

Research Article

Stem Control of a Sliding-Stem Pneumatic Control Valve Using a Recurrent Neural Network

Mohammad Heidari¹ and Hadi Homaei²

¹ Mechanical Engineering Group, Aligudarz Branch, Islamic Azad University, P.O. Box 159, Aligudarz, Iran

² Faculty of Engineering, Shahrekord University, P.O. Box 115, Shahrekord, Iran

Correspondence should be addressed to Mohammad Heidari; moh104337@yahoo.com

Received 14 March 2013; Revised 1 June 2013; Accepted 6 June 2013

Academic Editor: Chao-Ton Su

Copyright © 2013 M. Heidari and H. Homaei. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a neural scheme for controlling an actuator of pneumatic control valve system. Bondgraph method has been used to model the actuator of control valve, in order to compare the response characteristics of valve. The proposed controller is such that the system is always operating in a closed loop, which should lead to better performance characteristics. For comparison, minimum- and full-order observer controllers are also utilized to control the actuator of pneumatic control valve. Simulation results give superior performance of the proposed neural control scheme.

1. Introduction

Process plants consist of hundreds, or even thousands, of control loops all networked together to produce a product to be offered for sale. Each of these control loops is designed to keep some important process variables such as pressure, flow, level, and temperature within a required operating range to ensure the quality of the end product. Each of these loops receives and internally creates disturbances that detrimentally affect the process variable, and interaction from other loops in the network provides disturbances that influence the process variable. To reduce the effect of these load disturbances, sensors and transmitters collect information about the process variable and its relationship to some desired set points. A controller then processes this information and decides what must be done to get the process variable back to where it should be after a load disturbance occurs. When all the measuring, comparing, and calculating are done, some type of final control element must implement the strategy selected by the controller. The most common final control element in the process control industries is the control valve. The control valve manipulates a flowing fluid, such as gas, steam, water, or chemical compounds, to compensate for the load disturbance and keep the regulated process variable as close as possible to the desired set point. Control valves adjust

the temperature, pressure, flow rate, and so forth by changing the flow rate. Figure 1 shows a reverse-acting diaphragm actuator of pneumatic control valve. Pneumatic control valves are still the most used valves in the process industries, due to their low cost and simplicity. Pneumatic valves are used extensively in various industries today. Industry standard has been established that details the vibration, humidity, thermal, salt spray, and temperature extremes that these valves must operate within. This makes the design of valve control systems a very challenging task. Control valves have two major components, valve body housing and the actuation unit. One factor in the quality of the final end product is the improvement of the control loop performance. A critical component in the loop is the final control element, the control valve package. Optimized actuator parameters play a vital role in the dynamic performance of the pneumatic control valve. Hägglund [1] presented a procedure that compensates for static friction (stiction) in pneumatic control valves.

The compensation is obtained by adding pulses to the control signal. The characteristics of the pulses are determined from the control action. The compensator is implemented in industrial controllers and control systems, and the industrial experiences show that the procedure reduces the control error during stick-slip motion significantly compared to standard control without stiction compensation.

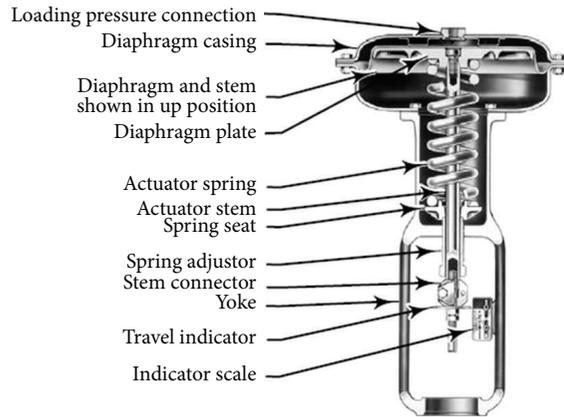


FIGURE 1: Reverse-acting diaphragm actuator of a pneumatic control valve.

The oscillations caused by static friction (stiction) in pneumatic control valves cause losses in quality and expense of raw materials. The input-output behavior of a pneumatic control valve is affected by stiction in valve. De Souza et al. [2] presented a well-known stiction compensation method that reduced variability both at process variable and pneumatic valve stem movement. The two-move method was revisited in their research and it was shown that assumptions on the knowledge of steady-state stem position of control valve that assured equality of set point and the controlled variable was not easily achievable.

Champagne and Boyle [3] reviewed the pneumatic actuator and positioner parameters that affect the control package performance. This was done through the use of a control valve package computer model to assess the dynamic performance. The attributes of spring return versus double-acting actuators were illustrated. The effects of supply pressure, step size, load margin, flow, actuator volume, and design style were investigated through the use of mathematical simulations of pneumatic control valve dynamic performance.

Bondgraph is a graphical representation of a physical dynamics system. It is similar to the better known block diagram and signal flow, with the major difference that the arcs in bondgraphs represent bidirectional exchange of physical energy, while those in block diagrams and signal-flow graphs represent unidirectional flow of information. Also, bondgraphs are multidomain and domain neutral. This means that a bondgraph can incorporate multiple domains simultaneously. The fundamental idea of a bondgraph is that power is transmitted between connected components by a combination of “effort” and “flow” (generalized effort & generalized flow). Bondgraphs were devised by Paynter [4] at MIT in April 1959 and subsequently developed into a methodology together with Karnopp et al. [5]. Early prominent promoters of bondgraph modeling techniques among others were Thoma [6], Dixhoorn, and Dransfield. They contributed substantially to the dissemination of bondgraph modeling in Europe, Australia, Japan, China and India. Athanasatos and Costopoulos [7] used the bondgraph method for finding the proactive fault in 4/3 way direction

control valve of a high-pressure hydraulic system. The accuracy of the bondgraph model was verified by comparing its response to the response of an actual hydraulic system. Zuccarini et al. [8] utilized the bondgraph as boundary condition for a detailed model of an idealized mitral valve. A specific application in cardiovascular modeling was demonstrated by focusing on a specific example, a 3D model of the mitral valve coupled to a lumped parameter model of the left ventricle. Ekren et al. [9] used three different control algorithms such as proportional, integral, differential (PID), fuzzy logic, and artificial neural network (ANN) for control of a variable speed compressor and electronic expansion valve in a chiller system. The results showed that ANN controller has lower power consumption of 8.1 percent and 6.6 percent than both PID and fuzzy controllers, respectively. Choi et al. [10] have modeled a system of position control which uses a single-rod cylinder activated by an electrorheological (ER) valve. From the state-space model for the governing equations, a neural network control scheme has been synthesized to achieve the position control of the cylinder system. The results showed the effectiveness of the proposed methodology.

In this paper, a control scheme was investigated to control a pneumatic control valve system. The robustness of the proposed scheme was presented through computer simulation and the efficacy of the scheme is shown both in the time and amplitude domains. A sliding-stem pneumatic control valve is modeled by bondgraph method. Then, several control schemes have been used for control of valve in order to compare the response characteristics of these different schemes.

This research is organized as follows. Section 2 recalls the bondgraph model of valve and proposes equations of motion of the valve. Section 3 develops the control schemes of the valve. Simulation results and discussion of the control schemes are given in Section 4. The paper is concluded with Section 5.

2. Bondgraph Model of Valve and Equations

The bondgraph model of the valve is shown in Figure 2. In this model, SE is the inlet pressure of the system. The pressure changes to force by multiplying in effect area of the diaphragm. In bondgraph, this transformer is modeled by TF . Element R is the friction of the system. Element I is the movable mass of valve and diaphragm Element C represents the spring of the valve actuator.

Also 1-junction is a common flow junction. 1-junctions have equality of flows and the efforts sum up to zero with the same power orientation. In fact, junctions can connect two or more bonds. The direction of the half arrows (\rightarrow) denotes the direction of power flow given by the product of the effort and flow variables associated with the power bond. The bonds in a bondgraph may be numbered sequentially using integers starting with 1. The two 1-junctions in the bondgraph shown can be uniquely identified as (S 1 2) and (S 4 5 6); similarly symbols like SE_1 , R_6 can be used to identify a particular element. This system has two state variable P_4 and q_5 . q_5 is

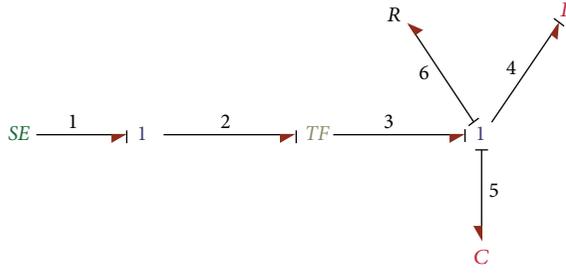


FIGURE 2: Bondgraph control valve actuator.

the displacement of valve stem and the variation of the spring length. Also $v_4 = \dot{P}_4/I_4$ is the velocity of the valve stem. The equations of motion are derived using bondgraph method as below:

$$\dot{P}_4 = A \times SE_1 - K_3 q_5 - \frac{R_6}{I_4} P_4, \quad (1)$$

$$\dot{q}_5 = \frac{P_4}{I_4}. \quad (2)$$

Now, if the velocity and position of stem are zero in the initial condition, $X(0)$, then we have the following:

$$X(0) = \begin{bmatrix} P_4(0) \\ q_5(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (3)$$

By derivation of relation (2) with respect to time we have the following:

$$\ddot{q}_5 = \frac{\dot{P}_4}{I_4}. \quad (4)$$

By substitution of \dot{P}_4 from (1) into (4) we have the following:

$$\ddot{q}_5 = \frac{A \times SE_1 - K_3 q_5 - (R_6/I_4) P_4}{I_4}. \quad (5)$$

By substitution of P_4 from (2) into (5) we have the following:

$$\ddot{q}_5 = \frac{1}{I_4} (A \times SE_1 - K_3 q_5 - R_6 \dot{q}_5). \quad (6)$$

Using Laplace transformation of (6), we have the following:

$$\frac{q_5(s)}{SE_1(s)} = \frac{(A/I_4)}{s^2 + (R_6/I_4)s + (K_3/I_4)}. \quad (7)$$

Equation (7) is the transfer function of the valve. The results of bondgraph model of valve shows that the response of the system is identical with the result in [11].

3. Control Schemes

For completeness, this section briefly reviews the control schemes, which are the observer control and the NN control proposed in this paper.

3.1. State Observer. The state observer estimates the state variables using the output and control input value. At this time, it can be configured only when the system is observable. \tilde{x} is the observed state vector. Let us consider the system that is defined in form of a state-space representation as follows:

$$\dot{x} = Ax + Bu, \quad (8)$$

$$y = Cx + Du, \quad (9)$$

where x is called the state vector, \dot{x} the derivative of the state vector with respect to time, y the output vector, and $u(t)$ the input or control vector. Also A is the system matrix, B the input matrix, C the output matrix, and D direct transmission matrix. Suppose that \tilde{x} is an estimation for the state vector x . Then, we have the following:

$$\dot{\tilde{x}} = A\tilde{x} + Bu + K_e(y - C\tilde{x}). \quad (10)$$

Formula (10) indicates the state observer. y and u are inputs and \tilde{x} is output. The term at the right end of (10) corrects the difference between the measured output y and the estimated output \tilde{x} . K_e works as the weighting matrix. Even though there is a difference between the matrix A and B of the actual system and the model, the influence is reduced by this correction. Deduct formula (10) from formula (8) to get the error of the observer. Then,

$$\dot{x} - \dot{\tilde{x}} = (A - K_e C)(x - \tilde{x}). \quad (11)$$

The result is formula (11). Suppose that the observer error ($x - \tilde{x}$) is e ; that is to say, $e = (x - \tilde{x})$, formula (12) is arranged as follows:

$$\dot{e} = (A - K_e C)e. \quad (12)$$

From formula (12), we can recognize that the dynamic characteristics of the observer error are determined by the eigenvalue of $A - K_e C$. If $A - K_e C$ is a stable matrix, the error vector approaches zero related to any initial value $e(0)$. That is to say, $\tilde{x}(t)$ approaches $x(t)$ irrespective of the value of $x(0)$ and $\tilde{x}(0)$. If we select the eigenvalue of this matrix well, the error vector can approach to zero fast. If the given system is completely observable, we can make the eigenvalue of $A - K_e C$ as we want as selecting K_e well. That is to say, K_e is selected to make $A - K_e C$ as we want and this is the gain matrix of the state observer. The observer discussed thus far is designed to reconstruct all of the state variables. In practice, some of the state variables may be accurately measured. Such accurately measurable state variables need not to be estimated. An observer that estimates fewer than n state variables, where n is the dimension of the state vector, is called a reduced-order observer. The details of designing a minimum-order observer have been presented in [12].

3.2. Proposed Neural Controller. The neural networks employed in this work were of the recurrent type. Recurrent networks have the advantage of being able to model dynamic systems accurately and in a compact form [13]. A recurrent network can be represented in a general diagrammatic form

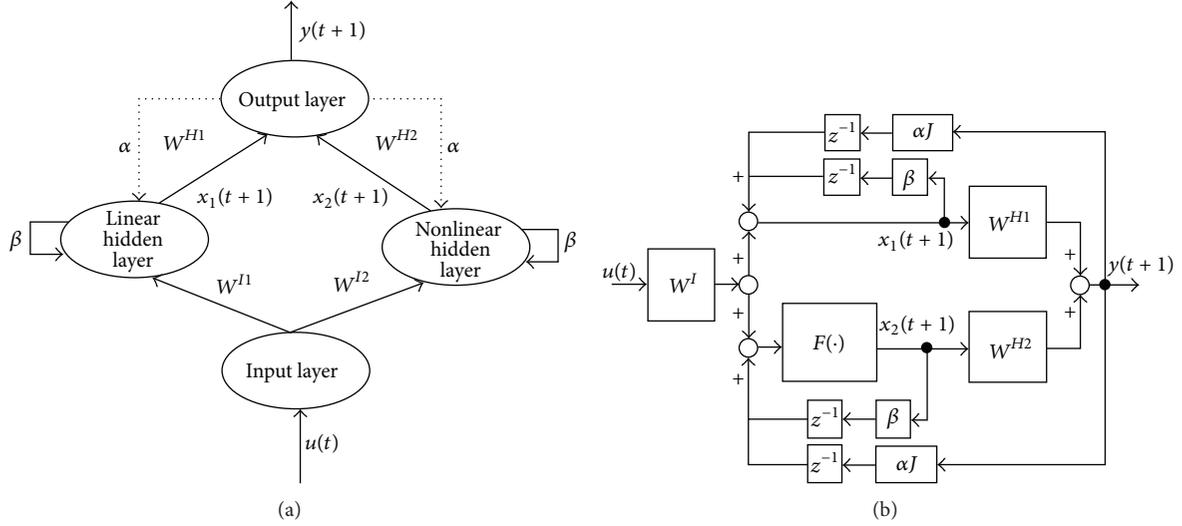


FIGURE 3: (a) Recurrent hybrid network structure. (b) Block diagram of recurrent hybrid network.

as illustrated in Figure 3(a). This diagram depicts the hybrid hidden layer as comprising a linear part and a nonlinear part and shows that, in addition to the usual feedforward connections, the networks also have feedback connections from the output layer to the hidden layer and self-feedback connections in the hidden layer. The reason for adopting a hybrid linear/non-linear structure for the hidden layer will be evident later.

At a given discrete time t , let $u(t)$ be the input to a recurrent hybrid network, $y(t)$ the output of the network, $x_1(t)$ the output of the linear part of the hidden layer, and $x_2(t)$ the output of the nonlinear part of the hidden layer. The operation of the network is summarized by the following equations (also see Figure 3(b)):

$$\begin{aligned} x_1(t+1) &= W^{I1}u(t+1) + \beta x_1(t) + \alpha J_1 y(t), \\ x_2(t+1) &= F\{W^{I2}u(t+1) + \beta x_2(t) + \alpha J_2 y(t)\}, \\ y(t+1) &= W^{H1}x_1(t+1) + W^{H2}x_2(t+1), \end{aligned} \quad (13)$$

where W^{I1} is the matrix of weights of connections between the input layer and the linear hidden layer, W^{I2} is the matrix of weights of connections between the input layer and the non-linear hidden layer, W^{H1} is the matrix of weights of connections between the linear hidden layer and the output layer, W^{H2} is the matrix of weights of connections between the non-linear hidden layer and the output layer, $F\{\cdot\}$ is the activation function of neurons in the non-linear hidden layer and α and β are the weights of the self-feedback and output feedback connections. J_1 and J_2 are, respectively, $(n_{H1} \times n_O)$ and $(n_{H2} \times n_O)$ matrices with all elements equal to 1, where n_{H1} and n_{H2} are the numbers of linear and non-linear hidden neurons, and n_O is the number of output neurons. If only

linear activation is adopted for the hidden neurons, the above equations are simplified to

$$y(t+1) = W^{H1}x(t+1), \quad (14)$$

$$x(t+1) = W^{I1}u(t+1) + \beta x(t) + \alpha J_1 y(t). \quad (15)$$

Replacing $y(t)$ by $W^{H1}x(t)$ in (15) gives

$$x(t+1) = W^{I1}u(t+1) + (\beta I + \alpha J_1 W^{H1})x(t), \quad (16)$$

where I is a $(n_{H1} \times n_{H1})$ identity matrix.

Equation (16) is of the form

$$x(t+1) = Ax(t) + Bu(t+1), \quad (17)$$

where $A = \beta I + \alpha J_1 W^{H1}$ and $B = W^{I1}$. Equation (17) represents the state equation of a linear system of which x is the state vector. The elements of A and B can be adjusted through training so that any arbitrary linear system of order n_{H1} can be modelled by the given network. When non-linear neurons are adopted, this gives the network the ability to perform non-linear dynamic mapping and thus model non-linear dynamic systems [14, 15]. The existence in the recurrent network of a hidden layer with both linear and non-linear neurons facilitates the modeling of practical non-linear systems comprising linear and non-linear parts. Figure 4 shows the proposed control system for a pneumatic valve actuator. The system comprises a PD controller and an NN controller, which is a recurrent hybrid network used to model inverse dynamics of the valve. The NN is trained online during the control to make the system able to adapt to changes. The control architecture illustrated in Figure 3 was implemented on a personal computer using Neural Network Toolbox of MATLAB [16].

4. Results and Discussion

Table 1 shows the parameters of a sliding-stem pneumatic control valve.

TABLE 1: Valve parameters [11].

Name of variable	Parameter	Value
Effective area of diaphragm	A	0.196 ft ²
Spring constant	K	6790
Movable mass	$I (M)$	0.03 slug
Resistance and friction coefficient	R	1 lb · s/ft
Air pressure	SE	140 lb/ft ²

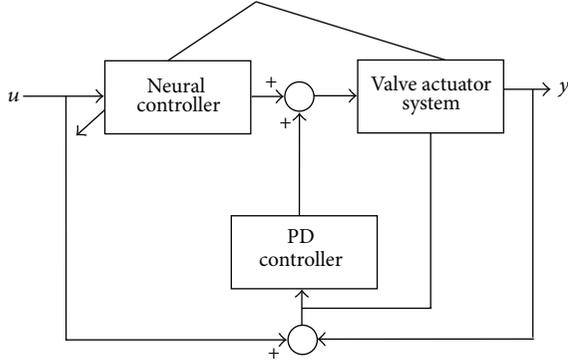


FIGURE 4: Block diagram of the proposed neural control system.

By substitution of Table 1 into (7), we have the following:

$$\frac{q_5(s)}{SE_1(s)} = \frac{6.53}{s^2 + 33.33s + 226333.3}. \quad (18)$$

Matrices of the state space equations of the valve are as follows:

$$A = \begin{bmatrix} 0 & 1 \\ -226333.33 & -33.33 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 914.2 \end{bmatrix}, \quad (19)$$

$$C = [1 \ 0], \quad D = 0.$$

In this section, the design of control system with full- and minimum order observers is considered, when the system has reference inputs or command inputs. The output of the control system should follow the input. In following, the command input, the system should exhibit satisfactory performance (a reasonable rise time, overshoot, settling time, and so on). We consider control systems that are designed by use of the pole placement with observer approach. When a system has a reference input, several different block diagram configurations are conceivable, each having an observer controller. Two of these configurations are feedforward and feedback path. We would like to design the full- and minimum order observer controllers such that in the unit step response the maximum overshoot is less than 10% and settling time is about 0.5 sec. We first design the controller by finding the desired characteristic equation. A 10% overshoot and a settling time of 0.5 second yield $\xi = 0.591$ and $\omega_n = 13.53$; thus, the characteristic equation for dominant poles is $s^2 + 16s + 183.1 = 0$, where the dominant poles are located at $-8 \pm j10.91$ [17]. Hence, choose the desired closed-loop poles at $s = \mu_i$ ($i = 1, 2$), where

$$\mu_1 = -8 + j10.91, \quad \mu_2 = -8 - j10.91. \quad (20)$$

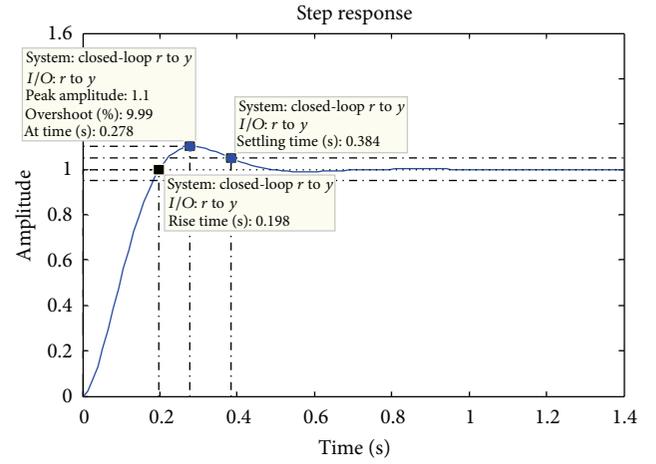


FIGURE 5: The unit step response of control system with full-order observer in feedforward path.

Also, we choose the desired observer poles at $s = -680$, $s = -680$. These poles create a stable controller. The state feedback gain matrix K and the observer gain matrix K_e can be obtained as follows:

$$K = [-247.3389 \quad -0.019], \quad (21)$$

$$K_e = [1326.67 \quad 191882.08]^T.$$

The transfer function of the controller observer is obtained as follows:

$$G_s(s) = \frac{-331800s - 52710000}{s^2 + 1343s + 213300}. \quad (22)$$

If the full-order observer controller is placed in the feedforward path, from this block diagram, the closed-loop transfer function (CLTF) is obtained as follows:

$$CLTF = \frac{N(s^2 9.43e - 9 - 303300000s - 4.818e10)}{(s^4 + 1376s^3 + 484300s^2 + 7647000s + 84630000)}. \quad (23)$$

We can determine the value of the gain N , such that for a unit step input r , the output y is unity as time approaches infinity. Thus we choose the following:

$$N = \frac{84630000}{-4.818e10} = -1.7565e - 3. \quad (24)$$

The unit step response of the system is shown in Figure 5. Notice that the maximum overshoot is 9.99% and the settling and rise times are 0.384 and 0.198 second, respectively.

If we choose the full observer controller in the feedback path, then, the closed-loop transfer function is obtained as follows:

$$CLTF = \frac{N(914.2s^2 + 1227000s + 1.95e8)}{(s^4 + 1376s^3 + 484300s^2 + 7647000s + 84630000)}. \quad (25)$$

TABLE 2: Structural and training parameters of neural controllers.

Controller	η	μ	α	β	n	N	A_F
NC	0.0001	0.01	0.9	0.8	8 + 8	30000	H_T

Note. η : learning term; μ : momentum term; α : feedback gain from output layer to hidden layer; β : feedback gain from hidden layer to itself; n : linear + nonlinear neurons in the hidden layer; N : iteration numbers; A_F : activation function for non-linear neurons; H_T : hyperbolic tangent.

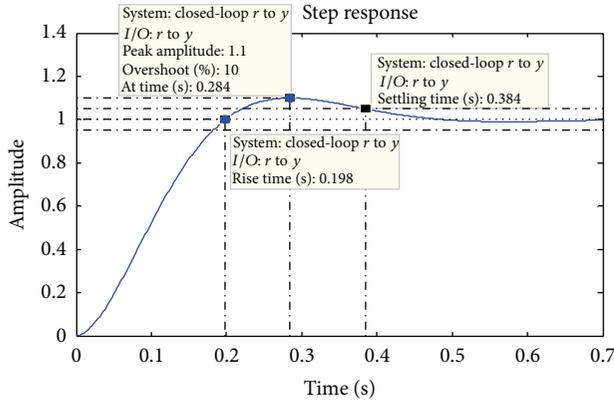


FIGURE 6: The unit step response of control system with full-order observer in feedback path.

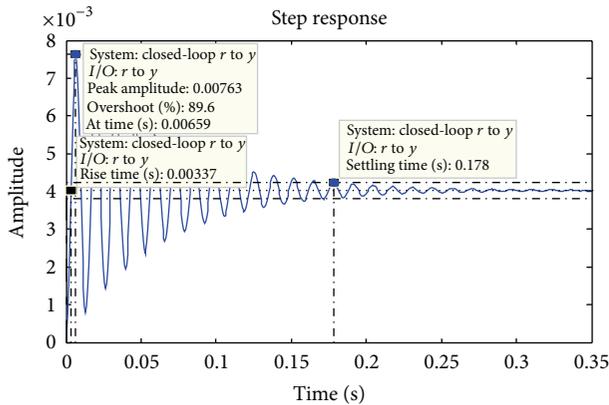


FIGURE 7: The unit step response of valve without any controller.

Thus, we can determine the value of the gain N as follows:

$$N = \frac{84630000}{1.95e8} = 0.434. \quad (26)$$

The unit step response of control system with full-order observer in feedback path is shown in Figure 6. Notice that the maximum overshoot is 10% and the settling and rise times are 0.384 and 0.198 second, respectively.

Figure 7 shows the step response of valve stem without any controller. Here, the overshoot is too high and the settling time is also 0.178 second. The output has an overshoot less than 90% and rise time is 0.00337 second. But the design of full- and minimum order observer controller for valve led to the well unit step response characteristic.

The valve system was controlled using the proposed neural control system. Structural and learning parameters of the proposed neural network are given in Table 2.

If the learning rate is made too large, the algorithm becomes unstable. If the learning rate is set too small, the algorithm takes a long time to converge. Usually the learning rate is a constant real number between 0.1 and 1. It is not practical to determine the optimal setting for the learning rate before training and, in fact, the optimal learning rate changes during the training process, as the algorithm moves across the performance surface. The Mean square error (MSE) is defined as performance of the net. A backpropagation (BP) algorithm is designed to reduce error between the actual output and the desired output of the network in a gradient descent manner [13]. The hidden layer is responsible for internal representation of data and the information transformation between input and output layers. If there are too few neurons in the hidden layer, the network may not contain sufficient degrees of freedom to form a representation (i.e., in sufficient learning capacity). If too many neurons are defined, the network may become over trained (i.e., they classify training patterns well but lack the ability to generalize other independent data). Therefore, an interesting design for the number of neurons in the hidden layer, to determine the optimum network, will be important. Momentum can be added to backpropagation learning by making weight changes equal to the sum of a fraction of the last weight change and the new change suggested by the back propagation rule. The magnitude of the effect that the last weight change is allowed to have is mediated by a momentum constant, which can be any number between 0 and 1. So, here, different forms in the hidden layers are considered for the network. As a final result, the network with 16 neurons in hidden layer, learning, and momentum term in Table 2 has minimum performance with $MSE = 0.02$.

From Figure 8, the overshoot is initially very small. The output has an overshoot less than 7% and settling time shorter than 0.4 s.

5. Conclusion

Actuator valve systems are multivariable dynamic systems for which it is difficult to derive mathematical models. Therefore, analytical control schemes based on such models are complex to construct and generally do not perform well in practice. This paper has described a proposed neural control scheme for actuator of the pneumatic control valve. The aim of this study is the development of design procedure of an NN controller to meet transient response specifications of a sliding-stem pneumatic control valve. A type of actuator pneumatic control valve is modelled using bondgraph method. For comparison, full- and minimum order observer controllers are applied to the control of stem position in

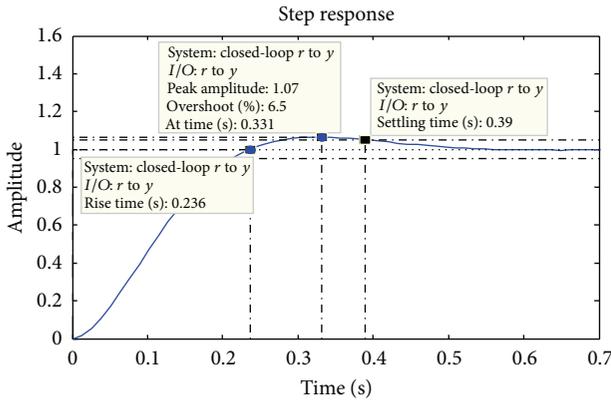


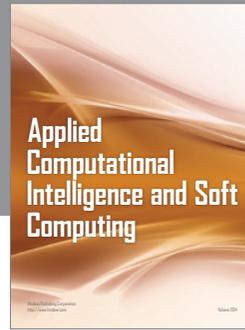
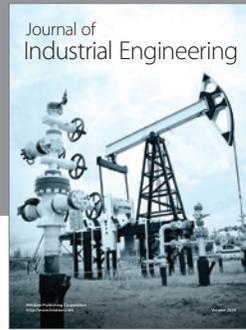
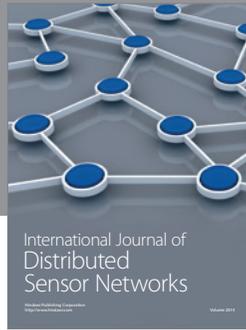
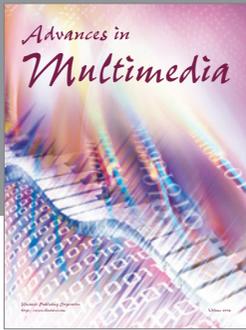
FIGURE 8: The unit step response of the valve using proposed neural controller.

a pneumatic control valve system. From this study, the following conclusions can be drawn.

- (1) NN controller has smaller overshoot rather than minimum- and full-order observer.
- (2) Minimum observer controller has smaller settling time and rise time respect to full-order observer.
- (3) The full- and minimum order observers in feedback and feedforward path have the same overshoot, rise time, and settling time, if the normalization constants (N) are not the same.

References

- [1] T. Hägglund, "A friction compensator for pneumatic control valves," *Journal of Process Control*, vol. 12, no. 8, pp. 897–904, 2002.
- [2] M. A. De Souza L. Cuadros, C. J. Munaro, and S. Munareto, "Improved stiction compensation in pneumatic control valves," *Computers and Chemical Engineering*, vol. 38, pp. 106–114, 2012.
- [3] R. P. Champagne and S. J. Boyle, "Optimizing valve actuator parameters to enhance control valve performance," *ISA Transactions*, vol. 35, no. 3, pp. 217–223, 1996.
- [4] H. Paynter, *Analysis and Design of Engineering Systems*, MIT Press, Cambridge, Mass, USA, 1959.
- [5] D. C. Karnopp, R. C. Rosenberg, and D. L. Margolis, *System Dynamics: Modeling, Simulation, and Control of Mechatronic Systems*, John Wiley & Sons, New York, NY, USA, 5th edition, 2012.
- [6] J. U. Thoma, *Simulation by Bondgraphs: Introduction to a Graphical Method*, Springer, Berlin, Germany, 2012.
- [7] P. Athanasatos and T. Costopoulos, "Proactive fault finding in a 4/3-way direction control valve of a high pressure hydraulic system using the bondgraph method with digital simulation," *Mechanism and Machine Theory*, vol. 50, pp. 64–89, 2012.
- [8] V. D. Zuccarini, D. Rafirou, J. LeFevre, D. R. Hose, and P. V. Lawford, "Systemic modelling and computational physiology: the application of bondgraph boundary conditions for 3D cardiovascular models," *Simulation Modelling Practice and Theory*, vol. 17, no. 1, pp. 125–136, 2009.
- [9] O. Ekren, S. Sahin, and Y. Isler, "Comparison of different controllers for variable speed compressor and electronic expansion valve," *International Journal of Refrigeration*, vol. 33, no. 6, pp. 1161–1168, 2010.
- [10] S. B. Choi, C. C. Cheong, J. M. Jung, and Y. T. Choi, "Position control of an er valve-cylinder system via neural network controller," *Mechatronics*, vol. 7, no. 1, pp. 37–52, 1997.
- [11] J. C. Mackanic, *Design, Construction and Evaluation of a Simulated Geothermal Flow System*, University of California, Berkeley, Calif, USA, 1980.
- [12] K. Ogata, *Modern Control Engineering*, Prentice Hall, Upper Saddle River, NJ, USA, 5th edition, 2010.
- [13] S. Zerkaoui, F. Druaux, E. Leclercq, and D. Lefebvre, "Stable adaptive control with recurrent neural networks for square MIMO non-linear systems," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 4-5, pp. 702–717, 2009.
- [14] M. Hernandez and Y. Tang, "Adaptive output-feedback decentralized control of a class of second order nonlinear systems using recurrent fuzzy neural networks," *Neurocomputing*, vol. 73, no. 1-3, pp. 461–467, 2009.
- [15] H. W. Ge, W. L. Du, F. Qian, and Y. C. Liang, "Identification and control of nonlinear systems by a time-delay recurrent neural network," *Neurocomputing*, vol. 72, no. 13-15, pp. 2857–2864, 2009.
- [16] H. Demuth and M. Beale, *Matlab Neural Networks Toolbox, User's Guide*, The MathWorks, Natick, Mass, USA, 2001, <http://www.mathworks.com/>.
- [17] N. S. Nise, *Control System Engineering*, John Wiley & Sons, New York, NY, USA, 6th edition, 2010.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

