

A Study of PID Neural Network Control Algorithm for Electric Gas Regulator

Yuanchang Zhong^{1,2*}, Shaojing Xing¹, Guolong Zhao¹, Xiaochen Zhang¹
and Jing Qiao¹

¹College of Communication Engineering, Chongqing University,
Chongqing, 400044, China

²School of Automation, Chongqing University, Chongqing, 400044, China
{zyc, 20131202027, 20131213076, 20131202030, 201312020222}@cqu.edu.cn

Abstract

To improve the accuracy and reduce the response time of traditional gas regulator, a Proportional Integral Derivative (PID) neural network controller of electric gas regulator is proposed by combining conventional digital PID with neural network. And the parallel architecture of the PID neural network is implemented by using Field Programmable Gate Array (FPGA). The neural network is demonstrated in a closed loop system of electric gas regulator and the ideal system output can be obtained by using this improved neural network algorithm. Theoretical analysis and simulation results show that the PID neural network pressure controller based on FPGA can achieve faster response speed and higher control accuracy compared with those based on software, and be obvious to improve the efficiency and security of the electric gas regulator.

Keywords: PID; neural network; FPGA; electric gas regulator

1. Introduction

In recent years, the development and utilization of gas have played an important role in social economic development and environmental protection [1, 2]. The existing gas distribution system usually adopts conventional mechanical (direct or indirect) gas regulators, which have low accuracy and are easy to be affected by the external environment. Low regulated pressure accuracy makes the actual outlet pressure to deviate from the predefined range, which result in insufficient burning of gas, causes consumption of energy sources, environmental pollution and even gas security of supply [3, 4]. To solve these problems, electric gas pressure regulator technology has been proposed to replace the conventional mechanical gas regulators. But how to improve the precision and reduce the reaction time is still difficult to realize, and it is a hot topic in this research field [5]. For these problems, the author has been made the research in [6] which proposed a scheme of multi-sensor information fusion based on the intelligent air pressure sensor array to detect the inlet and outlet pressure accurately of the electric gas pressure regulator to get the pressure value which is more reliable.

The existing gas pressure regulator mainly adopts PID controller, is widely used in the industry closed loop control system, which has good robustness and easy to be implemented [7, 8]. Since artificial neural network can sufficiently approach any complicated nonlinear function, it is introduced to construct the infrastructure of the gas pressure regulator and has achieved greater fault tolerance and robustness [9, 10]. So far, the main realization method of artificial neural network is software programming. But it cannot reflect the characteristic of high-speed parallel information processing by using the serial method, and it is also unrealistic to guarantee the real-time control [11]. As the FPGA has rich programmable resource and the capacity of parallel processing, it is very

suitable for the realization of the artificial neural network [12, 13]. In this paper, we design a PID neural network controller for electric gas regulator by combining the neural network with digital PID. And we evaluate our scheme through plentiful simulations in the end.

2. The Control System Model of Electric Gas Regulator

The current gas pressure regulator is part of the mechanical pressure regulating. The basic principle of it is mechanical balance, and the internal structure is shown in Figure 1. When the outlet pressure of P2 (dark pink in Figure 1) applied the film is equal to the spring force, the outlet pressure P2 stay in balance. If the pressure P2 decreases, the spring force is greater than the upward force of the film, the spring will move down and pull the valve to move upward to increase valve opening and the downstream flow, thereby increasing the gas outlet pressure until the system reaches a new mechanical equilibrium. Obviously, the realization of this mechanical pressure regulating method depends on adjusting the related parameters of the spring and the heavy block. As time goes on, the loss of accuracy of the regulator occurs because of the reduced elasticity of the spring and the friction's damage on diaphragm [3, 5].

Based on the disadvantages of mechanical pressure regulator, we replace the original self-operated pressure regulator (Figure 2) with stepping motor and driving system (Figure 2) thereby we rebuild the core parts of pressure regulator (part 1 Figure 1), to further improve the accuracy of gas regulator by controlling stepping motor with optimized PID neural network.

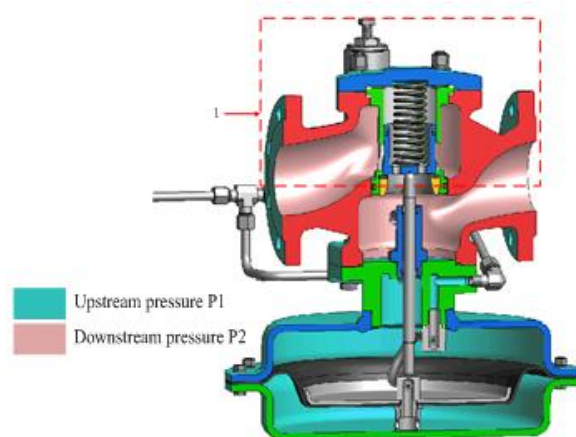


Figure 1. The Module of the Self-Operated Pressure Regulator

The self-regulating valve part 1 shown as in Figure 2, is replaced by the electric pressure value to form a linear motion electric pressure regulator. It converts the rotary motion of the gear to the nut into the linear motion of the screw shaft. And then it drives the valve to run up-and-down motion through the linear motion of screw shaft, achieving the purpose of changing the valve opening to regulate pressure. When the actual detected value is less than the pressure set point, the motor rotate is forward to increase the opening of the pressure regulating valve, and when the actual detected value is larger than the pressure set point, the motor rotate in reverse reduces the opening of the pressure regulating valve, thereby maintaining the dynamic balance of the regulator outlet pressure around the set point. How to reduce the dynamic range of dynamic equilibrium is the main research of this paper.

In order to analyze the response characteristics of the electric pressure controller system, and validate subsequent control algorithm, we need to establish corresponding mathematical models for the entire electric pressure controller system. According to its working principles, the drive motor and the control system of electric pressure regulator should be modeled, respectively.

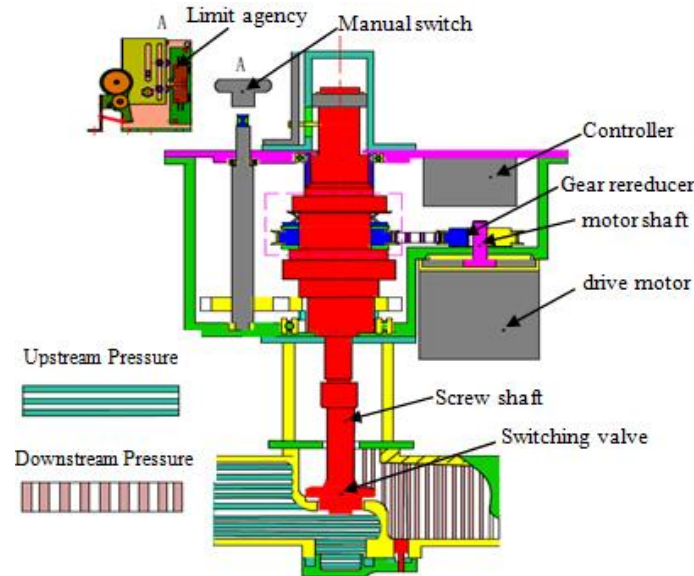


Figure 2. Electric Actuator Sketch

2.1. The Mathematical Model of Driving Motor

In this paper, driving motor adopts two phase hybrid stepping motor which has the advantages of high step frequency and fast response speed, especially it has obvious zero current position torque.

The dynamic equation of the stepping motor consists of motion equation and voltage balance equation. Assumption the number of phases of the motor is m , we can get

$$\begin{cases} T - T_L = J\ddot{\theta} \\ \dot{\psi}_k + i_k R = U_k \quad (k = 1 \dots m) \end{cases} \quad (1)$$

where T is the electromagnetic torque of the motor which consists of synchronous torque T_{e1} and asynchronous torque T_{e2} , that is $T = T_{e1} + T_{e2}$. T_L is load torque, $T_L = T_{11} + T_{12} + T_{13}$, T_{11} is dry friction torque, T_{12} is the load torque of fixed direction, T_{13} is external damping torque. J is the moment of inertia of the motor. ψ_k is the k phases magnetic linkage. i_k is k phases current. R is phase winding resistance. U_k is k phases voltage.

In the dynamic equation of the stepping motor, we assume that the flux of the permanent magnet motor hinge is Φ_m , so we can get when A phase is connected with electricity, torque formula of A and B phase tooth are as follows [14,15].

$$T_A = -E_r \Phi_m i_A \sin(E_r \theta) \quad (2)$$

$$T_B = -E_r \Phi_m i_B \sin[E_r(\theta - \lambda)] \quad (3)$$

where E_r is the number of motor rotor teeth.

Then the equation of motion of the rotor is

$$J\ddot{\theta} + D\dot{\theta} = T_A + T_B \quad (4)$$

$$J\ddot{\theta} + D\dot{\theta} + E_r\Phi_m i_A \sin(E_r\theta) + E_r\Phi_m i_B \sin[E_r(\theta - \lambda)] = 0 \quad (5)$$

where D is the viscous friction coefficient. i_A and i_B are the winding phase currents of A phase and B phase. θ is the output angle of the rotor. λ is the polar angle.

The voltage balance equations of A and B are as follows

$$\begin{cases} U - Ri_A - Li_A \dot{i}_A - Mi_B \dot{i}_B - \Phi_m E_r \sin(E_r\theta)\dot{\theta} = 0 \\ U - Ri_B - Li_B \dot{i}_B - Mi_A \dot{i}_A - \Phi_m E_r \sin[E_r(\theta - \lambda)]\dot{\theta} = 0 \end{cases} \quad (6)$$

where U is the phase winding voltage. L is the phase winding self-induction. M is the mutual induction between A and B phase. R is the phase winding resistance.

The above description is divided into two phase electricity condition. When single-phase of the stepper motor power is on, the energized sequence is given as $A - B - \bar{B} - \bar{A}$ and we suppose $\lambda = 0$, the constant current which flows through A phase and B phase is $i_A = i_0$, (5) can be rewritten in the following form

$$J\ddot{\theta} + D\dot{\theta} + 2E_r\Phi_m i_0 \sin(E_r\theta) = 0 \quad (7)$$

Let $\theta = \theta_0 - \theta_i$, $\sin(E_r\theta) \approx E_r\theta$, then (7) can be rewritten as

$$J\ddot{\theta} + D\dot{\theta} + 2E_r^2\Phi_m i_0 E_r\theta_0 = 2E_r^2\Phi_m i_0 \theta_i \quad (8)$$

where θ_i is target mechanical angle of the stepping motor, which was illustrated as the input value. θ_0 is mechanical angle of the controlled quantity, is given as the output value of the stepping motor.

In the zero initial conditions, we use Laplace transform to (8)

$$(s^2 J + sD + 2E_r^2\Phi_m i_0) \Phi_0(s) = 2E_r^2\Phi_m i_0 \theta_i(s) \quad (9)$$

So we can get the transfer function of motor as follows

$$G_\theta(s) = \frac{\theta_0(s)}{\theta_i(s)} = \frac{\omega_n^2}{s^2 + \xi\omega_n s + \omega_n^2} \quad (10)$$

where ω_n is natural angular frequency of the system, and $\xi = \frac{D}{2J\omega_n}$ is attenuation coefficient.

2.2. The Control System Model of Electric Gas Regulator

According to the model of driving motor and other system model, we can get the system of electric pressure regulator for the open-loop transfer function.

$$G(s) = \frac{Y_1 K_1}{Z_4 s^4 + Z_3 s^3 + Z_2 s^2 + Z_1 s + Z_0} \quad (11)$$

The model of pressure regulating valve is shown in Figure 3. In the system model, the pressure sensor is mainly used for monitoring the output gas pressure of the electric gas regulator and then converts the measured pressure parameter into electrical parameters. The pressure measurement is mainly used to convert the electrical parameters into the corresponding pressure value. After we send both the pressure value and the preset number into the PID neural network, the output of the PID neural network controls the rotation of the driving motor. It realizes the control of air pressure by adjusting angle of the rotor and then control valve opening through transmission system so as to reduce the gas outlet pressure in a preset value to ensure a stable air/fuel ratio in the downstream of the gas system. Finally, we realize the module of the intelligence pressure regulation of the gas to make ensure that the regulator generates a steady pressure.

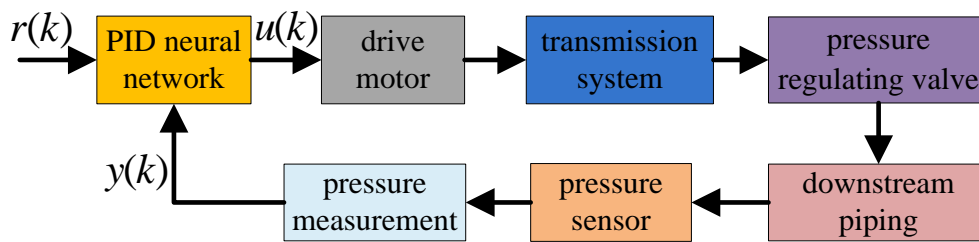


Figure 3. The Model of Electric Gas Regulator Control System

3. The PID Controller based on the Neural Network

The electric gas pressure regulator system is a typical nonlinear, time-varying and unstable system, which is unrealistic to tune the parameters. The incremental digital PID controller is difficult to achieve the ideal control effect because of the bad parameters tuning, the poor performance and the poor operation conditions adaptability [16, 17]. Therefore, we use Back Propagation (BP) neural network to tune the digital PID parameters online, which can increase the accuracy and stability of the control system.

3.1. The Back Propagation Algorithm of PID Neural Network

BP network is a kind of multilayer feedforward neural network with back propagation learning algorithm. Network not only has input and output nodes, but also owns one or more hidden layer nodes. The hidden layer nodes in the same layer don't have any coupling [18, 19]. PID neural network has a good nonlinear mapping ability, which is primarily used to constitute the control system to complete the control task of all kinds of systems. The electric gas pressure regulator system designed by using PID neural network, which is a three layers feedforward neural network with nonlinear characteristics. And it is formed by integrating the PID control law into the BP neural network [20].

The output of the electric gas regulator is an accurate and reliable pressure regulating signal to drive the electric pressure regulator to work, so the number of neuron nodes in the output layer is 1. The neurons in the hidden layer are made of proportional (P), integral (I) and derivative (D) which complete proportional, integral and derivative operation, so the number of nodes in the hidden layer neurons is 3. The input of the PID neural network is the combination of preset pressure of the electric gas pressure regulator and the actual value of the pressure detection, so the input layer has 2 neural nodes. Therefore, the structure of PID neural network is determined as $2 \times 3 \times 1$, which is shown in Figure 4. In the input layer, $r(k)$ is the preset value and $y(k)$ is the practical detection value of the controlled system. In the hidden layer, P, I and D are to complete the proportion, integral and derivative operation. In the output layer, $u(k)$ is formed to be an accurate and reliable pressure regulating signal by PID neural network control to complete a comprehensive of control law and output. S is the actual controlled system that is electric pressure regulator.

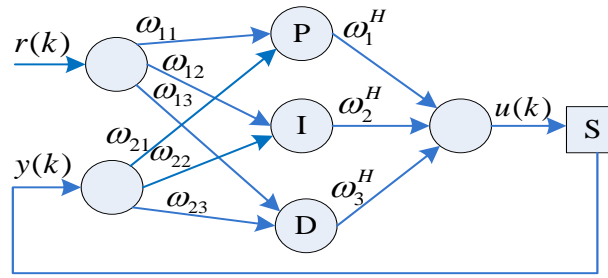


Figure 4. The Model of PID Neural Network

3.1.1. Positive Propagation of the Working Signal: (1) Input Layer

The input of the neural of the input layer, that is, the preset pressure value and the actual detection pressure value of the system are $r(k)$ and $y(k)$, respectively.

$$Net_1(k) = r(k) \quad (12)$$

$$Net_2(k) = y(k) \quad (13)$$

$$I_i(k) = Net_i(k) \quad i = 1, 2 \quad (14)$$

The output of the input layer neurons is

$$O_i(k) = \begin{cases} -1, & Net_i(k) < -1 \\ Net_i(k), & -1 < Net_i(k) < 1 \\ 1, & Net_i(k) > 1 \end{cases} \quad (15)$$

(2) Hidden Layer

The input of the hidden layer neurons is

$$I_j^H(k) = \sum_{i=1}^2 \omega_{ij} O_i(k) \quad j = 1, 2, 3 \quad (16)$$

where ω_{ij} is the weight between input layer and hidden layer, j is the serial number of the hidden layer. Three nodes of the hidden layer are proportional, integral and derivative, respectively.

The state of proportion:

$$O_1^H(k) = \begin{cases} -1, & O_1^H(k) < -1 \\ I_1^H(k), & -1 < O_1^H(k) < 1 \\ 1, & O_1^H(k) > 1 \end{cases} \quad (17)$$

The state of integral:

$$O_2^H(k) = \begin{cases} -1, & O_2^H(k) < -1 \\ O_2^H(k-1) + I_2^H(k), & -1 < O_2^H(k) < 1 \\ 1, & O_2^H(k) > 1 \end{cases} \quad (18)$$

The state of derivative:

$$O_3^H(k) = \begin{cases} -1, & O_3^H(k) < -1 \\ I_3^H(k) - I_3^H(k-1), & -1 < O_3^H(k) < 1 \\ 1, & O_3^H(k) > 1 \end{cases} \quad (19)$$

(3) Output Layer

The output layer contains only one neuron and its input is the weighted sum of the output of the hidden layer nodes.

$$Net_0(k) = \sum_{j=1}^3 \omega_j^H O_j^H(k), \quad j=1,2,3 \quad (20)$$

where ω_j^H is the weight between the hidden layer and the output layer, j is the serial number of the hidden layer.

The output of the output layer is the output of the whole neural network, that is, its output value is a accurate and reliable pressure regulating signal to control the electric gas pressure regulator to work, which is shown as followes

$$u(k) = \begin{cases} -1, & Net_0(k) < -1 \\ Net_0(k), & -1 < Net_0(k) < 1 \\ 1, & Net_0(k) > 1 \end{cases} \quad (21)$$

The error of the output layer neural in j is expressed as follows

$$E_j(k) = r_j(k) - y_j(k) \quad (22)$$

The total error of the network is expressed as follows

$$J = \frac{1}{2} \sum_{k=1}^L E^2(k) = \frac{1}{2} \sum_{k=1}^L [r(k) - y(k)]^2 \quad (23)$$

3.1.2. Back Propagation of the Error Signal

PID neural network uses BP learning algorithm, which is to reversely calculate the error signal(the difference between the sample output and the network output)according to its original connection route to adjust the weight or threshold value of the neuron of each layer by using the gradient decent method, to reduce the error signal.

The back propagation learning algorithm is used for adjusting PID parameters and the performance indicators of the tuning of PID parameters are to make the total error function of network within the limit, that is, J is given by

$$J = \sum_{k=1}^L E(k) = \frac{1}{2} \sum_{k=1}^L [r(k) - y(k)]^2 \leq \varepsilon \quad (24)$$

(1) First of all, we adjust the weight ω_j^H between the hidden layer and the output layer.

According to gradient descent, we should calculate the gradient $\frac{\partial J}{\partial \omega_j^H}$, which is error to

ω_j^H , and then reverse adjust along the direction, which are shown as follows

$$\Delta \omega_j^H = -\eta_2 \frac{\partial J}{\partial \omega_j^H} \quad (25)$$

$$\omega_j^H(k+1) = \Delta \omega_j^H(k) + \omega_j^H(k) = -\eta_2 \frac{\partial J}{\partial \omega_j^H} + \omega_j^H(k) \quad (26)$$

where k is learning step, η_2 is the learning rate.

Gradient can be obtained by partial derivative and according to the chain rule of derivative we can get

$$\frac{\partial J}{\partial \omega_j^H} = \frac{\partial J}{\partial y} \cdot \frac{\partial y}{\partial u} \cdot \frac{\partial u}{\partial Net_0} \cdot \frac{\partial Net_0}{\partial \omega_j^H} \quad (27)$$

Due to the characteristics of the object is unknown, $\frac{\partial y}{\partial u}$ can be expressed as follows

$$\frac{\partial y}{\partial u} = \text{sgn} \frac{y(k) - y(k-1)}{u(k-1) - u(k-2)} \quad (28)$$

Its positive and negative result can affect the convergence direction, substitute it into formula (27), we can get gradient which is shown as follows

$$\frac{\partial J}{\partial \omega_j^H} = -\sum_{k=1}^L \delta^2(k) O_j^H(k) \quad (29)$$

where

$$\delta^2(k) = (r - y) \text{sgn} \frac{y(k) - y(k-1)}{u(k-1) - u(k-2)} \quad (30)$$

Therefore, the value of weight modification is

$$\Delta \omega_j^H = \eta_2 \sum_{k=1}^L \delta^2(k) O_j^H(k) \quad (31)$$

(2) The error signal propagates forward and we adjust algorithms of weight between input layer and hidden layer, similarly, we have

$$\Delta \omega_{ij} = -\eta_1 \frac{\partial J}{\partial \omega_{ij}} \quad (32)$$

$$\omega_{ij}(k+1) = \Delta \omega_{ij}(k) + \omega_{ij}(k) = -\eta_1 \frac{\partial J}{\partial \omega_{ij}} + \omega_{ij}(k) \quad (33)$$

where k is learning step, η_1 is the learning rate.

$$\frac{\partial J}{\partial \omega_{ij}} = \frac{\partial J}{\partial y} \cdot \frac{\partial y}{\partial u} \cdot \frac{\partial u}{\partial \text{Net}_0} \cdot \frac{\partial \text{Net}_0}{\partial O_j^H} \cdot \frac{\partial O_j^H}{\partial I_j^H} \cdot \frac{\partial I_j^H}{\partial \omega_{ij}} \quad (34)$$

where

$$\frac{\partial O_j^H}{\partial I_j^H} = \text{sgn} \frac{O_j^H(k) - O_j^H(k-1)}{I_j^H(k-1) - I_j^H(k-2)} \quad (35)$$

The formula (34) can be rewritten as

$$\frac{\partial J}{\partial \omega_{ij}} = -\sum_{k=1}^L \delta^1(k) r(k) \quad (36)$$

where

$$\delta^1(k) = \delta^2(k) \cdot \omega_j^H \cdot \text{sgn} \frac{O_j^H(k) - O_j^H(k-1)}{I_j^H(k-1) - I_j^H(k-2)} \quad (37)$$

Therefore, the value of weight modification is

$$\Delta \omega_{ij} = \eta_1 \sum_{k=1}^L \delta^1(k) r(k) \quad (38)$$

So far, we have completed a round weights adjustment of BP network. The performance indicators of PID that is network total error of the system have been further reduced, which is further close to the preset goal and ultimately forms an accurate and reliable pressure regulating signal.

3.2. The Structure of PID Control System based on the BP Neural Network

The structure of the PID control system based on the BP neural network is shown in Figure 5. The controller consists of the incremental digital PID controller and the BP neural network. The PID controller directly controls the driving motor to realize closed loop control and tune K_p, K_i, K_d the three parameters online. In order to optimize a certain performance index, neural network on the basis of running state of the system adjusts the parameters of PID controller to make the output state of the output layer neurons correspond to the three adjustable parameters K_p, K_i, K_d of PID controller [21]. Through self-learning of the neural network and adjusting the weighting coefficients, we make it steady correspond to a certain optimal combination of PID control.

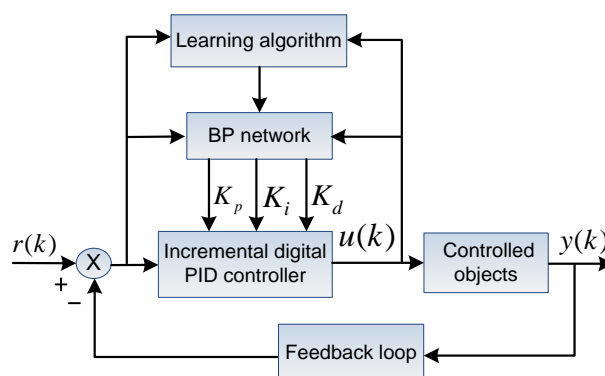


Figure 5. The Structure of PID Control System based on the BP Neural Network

4. The Realization of PID Neural Network based on FPGA

Based on FPGA hardware platform this paper designs a PID neural network controller of electric gas regulator and simulation analysis, and uses FPGA to realize the parallel structure of PID neural network. The adders, multipliers, triggers in the PID neural network are used in 32-bit with the standard IEEE754 single precision floating point number.

For the FPGA implementation of the PID neural network, we use EP1S20F484C5 of Stratix series. The input layer, hidden layer and output layer are designed respectively in a top-down approach based on the structure of the neural network. Each layer is integrated into an IP core for the overall design. The synthesized Register-Transfer-Level (RTL) structure is shown in Figure 6.

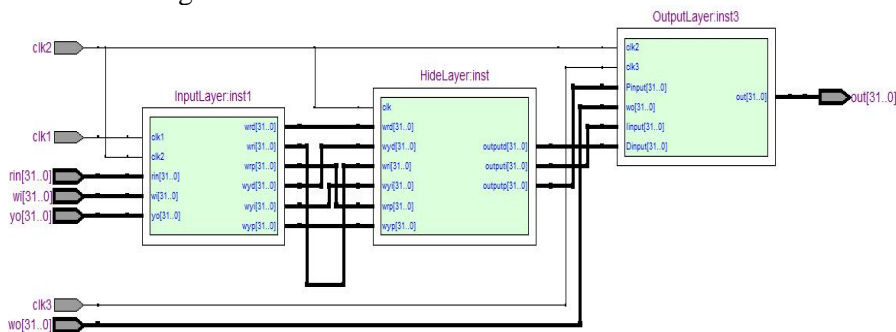


Figure 6. The RTL Structure of PID Neural Network

In Figure 6, $clk1$, $clk2$ and $clk3$ are system clock input, $rin[31..0]$ is the preset value, $yo[31..0]$ is the feedback value, $wi[31..0]$ is the weight input between the input layer and hidden layer, $wo[31..0]$ is the weight output between the hidden layer and output layer, $out[31..0]$ is system output. The value $wrp[31..0]$ of the Input Layer is the output value which is the preset value after the weighted calculation sends to the proportional neurons, $wri[31..0]$ is the output value which is the preset value after the weighted calculation sends to the integration neurons, $wrd[31..0]$ is the output value which is the preset value after the weighted calculation sends to the derivative neurons, $wyp[31..0]$ is the output value which is the detection value after the weighted calculation sends to the proportional neurons, $wyi[31..0]$ is the output value which is the detection value after the weighted calculation sends to the integration neurons, $wyd[31..0]$ is the output value which is the detection value after the weighted calculation sends to the derivative neurons. The input layer module mainly completes the data input of the neural network, and the weighted operation on input data, outputs the value to the next module. In the hidden layer, $wrp[31..0]$ receives the result of the preset value and the input weight, $wyp[31..0]$ receives the result of the detection value and the input weight, and then the data is sent to the proportional neuron to complete the calculation, furthermore the proportional calculation is output through the $outputp[31..0]$ to the next module. $wri[31..0]$ receives the result of the preset value and the input weight, $wyi[31..0]$ receives the result of the detection value and the input weight, and the data is sent to the integral neuron to complete the calculation, besides the integral calculation is output through the $outputi[31..0]$ to the next module. $wrd[31..0]$ receives the result of the preset value and the input weight, $wyd[31..0]$ receives the result of the detection value and the input weight, and the data is sent to the derivative neuron to complete the calculation, and then the derivative calculation is output through the $outputd[31..0]$ to the next module. In the Output Layer (OutLayer), $wo[31..0]$ is the input of the output weight value, $Pinput[31..0]$ is used to receive the proportional output of the hidden layer. $Iinput[31..0]$ is used to receive the integral output of the hidden layer. $Dinput[31..0]$ is used to receive the derivative output of the hidden layer. After operations of the output layer, $out[31..0]$ is to accomplish the output of the whole PID neural network and to control the pressure regulating system. The resource consumption of PID neural network controller is shown in Figure 7.

The chip EP1S20F484C5 includes 18460 logic elements (LE), 362 I/O pins, 1669248 RAMs, 80 DSPs, consuming 9412(51%), 163(45%), 2162(<1%) and 72(90%), phase-locked loop PLL and DLL are not needed [22].

| | |
|--------------------------|--|
| Flow Status | Successful - Fri Dec 26 15:43:43 2014 |
| Quartus II Version | 9.0 Build 132 02/25/2009 SJ Full Version |
| Revision Name | PID_neural_network |
| Top-level Entity Name | PID_neural_network |
| Family | Stratix |
| Met timing requirements | Yes |
| Total logic elements | 9,412 / 18,460 (51 %) |
| Total pins | 163 / 362 (45 %) |
| Total virtual pins | 0 |
| Total memory bits | 2,162 / 1,669,248 (< 1 %) |
| DSP block 9-bit elements | 72 / 80 (90 %) |
| Total PLLs | 0 / 6 (0 %) |
| Total DLLs | 0 / 2 (0 %) |
| Device | EP1S20F484C5 |
| Timing Models | Final |

Figure 7. Resource Consumption of PID Neural Network Controller

5. Analysis and Comparison of Simulation

The frequency of three clocks (*clk1*, *clk2* and *clk3*) is set to 10 ns, and then is simulated by a no-load way, which is shown in Figure 8.

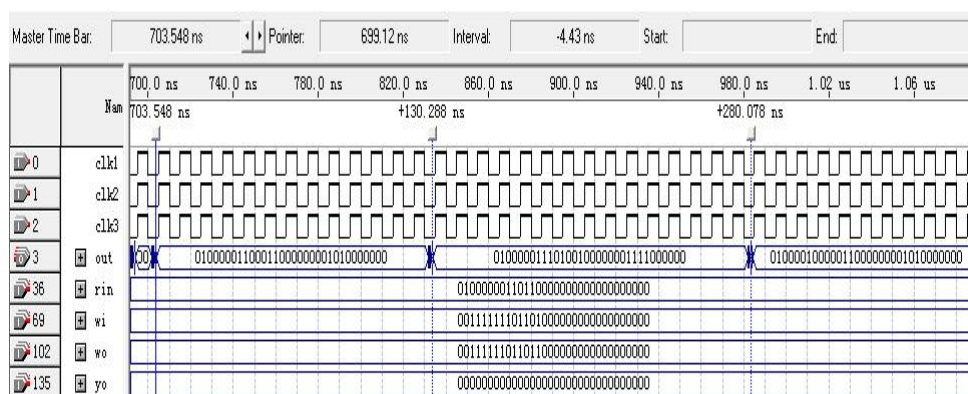


Figure 8. The Waveform No-load Simulation

Since it is a no-load simulation, the input value *yo* is 0. The preset value *rin* is 01000001101100000000000000000000, and according to the 32-bit float counting rules, its decimal number is 6.75. The weight *wi* between the input layer and the hidden layer is 00111111101101000000000000000000, and its decimal number is 1.40625. The weight *wo* between the hidden layer and output layer is 00111111011011000000000000000000, and its decimal number is 0.921875.

At 0ns, PID neural network inputs data, and outputs after 703.548ns. Since it is a no-load simulation, the connection weights of the whole neural network are constant, so the derivative value is 0. The output value depends only on proportional element and integral element, and the theoretical output value should be $2 \times 6.75 \times 1.40625 \times 0.921875 = 17.5012207$. The actual output value is 01000001100010000000000000000000, and its decimal number is 17.5. The error between the theoretical value and the actual value is $17.5012207 - 17.5 = 0.001220703$. Because of the cumulative effect of integral element, the second theoretical output should be $3 \times 6.75 \times 1.40625 \times 0.921875 = 26.25183105$, and the actual output is 01000001110100100000001111000000. Its decimal number is 26.2520752, so the error is

$26.25183105 - 26.2520752 = -0.00024415$. The third theoretical output should be $4 \times 6.75 \times 1.40625 \times 0.921875 = 35.00244141$, while the actual output is 01000010000011000000001010000000, its decimal number is 35.00244141, so the error is 0.

According to the comparison of simulation data and the theoretical analysis, PID neural networks can satisfy the design requirements. As achieves good real-time effect and small error, it can control the air pressure quickly and accurately.

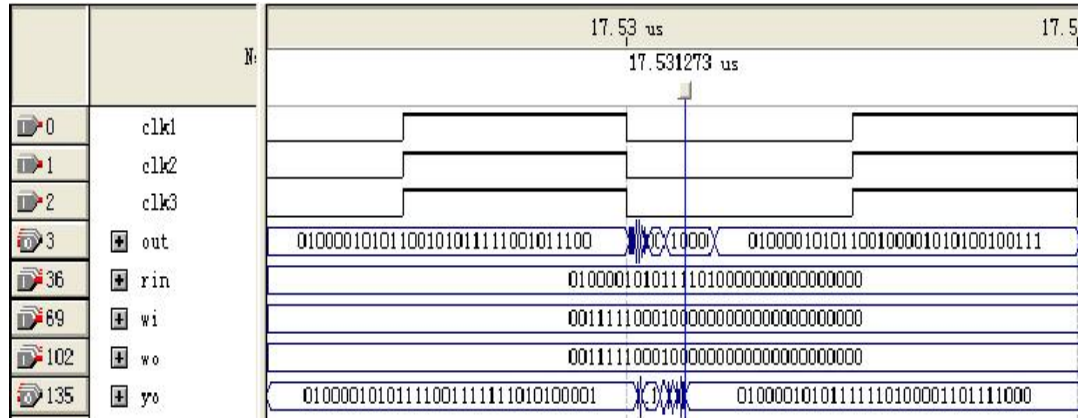


Figure 9. The Waveform of Global Simulation

In Figure 9, the preset value *rin* is 01000010101111010000000000000000, and its decimal number is 0.03125. The weight *wi* between the input layer and the hidden layer is identical with the weight *wo* between the hidden layer and the output layer, and both are 00111110001000000000000000000000, and their decimal number is 94.5. At approximately 17.5 seconds, the value *yo* is 0100001010111110100001101111000, and its decimal number is 95.15234375.

In Table 1, the PID neural network that is designed in this paper has a higher precision pressure regulator and closed precision pressure regulator than the traditional gas pressure regulator. In terms of response speed, the traditional gas pressure regulator is generally 1~3s, while in Figure 9, the PID neural networks which is based on FPGA hardware platform can quickly make changes to feedback data, and the response speed is generally less than 1s.

Table 1. Performance Comparison Regulator

| Type | Regulating Accuracy (%) | Closing Accuracy (%) |
|--------------------|-------------------------|----------------------|
| Reno | ± (10~20) | 15~20 |
| T-type | ±10 | 20~25 |
| Self-operated | ± (5~10) | 20 |
| Direct Acting | ±5 | 10 |
| PID Neural Network | ± (1~5) | 5 |

6. Conclusion

The existing electric gas pressure regulator widely adopts conventional incremental control model which is simple and easy to be implemented. Since the system of the whole pressure regulator is nonlinear and time-varying, it is difficult to establish an accurate mathematical model. The application of current neural networks is mostly realized through software or DSP, and it is difficult to be implemented in a large-scale application due to the low processing speed of the microprocessor. In this paper, we realize PID

neural networks controller by combining PID control with neural networks, employing the original digital incremental PID control mode based on FPGA hardware platform. Our scheme achieves high accuracy, and it comprehensively reflects the parallel computing speed of neural networks as its running speed is ten times even hundred times faster than that of software or microcomputer. At the same time, the accuracy and the response speed of the electric gas pressure regulator are improved. PID controller that we designed is not only extensively used in electric gas pressure regulator, but also provides an efficient and viable solution for the intelligent control of the industry in the future.

Acknowledgments

This work is supported in part by the Major State Basic Research Development Program 973 (No. 2012CB215202).

References

- [1] X. Wang and J. Xiao, "The artificial neural networks for real-time operation of natural gas production and sale", Environmental Science and Information Application Technology (ESIAT), 2010 International Conference on IEEE, Wuhan, China, (2010), July 7-18.
- [2] S. H. Mohr and G. M. Evans, "Long term forecasting of natural gas production", Energy Policy, vol. 39, no. 9, (2011), pp. 5550-5560.
- [3] T. Morley, M. Dodd, K. Gagnon, V. Hanemaayer and J. Wilson, "An automated module for the separation and purification of cyclotron-produced $^{99m}\text{TcO}_4^-$ ", Nuclear Medicine and Biology, vol. 39, no. 4, (2012), pp. 551-559.
- [4] H. Kum, O. Ocal and A. Aslan, "The relationship among natural gas energy consumption, capital and economic growth: Bootstrap-corrected causality tests from G-7 countries", Renewable and Sustainable Energy Reviews, vol. 16, no. 5, (2012), pp. 2361-2365.
- [5] Y. D. Xie, Y. Wang and Y. J. Liu, "Overview of Control Valve Technology", Control and Instruments in Chemical Industry, vol. 39, no. 9, (2012), pp. 1111-1114.
- [6] Y. C. Zhong, F. Wang, L. Zhang and C. X. Tao, "The Information Fusion Algorithm on Gas Pressure Regulator Inlet/Outlet Detection", Journal of Computational Information Systems, vol. 10, no. 5, (2014), pp. 2145-2153.
- [7] C. D. Jung, J. Mook, Kang and C. H. Park, "Study of Engine Oil Replacement Time Estimate Method using Fuzzy and Neural Network Algorithm in Ubiquitous Environment", International Journal of Control and Automation (IJCA), vol. 6, no. 3, (2013), pp. 267-280.
- [8] M. Farahani, S. Ganjefar and M. Alizadeh, "PID controller adjustment using chaotic optimization algorithm for multi-area load frequency control", IET Control Theory & Applications., vol. 6, no. 13, (2012), pp. 1984-1992.
- [9] J. Rodger, "A fuzzy nearest neighbor neural network statistical model for predicting demand for natural gas and energy cost savings in public buildings", Expert Systems with Applications, vol. 41, no. 4, (2014), pp. 1813-1829.
- [10] R. Taormina, K.-W. Chau and R. Sethi, "Artificial neural network simulation of hourly groundwater levels in a coastal aquifer system of the Venice lagoon", Engineering Applications of Artificial Intelligence, vol. 25, no. 8, (2012), pp. 1670-1676.
- [11] C. M. David, V. R. Martin, G. P. Artur, A. O. R. Roque and J. R. T. Rene, "Empirical Mode Decomposition and Neural Networks on FPGA for Fault Diagnosis in Induction Motors", The Scientific World Journal, vol. 2014, (2014), pp. 1-17.
- [12] O. K. Teresa and K. Marcin, "FPGA Implementation of the Multilayer Neural Network for the Speed Estimation of the Two-Mass Drive System", IEEE Transactions on Industrial Informatics, vol. 7, no. 3, (2011), pp. 436-445.
- [13] H. Rostro-Gonzalez, B. Cessac, B. Girau and C. Torres-Huitzil, "The role of the asymptotic dynamics in the design of FPGA-based hardware implementations of gIF-type neural networks", Journal of Physiology-Paris, vol. 105, no. 1-3, (2011), pp. 91-97.
- [14] S. Seshagiri, "Position control of permanent magnet stepper motors using conditional servocompensators", IET Control Theory & Applications, vol.3, no. 9, (2013), pp. 1196-1208.
- [15] R. Delpoux, M. Bodson and T. Floquet, "Parameter estimation of permanent magnet stepper motors without position or velocity sensors", in Proceedings of the IEEE Control Engineering Practice, Montreal, Montreal, Canada, (2012), June 27-29.
- [16] D. J. Wang, "Stabilising regions of PID controllers for nth-order all pole plants with dead-time", IET Control Theory & Applications, vol.1, no.4, (2007), pp. 1068-1074.
- [17] M. Shamsuzzoha and S. Skogestad, "The setpoint overshoot method: A Simple and Fast Closed Loop Approach for PID Tuning", Journal of Process Control, vol. 20, no. 10, (2011), pp. 1220-1234.

- [18] Z. Guo, J. Wu, H. Lu and J.Z. Wang, "A case study on a hybrid wind speed forecasting method using BP neural network", *Knowledge-Based Systems*, vol. 24, no. 7, (2011), pp. 1048-1056.
- [19] Z. Xiao, S. J. Ye, B. Zhong and C. X. Sunc, "BP neural network with rough set for short term load forecasting", *Expert Systems with Applications*, vol. 36, no. 1, (2009), pp. 273-279.
- [20] J. Zhang, S. C. Wang, Y. Y. Jing and X. Zhang, "Induction Motor Control Based on BP Neural Network and Adaptive PID' International Conference on Computer Science and Network Technology", pp. 959-963, in *Proceedings of the International Conference on Computer Science and Network Technology*, Changchun, China, (2012), December 13-14.
- [21] R. R. Sumar, A. A. R. Coelho and L. D. S. Coelho, "Computational intelligence approach to PID controller design using the universal model", *Information Sciences*, vol. 180, no. 20, (2010), pp. 3980-3991.
- [22] R. Horvat, K. Jezernik and M. Curkovic, "An Event-Driven Approach to the Current Control of a BLDC Motor Using FPGA", *IEEE Transactions on Industrial Electronics*, vol. 61, no. 7, (2014), pp. 3719-3726.

Authors



Yuanchang Zhong, he was born in Nanchong City, Sichuan Province, in 1965. He received M.S. degrees in 2004 and the Ph.D. degree in 2011 from Chongqing University.

From 2003 to 2011, he was an Assistant Professor with the College of Communication engineering, since 2012, he has been a Professor at Chongqing University, where he leads the Networks and Mobile Systems group at CTTC (communication and tracking telemetering & command). He is the author of two books, more than 100 articles, and more than 10 inventions. His research is in the area of communication and tracking telemetering & command; Circuits and systems; wireless sensor networks. In addition to many widely cited papers, several systems developed as part of his research are available in the public domain. Some are in production or commercial use.



Shaojing Xing, he was born in Sichuan Province, China, in 1990. He is currently pursuing the M.S. degree in the College of Communication Engineering of Chongqing University, Chongqing, China. His research interests include automatic control and Big Data.



Guolong Zhao, he was born in Gansu Province, China, in 1988. He is currently pursuing the M.S. degree in the College of Communication Engineering of Chongqing University, Chongqing, China. His research interests include control systems and time delay systems



Xiaochen Zhang, he was born in Anhui Province, China, in 1988. He is currently pursuing the M.S. degree in the College of Communication Engineering of Chongqing University, Chongqing, China. His research interests include automatic control and wireless sensor networks.



Jing Qiao, she was born in Henan Province, China, in 1989. She is currently pursuing the M.S. degree from Chongqing University, Chongqing, China. Her research interests include control systems and Big Data

