Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Model predictive control of servo motor driven constant pump hydraulic system in injection molding process based on neurodynamic optimization*

Yong-gang PENG†1, Jun WANG[2], Wei WEI[1]
([1]College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China)
([2]Department of Mechanical and Automation Engineering, Chinese University of Hong Kong, Hong Kong, China)
†E-mail: pengyg@zju.edu.cn

Received July 9, 2013; Revision accepted Nov. 18, 2013; Crosschecked Jan. 15, 2014

Abstract: In view of the high energy consumption and low response speed of the traditional hydraulic system for an injection molding machine, a servo motor driven constant pump hydraulic system is designed for a precision injection molding process, which uses a servo motor, a constant pump, and a pressure sensor, instead of a common motor, a constant pump, a pressure proportion valve, and a flow proportion valve. A model predictive control strategy based on neurodynamic optimization is proposed to control this new hydraulic system in the injection molding process. Simulation results showed that this control method has good control precision and quick response.

Key words: Model predictive control, Recurrent neural network, Neurodynamic optimization, Injection molding machine
doi:10.1631/jzus.C1300182 **Document code:** A **CLC number:** TP273.1

1 Introduction

The injection molding process can quickly shape a variety of complex configurations at a single time and it is one of the most important ways of polymer forming. Injection molding machines are mostly driven by hydraulic systems. A typical injection molding machine has four components: injection unit, mold clamping unit, hydraulic unit, and control unit. The injection molding process is a cyclic process, which consists of six phases: mold clamping, injection, dwelling, plastication, cooling, and mold opening. The injection molding process consumes high energy. The hydraulic system consumes most of the energy input of the injection molding machine. Thus, control precision and energy-saving are development tendencies of the injection molding industry.

In the traditional hydraulic system of the injection molding machine, a three-phase motor drives a constant pump, and a pressure proportional valve and a flow proportional valve are used to control the pressure and flow of the hydraulic system. The pressure and flow of different phases in the injection molding process are different, especially in the cooling phase when the needed pressure and flow are zero. However, the three-phase motor is always running at the rated state. To save energy, a matching pressure and flow should be given for injection molding process. In this paper we propose a servo motor driven constant pump hydraulic system for the precision injection molding process, which uses a servo motor, a constant pump, and a pressure sensor, instead of a common motor, a constant pump, a pressure proportion valve, and a flow proportion valve. In this hydraulic system the controller can give proper pressure

* Project supported by the National Natural Science Foundation of China (No. 61203299), the Fundamental Research Funds for the Central Universities (No. 2013QNA4021), the Natural Science Foundation of Zhejiang Province (Nos. Y1110135 and LY12F03018), and the Qianjiang Talents Program of Zhejiang Province, China (No. 2013R10047)

and flow according to the need of the injection molding process.

The injection molding process is a periodic complex dynamic process and its accurate control is difficult. Pressure, speed, and temperature are the three most important control variables in the injection molding process and they affect directly the quality of the molded plastic product. Considering the nonlinearity of the injection molding process, Yang and Gao (2000) and Tan *et al*. (2001) presented an adaptive control strategy to control the injection filling velocity of the injection molding process and achieved better tracking performance than using traditional control methods such as the proportional-integral-derivative (PID) controller. Tan and Tang (2002) suggested learning-enhanced proportional integral (PI) control to control the ram velocity of the injection molding process. They used a learning strategy to reject external disturbances, which can compensate for the nonlinearities of the system. Gao *et al.* (2001) addressed a robust iterative learning control method for the injection molding process. By contrast, Gao *et al*. (1996) proposed self-tuning control to dynamically control the cavity pressure at the filling and packing phases of the thermoplastics injection molding process, and Varela (2000) used a self-tuning method to control the injection molding cavity during the filling phase.

Model predictive control (MPC) is an efficient intelligent control technique and has been widely used in petroleum and chemical industry since it arose in the 1960s (Morari and Lee, 1999; Qian and Badgwell, 2003). MPC is a kind of computer control method that uses an explicit process model to predict the future output of an object. The melt temperature object in the injection molding process is a nonlinear, strong coupling, and time delay object, and also a long time constant object. So, MPC was first used to control the temperature of barrel and melt polymer in the injection molding process. For example, to control the melt temperature, Huang *et al*. (1999b) used adaptive generalized predictive control (GPC) and Dubay (2002) used self-optimizing MPC. The optimization process of MPC is time-consuming, which makes MPC difficult to use in the control of the injection molding machine. Pandelidis and Agrawal (1988) first attempted optimal anticipatory control of ram velocity. Then Liao *et al*. (1996) proposed adaptive generalized predictive PI control. Huang *et al*. (1999a) proposed a predictive control method to control the ram velocity of the injection molding process. These methods, however, are based on traditional PID control, and PID here is used to quickly stabilize the system. Only when an effective online optimization tool is ready, can a complete MPC be used in a quick response process such as the injection molding process.

The neural network (NN) is a promising approach to dynamic online optimization. It can process information in parallel, and its convergence rate does not decrease when the size of the optimization problem increases. Hopfield and Tank (1985) and Tank and Hopfield (1986) applied the Hopfield networks to solve a linear program and the traveling salesman problem. Since then many NNs have been proposed to handle optimization problems. Kennedy and Chua (1988) developed an NN to solve the nonlinear programming problem. The proposed NN contains a finite number of parameters, and its equilibrium points correspond to a near optimal solution. Zhang and Constantinides (1992) proposed a Lagrangian network for quadratic programming. Another two typical NNs are the dual networks proposed by Xia and Wang (2001) and the simplified dual network presented by Liu and Wang (2006). These two NNs have low computational complexity and high global convergence speed.

Several studies on NN based MPC have been presented (Draeger *et al*., 1995; Mahadevan and Doyle, 2003; Akesson and Toivonen, 2006; Pan and Wang, 2009). In these works, NNs were successfully applied for modeling and control of dynamic systems. Huang *et al*. (2004) proposed a predictive learning control method for the ram velocity control of the injection molding process based on NN. In this paper, we consider an NN based nonlinear model predictive control (NMPC) scheme to control the injection molding process. A recurrent neural network model is proposed to solve the associated quadratic minimization problem.

2 Structure and model of the new hydraulic system for the injection molding process

In this section a new servo motor driven constant pump hydraulic system is proposed for the injection molding process, and the model based on the proposed hydraulic system is obtained.

A traditional hydraulic system for an injection molding machine has huge power consumption as well as power waste because the input power of the hydraulic system does not match the needed power of the injection molding process. A servo motor driven hydraulic system can flexibly change the flow of the hydraulic system and has the natural power saving advantage. So, we design a servo motor driven constant pump hydraulic system for the injection molding process (Fig. 1).

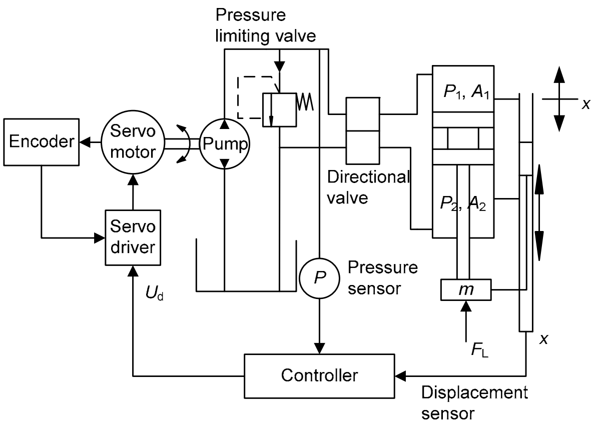

Fig. 1 Structure of our proposed servo motor driven constant pump hydraulic system

U_d is the control input of the servo motor, and F_L is the load pressure

A servo motor drives a constant pump and provides the hydraulic system with the needed flow. The flow of the hydraulic system and the speed of the injection molding process are controlled by the servo driver and the controller. The pressure of the hydraulic system is sampled by a pressure sensor and controlled by the controller in a closed loop. This hydraulic system can supply the needed flow and has little waste.

We can model the proposed hydraulic system mathematically. The servo motor can be described as

$$\frac{d\omega}{dt} = -\frac{\omega}{T_D} + \frac{K_V}{T_D}u, \tag{1}$$

where ω is the speed of the servo motor, T_D is the motor time constant, K_V is the motor torque gain, and u is the input control voltage.

The relationship between the flow of the pump and the speed of the pump can be described as

$$q = C\omega V, \tag{2}$$

where q is the flow of the pump, C is the volumetric efficiency of the pump, and V is the specified discharge of the pump.

Theory and practice indicate that the injection molding process has obvious time-varying and nonlinear characteristics. During the injection phase and holding phase, the melt polymer affects the dynamics of the injection molding process. Rafizadeh *et al.* (1996) modeled this nonlinear process as follows:

$$\dot{x} = v, \tag{3}$$

$$\dot{v} = \frac{1}{M}\left[P_1A_1 - P_2A_2 - 2\pi\eta R^{1-n}(l_0 + x)\left(\frac{(s-1)v}{K_r^{1-s}}\right)^n\right], \tag{4}$$

$$\dot{P}_1 = \frac{\beta_1}{\alpha_{10} + A_1x}(q - A_1v), \tag{5}$$

$$\dot{P}_2 = \frac{\beta_2}{\alpha_{20} - A_2x}(A_2x - Q_p), \tag{6}$$

where x and v are the displacement and speed of the injection hydraulic actuating cylinder, respectively, and P_1 and P_2 are both pressure of the injection hydraulic actuating cylinder. P_2 is also the pressure of the injection nozzle. The meanings of the other symbols are listed in Table 1.

Let $x_1=\omega$, $x_2=x$, $x_3=v$, $x_4=P_1$, and $x_5=P_2$. The following state equations can be obtained for the injection molding process based on the proposed hydraulic system:

$$\dot{x}_1 = -\frac{x_1}{T_D} + \frac{K_V}{T_D}u, \tag{7}$$

$$\dot{x}_2 = x_3, \tag{8}$$

$$\dot{x}_3 = \frac{1}{M}\left[A_1x_4 - A_2x_5 - 2\pi\eta R^{1-n}(l_0 + x_2)\left(\frac{(s-1)x_3}{K_r^{1-s}}\right)^n\right], \tag{9}$$

$$\dot{x}_4 = \frac{\beta_1}{\alpha_{10} + A_1x_2}(CVx_1 - A_1x_3), \tag{10}$$

$$\dot{x}_5 = \frac{\beta_2}{\alpha_{20} - A_2x_2}(A_2x_2 - Q_p). \tag{11}$$

Table 1 The physical meanings of symbols in Eqs. (3)–(6)

Symbol	Physical interpretation
A_1	Cylinder cross-sectional area
A_2	Barrel cross-sectional area
K_r	Ratio of the screw radius to the nozzle radius
l_0	Initial length of the screw
M	Mass of actuator-screw assembly
R	Nozzle radius
α_{10}	Volume of oil on the injection side
α_{20}	Volume of polymer in the barrel
β_1	Hydraulic fluid bulk modulus
β_2	Nozzle bulk modulus
n	Power law index for polymer melt
s	$1/n$
η	Polymer viscosity
Q_p	Polymer melt flow rate

Eqs. (7)–(11) show that the injection molding process based on the proposed hydraulic system is a strong nonlinear system.

3 Problem formulation and neural network optimization

In this section a model predictive control based on a recurrent neural network optimization approach is presented. First, the MPC of the nonlinear affine system is formulated for quadratic programming (QP) problems. Then a neurodynamic optimization approach is introduced to solve the QP problem of MPC.

3.1 Problem formulation

Many nonlinear systems can be described as the following discrete-time nonlinear affine system:

$$\begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{f}(\boldsymbol{x}(k)) + \boldsymbol{g}(\boldsymbol{x}(k))\boldsymbol{u}(k), \\ \boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k), \end{cases} \tag{12}$$

with constraints

$$\begin{cases} \boldsymbol{u}_{\min} \le \boldsymbol{u}(k) \le \boldsymbol{u}_{\max}, \\ \boldsymbol{y}_{\min} \le \boldsymbol{y}(k) \le \boldsymbol{y}_{\max}, \end{cases} \tag{13}$$

where $\boldsymbol{x}(k)\in\mathbb{R}^n$ (note that this n is different from that in Table 1) is the state vector, $\boldsymbol{u}(k)\in\mathbb{R}^m$ is the input vector, $\boldsymbol{y}(k)\in\mathbb{R}^p$ is the output vector, $\boldsymbol{f}(\cdot)$ and $\boldsymbol{g}(\cdot)$ are nonlinear functions, $\boldsymbol{C}\in\mathbb{R}^{p\times n}$, $\boldsymbol{u}_{\min}$ and $\boldsymbol{y}_{\min}$ are vectors of lower bounds, and $\boldsymbol{u}_{\max}$ and $\boldsymbol{y}_{\max}$ are vectors of upper bounds.

MPC is an optimization control method that uses the current state and a predictive model to obtain the optimal input vector by solving an optimization problem. The following optimal criterion is often used in MPC:

$$J(k) = \sum_{j=1}^{N_p} \left\| \boldsymbol{r}(k+j) - \boldsymbol{y}(k+j) \right\|_{\boldsymbol{Q}}^2 + \sum_{j=0}^{N_u-1} \left\| \boldsymbol{u}(k+j) \right\|_{\boldsymbol{R}}^2, \tag{14}$$

where $\boldsymbol{r}(k+j)$ denotes the reference trajectory of the output signal, $\boldsymbol{y}(k+j)$ denotes the predictive output, and $\boldsymbol{u}(k+j)$ denotes the predictive input. N_p ($N_p\geq1$) is the predictive horizon and N_u ($0<N_u\leq N_p$) is the control horizon. $\boldsymbol{Q}$ and $\boldsymbol{R}$ are appropriate weighting matrices.

According to model (12), we have

$$\begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{f}(\boldsymbol{x}(k)) + \boldsymbol{g}(\boldsymbol{x}(k))\boldsymbol{u}(k), \\ \boldsymbol{x}(k+2) = \boldsymbol{f}(\boldsymbol{x}(k+1)) + \boldsymbol{g}(\boldsymbol{x}(k+1))\boldsymbol{u}(k+1), \\ \quad\vdots \\ \boldsymbol{x}(k+N_u) = \boldsymbol{f}(\boldsymbol{x}(k+N_u-1)) \\ \qquad + \boldsymbol{g}(\boldsymbol{x}(k+N_u-1))\boldsymbol{u}(k+N_u-1), \\ \quad\vdots \\ \boldsymbol{x}(k+N_p) = \boldsymbol{f}(\boldsymbol{x}(k+N_p-1)) \\ \qquad + \boldsymbol{g}(\boldsymbol{x}(k+N_p-1))\boldsymbol{u}(k+N_p-1). \end{cases} \tag{15}$$

We define the following vectors:

$$\begin{cases} \tilde{\boldsymbol{y}}(k) = [\boldsymbol{y}(k+1), \boldsymbol{y}(k+2), \cdots, \boldsymbol{y}(k+N_p)]^{\mathrm{T}} \in \mathbb{R}^{N_p\times p}, \\ \tilde{\boldsymbol{u}}(k) = [\boldsymbol{u}(k),\ \boldsymbol{u}(k+1),\ \cdots,\ \boldsymbol{u}(k+N_u)]^{\mathrm{T}} \in \mathbb{R}^{N_u\times m}, \\ \tilde{\boldsymbol{r}}(k) = [\boldsymbol{r}(k+1), \boldsymbol{r}(k+2), \cdots, \boldsymbol{r}(k+N_p)]^{\mathrm{T}} \in \mathbb{R}^{N_p\times p}, \\ \tilde{\boldsymbol{x}}(k) = [\boldsymbol{x}(k+1), \boldsymbol{x}(k+2), \cdots, \boldsymbol{x}(k+N_p)]^{\mathrm{T}} \in \mathbb{R}^{N_p\times n}. \end{cases} \tag{16}$$

The predictive output $\tilde{\boldsymbol{y}}(k)$ can be expressed in the following form:

$$\tilde{\boldsymbol{y}}(k) = \tilde{\boldsymbol{C}}\tilde{\boldsymbol{x}}(k) = \tilde{\boldsymbol{C}}(\tilde{\boldsymbol{f}} + \tilde{\boldsymbol{G}}\tilde{\boldsymbol{u}}), \tag{17}$$

where

$$\tilde{\boldsymbol{C}} = \begin{bmatrix} \boldsymbol{C} & \cdots & \boldsymbol{0} \\ \vdots & & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{C} \end{bmatrix} \in \mathbb{R}^{N_p p\times N_p n},$$

$$\boldsymbol{G}=\begin{bmatrix} \boldsymbol{g}(\boldsymbol{x}(k)) & \boldsymbol{0} & & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{g}(\boldsymbol{x}(k+1)) & & \cdots & \boldsymbol{0} \\ \vdots & & \ddots & & \\ \boldsymbol{0} & \cdots & \boldsymbol{g}(\boldsymbol{x}(k+N_{\mathrm{u}}-1)) & \cdots & \boldsymbol{0} \\ \vdots & & & \ddots & \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & & \boldsymbol{g}(\boldsymbol{x}(k+N_{\mathrm{p}}-1)) \end{bmatrix} \in \mathbb{R}^{N_{\mathrm{p}}n\times N_{\mathrm{u}}m},$$

$$\tilde{\boldsymbol{f}}=\begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}(k)) \\ \boldsymbol{f}(\boldsymbol{x}(k+1)) \\ \vdots \\ \boldsymbol{f}(\boldsymbol{x}(k+N_{\mathrm{p}}-1)) \end{bmatrix} \in \mathbb{R}^{N_{\mathrm{p}}n}.$$

Hence, Eq. (14) becomes

$$\begin{aligned} &\min\left(\left\|\tilde{\boldsymbol{r}}(k)-\tilde{\boldsymbol{C}}\tilde{\boldsymbol{f}}-\tilde{\boldsymbol{C}}\boldsymbol{G}\tilde{\boldsymbol{u}}\right\|_{\boldsymbol{Q}}^{2}+\left\|\tilde{\boldsymbol{u}}(k)\right\|_{\boldsymbol{R}}^{2}\right) \\ &\text{s.t.} \quad \tilde{\boldsymbol{u}}_{\min}\le\tilde{\boldsymbol{u}}(k)\le\tilde{\boldsymbol{u}}_{\max}, \\ &\qquad \tilde{\boldsymbol{y}}_{\min}\le\tilde{\boldsymbol{C}}\tilde{\boldsymbol{f}}+\tilde{\boldsymbol{C}}\boldsymbol{G}\tilde{\boldsymbol{u}}\le\tilde{\boldsymbol{y}}_{\max}. \end{aligned} \tag{18}$$

By defining $\boldsymbol{z}=\tilde{\boldsymbol{u}}(k)$, problem (18) can be rewritten as a QP problem:

$$\begin{aligned} &\min\left(\frac{1}{2}\boldsymbol{z}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{z}+\boldsymbol{c}^{\mathrm{T}}\boldsymbol{z}\right) \\ &\text{s.t.} \quad \boldsymbol{b}_{\mathrm{l}}\le\boldsymbol{E}\boldsymbol{z}\le\boldsymbol{b}_{\mathrm{h}}, \end{aligned} \tag{19}$$

where

$$\boldsymbol{W}=2(\boldsymbol{G}^{\mathrm{T}}\tilde{\boldsymbol{C}}^{\mathrm{T}}\boldsymbol{Q}\tilde{\boldsymbol{C}}\boldsymbol{G}+\boldsymbol{R})\in\mathbb{R}^{N_{\mathrm{u}}m\times N_{\mathrm{u}}m},$$

$$\boldsymbol{c}=-2\boldsymbol{G}^{\mathrm{T}}\tilde{\boldsymbol{C}}^{\mathrm{T}}(\tilde{\boldsymbol{r}}-\tilde{\boldsymbol{C}}\tilde{\boldsymbol{f}})\in\mathbb{R}^{N_{\mathrm{u}}m},$$

$$\boldsymbol{E}=[\tilde{\boldsymbol{I}}\ \ \tilde{\boldsymbol{C}}\boldsymbol{G}]^{\mathrm{T}}\in\mathbb{R}^{(N_{\mathrm{u}}m+N_{\mathrm{p}}p)\times N_{\mathrm{u}}m},$$

$$\tilde{\boldsymbol{I}}=\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{I} \end{bmatrix}\in\mathbb{R}^{N_{\mathrm{u}}m\times N_{\mathrm{u}}m},$$

$$\boldsymbol{b}_{\mathrm{l}}=\begin{bmatrix} \tilde{\boldsymbol{u}}_{\min} \\ \tilde{\boldsymbol{y}}_{\min}-\tilde{\boldsymbol{C}}\tilde{\boldsymbol{f}} \end{bmatrix}\in\mathbb{R}^{N_{\mathrm{u}}m+N_{\mathrm{p}}p},\ \boldsymbol{b}_{\mathrm{h}}=\begin{bmatrix} \tilde{\boldsymbol{u}}_{\max} \\ \tilde{\boldsymbol{y}}_{\max}-\tilde{\boldsymbol{C}}\tilde{\boldsymbol{f}} \end{bmatrix}\in\mathbb{R}^{N_{\mathrm{u}}m+N_{\mathrm{p}}p}.$$

So, the solution of QP problem (19) is the control action output vector $\tilde{\boldsymbol{u}}(k)$. In MPC the first element of $\tilde{\boldsymbol{u}}(k)$ is used to control the objective system.

3.2 Neural network optimization approach

For many real-time solutions to optimization problems, traditional optimization methods will be no more effective when the problem dimensionality increases. Many types of NN have thus been developed for real-time optimization problems (Hopfield and Tank, 1985; Tank and Hopfield, 1986; Kennedy and Chua, 1988; Zhang and Constantinides, 1992; Xia and Wang, 2001; Liu and Wang, 2006). As NN can process information in parallel, its convergence rate does not decrease when the size of the problem increases.

The optimization problem of MPC has been formulated as a QP problem. Many NN models have been proposed to solve QP problems. For example, Liu and Wang (2008) proposed a one-layer NN with a hard limiting activation function. Their approach, however, has difficulty in handling general inequality constraints.

Hu and Wang (2008) proposed a simplified dual neural network for optimal QP problems. Its dynamic equations are described as follows:

State equation:

$$\varepsilon\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t}=-\boldsymbol{E}\boldsymbol{z}+\boldsymbol{h}(\boldsymbol{E}\boldsymbol{z}-\boldsymbol{y}). \tag{20}$$

Output equation:

$$\boldsymbol{z}=\boldsymbol{W}^{-1}(\boldsymbol{E}^{\mathrm{T}}\boldsymbol{y}-\boldsymbol{c}). \tag{21}$$

Herein ε is a positive constant, $\boldsymbol{E}$ is the inequality constraint matrix as given in Eq. (19), $\boldsymbol{h}(\boldsymbol{x})$ is the activation function of the neuron, with

$$h(x_i)=\begin{cases} b_{\mathrm{l}i}, & x_i<b_{\mathrm{l}i}, \\ x_i, & b_{\mathrm{l}i}\le x_i\le b_{\mathrm{h}i}, \\ b_{\mathrm{h}i}, & x_i>b_{\mathrm{h}i}, \end{cases} \tag{22}$$

$\boldsymbol{y}$ is the dual variable of the original optimization problem, and $\boldsymbol{z}$ is the original optimal variable. In this dual neural network, $\boldsymbol{y}$ is the state vector and $\boldsymbol{z}$ is the output vector.

This simplified dual neural network can be used to deal with optimal QP problems with inequality constraints. In this study this neural network is used to solve QP problem (19) in MPC.

The MPC scheme for nonlinear affine systems based on recurrent neural network (20) is summarized as follows:

1. Let k=1, and set MPC parameters prediction horizon N_p, control horizon N_u, sample period τ, weight matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, and neural network parameter ε.

2. Calculate process values $\boldsymbol{G}$, $\tilde{\boldsymbol{C}}$, $\tilde{\boldsymbol{f}}$, $\boldsymbol{W}$, $\boldsymbol{c}$, $\boldsymbol{E}$, and $\boldsymbol{b}$.

3. Solve quadratic problem (19) using neural network (20) to obtain optimal control action $\tilde{\boldsymbol{u}}(k)$.

4. Use the first element $\boldsymbol{u}(k)$ of $\tilde{\boldsymbol{u}}(k)$ to control the system.

5. k=k+1, and go to step 2.

Because of the inherent nature of parallel and distributed information processing in neural networks, the proposed MPC based on neurodynamic optimization has much higher optimization speed than the normal MPC. Moreover, a neural network can be implemented by hardware, with much higher running speed than on an electronic computer.

4 Simulations

Simulations were implemented using the proposed MPC method based neural network optimization. The model (7)–(11) of the injection molding process was discretized as a discrete-time nonlinear affine system using a sampling period τ=0.005 s and Euler approximation:

$$\boldsymbol{f}(k)=\begin{bmatrix} x_1(k)-\dfrac{\tau}{T_{\mathrm{D}}}x_1(k) \\ x_2(k)+\tau x_3(k) \\ x_3(k)+\dfrac{\tau}{M}\left[A_1x_4(k)-A_2x_5(k)-2\pi\eta R^{1-n}\right. \\ \left.\cdot(l_0+x_2(k))\left(\dfrac{(s-1)x_3(k)}{K_{\mathrm{r}}^{1-s}}\right)^n\right] \\ x_4(k)+\dfrac{\tau\beta_1}{\alpha_{10}+A_1x_2(k)}(CVx_1(k)-A_1x_3(k)) \\ x_5(k)+\dfrac{\tau\beta_2}{\alpha_{20}-A_2x_2(k)}(A_2x_2(k)-Q_{\mathrm{p}}) \end{bmatrix}, \tag{23}$$

$$\boldsymbol{g}=[K_{\mathrm{v}}/T_{\mathrm{D}},0,0,0,0]^{\mathrm{T}}. \tag{24}$$

System output is x_3, which is the speed of the injection hydraulic actuating cylinder. $\boldsymbol{C}$=[0, 0, 1, 0, 0]. Parameters used in the simulations are given in Table 2.

Table 2 Values of simulation parameters

Symbol	Value
A_1	3342 mm^2
A_2	201 mm^2
K_r	0.9
l_0	100 mm
M	8.663 μg
R	2 mm
α_{10}	17 045 mm^3
α_{20}	11 678 mm^3
β_1	1120 MPa
β_2	1120 MPa
n	0.822
s	1.216
η	4.6×10^{-3} MPa·s
Q_p	16.67 mm^3/s

The constraints were set as $-500\le u\le 500$, $0\le y_1\le$ 500 mm/s. In the simulation, the initial states were $[0, 0, 0, 0, 0]^{\mathrm{T}}$, the prediction horizon and control horizon were N_p=20 and N_u=5, respectively, the weighting matrices were $\boldsymbol{Q}=\boldsymbol{I}$ and $\boldsymbol{R}=0.1\boldsymbol{I}$, and $\varepsilon=10^4$.

First, a step response with the proposed MPC method and PID method was tested and the results are shown in Fig. 2. PID parameters were adjusted using the Simulink tool in MATLAB.

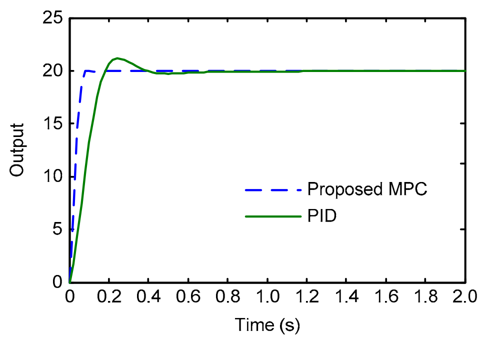

Fig. 2 Step response with the proposed method and PID

Then a varying periodic response was tested with the proposed MPC method and PID method. The set outputs were 20, 40, and 30 mm/s. The output response and optimal input are shown in Figs. 3 and 4. Fig. 3 is the periodic response comparison of PID and MPC, and Fig. 4 shows the periodic response outputs and optimal control outputs of the proposed MPC.

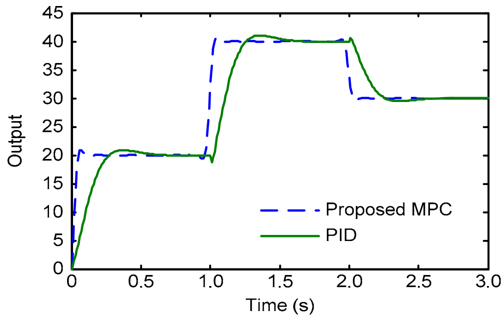

Fig. 3 Periodic response comparison of PID and MPC

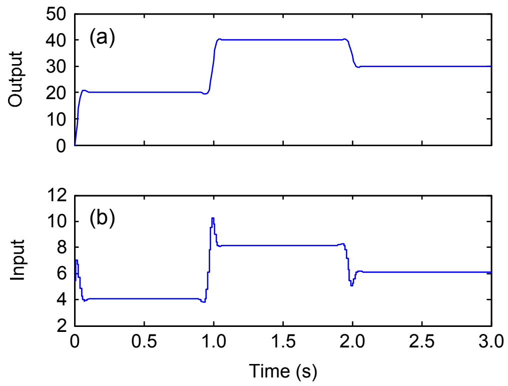

Fig. 4 Periodic response with the proposed MPC method
(a) Periodic response output; (b) Optimization control input

Fig. 2 shows that the proposed MPC method has a smaller response time and smaller overshoot than the PID method. Fig. 3 shows that the response speed of MPC is much higher than that of PID in terms of periodic response. Fig. 4 shows that the proposed MPC method follows the set points exactly. The simulation results show that the proposed MPC method based on the neural network results in a more desirable performance.

5 Conclusions

In this paper we propose a servo motor driven constant pump hydraulic system for the injection molding machine. The model of the injection molding process based on the proposed hydraulic system was described. An MPC based on neurodynamic optimization was used to control this injection molding process. Simulation results showed that the proposed MPC method based on neurodynamic optimization has achieved a satisfactory performance. Using the neurodynamic optimization method as an online optimizer of MPC improves the optimization speed of MPC. The proposed MPC based on the neurodynamic optimization method can be used in some quick response process fields.

References

Akesson, B.M., Toivonen, H.T., 2006. A neural network model predictive controller. *J. Process Contr.*, **16**(9):937-946. [doi:10.1016/j.jprocont.2006.06.001]

Draeger, A., Engell, S., Ranke, H., 1995. Model predictive control using neural networks. *IEEE Contr. Syst.*, **15**(5): 61-66. [doi:10.1109/37.466261]

Dubay, R., 2002. Self-optimizing MPC of melt temperature in injection moulding. *ISA Trans.*, **41**(1):81-94. [doi:10.1016/S0019-0578(07)60204-3]

Gao, F., Patterson, W.I., Kamal, M.R., 1996. Cavity pressure dynamics and self-tuning control for filling and packing phases of thermoplastics injection molding. *Polym. Eng. Sci.*, **36**(9):1272-1285. [doi:10.1002/pen.10521]

Gao, F.R., Yang, Y., Shao, C., 2001. Robust iterative learning control with applications to injection molding process. *Chem. Eng. Sci.*, **56**(24):7025-7034. [doi:10.1016/S0009-2509(01)00339-6]

Hopfield, J.J., Tank, D.W., 1985. "Neural" computation of decisions in optimization problems. *Biol. Cybern.*, **52**(3): 141-152. [doi:10.1007/BF00339943]

Hu, X.L., Wang, J., 2008. An improved dual neural network for solving a class of quadratic programming problems and its k-winners-take-all application. *IEEE Tran. Neur. Networks*, **19**(12):2022-2031. [doi:10.1109/TNN.2008.2003287]

Huang, S.N., Tan, K.K., Lee, T.H., 1999a. Predictive control of ram velocity in injection molding. *Polym. Plast. Technol. Eng.*, **38**(2):285-303. [doi:10.1080/03602559909351578]

Huang, S.N., Tan, K.K., Lee, T.H., 1999b. Adaptive GPC control of melt temperature in injection moulding. *ISA Trans.*, **38**(4):361-373. [doi:10.1016/S0019-0578(99)00029-4]

Huang, S.N., Tan, K.K, Lee, T.H., 2004. Neural-network-based predictive learning control of ram velocity in injection molding. *IEEE Trans. Syst. Man Cybern. C*, **34**(3): 363-368. [doi:10.1109/TSMCC.2004.829304]

Kennedy, M.P., Chua, L.O., 1988. Neural networks for nonlinear programming. *IEEE Trans. Circ. Syst.*, **35**(5): 554-562. [doi:10.1109/31.1783]

Liao, J.H., Tsai, C.C., Chang, H.C., 1996. Adaptive generalized predictive PI control of injection molding processes. Proc. 4th Int. Conf. on Automation Technology, p.711-718.

Liu, Q.S., Wang, J., 2008. A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming. *IEEE Trans. Neur.*

Networks, **19**(4):558-570. [doi:10.1109/TNN.2007.910736]

Liu, S.B., Wang, J., 2006. A simplified dual neural network for quadratic programming with its KWTA application. *IEEE Trans. Neur. Networks*, **17**(6):1500-1510. [doi:10.1109/TNN.2006.881046]

Mahadevan, R., Doyle, F.J., 2003. Efficient optimization approaches to nonlinear model predictive control. *Int. J. Robust Nonl. Contr.*, **13**(3-4):309-329. [doi:10.1002/rnc.820]

Morari, M., Lee, J.H., 1999. Model predictive control: past, present and future. *Comput. Chem. Eng.*, **23**(4-5):667-682. [doi:10.1016/S0098-1354(98)00301-9]

Pan, Y.P., Wang, J., 2009. Model predictive control for nonlinear affine systems based on the simplified dual neural network. IEEE Int. Symp. on Intelligent Control and Control Applications, p.683-688. [doi:10.1109/CCA.2009.5281106]

Pandelidis, I.O., Agrawal, A.R., 1988. Optimal anticipatory control of ram velocity in injection molding. *Polym. Eng. Sci.*, **28**(3):147-156. [doi:10.1002/pen.760280305]

Qian, S.J., Badgwell, T.A., 2003. A survey of industrial model predictive control technology. *Contr. Eng. Pract.*, **11**(7):733-764. [doi:10.1016/S0967-0661(02)00186-7]

Rafizadeh, M., Patterson, W.I., Kamal, M.R., 1996. Physically-based model of thermoplastics injection molding for control applications. *Int. Polym. Proc.*, **11**(4):352-362.

Tan, K.K., Tang, J.C., 2002. Learning-enhanced PI control of ram velocity in injection molding machines. *Eng. Appl. Artif. Intell.*, **15**(1):65-72. [doi:10.1016/S0952-1976(02)00032-5]

Tan, K.K., Huang, S.N., Jiang, X., 2001. Adaptive control of ram velocity for the injection moulding machine. *IEEE Tran. Contr. Syst. Tech.*, **9**(4):663-671. [doi:10.1109/87.930978]

Tank, D., Hopfield, J.J., 1986. Simple 'neural' optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit. *IEEE Trans. Circ. Syst.*, **33**(5):533-541. [doi:10.1109/TCS.1986.1085953]

Varela, A.E., 2000. Self-tuning pressure control in an injection moulding cavity during filling. *Chem. Eng. Res. Des.*, **78**(1):79-86. [doi:10.1205/026387600526906]

Xia, Y., Wang, J., 2001. A dual neural network for kinematic control of redundant robot manipulators. *IEEE Trans. Syst. Man Cybern. B*, **31**(1):147-154. [doi:10.1109/3477.907574]

Yang, Y., Gao, F., 2000. Adaptive control of the filling velocity of thermoplastics injection molding. *Contr. Eng. Pract.*, **8**(11):1285-1296. [doi:10.1016/S0967-0661(00)00060-5]

Zhang, S., Constantinides, A.G., 1992. Lagrange programming neural networks. *IEEE Trans. Circ. Syst. II*, **39**(7):441-452. [doi:10.1109/82.160169]