Neural Comput & Applic (2018) 29:585–596
https://doi.org/10.1007/s00521-016-2556-5

ORIGINAL ARTICLE

A precise BP neural network-based online model predictive control strategy for die forging hydraulic press machine

Y. C. Lin[1,2,3] · Dong-Dong Chen[1,2] · Ming-Song Chen[1,2] · Xiao-Min Chen[1,2] · Jia Li[1,2]

Received: 29 January 2016 / Accepted: 12 August 2016 / Published online: 6 September 2016

Abstract The time variance and nonlinearity of forging processes pose great challenges to high-quality production. In this study, a one-step-ahead model predictive control (MPC) strategy based on backpropagation (BP) neural network is proposed for the precise forging processes. Two online updated BP neural networks, predictive neural network (PNN) and control neural network (CNN), are developed to accurately control the die forging hydraulic press machine. The PNN and CNN are utilized to predict the output (the velocity of upper die) and determine the input (the oil pressure of driven cylinders), respectively. The weights of neural networks are initially trained offline and then updated online according to an error backpropagation algorithm. In the proposed control strategy, only the input and output are required, which makes the forging process easy to be controlled. In addition, because of the generalized ability and adaptability of neural networks, the proposed predictive controller can well deal with the time variance and nonlinearity of forging process. Two forging experiments demonstrate the feasibility and effectiveness of the proposed strategy. Moreover, comparing the proposed MPC strategy with the traditional MPC approach and PID controller, it can be found that the proposed MPC strategy is the most effective control approach for the practical forging process.

✉ Y. C. Lin
yclin@csu.edu.cn; linyongcheng@163.com

1 School of Mechanical and Electrical Engineering, Central South University, Changsha 410083, China

2 State Key Laboratory of High Performance Complex Manufacturing, Changsha 410083, China

3 Light Alloy Research Institute, Central South University, Changsha 410083, China

Keywords BP neural networks · Model predictive control · Forging process

1 Introduction

The forging technology has been widely used to manufacture the critical components with high performance in modern industries [1–3]. The hydraulic press machine (HPM), which provides the forging force to shape forgings, is one of key equipments in forging technology [2–5]. The diagram of a typical forging process is shown in Fig. 1, where the upper die of HPM is driven by three driving cylinders to make the billet deformed. The cylinders are driven by the corresponding hydraulic system, including pumps, valves, and pipes. In order to guarantee the quality of forgings, the velocity and position of upper die must be accurately predicted and controlled by servo valves of hydraulic system.

In the practical forging process, the shape of forging is often irregular. Thus, the deformation force is nonuniform. Moreover, the deformation behaviors of billet are very complex and time-variant, such as the complex microstructural evolution [6–10] and irregular metal flow [11–15]. In addition, the driving system of the HPM is strongly nonlinear [16, 17]. The coupling between the mechanical and hydraulic systems easily makes the forging process complex and nonlinear. Therefore, due to the time variance and nonlinearity of forging process, the accurate control of HPM is a great challenge for the precise forging process [4, 18]. In the past, some control strategies have been developed to control HPM. The PI control [19], one of the traditional control approaches, is widely applied in forging process. Also, the iterative learning control [20] and sliding mode control [21] are often used to control the

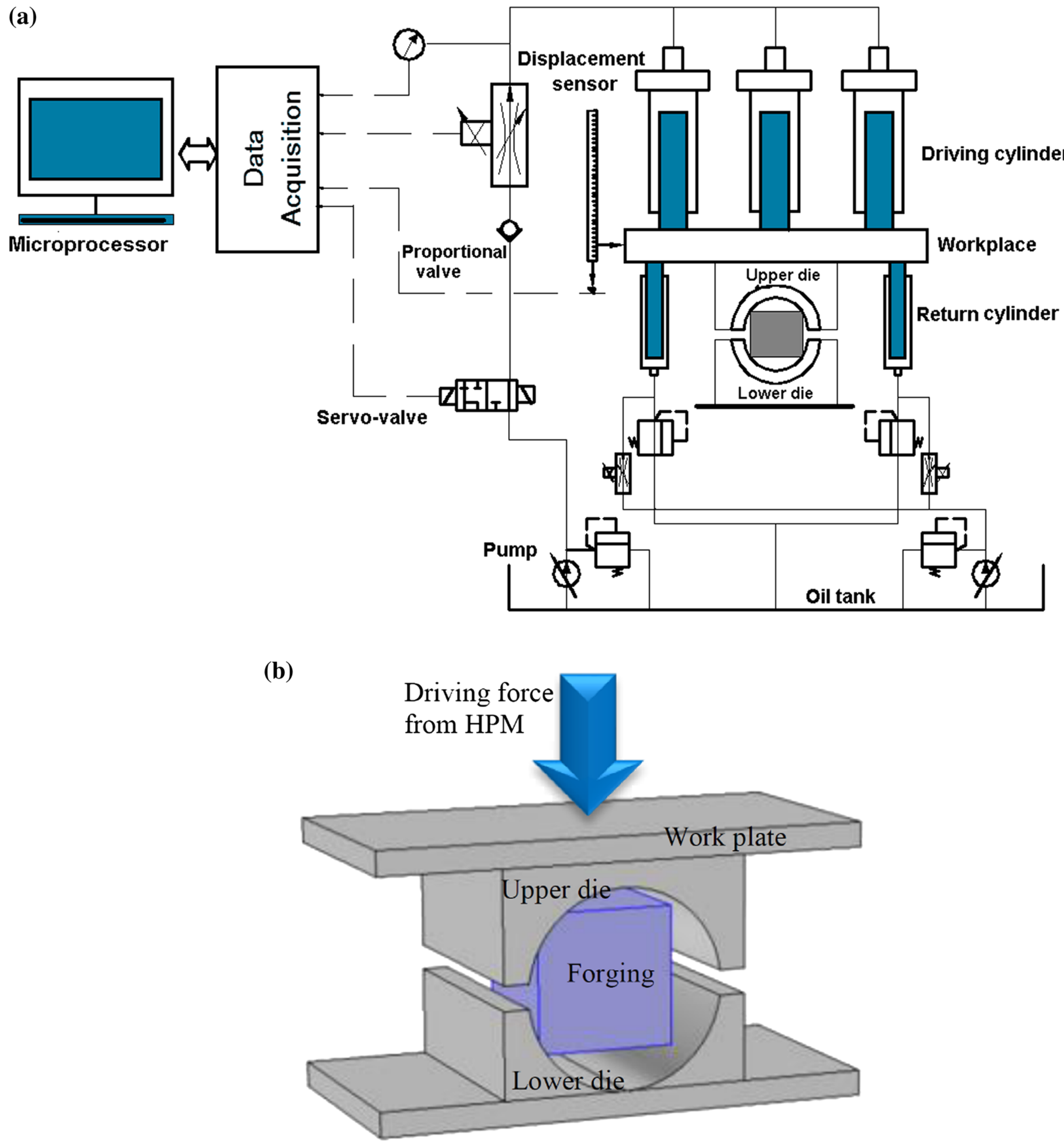

Fig. 1 Diagram of the forging process: **a** diagram of HPM and control system; **b** forging process

HPM. However, these control strategies simplify the forging process to be a linear model which ignores the influence of unknown disturbances. Therefore, these control methods cannot meet the requirements of the complex nonlinear forging process. To better control forging process, new control strategies or intelligent control methods have been developed in recent years. To reduce the difficulty of modeling and controlling forging process, the system-decomposition-based multi-level control method [4] and multi-domain modeling method [22] were proposed to decompose the complex nonlinear system into a series of linear systems or partition the whole operation region into some local regions. Meanwhile, due to their superior data processing ability, the intelligent control methods, including neural network method [23–25], fuzzy method [17, 26–28], and support vector machine method [29–31], have been gradually applied in the modeling and controlling of different industrial processes. However, these methods are hard to be achieved online. Therefore, it is necessary to develop a precise online control strategy for the time-variant and nonlinear forging process.

Model predictive control (MPC) presents a dramatic advance in theory and application of modern automatic control [32–35]. Now, MPC has been widely applied in many industrial fields, such as manufacturing industry [36, 37], chemical control engineering [38–41]. One-step-ahead MPC is one of the simplest MPC strategies. Due to the simplicity and effectiveness, one-step-ahead MPC has been successfully used in the predictive control of wind power [42], hydroturbine governor [43], and uninterruptible power supply system [44]. In recent years, the neural network-based model predictive control (NNMPC) strategy has been proposed and widely applied in industrial process control [45–47]. Up to now, one-step-ahead NNMPC has not been applied in the predictive control of HPM in the forging process.

In this study, a backpropagation (BP) neural network-based online MPC strategy is firstly developed to control HPM in the time-variant and nonlinear forging process. The developed control strategy employs two neural networks, predictive neural network (PNN) and control neural network (CNN), to simplify the traditional MPC. The PNN is trained to predict the output of system, while the CNN is updated to determine the optimal input. Finally, the feasibility and effectivity of the developed control strategy are verified by two practical forging experiments on 4000T HPM. Moreover, the performances of the proposed and traditional MPC approaches and PID controller are compared and analyzed.

2 BP neural network-based online model predictive control

2.1 Model predictive control

Briefly, the flowchart of traditional MPC is shown in Fig. 2, where u, y, y_d, y_r, y_m, and y_p denote the input, output, set value, reference, predictive output, and revised predictive output, respectively. MPC is mainly composed of predictive model, reference target planning, feedback compensation, and rolling optimization [48]. At each control level, the predictive output y_m, which is obtained by predictive model, is used to determine the revised predictive output y_p by feedback compensation. Then, the input u is optimally determined by comparing the revised predictive output y_p

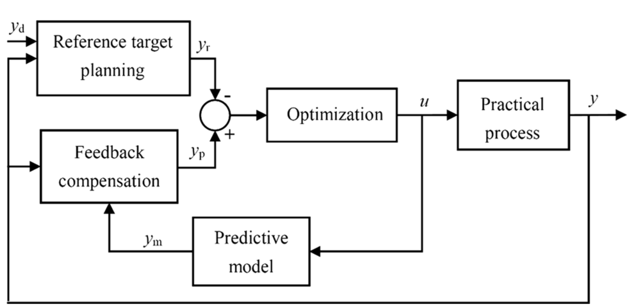

Fig. 2 Flowchart of traditional MPC (u, y, y_d, y_r, y_m, and y_p denote the input, output, set value, reference, predictive output, and revised predictive output)

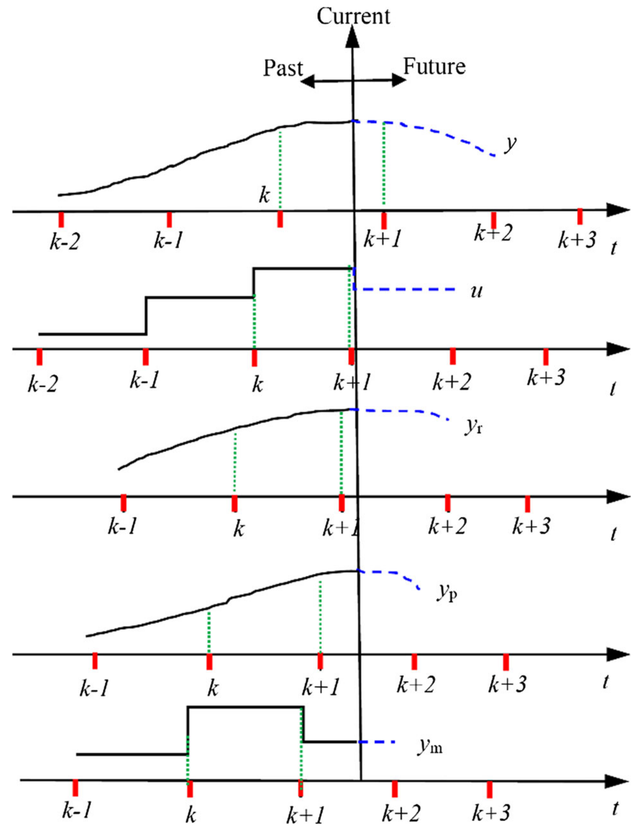

Fig. 3 Sequence for determining the predictive output y_m, revised predictive output y_p, reference y_r, input u, and output y

and reference y_r. The whole process will be online iterative until the output of system reaches the set value y_d.

In this study, the BP neural network-based MPC, which follows the course of traditional MPC, is a one-step-ahead control strategy. The sequence for determining the predictive output y_m, revised predictive output y_p, reference y_r, input u, and output y is shown in Fig. 3. The solid line and dash line represent the past and future processes, respectively. At each control level, the predictive output y_m is firstly determined by the predictive model, and then the revised predictive output y_p, reference y_r, and input u are obtained by the feedback compensation, reference target planning, and rolling optimization, respectively. Finally, the output y is obtained in the process. Meanwhile, the next control level is starting and the new predictive output y_m is predicted by the predictive model.

2.2 Predictive neural network

Although the proposed control strategy follows the structure of traditional MPC, the PNN is used to replace the predictive model in traditional MPC to predict the output of system. Then, the control of forging process can be simplified to some extent. The architecture of PNN is shown in Fig. 4. Here, the PNN has 5 inputs, 11 neurons in the

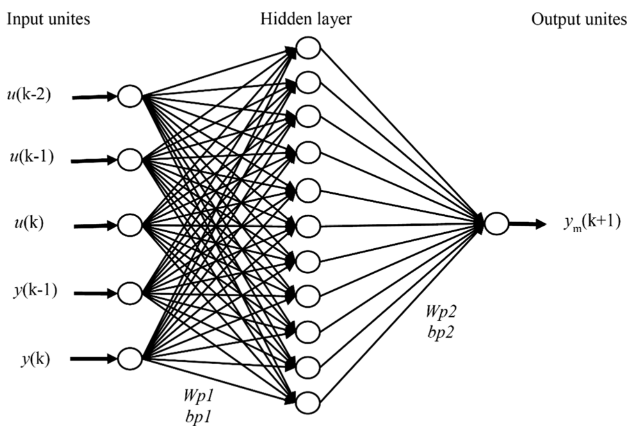

Fig. 4 Architecture of PNN

hidden layer, and 1 output. Additionally, the PNN model can be expressed as:

$$y_{\mathrm{m}}(k+1) = f_{\mathrm{p}}[u(k-2), u(k-1), u(k), y(k-1), y(k)] \tag{1}$$

where $y(k-1)$ and $y(k)$ are the $(k-1)$th and kth practical outputs of the system, respectively. $u(k-2)$, $u(k-1)$, and $u(k)$ are the $(k-2)$th, $(k-1)$th, and kth inputs for the system, respectively. $y_{\mathrm{m}}(k+1)$ denotes the $(k+1)$th predictive output of the system.

In the PNN, $\mathrm{Wp1} \in R^{11\times5}$ and $\mathrm{Wp2} \in R^{1\times11}$ denote the input–hidden and hidden–output connection weight matrices, respectively. Also, $\mathrm{bp1} \in R^{11\times1}$ and $\mathrm{bp2} \in R^{1\times1}$ represent the input–hidden and hidden–output bias terms. The activation function used in PNN is shown as:

$$g(x) = \frac{1}{1+e^{-x}} \tag{2}$$

Then, the state and output equations of PNN are

$$\mathrm{np1}_i = \sum_{j=1}^{5} Wp1_{ij}U_j + \mathrm{bp1}_i \quad i = 1, 2, \cdots 11 \tag{3}$$

$$\mathrm{hp1}_i = g(\mathrm{np1}_i) \quad i = 1, 2, \cdots 11 \tag{4}$$

$$\mathrm{np2} = \sum_{i=1}^{11} \mathrm{Wp2}_i \mathrm{hp1}_i + \mathrm{bp2} \tag{5}$$

$$y_m(k+1) = g(\mathrm{np2}) \tag{6}$$

where U represents the inputs of the PNN $[u(k-2), u(k-1), u(k), y(k-1), y(k)]$; np1 and hp1 denote the input and output of the hidden layer, respectively; np2 and y_{m} denote the input and output of the output layer, respectively.

2.3 Control neural network

In the proposed control strategy, the CNN is used to generate the optimal control signal, which is simpler and lower time-consuming compared to the traditional rolling optimization method. The architecture of CNN is shown in Fig. 5. Here, the CNN has 4 inputs, 9 neurons in the hidden layer, and 1 output. Then, the CNN model can be expressed as:

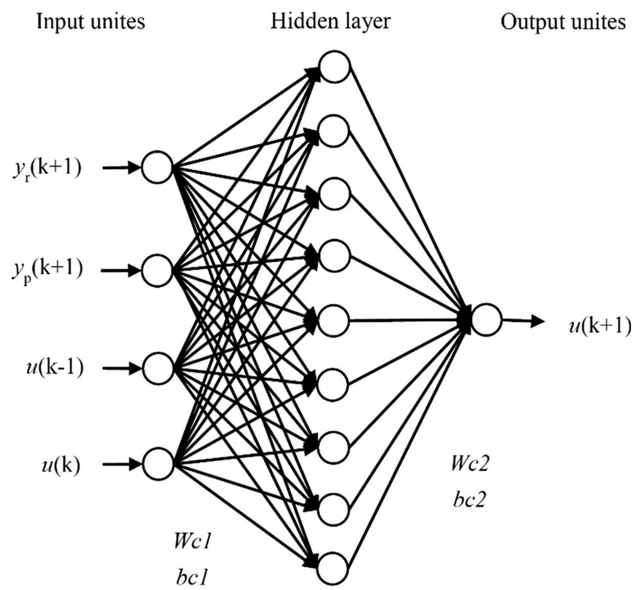

Fig. 5 Architecture of CNN

$$u(k+1) = f_{\mathrm{c}}[y_{\mathrm{r}}(k+1), y_{\mathrm{p}}(k+1), u(k-1), u(k)] \tag{7}$$

where $u(k-1)$, $u(k)$, and $u(k+1)$ are the $(k-1)$th, kth, and $(k+1)$th practical inputs of the system, respectively. $y_r(k+1)$ and $y_p(k+1)$ denote the $(k+1)$th reference and revised predictive output of the system, respectively.

In the CNN, $\mathrm{Wc1} \in R^{9\times4}$ and $\mathrm{Wc2} \in R^{1\times9}$ denote the input–hidden and hidden–output connection weight matrices, respectively. Also, $\mathrm{bc1} \in R^{9\times1}$ and $\mathrm{bc2} \in R^{1\times1}$ represent the input–hidden and hidden–output bias terms. The activation function used in CNN is the same as that used in PNN.

So, the state and output equations of CNN are

$$\mathrm{nc1}_i = \sum_{j=1}^{4} \mathrm{Wc1}_{ij}Y_j + \mathrm{bc1}_i \quad i = 1, 2, \cdots 9 \tag{8}$$

$$\mathrm{hc1}_i = g(\mathrm{nc1}_i) \quad i = 1, 2, \cdots 9 \tag{9}$$

$$\mathrm{nc2} = \sum_{i=1}^{9} \mathrm{Wc2}_i \mathrm{hc1}_i + \mathrm{bc2} \tag{10}$$

$$u(k+1) = g(\mathrm{nc2}) \tag{11}$$

where Y represents the inputs of CNN $[y_r(k+1), y_p(k+1), u(k-1), u(k)]$, and nc1, hc1, nc2, and $u(k+1)$ denote the input of hidden layer, the output of hidden layer, the input of output layer, and the output of output layer, respectively.

2.4 Online update of PNN

According to the procedure of MPC, when the (k + 1)th practical output of system, i.e., $y(k+1)$, is observed, the weight matrices and bias terms of the PNN can be updated online to predict the next time predictive output of system. For the PNN, the cost function can be defined as:

$$E = \frac{1}{2}[y(k+1) - y_m(k+1)]^2 \tag{12}$$

According to the backpropagation algorithm, the weights of PNN can be updated as:

$$w_i(k+1) = w_i(k) + \Delta w_i(k) \tag{13}$$

$$\Delta w_i(k) = -\eta \frac{\partial E(k)}{\partial w_i(k)} \tag{14}$$

where η represents the learning rate.

2.4.1 Update of hidden–output weights

The weight-updating formula of hidden–output can be expressed as:

$$\Delta \mathrm{Wp2}_i = -\eta \frac{\partial E}{\partial \mathrm{Wp2}_i} = -\eta \frac{\partial E}{\partial y_m(k+1)} \frac{\partial y_m(k+1)}{\partial \mathrm{np2}} \frac{\partial \mathrm{np2}}{\partial \mathrm{Wp2}_i} \tag{15}$$

where $\frac{\partial E}{\partial y_m(k+1)} = -[y(k+1) - y_m(k+1)]$, $\frac{\partial y_m(k+1)}{\partial \mathrm{np2}} = y_m(k+1) \times (1 - y_m(k+1))$, $\frac{\partial \mathrm{np2}}{\partial \mathrm{Wp2}_i} = \mathrm{hp1}_i$

So, $\Delta \mathrm{Wp2}_i$ can be expressed by,

$$\begin{aligned}\Delta \mathrm{Wp2}_i = \eta \times [y(k+1) - y_m(k+1)] \times y_m(k+1) \\ \times (1 - y_m(k+1)) \times \mathrm{hp1}_i\end{aligned} \tag{16}$$

Similarly, the bias-term-updating formula of hidden–output can be expressed as:

$$\Delta \mathrm{bp2} = -\eta \frac{\partial E}{\partial \mathrm{bp2}} = -\eta \frac{\partial E}{\partial y_m(k+1)} \frac{\partial y_m(k+1)}{\partial \mathrm{np2}} \frac{\partial \mathrm{np2}}{\partial \mathrm{bp2}} \tag{17}$$

where $\frac{\partial \mathrm{np2}}{\partial \mathrm{bp2}} = 1$.

Therefore, $\Delta \mathrm{bp2}$ can be calculated as,

$$\begin{aligned}\Delta \mathrm{bp2} = \eta \times [y(k+1) - y_m(k+1)] \times y_m(k+1) \\ \times (1 - y_m(k+1))\end{aligned} \tag{18}$$

2.4.2 Update of input–hidden weights

The weight-updating formula of input–hidden can be expressed as:

$$\begin{aligned}\Delta \mathrm{Wp1}_{ij} &= -\eta \frac{\partial E}{\partial \mathrm{Wp1}_{ij}} \\ &= -\eta \frac{\partial E}{\partial y_m(k+1)} \frac{\partial y_m(k+1)}{\partial \mathrm{np2}} \frac{\partial \mathrm{np2}}{\partial \mathrm{hp1}_i} \frac{\partial \mathrm{hp1}_i}{\partial \mathrm{np1}_i} \frac{\partial \mathrm{np1}_i}{\partial \mathrm{Wp1}_{ij}}\end{aligned} \tag{19}$$

where $\frac{\partial \mathrm{np2}}{\partial \mathrm{hp1}_i} = \mathrm{Wp2}_i$, $\frac{\partial \mathrm{hp1}_i}{\partial \mathrm{np1}_i} = \mathrm{hp1}_i \times (1 - \mathrm{hp1}_i)$, $\frac{\partial \mathrm{np1}_i}{\partial \mathrm{Wp1}_{ij}} = U_j$.

So, $\Delta \mathrm{Wp1}_{ij}$ can be expressed by,

$$\begin{aligned}\Delta \mathrm{Wp1}_{ij} = \eta \times [y(k+1) - y_m(k+1)] \times y_m(k+1) \\ \times (1 - y_m(k+1)) \times \mathrm{Wp2}_i \times \mathrm{hp1}_i \times (1 - \mathrm{hp1}_i) \times U_j\end{aligned} \tag{20}$$

Similarly, the bias-term-updating formula of input–hidden can be expressed as:

$$\begin{aligned}\Delta \mathrm{bp1}_i &= -\eta \frac{\partial E}{\partial \mathrm{bp1}_i} \\ &= -\eta \frac{\partial E}{\partial y_m(k+1)} \frac{\partial y_m(k+1)}{\partial \mathrm{np2}} \frac{\partial \mathrm{np2}}{\partial \mathrm{hp1}_i} \frac{\partial \mathrm{hp1}_i}{\partial \mathrm{np1}_i} \frac{\partial \mathrm{np1}_i}{\partial \mathrm{bp1}_i}\end{aligned} \tag{21}$$

where $\frac{\partial \mathrm{np1}_i}{\partial \mathrm{bp1}_i} = 1$.

Therefore, $\Delta \mathrm{bp1}_i$ can be calculated as,

$$\begin{aligned}\Delta \mathrm{bp1}_i = \eta \times [y(k+1) - y_m(k+1)] \times y_m(k+1) \\ \times (1 - y_m(k+1)) \times \mathrm{Wp2}_i \times \mathrm{hp1}_i \times (1 - \mathrm{hp1}_i)\end{aligned} \tag{22}$$

2.5 Online update of CNN

In the traditional MPC, the control signal is always obtained by rolling optimization method in real time. Then, the weight matrices and bias terms of CNN should also be updated online to determine the next time input of system. For the CNN, according to the objective function of rolling optimization method, the cost function is defined as:

$$J = \frac{1}{2}\left[y_r(k+1) - y_p(k+1)\right]^2 \tag{23}$$

According to the backpropagation algorithm, the weights of CNN can be updated as:

$$w_i(k+1) = w_i(k) + \Delta w_i(k) \tag{24}$$

$$\Delta w_i(k) = -\eta \frac{\partial J(k)}{\partial w_i(k)} \tag{25}$$

2.5.1 Update of hidden–output weights

The weight-updating formula of hidden–output can be expressed as:

$$\begin{aligned}\Delta \mathrm{Wc2}_i &= -\eta \frac{\partial J}{\partial \mathrm{Wc2}_i} \\ &= -\eta \frac{\partial J}{\partial y_p(k+1)} \frac{\partial y_p(k+1)}{\partial y_m(k+1)} \frac{\partial y_m(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial \mathrm{Wc2}_i}\end{aligned} \tag{26}$$

where $\frac{\partial J}{\partial y_p(k+1)} = -\left[y_r(k+1) - y_p(k+1)\right]$, $\frac{\partial y_p(k+1)}{\partial y_m(k+1)} = 1$,

$$\frac{\partial y_m(k+1)}{\partial u(k)} = \frac{\partial y_m(k+1)}{\partial \mathrm{np2}}\frac{\partial \mathrm{np2}}{\partial \mathrm{hp1}}\frac{\partial \mathrm{hp1}}{\partial \mathrm{np1}}\frac{\partial \mathrm{np1}}{\partial u(k)} = y_m(k+1) \times (1 - y_m(k+1)) \times \mathrm{Wp2} \times \mathrm{hp1} \times (1 - \mathrm{hp1}) \times \mathrm{Wp1}_{i3}$$

$$\frac{\partial u(k)}{\partial \mathrm{Wc2}_i} = \frac{\partial u(k)}{\partial \mathrm{nc2}}\frac{\partial \mathrm{nc2}}{\partial \mathrm{Wc2}_i} = u(k) \times (1 - u(k)) \cdot \mathrm{hc1}_i$$

So, $\Delta\mathrm{Wc2}_i$ can be expressed by,

$$\Delta\mathrm{Wc2}_i = \eta \times [y_r(k+1) - y_p(k+1)] \times y_m(k+1) \times (1 - y_m(k+1)) \times \mathrm{Wp2} \cdot \mathrm{hp1} \times (1 - \mathrm{hp1}) \cdot \mathrm{Wp1}_{i3} \cdot u(k) \cdot (1 - u(k)) \cdot \mathrm{hc1}_i \tag{27}$$

Similarly, the bias-term-updating formula of hidden–output can be expressed as:

$$\Delta\mathrm{bc2} = -\eta\frac{\partial J}{\partial \mathrm{bc2}} = -\eta\frac{\partial J}{\partial y_p(k+1)}\frac{\partial y_p(k+1)}{\partial y_m(k+1)}\frac{\partial y_m(k+1)}{\partial u(k)}\frac{\partial u(k)}{\partial \mathrm{bc2}} \tag{28}$$

where $\frac{\partial u(k)}{\partial \mathrm{bc2}} = \frac{\partial u(k)}{\partial \mathrm{nc2}}\frac{\partial \mathrm{nc2}}{\partial \mathrm{bc2}} = u(k) \times (1 - u(k)) \cdot 1$.

Therefore, $\Delta\mathrm{bc2}$ can be calculated as,

$$\Delta\mathrm{bc2} = \eta \times \left[y_r(k+1) - y_p(k+1)\right] \times y_m(k+1) \times (1 - y_m(k+1)) \times \mathrm{Wp2} \times \mathrm{hp1} \times (1 - \mathrm{hp1}) \times \mathrm{Wp1}_{i3} \times u(k) \times (1 - u(k)) \tag{29}$$

2.5.2 *Update of input–hidden weights*

The weight-updating formula of input–hidden can be expressed as:

$$\Delta\mathrm{Wc1}_{ij} = -\eta\frac{\partial J}{\partial \mathrm{Wc1}_{ij}} = -\eta\frac{\partial J}{\partial y_p(k+1)}\frac{\partial y_p(k+1)}{\partial y_m(k+1)}\frac{\partial y_m(k+1)}{\partial u(k)}\frac{\partial u(k)}{\partial \mathrm{Wc1}_{ij}} \tag{30}$$

where $\frac{\partial u(k)}{\partial \mathrm{Wc1}_{ij}} = \frac{\partial u(k)}{\partial \mathrm{nc2}}\frac{\partial \mathrm{nc2}}{\partial \mathrm{hc1}_i}\frac{\partial \mathrm{hc1}_i}{\partial \mathrm{nc1}_i}\frac{\partial \mathrm{nc1}_i}{\partial \mathrm{Wc1}_{ij}} = u(k) \times (1 - u(k)) \times \mathrm{Wc2}_i \times \mathrm{hc1}_i \times (1 - \mathrm{hc1}_i) \times Y_j$.

So, $\Delta\mathrm{Wc1}_{ij}$ can be expressed by,

$$\Delta\mathrm{Wc1}_{ij} = \eta \times [y_r(k+1) - y_p(k+1)] \times y_m(k+1) \times (1 - y_m(k+1)) \times \mathrm{Wp2} \times \mathrm{hp1} \times (1 - \mathrm{hp1}) \times \mathrm{Wp1}_{i2} \times u(k) \times (1 - u(k)) \times \mathrm{Wc2}_i \times \mathrm{hc1}_i \times (1 - \mathrm{hc1}_i) \times Y_j \tag{31}$$

Similarly, the bias-term-updating formula of input–hidden can be expressed as:

$$\Delta\mathrm{bc1}_i = -\eta\frac{\partial J}{\partial \mathrm{bc1}_i} = -\eta\frac{\partial J}{\partial y_p(k+1)}\frac{\partial y_p(k+1)}{\partial y_m(k+1)}\frac{\partial y_m(k+1)}{\partial u(k)}\frac{\partial u(k)}{\partial \mathrm{bc1}_i} \tag{32}$$

where $\frac{\partial u(k)}{\partial \mathrm{bc1}_i} = \frac{\partial u(k)}{\partial \mathrm{nc2}}\frac{\partial \mathrm{nc2}}{\partial \mathrm{hc1}_i}\frac{\partial \mathrm{hc1}_i}{\partial \mathrm{nc1}_i}\frac{\partial \mathrm{nc1}_i}{\partial \mathrm{bc1}_i} = u(k) \times (1 - u(k)) \times \mathrm{Wc2}_i \times \mathrm{hc1}_i \times (1 - \mathrm{hc1}_i) \cdot 1$.

Therefore, $\Delta\mathrm{bc1}_i$ can be calculated as,

$$\Delta\mathrm{bc1}_i = \eta \cdot [y_r(k+1) - y_p(k+1)] \times y_m(k+1) \times (1 - y_m(k+1)) \times \mathrm{Wp2} \times \mathrm{hp1} \times (1 - \mathrm{hp1}) \times \mathrm{Wp1}_{i2} \times u(k) \times (1 - u(k)) \times \mathrm{Wc2}_i \times \mathrm{hc1}_i \times (1 - \mathrm{hc1}_i) \tag{33}$$

2.6 Implementation of BP neural network-based online MPC strategy

The BP neural network-based online MPC strategy follows the course of traditional MPC. However, two improvements in the proposed control strategy have been achieved, i.e., simplifying the controller and reducing time consumption, which can satisfy the online and accurate control of the time-variant and nonlinear forging process. The flowchart of BP neural network-based MPC strategy is shown in Fig. 6. It can be summarized as follows:

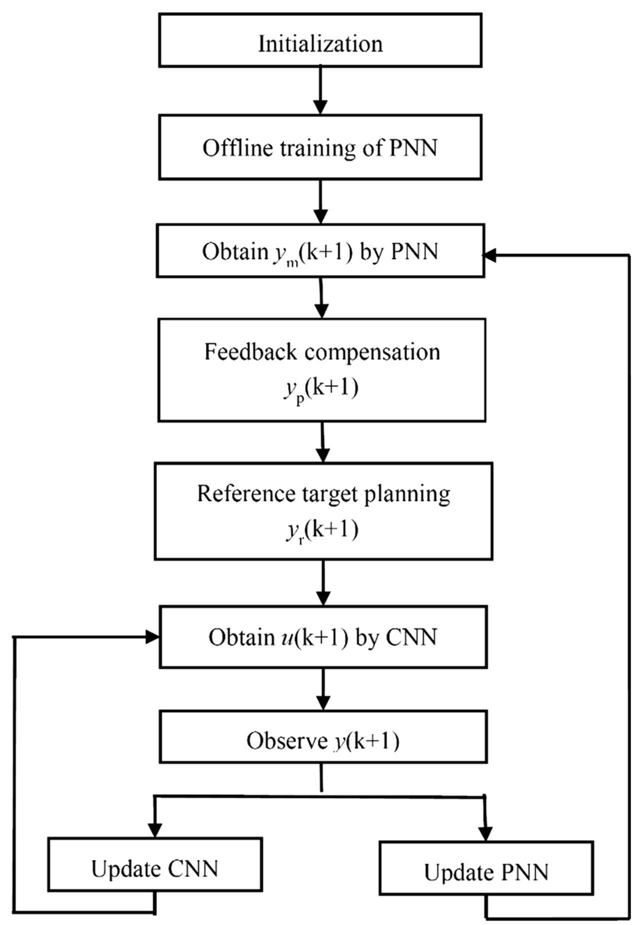

Fig. 6 Flowchart of BP neural network-based MPC strategy

Step 1 Initialization of model parameters. Initialize the learning rate η, softness parameter α, and initial input $u(1)$.

Step 2 Offline training of PNN. At the first control level, the PNN should be trained to determine the initial predictive output $y_{\mathrm{m}}(1)$.

Step 3 Obtain $y_{\mathrm{m}}(k+1)$ by PNN. In the whole process, the predictive output $y_{\mathrm{m}}(k+1)$ can be iteratively determined by PNN.

Step 4 Feedback compensation. Due to the nonlinearity, time-variant characteristic, and unknown disturbances of practical forging process, the predictive output $y_{\mathrm{m}}(k+1)$ is hard to match the actual output $y(k+1)$. It may be make the control process unstable. The predictive error between the actual and predictive outputs should be used to revise the predictive output. The formula can be given by:

$$y_p(k+1) = y_m(k+1) + h(y(k) - y_m(k)) \tag{34}$$

where h is the weight coefficient. Generally, $h = 1$.

Step 5 Reference target planning. In each control horizon, the reference target $y_r(k+1)$ should be presented before the practical output $y(k+1)$ is observed. The reference target planning formula is expressed as:

$$y_r(k+1) = \alpha y(k) + (1-\alpha) y_d \tag{35}$$

where y_{d} is the value of goal setting and α $(0 < \alpha < 1)$ is the softness parameter. If α is large, the predictive control is strongly robust. However, it will lead to the slow response speed of system. Otherwise, the response speed of system will become fast, but it will cause the system overshoot and oscillation. In this study, the softness parameter α is set as 0.1 in order to quickly reach the goal.

Step 6 Obtain $u(k+1)$ by CNN. The initial input $u(1)$ is determined by initialization. However, in the later process, the input $u(k+1)$ is iteratively obtained by CNN.

Step 7 Observe $y(k+1)$. The practical output $y(k+1)$ is utilized in the next control horizon.

Step 8 Update the weights of neural networks. The methods to update online the weights of PNN and CNN are presented in Sect. 2.4 and 2.5, respectively.

Step 9 Return to Step 3 to carry out the next time step prediction and control.

Remark 1 Generally, for the traditional MPC, the rolling optimization mainly deals with the optimal problem with constraints, and the quadratic programming (QP) method is often used to solve the problem. However, in this study, the CNN is used to replace QP method to obtain the optimal input for the system. This is because the QP method costs much more time than the CNN. Moreover, the constraints for this problem are satisfied naturally when training the neural networks. So, the CNN not only simplifies the complex problem to be an input–output question, but also reduces the time consumption.

3 Experimental verifications

Two experiments were performed on 4000T HPM (as shown in Fig. 7) to confirm the feasibility and effectiveness of the proposed control strategy. The pump station, which can produce the maximal 25 MPa oil pressure, provides power for the entire system. The oil pressures of three cylinders located above the work plate are controlled by servo valves. These servo valves receive control signals from a PLC (SIMATICS7-300), a control panel equipped with a PC, and the data acquisition board for pressure, displacement, and velocity. The pressure sensors (E-ART-6/400, range 0–400 bar) installed at the inlet of the driven cylinders are used to collect pressure data. The displacement sensors (magnetostrictive sensors: RPS 1650M D70 1S1 G8400, resolution 0.001 mm) installed at the vertical columns are used to collect displacement data. The sampling period of all sensors is one second. The practical pressure of driven cylinders and the velocity of upper die are defined as the input and output of the system, respectively. In this study, the first experiment is mainly utilized to validate the feasibility, while the second experiment is used to confirm the effectiveness of the proposed control strategy.

3.1 Experiment 1

The first experiment data are shown in Figs. 8 and 9. In order to validate the feasibility of the proposed control strategy, the input (the pressure of driven cylinders) and the output (the velocity of upper die) are used to train the PNN initially. Then, according to the proposed control strategy, the forging process is simulated using the trained PNN. The values of parameters $\eta = 0.005$, $h = 1$, and $\alpha = 0.1$. The value of goal setting is shown as:

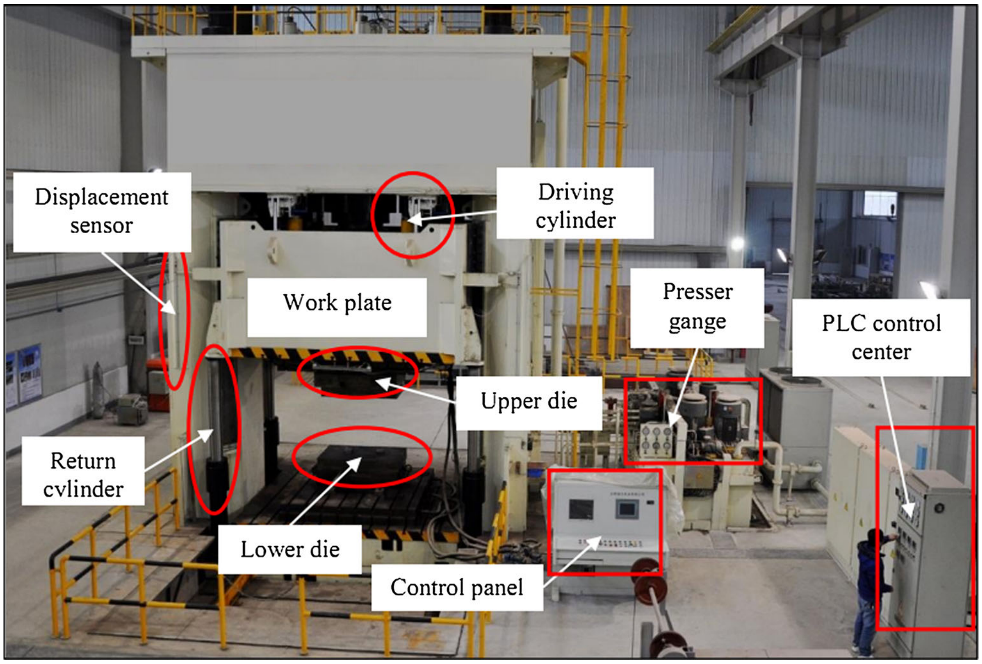

Fig. 7 Practical 4000T HPM

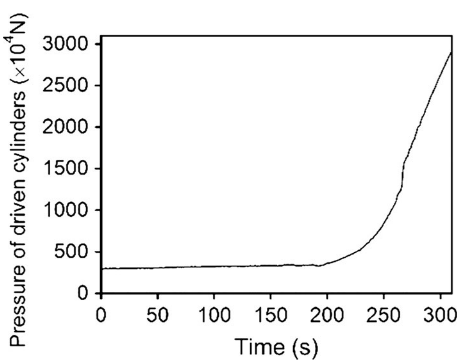

Fig. 8 Input of experiment 1

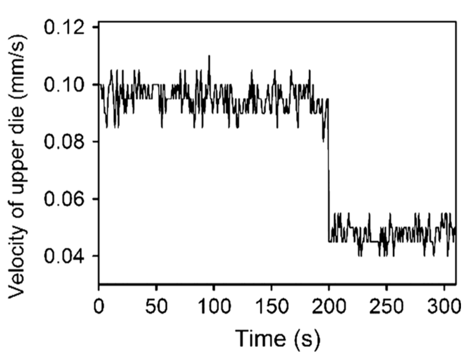

Fig. 9 Output of experiment 1

$$y_d = \begin{cases} 0.1 \text{ mm/s} & 0 < t \leq 200 \text{ s} \\ 0.05 \text{ mm/s} & 200 \text{ s} < t \leq 300 \text{ s} \end{cases} \tag{36}$$

Figure 10 shows the comparison of the real and predictive outputs. It is clear that the predictive output is closely consistent with the real output. Furthermore, the predictive output is more stable than the real output. In addition, the predictive output is closer to the reference than the real output. Due to the predictive characteristic of MPC approach, the control system can timely change the velocity of upper die from 0.1 to 0.05 mm/s. Meanwhile, there is no large oscillation in the change process. Also, the errors between the real and predictive outputs are mainly distributed around 0 except the sudden change point. So, the BP neural network-based MPC strategy is feasible, and it can effectively predict and control the forging process on 4000T HPM.

3.2 Experiment 2

The second experiment is mainly utilized to confirm the effectiveness of the proposed control strategy. Then, the weight matrices and bias terms of PNN and CNN, which were trained in the first experiment, are utilized as the initial weight matrices and bias terms of PNN and CNN for the second experiment. The initial weight matrices and

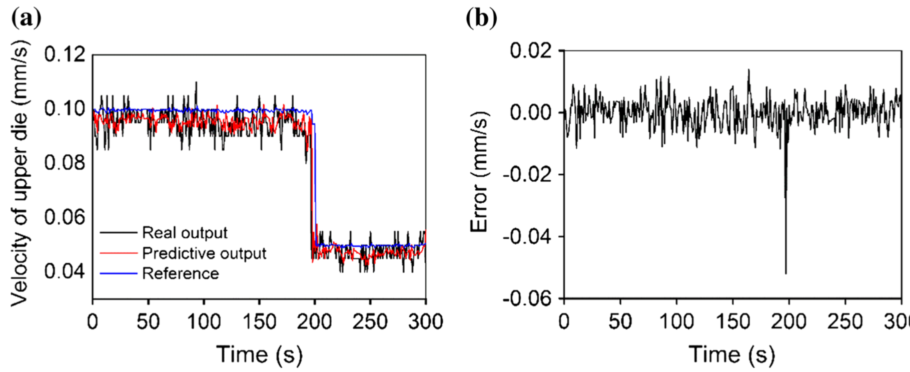

Fig. 10 Experiment 1: **a** the real and predictive outputs; **b** the error between the real and predictive outputs

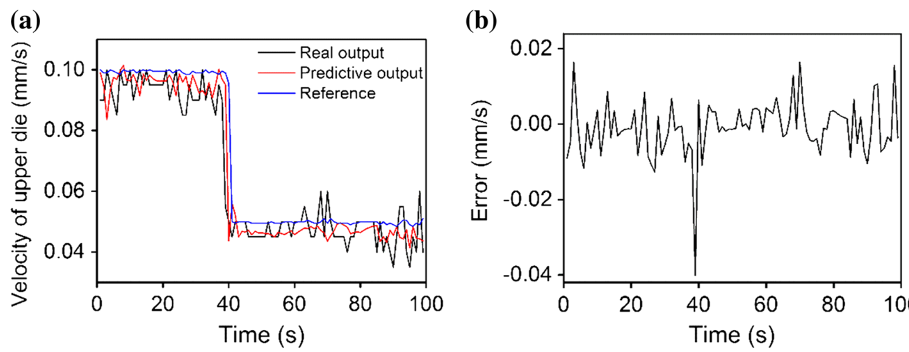

Fig. 11 Experiment 2: **a** the real output vs. predictive output; **b** the error between the real and predictive outputs

bias terms of PNN for the second experiment are shown as:

$$\mathrm{Wp1} = \begin{pmatrix} -152.0825 & 96.1193 & 46.6102 & 1.1991 & -4.1598 \\ 33.3517 & -22.6898 & -78.7413 & -265.7069 & -19.9291 \\ -185.9030 & 116.1661 & 188.3240 & -243.9329 & -105.1505 \\ -39.8815 & 5.1406 & 39.0562 & -11.8511 & 4.6203 \\ 55.9892 & -4.3900 & -53.6012 & 11.5856 & -6.6535 \\ -129.4756 & 74.6679 & 47.0959 & -0.7381 & -2.0258 \\ -242.9268 & -148.3639 & 186.7155 & -83.2754 & 29.2461 \\ -45.5106 & 2.1035 & 46.3862 & -11.4372 & 5.4132 \\ 121.6556 & -54.2411 & -154.8974 & 271.6618 & 114.3594 \\ -9.7958 & 71.5477 & -85.9566 & -13.9572 & -35.4460 \\ -6.0859 & 27.2676 & -42.9174 & -12.8681 & -32.3842 \end{pmatrix}$$

$$\mathrm{bp1} = \begin{pmatrix} 8.5749 \\ 42.5462 \\ 269.0954 \\ 1.5034 \\ -0.7899 \\ 7.2662 \\ 41.1325 \\ 1.2146 \\ -297.5819 \\ 32.8845 \\ 29.8884 \end{pmatrix}, \quad \mathrm{Wp2} = \begin{pmatrix} -55.8785 \\ 0.3617 \\ -62.4863 \\ -103.2579 \\ 70.4858 \\ 65.5863 \\ -0.8769 \\ 174.5329 \\ -62.1414 \\ 36.6314 \\ -40.8731 \end{pmatrix}, \quad \mathrm{bp2} = -16.2396$$

The initial weight matrices and bias terms of CNN for the second experiment are shown as:

$$\mathrm{Wc1} = \begin{pmatrix} -0.4657 & -0.1298 & 2.6725 & 5.0892 \\ -0.9548 & -0.2316 & -0.8359 & 0.5324 \\ 0.5629 & 0.8085 & -3.8450 & -3.6620 \\ 0.0350 & -0.2595 & -2.7894 & -3.2570 \\ -0.9671 & -0.3586 & -0.7828 & -1.5243 \\ -0.8506 & 0.6154 & -0.8587 & -0.1697 \\ 0.8945 & -0.0317 & -0.2684 & -0.8443 \\ 0.1653 & 0.2322 & -0.4686 & 1.5278 \\ 0.3535 & 0.1326 & -1.0371 & -0.4182 \end{pmatrix}$$

$$\mathrm{bc1} = \begin{pmatrix} 0.7020 \\ -0.8743 \\ -0.9597 \\ 6.4460 \\ -0.5119 \\ -1.0556 \\ 0.2901 \\ -0.0048 \\ 1.6241 \end{pmatrix}, \quad \mathrm{Wc2} = \begin{pmatrix} 4.4250 \\ 0.1883 \\ -3.0983 \\ -5.8189 \\ -0.7624 \\ -0.4408 \\ -2.0776 \\ 2.8230 \\ -1.2392 \end{pmatrix}, \quad \mathrm{bc2} = 1.5004$$

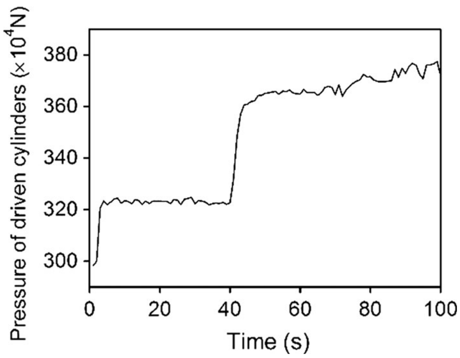

Fig. 12 Input of experiment 2

Also, the values of η, h, and α are the same with those of experiment 1. The value of goal setting is shown as:

$$y_d = \begin{cases} 0.1 \text{ mm/s} & 0 < t \leq 40 \text{ s} \\ 0.05 \text{ mm/s} & 40 \text{ s} < t \leq 100 \text{ s} \end{cases} \quad (37)$$

Figure 11 shows the comparisons between the real and predictive outputs in experiment 2. It is obvious that the predictive output agrees well with the real output, and the real and predictive outputs are all tracking the reference velocity. Moreover, the control system can timely change the velocity of upper die from 0.1 to 0.05 mm/s without large oscillation. Also, the errors between the real and predictive outputs are mainly distributed around 0. The input of experiment 2, as shown in Fig. 12, is obtained by the CNN developed in this control strategy. So, this experiment clearly indicates that the forging process can be effectively predicted and controlled by the proposed control strategy in this study.

3.3 Comparisons and discussions

In this section, the performance of the BP neural network-based online MPC approach is compared with those of the PID controller and traditional MPC approach. In this study, the parameters of PID controller (including proportional, integral, and derivative constants) are 105.32, 732.91, and 213.56, respectively. For the traditional MPC, the prediction and control horizons are 5 and 3, respectively. Figure 13 shows the tracking performance of the proposed MPC approach, PID controller, and traditional MPC approach. It is obvious that the velocity of upper die controlled by the proposed MPC approach is closer to the reference trajectory than those controlled by the PID controller, as well as traditional MPC approach. In particular, the PID controller depicts the higher overshoot than the proposed MPC. As shown in Fig. 13b, the tracking error of the proposed MPC approach is smaller than those of the PID controller and traditional MPC approach. Also, Table 1 shows the root-mean-square errors (RMSEs) of the proposed MPC approach, PID controller, and traditional MPC approach. It is clear that the RMSE of the proposed MPC is the smallest, which implies that the performance of the proposed MPC approach is the best.

The BP neural network-based online MPC approach employs the advantages of neural network, such as generalization capability, adaptation, and fault tolerance property. Due to these advantages, the proposed MPC approach could effectively handle the problems caused by unknown disturbances. This is because the proposed MPC approach is based on the practical experimental data, and the PNN and CNN have contained the influences of unknown disturbances on the inputs and outputs when they are trained. In addition, because the PNN and CNN are updated online, the proposed MPC approach is an online controller, which could timely identify the disturbances and update the weights of neural networks to eliminate the influences caused by unknown disturbances. However, the traditional MPC and PID controller cannot deal with the influences of unknown disturbances. Therefore, the proposed MPC approach is a very effective controller for the practical forging process.

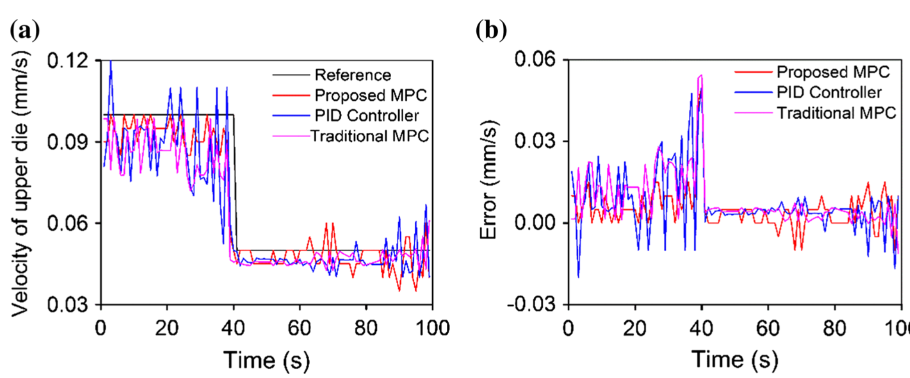

Fig. 13 Comparisons of the proposed MPC, PID controller, and the traditional MPC: **a** the real outputs and reference; **b** the tracking errors

Table 1 RMSEs of the proposed MPC approach, PID controller, and the traditional MPC approach

Method	Proposed MPC	PID controller	Traditional MPC
RMSE	0.0096	0.0136	0.0130

4 Conclusions

A precise BP neural network-based online MPC strategy is proposed to control the time-variant and nonlinear forging process on 4000T HPM in this study. Two BP neural networks, predictive neural network (PNN) and control neural network (CNN), are established to deal with the time-variant and nonlinear forging process. The PNN is used to predict the output of system, and the CNN is used to replace the traditional rolling optimization to determine the input of system. Due to the advantages of neural networks, such as generalization capability, high speed, adaptation, and fault tolerance property, the proposed control strategy has such features as simple structure, fast acting, and adaptability. Moreover, the constraints are satisfied and the unknown disturbances are compensated naturally by the feedback approach. Two forging experiments on 4000T HPM confirm the feasibility and effectiveness of the proposed control strategy. Compared to the traditional MPC approach and PID controller, the proposed MPC approach is the most effective control strategy for the practical forging process.

Acknowledgments This work was supported by the National Natural Science Foundation Council of China (Grant No. 51375502), the National Key Basic Research Program (Grant No. 2013CB035801), the Project of Innovation-driven Plan in Central South University (Grant No. 2016CX008), the Natural Science Foundation for Distinguished Young Scholars of Hunan Province (Grant No. 2016JJ1017), Program of Chang Jiang Scholars of Ministry of Education (No. Q2015140), and the Hunan Provincial Innovation Research Funds for Postgraduate (Grant No. CX2016B045), China.

References

1. Lin YC, Chen XM (2011) A critical review of experimental results and constitutive descriptions for metals and alloys in hot working. Mater Des 32(4):1733–1759
2. Lin YC, Wu XY (2015) A new method for controlling billet temperature during isothermal die forging of a complex superalloy casing. J Mater Eng Perform 24(9):3549–3557
3. Azari A, Poursina M, Poursina D (2014) Radial forging force prediction through MR, ANN, and ANFIS models. Neural Comput Appl 25(3–4):849–858
4. Lu XJ, Huang MH (2012) System-decomposition-based multilevel control for hydraulic press machine. IEEE Trans Ind Electron 59(4):1980–1987
5. Zhu PH, Zhang LH, Zhou R, Chen LH, Yu B, Xie QZ (2012) A novel sensitivity analysis method in structural performance of hydraulic press. Math Probl Eng 2012:1–21
6. Lin YC, Chen MS, Zhong J (2008) Microstructural evolution in 42CrMo steel during compression at elevated temperatures. Mater Lett 62(14):2132–2135
7. Bobbili R, Madhu V (2015) An investigation into hot deformation characteristics and processing maps of high-strength armor steel. J Mater Eng Perform 24(12):4728–4735
8. Chen F, Cui ZS, Chen J (2014) Prediction of microstructural evolution during hot forging. Manuf Rev 1:6
9. Kumar SSS, Raghu T, Bhattacharjee PP, Rao GA, Borah U (2016) Strain rate dependent microstructural evolution during hot deformation of a hot isostatically processed nickel base superalloy. J Alloys Compd 681:28–42
10. Lin YC, Li LT, Xia YC, Jiang YQ (2013) Hot deformation and processing map of a typical Al–Zn–Mg–Cu alloy. J Alloys Compd 550:438–445
11. Kumar SSS, Raghu T, Bhattacharjee PP, Rao GA, Borah U (2015) Constitutive modeling for predicting peak stress characteristics during hot deformation of hot isostatically processed nickel-base superalloy. J Mater Sci 50:6444–6456
12. Chen F, Liu J, Ou HG, Lu B, Cui ZS, Long H (2015) Flow characteristics and intrinsic workability of IN718 superalloy. Mater Sci Eng, A 642:279–287
13. Kotkunde N, Srinivasan S, Krishna G, Gupta AK, Singh SK (2016) Influence of material models on theoretical forming limit diagram prediction for Ti-6Al-4 V alloy under warm condition. Trans Nonferrous Met Soc China 26:736–746
14. Bobbili R, Madhu VGOGIAAK (2014) Neural network modeling to evaluate the dynamic flow stress of high strength armor steels under high strain rate compression. Def Technol 10(4):334–342
15. Wen DX, Lin YC, Li HB, Chen XM, Deng J, Li LT (2014) Hot deformation behavior and processing map of a typical Ni-based superalloy. Mat Sci Eng A 591:183–192
16. Pandian SR, Takemura F, Hayakawa Y, Kawamura S (2002) Pressure observer-controller design for pneumatic cylinder actuators. IEEE-ASME Trans Mech 7(4):490–499
17. Zheng JM, Zhao SD, Wei SG (2009) Application of self-tuning fuzzy PID controller for a SRM direct drive volume control hydraulic press. Control Eng Pract 17(12):1398–1404
18. Lin YC, Chen XM, Wen DX, Chen MS (2014) A physically-based constitutive model for a typical nickel-based superalloy. Comput Mater Sci 83:282–289
19. Cho SJ, Lee JC, Jeon YH, Jeon JW (2009) The development of a position conversion controller for hydraulic press systems. In: IEEE international conference on robotics and biomimetics, pp 2019–2022
20. Zhou YC, Liu SJ, Liu ZW, Huang MH (2008) Hydraulic position holding system of a huge water press based on iterative learning control. Mech Sci Technol Aerosp Eng 27:1130–1133
21. Ho TH, Ahn KK (2012) Speed control of a hydraulic pressure coupling drive using an adaptive fuzzy sliding-mode control. IEEE-ASME Trans Mech 17(5):976–986
22. Lu XJ, Li YB, Huang MH (2013) Operation-region-decomposition-based singular value decomposition/neural network modeling method for complex hydraulic press machines. Ind Eng Chem Res 52(48):17221–17228
23. Dong JR, Zheng CY, Kan GY, Zhao M, Wen J, Yu J (2015) Applying the ensemble artificial neural network-based hybrid data-driven model to daily total load forecasting. Neural Comput Appl 26(3):603–611
24. Das P, Banerjee I (2011) An hybrid detection system of control chart patterns using cascaded SVM and neural network–based detector. Neural Comput Appl 20(2):287–296
25. Awan SM, Aslam M, Khan ZA, Saeed H (2014) An efficient model based on artificial bee colony optimization algorithm with Neural Networks for electric load forecasting. Neural Comput Appl 25(7–8):1967–1978

26. Alberti N, Di Lorenzo R, Micari F, Teti R, Buonadonna P, Manzoni A (1998) Intelligent computation techniques for process planning of cold forging. J Intell Manuf 9(4):353–359
27. Wu X, Shen J, Li Y, Lee KY (2014) Fuzzy modeling and stable model predictive tracking control of large-scale power plants. J Process Contr 24(10):1609–1626
28. Rahmani R, Langeroudi NM, Yousefi R, Mahdian M, Seyedmahmoudian M (2014) Fuzzy logic controller and cascade inverter for direct torque control of IM. Neural Comput Appl 25(3–4):879–888
29. Wong PK, Tam LM, Li K, Vong CM (2010) Engine idle-speed system modelling and control optimization using artificial intelligence. Proc Inst Mech Eng D J Automob Eng 224:55–72
30. Rosillo R, Giner J, de la Fuente D (2014) The effectiveness of the combined use of VIX and support vector machines on the prediction of S&P 500. Neural Comput Appl 25(2):321–332
31. Ciccazzo A, Di Pillo G, Latorre V (2014) Support vector machines for surrogate modeling of electronic circuits. Neural Comput Appl 24(1):69–76
32. Vazquez S, Leon J, Franquelo LG, Rodriguez J, Young H, Marquez A, Zanchetta P (2014) Model predictive control: a review of its applications in power electronics. IEEE Ind Electron Mag 8(1):16–31
33. Dubljevic S (2010) Model predictive control of Kuramoto-Sivashinsky equation with state and input constraints. Chem Eng Sci 65(15):4388–4396
34. Wong PK, Wong HC, Vong CM, Xie ZC, Huang SJ (2016) Model predictive engine air-ratio control using online sequential extreme learning machine. Neural Comput Appl 27(1):79–92
35. Lee JH (2011) Model predictive control: review of the three decades of development. Int J Control Autom 9(3):415–424
36. Camacho EF, Bordons C (2012) Model predictive control in the process industry. Springer, London, pp 3–15
37. Beal CE, Gerdes JC (2013) Model predictive control for vehicle stabilization at the limits of handling. IEEE Trans Control Syst Technol 21(4):1258–1269
38. Hermansson AW, Syafiie S (2015) Model predictive control of pH neutralization processes: a review. Control Eng Pract 45:98–109
39. Ellis M, Christofides PD (2014) Integrating dynamic economic optimization and model predictive control for optimal operation of nonlinear process systems. Control Eng Pract 22:242–251
40. Bumroongsri P, Kheawhom S (2012) An ellipsoidal off-line model predictive control strategy for linear parameter varying systems with applications in chemical processes. Syst Control Lett 61(3):435–442
41. Liu L, Huang B, Dubljevic S (2014) Model predictive control of axial dispersion chemical reactor. J Process Control 24(11): 1671–1690
42. Venayagamoorthy GK, Rohrig K, Erlich I (2012) One step ahead: short-term wind power forecasting and intelligent predictive control based on data analytics. IEEE Power Energy Mag 10(5):70–78
43. Xiao ZH, Meng SL, Lu N, Malik OP (2015) One-step-ahead predictive control for hydroturbine governor. Math Probl Eng. doi:10.1155/2015/382954
44. Lim JS, Park C, Han J, Lee YI (2014) Robust Tracking control of a three-phase DC–AC inverter for UPS applications. IEEE T Ind Electron 61(8):4142–4151
45. Wang T, Gao HJ, Qiu JB (2016) A combined adaptive neural network and nonlinear model predictive control for multirate networked industrial process control. IEEE Trans Neural Net Learn 27(2):416–425
46. Cheng L, Liu WC, Hou ZG, Yu JZ, Tan M (2015) Neural-network-based nonlinear model predictive control for piezoelectric actuators. IEEE Trans Ind Electron 62(12):7717–7727
47. Daosud W, Jariyaboon K, Kittisupakorn P, Hussain MA (2016) Neural network based model predictive control of batch extractive distillation process for improving purity of acetone. Eng J 20(1):47–59
48. Tan GZ, Hao HQ, Wang YD (2011) Real time turning flow estimation based on model predictive control. Inf Technol Artif Intell Conf 1:356–360