,
Nanjing University of Aeronautics and Astronautics, China

Reviewed by:
Guang Wang,
North China Electric Power University, China
Yunsong Xu,
National University of Defense Technology, China

****Correspondence:***
Xianling Li
lixl_719@163.com

Specialty section:
This article was submitted to Adaptive, Robust and Fault Tolerant Control,
a section of the journal
Frontiers in Control Engineering

Received: *02 October 2021*
Accepted: *20 October 2021*
Published: *30 November 2021*

Citation:
Tao M, Gao T, Li X and Li K (2021) Broad Learning Aided Model Predictive Control With Application to Continuous Stirred Tank Heater.
Front. Control. Eng. 2:788492.
doi: 10.3389/fcteg.2021.788492

1 INTRODUCTION

Tracking control plays an important role in the modern industrial process and has drawn much attention both in academia and industry over these years (Yin et al., 2017). As an efficient feedforward control strategy, model predictive control (MPC) has shown its efficiency in the tracking control of multivariable systems with various constraints. Compared with other control strategies where the control law is calculated offline, the core idea of MPC is to calculate an optimal control sequence by solving a finite optimal control problem online (Mayne and Rawlings, 2001). MPC is proved to be easy for implementation and maintenance in practical industry (Hidalgo and Brosilow, 1990; Yin and Xiao, 2017). Due to these advantages, MPC has been successfully used in the tracking control of the hybrid energy storage system, modular multi-level converters, wind generation system and many other applications in recent literatures (Qin and Badgwell, 2003; Garcia-Torres et al., 2019; Guo et al., 2019; Kou et al., 2019).

Despite the successful applications of MPC, most predictive controllers rely heavily on the precision of the predictive model. For model-based MPC, the predictive model is usually generated using the first principle, *i.e.*, the controlled auto-regressive intergraded moving average (CARIMA) model (Zhang et al., 2018). However, it is arduous to set up a precise model for modern industrial process based on the first principle due to the complicated mechanism. Some alternative predictive models, such as the time-domain model, are also hard to obtain in practice (Patwardhan et al., 1998). Fortunately, there exists a huge number of underutilized process data with the development of the measurement and storage technology. Motivated by this, the data-driven predictive controller design has attracted the interest of researchers in recent years.

To propose the data-driven predictive control strategy, multivariable statistical analysis (MSA) techniques are used to identify the predictive model based on the available process data. Partial least square (PLS) and its modified forms have been paid more attention to due to the low computation complexity (Gao et al., 2018). To extract the dynamic characteristics of the process, a dynamic PLS based predictive control is proposed based on the assumed autoregressive exogenous (ARX) model (Kaspar and Ray, 1993). Subspace identification techniques are also used to design the predictive controller, where the state space model can be directly identified based on the process data (Yi, 2016; Garg and Mhaskar, 2017; Meidanshahi et al., 2017; Zhou et al., 2017). However, most investigations on subspace identification mainly focus on the identification of linear systems (Wu et al., 2013; Li et al., 2020). To cope with nonlinear systems, the neural network (NN) is often applied for the predictive controller design due to its strong learning ability in nonlinear model regression.

Over the past several years, fruitful achievements have been reported regarding the combination of NN and MPC. A dynamic neural network based MPC (DNN-MPC) is proposed and demonstrated to show superior control performance than the mathematical model based MPC for the industrial baker's yeast drying process (Yuzgec et al., 2008). A self-organizing recurrent neural network (RNN) aided nonlinear MPC is proposed (Han et al., 2016), where the control performance is evaluated through a wastewater treatment process. To improve the control performance for nonlinear process, the radial basis function neural network (RBF-NN) is applied to design the model predictive controller (Yu et al., 2011). Particularly, an adaptive splitting and merging RBF-NN based MPC has recently been successfully implemented in a zinc hydrometallurgy plant. However, these NN based MPC strategies often require much computation load, which limits its extension to the online implementation for real-time control performance.

In this paper, a random vector functional-link neural network (RVFLNN), *i.e.*, the broad learning system (Chen and Liu, 2018), is incorporated into the predictive controller design. By using the incremental broad learning algorithm, the predictive model can be identified adaptively based on the online process data. In addition, a Jacobian matrix is used to approximate the predictive model to enhance the precision of the predictive model. The main contributions can be summarized as follows:

1) A data-driven predictive control strategy is proposed based on the broad learning algorithm with no prior knowledge of the system model;
2) The proposed data-driven predictive controller can be updated online based on the incremental broad learning algorithm while requiring little computation load;
3) The proposed broad learning aided MPC strategy is verified on the continuous stirred tank heater (CSTH) benchmark for temperature adjustment.

The rest of this paper is organized as follows. **Section 2** introduces the basic knowledge of the model predictive control and the fundamental structure of the broad learning system. **Section 3** discusses the incremental broad learning algorithm for new training data and proposes the incremental broad learning based MPC. **Section 4** verifies the control performance of the proposed broad learning based MPC through the CSTH benchmark. **Section 5** concludes this paper.

2 PRELIMINARIES

2.1 Basic Knowledge of Model Predictive Control

Given a nonlinear process described as follows:

$$y(t) = f\left(y(t-1), \ldots, y\left(t-n_y\right), u(t), \ldots, \right. \tag{1}$$

$$\left. u(t-n_u)\right) + d(t) \tag{2}$$

where $y(t) \in \mathcal{R}^m$ is the output of the system, $u(t) \in \mathcal{R}^l$ is the output of the controller, n_u and n_y represent the order of the system.

The core idea of MPC is to optimize the following quadratic problem online

$$\begin{aligned} min\, J = & \sum_{i=1}^{n_p} \left(y_{sp}(k+i) - y(k+i)\right)^{\mathrm{T}} \lambda_1 \left(y_{sp}(k+i) - \right. \\ & \left. y(k+i)\right) + \sum_{i=1}^{n_c} \Delta u(k+i)^{\mathrm{T}} \lambda_2 \Delta u(k+i), \end{aligned} \tag{3}$$

or equivalently,

$$\underset{\Delta U}{min}\; J = \|Y_{sp} - \hat{Y}\|_{\lambda_1}^2 + \|\Delta U\|_{\lambda_2}^2, \tag{4}$$

$$\begin{aligned} & \qquad u_{min} \le \; u(t+j) \le u_{max} \\ s.t. \quad & \Delta u_{min} \le \Delta\; u(t+j) \le \Delta u_{max} \\ & \qquad y_{min} \le \; y(t+j) \le y_{max}, \end{aligned} \tag{5}$$

where Y_{sp} is the setpoint of the process, $\hat{Y}$ is the prediction of the output and ΔU is the incremental value of the controller output to be calculated. λ_1 and λ_2 are two weights to make a trade-off between the tracking error and the cost of the control action.

2.2 Basic Knowledge of Broad Learning

As an improved neural network generated from the RVFLNN, broad learning has been proven to be an efficient algorithm for regression and classification, where it is important to extract the features from the training data. A random initialization of the features is always a popular way due to its easy computation. However, the feature extracted by random projection may suffer from unpredictability and needs guidance. To avoid this, a set of sparse features can be extracted from the training data using the sparse autoencoder (SAE) discussed in Gong et al. (2015). In this way, the alternating direction method of multipliers (ADMM) technique can be used to solve the *lasso* problem in the broad learning algorithm (Chen and Liu, 2018) to fine-tune the

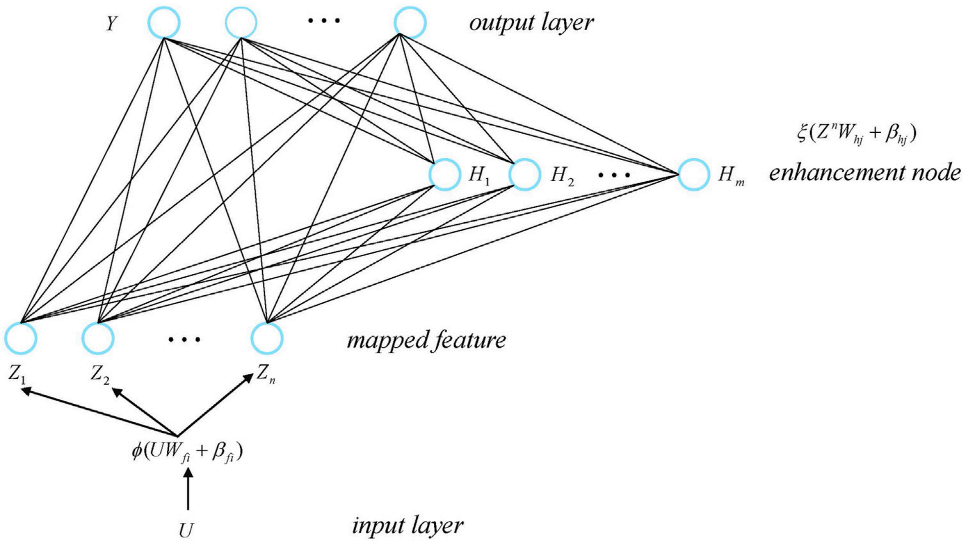


FIGURE 1 | Functional-linked broad learning model.

weight matrices. The details of the broad learning is given as below.

Assume that $U \in \mathcal{R}^{N\times l}$ and $Y \in \mathcal{R}^{N\times m}$ respectively denote the collected input and output data with N samples. The ith mapped feature $Z_i \in \mathcal{R}^{N\times\alpha}$ can then be calculated as

$$Z_i = \phi(UW_{ei} + \beta_{ei}), \tag{6}$$

where $\phi(\cdot)$ denotes the projection function, $W_{ei} \in \mathcal{R}^{l\times\alpha}$ and $\beta_{ei} \in \mathcal{R}^{N\times\alpha}$ are two random matrices.

Denote $Z^n \equiv [Z_1, Z_2, \ldots, Z_n]$ as the concentration of the mapped feature. Z^n can then be used to generate the jth enhancement node $H_j \in \mathcal{R}^{N\times\gamma}$ as

$$H_j = \zeta\left(Z^n W_{hj} + \beta_{hj}\right), \tag{7}$$

where $\zeta(\cdot)$ denotes the projection function, $W_{hj} \in \mathcal{R}^{\alpha\times\gamma}$ and $\beta_{hj} \in \mathcal{R}^{N\times\gamma}$ are two random matrices.

Similarly, denote $H^m \equiv [H_1, H_2, \ldots, H_m]$ as the collection of the enhancement nodes. In this way, the global model can be represented as

$$Y = [Z^n \mid H^m]W^m = A_n^m W^m. \tag{8}$$

where $A_n^m \in \mathcal{R}^{N\times(n\alpha+m\gamma)}$ denotes the input matrix and $W^m \in \mathcal{R}^{(n\alpha+m\gamma)\times p}$ denotes the weight matrix.

It is straightforward that W^m can be calculated as

$$W^m = \left(A_n^m\right)^+ Y. \tag{9}$$

where $(\cdot)^+$ represents the moore-penrose pseudoinverse. To reduce the computation complexity, the broad learning algorithm makes use of the ridge regression theory proposed in Hoerl and Kennard (2000).

Assume that A_n^m is of full column rank, we have

$$\left(A_n^m\right)^+ = \lim_{\lambda\to 0}\left(\lambda I + \left(A_n^m\right)^{\mathrm{T}} A_n^m\right)^{-1}\left(A_n^m\right)^{\mathrm{T}}, \tag{10}$$

and consequently,

$$W^m = \lim_{\lambda\to 0}\left(\lambda I + \left(A_n^m\right)^{\mathrm{T}} A_n^m\right)^{-1}\left(A_n^m\right)^{\mathrm{T}} Y. \tag{11}$$

The functional-linked broad learning model is shown in **Figure 1**. More details on the broad learning can be referred to Chen and Liu (2018).

3 BROAD LEARNING BASED PREDICTIVE CONTROL

3.1 Incremental Broad Learning Algorithm

Though broad learning has been proven to be efficient for classification and regression, it is not easy to regress the process model based on the online data due to the unpredictable noise, model uncertainty and dynamic characteristics. To implement the predictive control in the framework of the broad learning, it is essential to develop the equivalent incremental broad learning algorithm. By updating the parameters of the broad learning model, the precision of the regression model can be further enhanced. The incremental broad learning algorithm for new training data was briefly discussed in Chen and Liu (2018). The details of the incremental broad learning for new training data will be presented in this section.

According to Chen and Wan (1999), the stepwise updating algorithm of a functional-linked neural network for a new enhancement node can be equivalently represented as adding several columns to the input matrix. Denote the initial input matrix as A_n, the final input matrix is then given as $A_{n+1} = [A_n|a]$ when a new enhancement node a is added to the neural network.

Inspired by the moore-penrose pseudo inverse of partitioned matrix mentioned in Diaz and Gutierrez (2006), the pseudo inverse of A_{n+1} can be calculated as

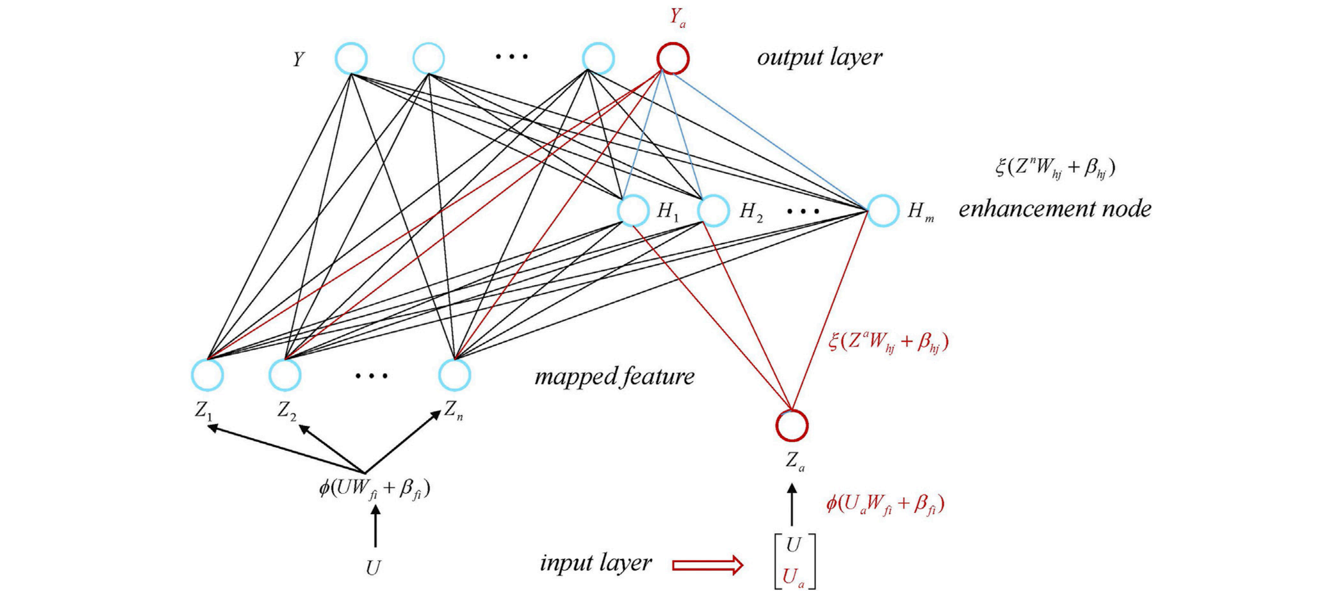

Y_a
Y
output layer
$\xi(Z^n W_{hj} + \beta_{hj})$
H_1
H_2
H_m
enhancement node
$\xi(Z^a W_{hj} + \beta_{hj})$
mapped feature
Z_1
Z_2
Z_n
Z_a
$\phi(UW_{fi} + \beta_{fi})$
$\phi(U_a W_{fi} + \beta_{fi})$
U
input layer
$\begin{bmatrix} U \\ U_a \end{bmatrix}$

FIGURE 2 | Incremental broad learning for new training data.

$$A_{n+1} = \begin{bmatrix} A_n^+ - db^\top \\ b^\top \end{bmatrix}, \tag{12}$$

where

$$b^\top = \begin{cases} c^+ & if\ c \neq 0 \\ (1 + d^\top d)^{-1} d^\top A_n^+ & if\ c = 0 \end{cases}, \tag{13}$$

$$d = A_n^+ a, \tag{14}$$

$$c = a - A_n d. \tag{15}$$

In this way, the new weights W_{n+1} can be calculated as

$$W_{n+1} = \begin{bmatrix} W_n - db^\top Y_n \\ b^\top Y_n \end{bmatrix}, \tag{16}$$

Denote X_a and Y_a as a group of new training input data and the corresponding output data. The incremental algorithm broad learning for new training data can be demonstrated as **Figure 2**.

The original combination of feature nodes and enhancement nodes forms the following input matrix A_n^m:

$$A_n^m = [\phi(UW_{e1} + \beta_{e1}), \ldots, \phi(UW_{en} + \beta_{en}) \mid \tag{17}$$

$$\xi(Z^n W_{h1} + \beta_{h1}), \ldots, \xi(Z^n W_{hm} + \beta_{hm})]. \tag{18}$$

The input matrix ${}^u A_n^m$ trained by the new input data can be calculated as

$$\begin{aligned} {}^u A_n^m = [&\phi({}^a U W_{e1} + {}^a\beta_{e1}), \ldots, \phi({}^a U W_{en} + {}^a\beta_{en}) \mid \\ &\xi({}^a Z^n W_{h1} + {}^a\beta_{h1}), \ldots, \xi({}^a Z^n W_{hm} + {}^a\beta_{hm})], \end{aligned} \tag{19}$$

where

$${}^a U = \begin{bmatrix} U \\ U_a \end{bmatrix}, {}^a\beta_{ei} = \begin{bmatrix} \beta_{ei} \\ \beta_{ei}^a \end{bmatrix}, i = 1, \ldots, n, \tag{20}$$

$${}^a Z^n = \begin{bmatrix} Z^n \\ Z^a \end{bmatrix}, {}^a\beta_{hi} = \begin{bmatrix} \beta_{hi} \\ \beta_{hi}^a \end{bmatrix}, i = 1, \ldots, m, \tag{21}$$

$$Z^a = [\phi(U_a W_{e1} + \beta_{e1}^a), \ldots, \phi(U_a W_{en} + \beta_{en}^a)). \tag{22}$$

It is obvious that ${}^u A_n^m$ can be equivalently constructed as adding several rows generated by U_a to the bottom of A_n^m.

Denote the input matrix generated by the new training data U_a as A_u, ${}^u A_n^m$ can then be represented as

$${}^u A_n^m = \begin{bmatrix} A_n^m \\ A_u \end{bmatrix}, \tag{23}$$

where

$$\begin{aligned} A_u = \ &[\phi(U_a W_{e1} + \beta_{e1}^a), \ldots, \phi(U_a W_{en} + \beta_{en}^a) \mid \\ &\xi(Z^a W_{h1} + \beta_{h1}^a), \ldots, \xi(Z^a W_{hm} + \beta_{hm}^a)]. \end{aligned} \tag{24}$$

The pseudoinverse of ${}^u A_n^m$ can be calculated as

$$\begin{aligned} ({}^u A_n^m)^+ &= \left(\left[(A_n^m)^\top \mid A_u^\top\right]^+\right)^\top \\ &= \begin{bmatrix} \left((A_n^m)^+\right)^\top - DB^\top \\ B^\top \end{bmatrix}^\top \\ &= \left[(A_n^m)^+ - BD^\top \mid B\right], \end{aligned} \tag{25}$$

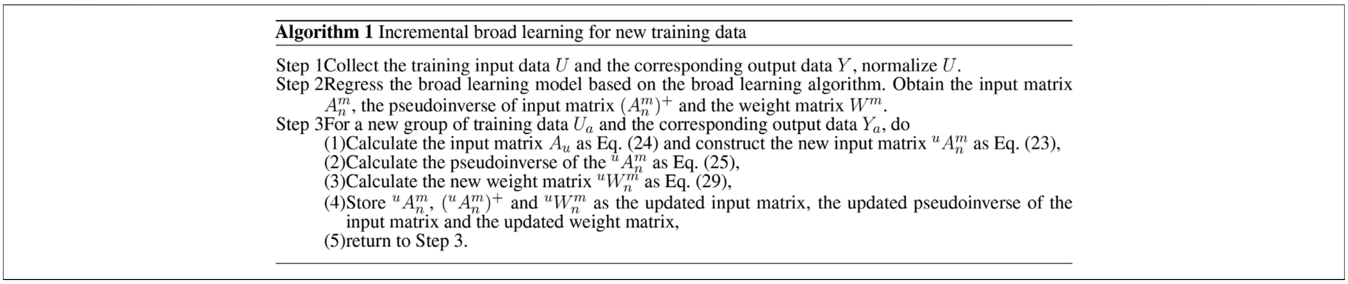
where

$$B = \begin{cases} (C^\top)^+ & if\ C \neq 0 \\ (A_n^m)^+ D(1 + D^\top D)^{-1} & if\ C = 0 \end{cases}, \tag{26}$$

$$D^\top = A_u (A_n^m)^+, \tag{27}$$

$$C^\top = A_u - D^\top A_n^m. \tag{28}$$

The weight matrix of the new broad learning model can be calculated as



```
Algorithm 1 Incremental broad learning for new training data

Step 1Collect the training input data $U$ and the corresponding output data $Y$, normalize $U$.
Step 2Regress the broad learning model based on the broad learning algorithm. Obtain the input matrix
    $A_n^m$, the pseudoinverse of input matrix $(A_n^m)^+$ and the weight matrix $W^m$.
Step 3For a new group of training data $U_a$ and the corresponding output data $Y_a$, do
    (1)Calculate the input matrix $A_u$ as Eq. (24) and construct the new input matrix ${}^uA_n^m$ as Eq. (23),
    (2)Calculate the pseudoinverse of the ${}^uA_n^m$ as Eq. (25),
    (3)Calculate the new weight matrix ${}^uW_n^m$ as Eq. (29),
    (4)Store ${}^uA_n^m$, $({}^uA_n^m)^+$ and ${}^uW_n^m$ as the updated input matrix, the updated pseudoinverse of the
       input matrix and the updated weight matrix,
    (5)return to Step 3.
```

$$
\begin{aligned}
{}^uW_n^m &= \left[\left(A_n^m\right)^+ - BD^\top \mid B \right] \begin{bmatrix} Y \\ Y_a \end{bmatrix} \\
&= \left(A_n^m\right)^+ Y - BD^\top Y + BY_a \\
&= W_n^m + B\left(Y_a - A_u W_n^m\right),
\end{aligned}
\tag{29}
$$

The incremental broad learning algorithm for new training data can be summarized in **Algorithm 1**.

Remark 1. *In fact, the broad learning algorithm is herein used to the establish the predictive model for control purpose. Note that the predictive model does not need to be retrained when the new data arrives if the incremental broad learning algorithm is applied. The main computation is spent on the update of the new weights* (*i.e.,* ${}^uW_n^m$) *and the pseudo inverse of input matrix* (*i.e.,* ${}^uA_n^m$) *based on the previous value when the predictive model needs to be updated, which requires less computational load compared with the basic broad learning algorithm* (*i.e., requires retraining the network model with full data*)

3.2 Incremental Broad Learning Based Predictive Control

For predictive control purpose, the cost function $J(k)$ to be minimized at the k^{th} instant can be given as follows,

$$
J(k) = \|Y_{sp}(k+1) - Y(k+1|k)\|_{\lambda_1}^2 + \|\Delta U(k+1)\|_{\lambda_2}^2, \tag{30}
$$

where

$$
Y_{sp}(k+1) = \begin{bmatrix} y_{sp}(k+1) \\ y_{sp}(k+2) \\ \vdots \\ y_{sp}\left(k+n_p\right) \end{bmatrix} \in \mathcal{R}^{\left(n_y \times n_p\right)\times 1}, \tag{31}
$$

$$
Y(k+1|k) = \begin{bmatrix} y(k+1|k) \\ y(k+2|k) \\ \vdots \\ y(k+n_c|k) \end{bmatrix} \in \mathcal{R}^{\left(n_y \times n_c\right)\times 1}, \tag{32}
$$

$$
\Delta U(k+1) = \begin{bmatrix} \Delta u(k+1) \\ \Delta u(k+2) \\ \vdots \\ \Delta u(k+n_c) \end{bmatrix} \in \mathcal{R}^{(n_u \times n_c)\times 1}, \tag{33}
$$

$$
\Delta u(k+i) = u(k+i) - u(k). \tag{34}
$$

and n_y, n_u, n_p, and n_c are the dimension of an output vector, the dimension of the input vector, the predictive horizon and the control horizon, respectively.

Remark 2. *It is worth mentioning that n_p should be set to be longer than the settling time in order to satisfy the convergence condition of the predictive control* (Rossiter, 2003). *n_p has no influence on the convergence of predictive control if n_p is greater than the settling time. However, n_p is often determined based on the practical experience since the settling time is unknown. In this paper, the length of the control horizon n_c and prediction horizon n_p are both set to be* 50.

In the broad learning algorithm, the relationship between input training data U and output training data Y can be represented as

$$
Y = UW_0 + \beta_0, \tag{35}
$$

where

$$
W_0 = [W_e, W_e W_h] W^m, \tag{36}
$$

$$
W_e = [W_{e1}, \ldots, W_{en}], \; W_h = [W_{h1}, \ldots, W_{hn}], \tag{37}
$$

$$
\beta_0 = [\beta_e, \beta_e W_h + \beta_h] W^m, \tag{38}
$$

$$
\beta_e = [\beta_{e1}, \ldots, \beta_{en}], \; \beta_h = [\beta_{h1}, \ldots, \beta_{hn}]. \tag{39}
$$

Therefore, regardless of the activation function, the correlation between $u(k)$ and $y(k)$ can be presented as

$$
y(k) = (W^m)^{\mathrm{T}} \begin{bmatrix} W_e^{\mathrm{T}} u(k) \\ W_h^{\mathrm{T}} W_e^{\mathrm{T}} u(k) \end{bmatrix}, \tag{40}
$$

However, the activation function $F(\cdot)$ is necessary to extract the nonlinear characteristics. In the broad learning, the prediction of $y(k)$ is calculated as

$$
\begin{aligned}
\hat{y}(k) &= (W^m)^{\mathrm{T}} \begin{bmatrix} W_e^{\mathrm{T}} u(k) \\ F(Wu(k)) \end{bmatrix} \\
&= (W^m)^{\mathrm{T}} \begin{bmatrix} W_e^{\mathrm{T}} u(k) \\ f\left(w_{1,1}u_1 + \cdots + w_{1,n_u}u_{n_u}\right) \\ \vdots \\ f\left(w_{h,1}u_1 + \cdots + w_{h,n_u}u_{n_u}\right) \end{bmatrix} \\
&= (W^m)^{\mathrm{T}} \begin{bmatrix} W_e^{\mathrm{T}} u(k) \\ f_1(u) \\ \vdots \\ f_h(u) \end{bmatrix},
\end{aligned}
\tag{41}
$$

where

$$
f(net) = \frac{1 - e^{net}}{1 + e^{net}}. \tag{42}
$$

Note that the nonlinearity brings challenges to the receding optimization of the predictive control since the cost function to be minimized is nonlinear. Therefore, an appropriate approximation of the predictive model is necessary.

According to **Eq. 41**, the prediction at the instant k can be rewritten as

$$\hat{y}(k) = G(u(k)), \tag{43}$$

Assume that $u(k)$ is known, $G(u(k))$ can then be calculated easily via the general inverse. However, the incremental ΔU is to be calculated by solving the quadratic problem as **Eq. 30**. Take the prediction of the $y(k+1)$ at the instant k as an example,

$$\begin{aligned} \hat{y}(k+1) &= G(u(k+1)) \\ &= G(u(k) + \Delta u(k+1)), \end{aligned} \tag{44}$$

where $\Delta u(k+1)$ is unknown.

Since $G(u)$ is continuous differentiable at any u, define the *Jacobian* matrix of $G(u)$ as

$$J_G(u) = G'(u),\ G'(u)_{ij} = \frac{\partial G_i}{\partial u_j}(u), \tag{45}$$

In this way, the approximation of $G(u(k+1))$ can be given as

$$\hat{G}(u(k+1)) = G(u(k)) + J_G(u(k))\Delta u(k+1). \tag{46}$$

According to the definition of $G(\cdot)$ and $f(net)$ as presented in **Eqs 41**, **42**. It holds that $f'(net) = 1 - f(net)^2 \le 1$ and $G(u)$ is continuously differentiable for any $u \in \mathcal{R}^{n_u}$. The *Jacobian* matrix $J_G(u)$ is *Lipschitz continuous* for $u \in \mathcal{R}^{n_u}$. Therefore, there exists a *Lipschitz* constant γ such that for any $v \in \mathcal{R}^{n_u}$

$$\|J_G(v) - J_G(u)\| \le \gamma \|v - u\|. \tag{47}$$

Theorem 1. *Let* **Eqs 41**, **42**, **47** *holds, replacing $G(u(k))$ with $\hat{G}(u(k))$ has no influence on the convergence of the quadratic problem as shown in* **Eq. 30**.

Proof. Define the deviation E_G between $\hat{G}(u(k+1))$ and $G(u(k+1))$ as

$$E_G = G(u(k+1)) - \hat{G}(u(k+1)), \tag{48}$$

According to the mean value theorem, we have

$$\begin{aligned} E_G = &\ G(u(k+1)) - G(u(k)) - J_G(u(k))\Delta u(k+1) \\ = &\left[\int_0^1 J_G(u(k) + t\Delta u(k+1))dt\right]\Delta u(k+1) \\ &-J_G(u(k))\Delta u(k+1) \\ = &\int_0^1 [J_G(u(k) + t\Delta u(k+1))]\Delta u(k+1)dt \\ &-J_G(u(k))\Delta u(k+1) \\ = &\int_0^1 [J_G(u(k) + t\Delta u(k+1)) - J_G(u(k))]\Delta u(k+1)dt, \end{aligned} \tag{49}$$

Consider the equivalence of Riemann sum and integral, it yields

$$\begin{aligned} \|E_G\| &= \|G(u(k+1)) - G(u(k)) - J_G(u(k))\Delta u(k+1)\| \\ &\le \int_0^1 \|J_G(u(k) + t\Delta u(k+1)) \\ &\quad -J_G(u(k))\|\|\Delta u(k+1)\|dt \\ &\le \int_0^1 \gamma \|t\Delta u(k+1)\|\|\Delta u(k+1)\|dt\Big] \\ &= \frac{\gamma}{2}\|\Delta u(k+1)\|^2, \end{aligned} \tag{50}$$

$$\begin{aligned} \hat{J}(k) &= \sum_{i=1}^{n_p} \|y_{sp}(k+i) - \hat{G}u(k+i)\|_{\lambda_1}^2 \\ &\quad + \sum_{i=1}^{n_p} \|\Delta u(k+i)\|_{\lambda_2}^2 \\ &\le \sum_{i=1}^{n_p} \|y_{sp}(k+i) - G(u(k+1))\|_{\lambda_1}^2 \\ &\quad + \sum_{i=1}^{n_p} \frac{\gamma}{2}\|\Delta u(k+i)\|_{\lambda_1}^2 + \sum_{i=1}^{n_p} \|\Delta u(k+i)\|_{\lambda_2}^2, \end{aligned} \tag{51}$$

Consider that λ_1 and λ_2 are two weight constants and it holds that $\lambda_2 \gg \lambda_1$, it yields

$$\begin{aligned} \hat{J}(k) \le &\sum_{i=1}^{n_p} \|y_{sp}(k+i) - G(u(k+1))\|_{\lambda_1}^2 \\ &+ \sum_{i=1}^{n_p} \|\Delta u(k+i)\|_{\lambda_1 + \frac{\gamma}{2}}^2 + \lambda_2 \\ \approx &\sum_{i=1}^{n_p} \|y_{sp}(k+i) - G(u(k+1))\|_{\lambda_1}^2 \\ &+ \sum_{i=1}^{n_p} \|\Delta u(k+i)\|_{\frac{\gamma}{2} + \lambda_2}^2 \\ = &\ \|Y_{sp}(k+1) - Y(k+1|k)\|_{\lambda_1}^2 + \|\Delta U(k+1)\|_{\frac{\gamma}{2} + \lambda_2}^2, \end{aligned} \tag{52}$$

Therefore, the convergence of $\hat{J}(k)$ is equivalent to the convergence of $J(k)$, which thus completes the proof.

Remark 3. Th*e Lipschitz constant γ is related to the spectral radius of W. In practice, Δu is often constrained due to the mechanical structure of the actuators. The deviation E_G can be acceptable with an appropriate W.*

Denote W as

$$W = \begin{bmatrix} W_1 \\ \vdots \\ W_h \end{bmatrix}, \tag{53}$$

The *Jacobian* matrix $J_G(u(k))$ can be constructed as

$$J_G(u(k)) = (W^m)^{\mathrm{T}} \cdot \begin{bmatrix} W_e^{\mathrm{T}} \\ F_W \circ W \end{bmatrix}, \tag{54}$$

where $\circ$ denotes the Hadamard product and

$$F_W = \begin{bmatrix} (1 - f^2(W_1 u(k))) & \dots & 1 - f^2(W_1 u(k)) \\ \vdots & \ddots & \vdots \\ (1 - f^2(W_h u(k))) & \dots & 1 - f^2(W_h u(k)) \end{bmatrix}. \tag{55}$$

In addition, the prediction of $y(k+i)$ at the k^{th} instant $y(k+i|k)$ can be given as

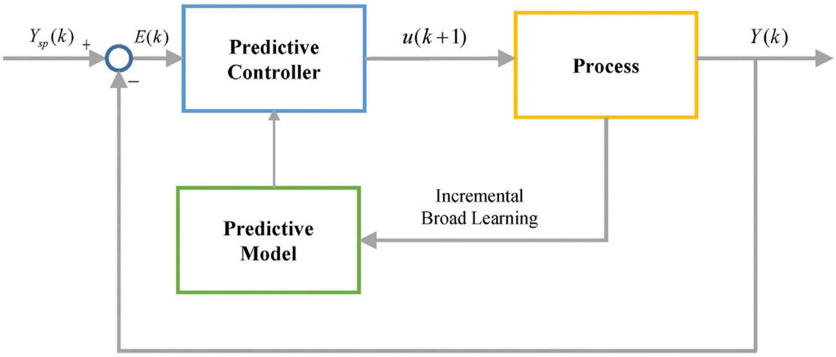


FIGURE 3 | Incremental broad learning based predictive control scheme.

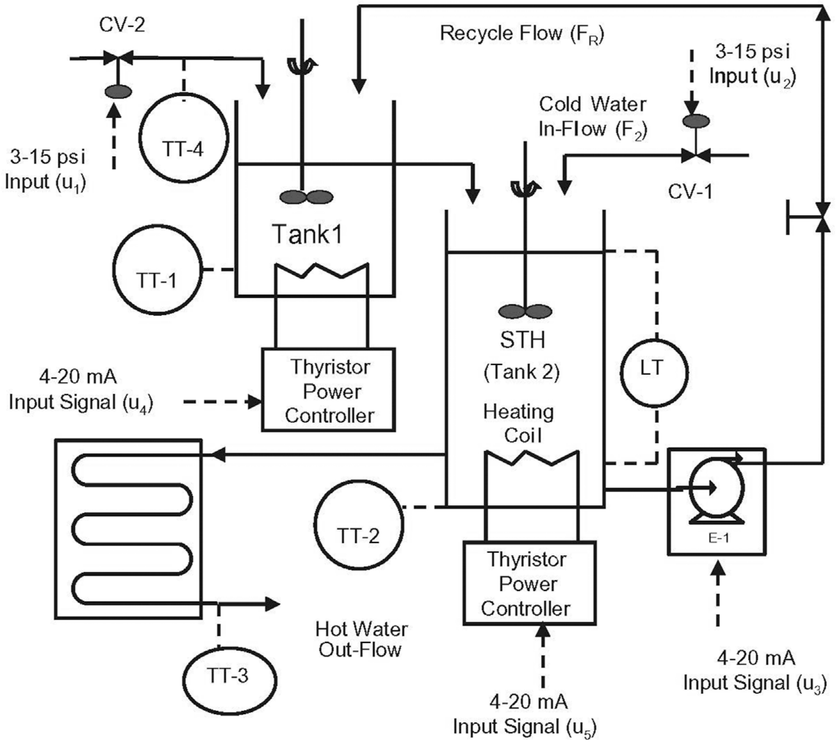


FIGURE 4 | CSTH system in IIT Bombay.

$$\begin{aligned} y(k+i|k) &= G(u(k)) + J_G(u(k))\Delta u(k+i) \\ &= y(k) + J_G(u(k))\Delta u(k+i), \end{aligned} \tag{56}$$

which leads to

$$Y(k+1|k) = Y(k) + M(k+1)\Delta U(k+1), \tag{57}$$

where

$$Y(k) = [y(k)\ldots, y(k)]^{\mathrm{T}} \in \mathcal{R}^{(n_y \times n_c)\times 1}, \tag{58}$$

$$M(k) = [J_G(u(k))\ldots, J_G(u(k))]^{\mathrm{T}} \in \mathcal{R}^{(n_y \times n_c)\times n_u}. \tag{59}$$

Substituting **Eqs 57**, **59** into **Eq. 30**, the cost function can be rewritten as

$$\begin{aligned} \underset{\Delta U(k+1)}{min}\ J(k) &= \|E(k) - M(k)\Delta U(k+1)\|_{\lambda 1}^2 \\ &+ \|\Delta U(k+1)\|_{\lambda 2}^2, \end{aligned} \tag{60}$$

subject to the same constraint as **Eq. 5**, where $E(k) = Y_{sp}(k) - Y(k)$. The proposed predictive control is illustrated in **Figure 3**.

4 BENCHMARK STUDY ON CONTINUOUS STIRRED TANK HEATER

In this section, the proposed predictive control is applied to tune the temperature in the continuous stirred tank

TABLE 1 | Nominal model parameters and steady state.

Parameter	Description	Value
V_1	Volume of tank 1	1.75×10^{-3} m^3
A_2	Cross sectional area of tank 2	7.854×10^{-3} m^2
r_2	Radius of tank 2	0.05 m
U	Heat transfer coefficient	235.1 W/m^{2}K
T_c	Cooling water temperature	30°C
T_a	Atmospheric temperature	25°C
u_1	Flow F_1(*%Input*)	60%
u_2	Flow F_2(*%Input*)	55%
u_3	Flow F_R(*%Input*)	50%
u_4	Heat input Q_1(*%Input*)	60%
u_5	Heat input Q_2(*%Input*)	80%
T_1	Steady state temperature(tank 1)	49.77°C
T_2	Steady state temperature(tank 2)	52.92°C
h_2	Steady state level	0.3599 m

heater (CSTH) benchmark. The performance of the proposed control strategy is compared with the partial form dynamic linearization aided model free control (PFDL-MFC) strategy proposed in Hou and Jin (2013) and Hou and Jin (2011). The PFDL-MFC algorithm is chosen for comparison because both algorithms need little prior knowledge of the system model and calculate the optimal control input sequence based on **Eq. 4**.

The continuous stirred tank heater (CSTH) is a typical pilot plant for control performance evaluation and process monitoring. The experimental CSTH system is developed in the Automation Laboratory at Department of Chemical Engineering, IIT Bombay as shown in **Figure 4** (Thornhill et al., 2008). It mainly consists of two separated water storage tanks. The water is heated separately and recycled between the two tanks. The CSTH benchmark can be regarded as a 5-input-3-output system. u_1, u_2, and u_3 represent the inputs controlled by separate valves. u_4 and u_5 represent the outputs of two heaters. The three outputs are the water temperature T_1 (Tank 1), T_2 (Tank 2) and the water level h_2 (Tank 2), respectively. **Table 1** gives the parameters of the CSTH benchmark in steady state.

Consider that the heat input is added to the state variables in the form of polynomial functions (Thornhill et al.,2008), the controlled system is nonlinear even if the flow rate is steady. For simplicity, we only use u_4 and u_5 as the controller outputs to be calculated. The designed predictive controller aims to make T_1 track its setpoint. If not specified, the other variables are set to be steady state in **Table 1**. The practical constraints in the CSTH benchmark are set to be

$$\begin{array}{c} 0 \leq u \leq 100 \\ -0.2 \leq \Delta u \leq 0.2 \end{array}. \tag{61}$$

Smooth approximation is a popular method to enhance the tracking performance (Chi and Liang, 2015). By smooth approximation, the setpoint can be changed smoothly from one desired state to the other. In the simulation, the setpoint is given as

$$y_{sp}(k) = \begin{cases} y_{s1}, \ 0 < k \leq 800 \\ \lambda_{sp} y_{sp}(t-1) + \left(1 - \lambda_{sp}\right) y_{s2}, \ 800 < k \leq 1400 \\ \lambda_{sp} y_{sp}(t-1) + \left(1 - \lambda_{sp}\right) y_{s1}, \ 1400 < k \leq 2000 \end{cases},$$

where $y_{s1} = 49$, $y_{s2} = 50$ and the smoothing coefficient λ_{sp} is set to be 0.98.

Consider that the proposed predictive control requires initializing its predictive model based on the available process data. In this simulation, the process inputs are set to be random square waves with white noise in the first 200 s. The initial predictive model is calculated using the process data collected in the first 200 s. After the 200^{th} instant, the predictive model is updated based on the online process data.

Define the prediction error E_p at the instant k as

$$E_p(k) = \frac{\sum_{i=1}^{n_p} \| y(k+i) - y(k+i|k) \|^2}{n_p}. \tag{62}$$

where n_c and n_p are both set to be 50.

Other parameters in the PF-MFPC algorithm are respectively set to be $\lambda = 27$, $\mu = 1$, $\rho = [1, 1, 1]$, $\eta = 1$, and $L = 3$. The illustrations of the designed parameters are shown in **Table 2**. The comparison of the prediction error is shown in **Figures 5**, **6**, from which it can be seen that there exists a fluctuation at the 200^{th}, 800^{th}, and 1400^{th} instant. The fluctuation is caused by the setpoint change. Therefore, both algorithms need to update the model dynamics based on the online process data. The comparison shows that the prediction of the proposed control strategy converges much faster than the PFDL-MFC algorithm. In addition, the fluctuation peaks of the proposed broad learning aided MPC is also much lower than the PFDL-MFC algorithm.

It is worth mentioning that the fluctuation of E_p occurs before the setpoint changes at the 1400^{th} and 4800^{th} instant since the proposed predictive controller is based on the multi-step prediction. In this simulation, it holds that $n_c = n_p = 50$, which implies that the prediction of the output and the incremental of the input can be foreseen in the future 50 sample instants.

The control performance of the proposed broad learning aided MPC and the PFDL-MFC algorithm are demonstrated in **Figures 7**, **8**, respectively. As the setpoint changes, the input signal converges after a few steps of adjustment. Both algorithms show the efficiency in the tracking control for nonlinear systems. However, the proposed algorithm

TABLE 2 | Illustration of the designed parameters.

Parameter	Description	Value
n_c	Length of control horizon	50
n_p	Length of prediction horizon	50
L	Order of pseudo-partial derivative	3
Λ	Weighting constant	27
M	Weighting factor	1
η	Step size constant	1
ρ	Step size vector	[1, 1, 1]

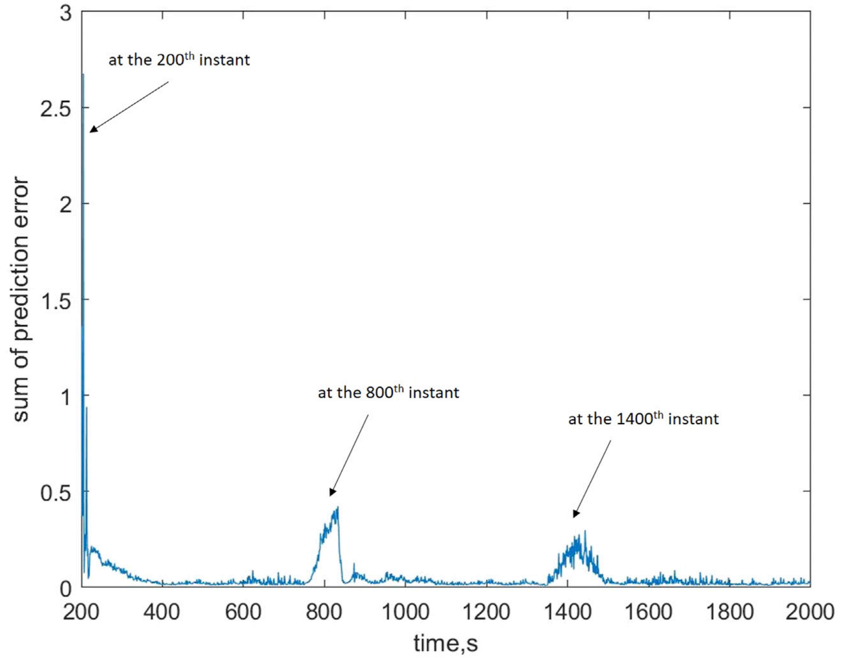


FIGURE 5 | Prediction error of broad learning aided predictive control.

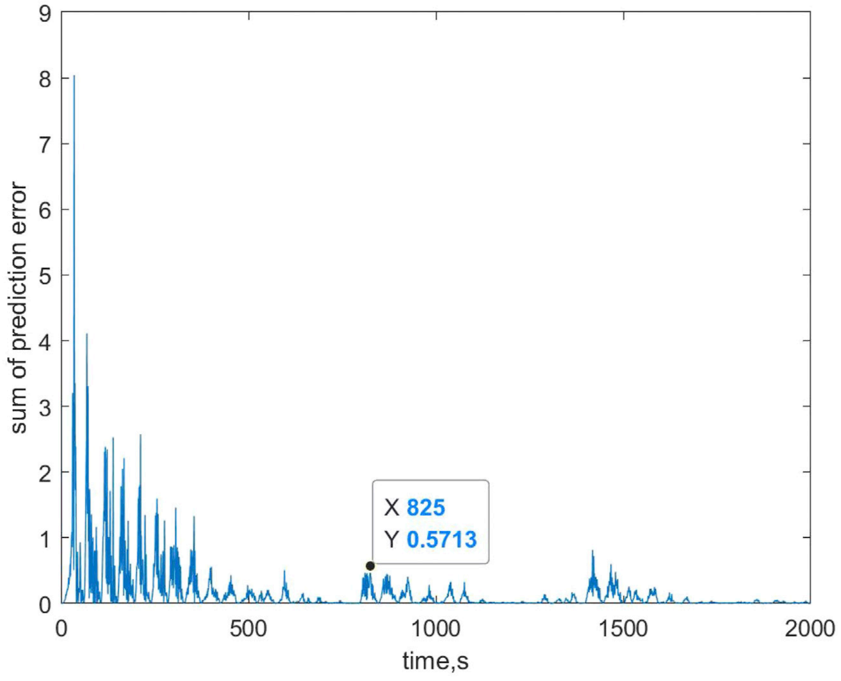


FIGURE 6 | Prediction error of PFDL-MFC algorithm.

converges much faster than the PFDL-MFC algorithm though it requires to collect the data during the first several instants. In addition, the parameters of the PFDL-MFC algorithm have a great influence on the control performance. In contrast, only the weight λ_1, λ_2 and the prediction horizon n_p can be adjustable in the proposed predictive control strategy, which shows minor influence on the control performance.

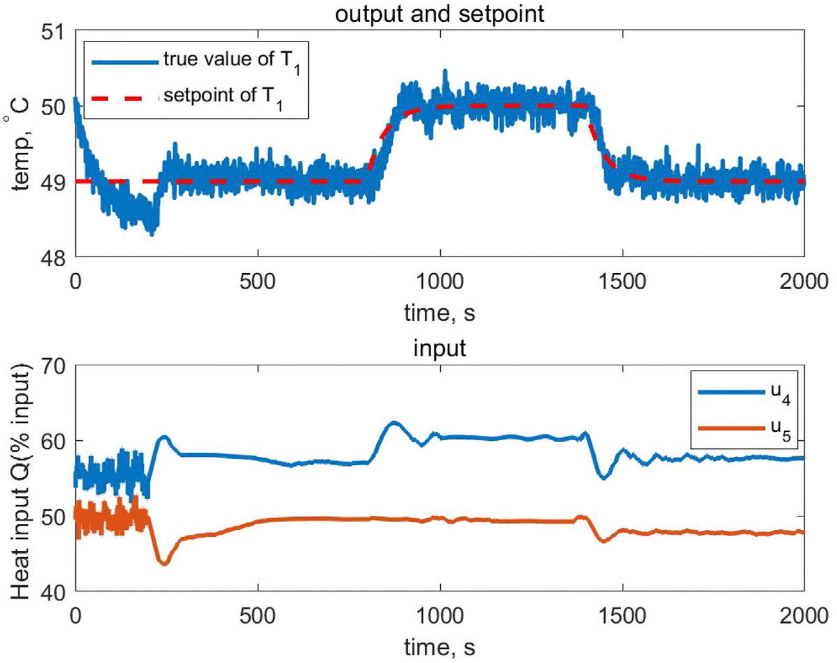


FIGURE 7 | Control performance of broad learning aided predictive control.

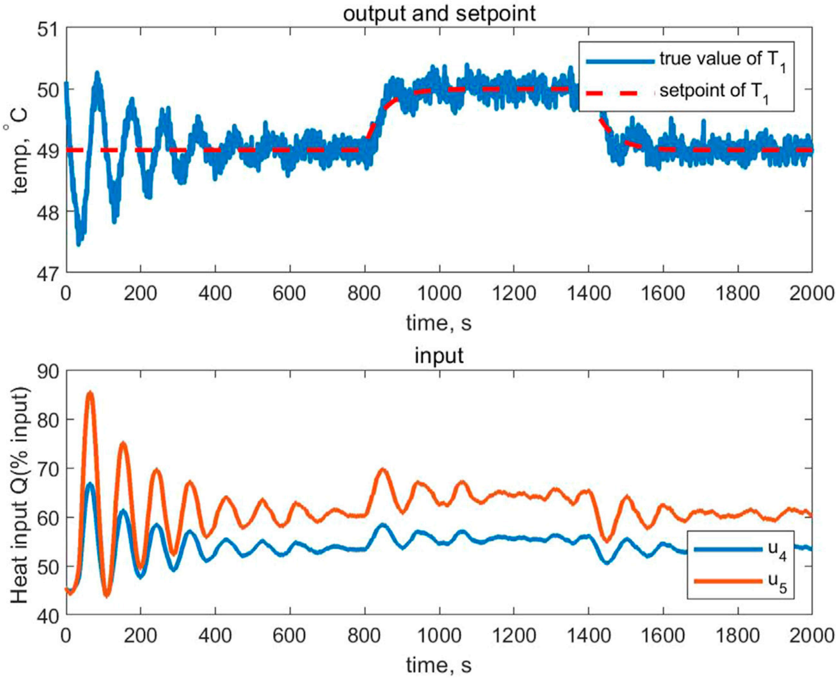


FIGURE 8 | Control performance of PFDL-MFC algorithm.

5 CONCLUSION

This paper proposes a broad learning based data-driven predictive control strategy without any prior knowledge of the system model. The online computation load of the proposed predictive controller is reduced by using the incremental broad learning algorithm. In addition, the prediction precision is enhanced by approximating the predictive model using a Jacobian matrix. The convergence of the predictive control strategy is discussed theoretically, which demonstrates that the approximation has no influence on the convergence of the predictive control algorithm. The effectiveness of the proposed broad learning based MPC is demonstrated for temperature adjustment through the continuous stirred tank heater benchmark.

Compared with other NN based predictive control strategies, the proposed strategy can update the predictive model online with less online computation burden. Meanwhile, the proposed predictive controller generates the local nonlinearity between the input and output data compared with the dynamic linearization based model free control strategy, which enhances the prediction precision and control performance. The simulation result shows that the proposed predictive control strategy shows faster convergence both in the control performance and the prediction than the PFDL-MFC algorithm. In the future, more work can be done to further reduce the online computation burden.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: http://www.ps.ic.ac.uk/~nina/CSTHSimulation/index.htm.

AUTHOR CONTRIBUTIONS

MT and TG: conceptualization, methodology, writing. XL: supervision. KL: writing. All authors contributed to the article and approved the submitted version.

ACKNOWLEDGMENTS

This work is funded by Open Foundation of Science and Technology on Thermal Energy and Power Laboratory under Grant TPL2018C02.

REFERENCES

Chen, C. L. P., and Liu, Z. (2018). Broad Learning System: An Effective and Efficient Incremental Learning System without the Need for Deep Architecture. *IEEE Trans. Neural Netw. Learn. Syst.* 29, 10–24. doi:10.1109/tnnls.2017.2716952

Chen, C. L. P., and Wan, J. Z. (1999). A Rapid Learning and Dynamic Stepwise Updating Algorithm for Flat Neural Networks and the Application to Time-Series Prediction. *IEEE Trans. Syst. Man. Cybern. B* 29, 62–72. doi:10.1109/3477.740166

Chi, Q., and Liang, J. (2015). A Multiple Model Predictive Control Strategy in the Pls Framework. *J. Process Control.* 25, 129–141. doi:10.1016/j.jprocont.2014.12.002

Díaz-García, J. A., and Gutiérrez-Jáimez, R. (2006). Distribution of the Generalised Inverse of a Random Matrix and its Applications. *J. Stat. Plann. Inference* 136, 183–192. doi:10.1016/j.jspi.2004.06.032

Gao, T., Yin, S., Qiu, J., Gao, H., and Kaynak, O. (2018). A Partial Least Squares Aided Intelligent Model Predictive Control Approach. *IEEE Trans. Syst. Man. Cybern, Syst.* 48, 2013–2021. doi:10.1109/tsmc.2017.2723017

Garcia-Torres, F., Bordons, C., and Ridao, M. A. (2019). Optimal Economic Schedule for a Network of Microgrids with Hybrid Energy Storage System Using Distributed Model Predictive Control. *IEEE Trans. Ind. Electron.* 66, 1919–1929. doi:10.1109/tie.2018.2826476

Garg, A., and Mhaskar, P. (2017). Subspace Identification and Predictive Control of Batch Particulate Processes."in American Control Conference (ACC). doi:10.23919/acc.2017.7963003

Gong, M., Liu, J., Li, H., Cai, Q., and Su, L. (2015). A Multiobjective Sparse Feature Learning Model for Deep Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* 26, 3263–3277. doi:10.1109/tnnls.2015.2469673

Guo, P., He, Z., Yue, Y., Xu, Q., Huang, X., Chen, Y., et al. (2019). A Novel Two-Stage Model Predictive Control for Modular Multilevel Converter with Reduced Computation. *IEEE Trans. Ind. Electron.* 66, 2410–2422. doi:10.1109/tie.2018.2868312

Han, H.-G., Zhang, L., Hou, Y., and Qiao, J.-F. (2016). Nonlinear Model Predictive Control Based on a Self-Organizing Recurrent Neural Network. *IEEE Trans. Neural Netw. Learn. Syst.* 27, 402–415. doi:10.1109/tnnls.2015.2465174

Hidalgo, P. M., and Brosilow, C. B. (1990). Nonlinear Model Predictive Control of Styrene Polymerization at Unstable Operating Points. *Comput. Chem. Eng.* 14, 481–494. doi:10.1016/0098-1354(90)87022-h

Hoerl, A. E., and Kennard, R. W. (2000). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics* 42, 80–86. doi:10.1080/00401706.2000.10485983

Hou, Z., and Jin, S. (2011). A Novel Data-Driven Control Approach for a Class of Discrete-Time Nonlinear Systems. *IEEE Trans. Contr. Syst. Technol.* 19, 1549–1558. doi:10.1109/tcst.2010.2093136

Hou, Z., and Jin, S. (2013). *Model Free Adaptive controlTheory and Applications*. CRC Press,Taylor & Francis Group.

Kaspar, M. H., and Harmon Ray, W. (1993). Dynamic PLS Modelling for Process Control. *Chem. Eng. Sci.* 48, 3447–3461. doi:10.1016/0009-2509(93)85001-6

Kou, P., Feng, Y., Liang, D., and Gao, L. (2019). A Model Predictive Control Approach for Matching Uncertain Wind Generation with PEV Charging Demand in a Microgrid. *Int. J. Electr. Power Energ. Syst.* 105, 488–499. doi:10.1016/j.ijepes.2018.08.026

Li, K., Luo, H., Yang, C., and Yin, S. (2020). Subspace-aided Closed-Loop System Identification with Application to DC Motor System. *IEEE Trans. Ind. Electron.* 67, 2304–2313. doi:10.1109/tie.2019.2907447

Mayne, D. Q., and Rawlings, J. B. (2001). Correction to "Constrained Model Predictive Control: Stability and Optimality". *Automatica* 37, 483. doi:10.1016/s0005-1098(00)00173-4

Meidanshahi, V., Corbett, B., Adams, T. A., and Mhaskar, P. (2017). Subspace Model Identification and Model Predictive Control Based Cost Analysis of a Semicontinuous Distillation Process. *Comput. Chem. Eng.* 103, 39–57. doi:10.1016/j.compchemeng.2017.03.011

Patwardhan, R. S., Lakshminarayanan, S., and Shah, S. L. (1998). Constrained Nonlinear Mpc Using Hammerstein and Wiener Models: Pls Framework. *Aiche J.* 44, 1611–1622. doi:10.1002/aic.690440713

Qin, S. J., and Badgwell, T. A. (2003). A Survey of Industrial Model Predictive Control Technology. *Control. Eng. Pract.* 11, 733–764. doi:10.1016/s0967-0661(02)00186-7

Rossiter, J. (2003). *Model-based Predictive Control: A Practical Approach*. Boca Raton, Florida): CRC Press.

Thornhill, N. F., Patwardhan, S. C., and Shah, S. L. (2008). A Continuous Stirred Tank Heater Simulation Model with Applications. *J. Process Control.* 18, 347–360. doi:10.1016/j.jprocont.2007.07.006

Wu, X., Shen, J., Li, Y., and Lee, K. Y. (2013). Data-Driven Modeling and Predictive Control for Boiler-Turbine Unit. *IEEE Trans. Energ. Convers.* 28, 470–481. doi:10.1109/tec.2013.2260341

Yi, L. (20162016). "System Modeling and Identification of Unmanned Quad Rotor Air Vehicle Based on Subspace and Pem," in *IEEE Systems And Technologies For Remote Sensing Applications through Unmanned Aerial Systems (STRATUS).* doi:10.1109/stratus.2016.7811137

Yin, S., and Xiao, B. (2017). Tracking Control of Surface Ships with Disturbance and Uncertainties Rejection Capability. *Ieee/asme Trans. Mechatron.* 22, 1154–1162. doi:10.1109/tmech.2016.2618901

Yin, S., Yu, H., Shahnazi, R., and Haghani, A. (2017). Fuzzy Adaptive Tracking Control of Constrained Nonlinear Switched Stochastic Pure-Feedback Systems. *IEEE Trans. Cybern.* 47, 579–588. doi:10.1109/tcyb.2016.2521179

Yu, H., Xie, T., Paszczynski, S., and Wilamowski, B. M. (2011). Advantages of Radial Basis Function Networks for Dynamic System Design*IEEE Trans. Ind. Electron.* 58, 5438–5450. IEEE International Conference on Mechatronics (ICM 2009)APR 14-17, 2009. doi:10.1109/tie.2011.2164773

Yuzgec, U., Becerikli, Y., and Turker, M. (2008). Dynamic Neural-Network-Based Model-Predictive Control of an Industrial Baker's Yeast Drying Process. *IEEE Trans. Neural Netw.* 19, 1231–1242. doi:10.1109/tnn.2008.2000205

Zhang, Y., Xu, D., and Huang, L. (2018). Generalized Multiple-Vector-Based Model Predictive Control for PMSM Drives. *IEEE Trans. Ind. Electron.* 65, 9356–9366. doi:10.1109/tie.2018.2813994

Zhou, P., Dai, P., Song, H., and Chai, T. (2017). Data-driven Recursive Subspace Identification Based Online Modelling for Prediction and Control of Molten Iron Quality in Blast Furnace Ironmaking. *IET Control. Theor. Appl.* 11, 2343–2351. doi:10.1049/iet-cta.2016.1474

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.