

Development of Quantification Algorithm for Control Valve Stiction

H. Zabiri*, A. Maulud, N. Omar, M. Ramasamy
 Chemical Engineering Department, Universiti Teknologi PETRONAS
 Bandar Sri Iskandar, Tronoh,
 31750 Perak
 MALAYSIA

*Corresponding author: haslindazabiri@petronas.com.my

ABSTRACT

Control valve stiction is the most commonly found valve problem in the process industry. Though many methods have been developed in detecting stiction, quantification of the actual amount of stiction presents in a loop is still an open research area. In this paper, Neural-network techniques are investigated in the development of a quantification algorithm for control valve stiction. It is shown that in the presence of a well-tuned controller, satisfactory performance of the proposed Neural-network based quantification algorithm can be achieved.

KEY WORDS

Control valve stiction, quantification, Neural-network

1. Introduction

Control valves constitute an important element in chemical process control systems. Through a control valve, control actions are implemented on the process. Due to their continuous motions, control valves tend to undergo wear and aging. In general, they contain static and dynamic nonlinearities including saturation, backlash, stiction, deadband and hysteresis [1-2].

Among the many types of nonlinearities in control valves, stiction is the most commonly encountered in the process industry [3]. In general, stiction is a phenomena that describes the valve's stem (or shaft) sticking when small changes attempted. Stiction causes fluctuation of process variables, which lowers productivity. The variability of process variables makes it difficult to keep operating conditions close to their constraints, and hence causes excessive or unnecessary energy consumption. It is therefore desirable to understand and quantify the dynamics behavior of stiction so that necessary actions can be implemented to eliminate or hinders its deleterious effect before it propagates.

Detection of stiction nonlinearity in a loop has been extensively studied in the literatures, however quantification of the actual amount of stiction presents is still an open research area [4]. Srinivasan [5] uses a Hammerstein model identification approach along with one parameter stiction model (stickband plus deadband, S estimation) to detect and quantify valve stiction. However

this method does not capture the true stiction behavior [4]. On the other hand, Choudhury [6] proposed three methods for quantifying stiction utilizing valve positioner data (mv), controlled output (pv) and valve input signal (op). Problems such as the unavailability of mv and process loop dynamics limit the performances of the proposed methods. An extended version of [6] that includes the loop dynamics is proposed in [4] and [7] using *two* parameter stiction estimation. Both these methods used Hammerstein model to simultaneously predict process model and quantify stiction in control valve.

In this paper, a similar algorithm used in [4] is adopted, and extended to investigate the probability of using Neural-network techniques in quantifying the control valve stiction. The outline of this paper is as follows: Section II describes stiction in general. In Section III, six Neural-Network algorithms considered in this paper are presented. Section IV illustrates the proposed quantification algorithm. Finally, the conclusions are drawn.

2. Control valve Stiction

Fig. 1 shows the general structure of a pneumatic control valve. Stiction happens when the smooth movement of the valve stem is hindered by excessive static friction at the packing area. The sudden slip of the stem after the controller output sufficiently overcomes the static friction caused undesirable effect to the control loop.

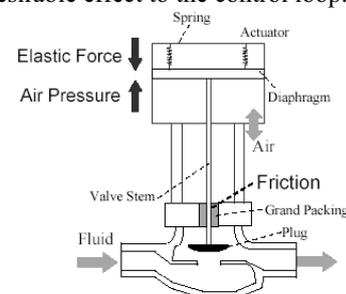


Fig. 1 Structure of pneumatic control valve adapted from [8].

Fig. 2 illustrates the input-output behavior for control valve with stiction. The dashed line represents the ideal control valve without any friction.

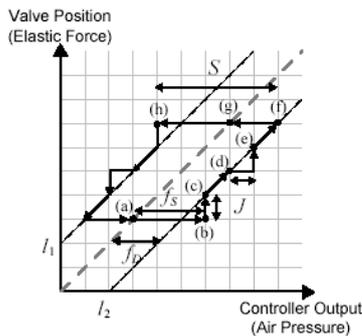


Fig. 2 Typical input-output behavior of a sticky valve adapted from [8].

Stiction consists primarily of deadband, stickband, slip jump and the moving phase [9]. For control valve under stiction resting at point (a), the valve position remains unchanged even when the controller output increases due to the deadband caused by the static friction. Only when the controller output exceeds the maximum static frictional force, f_s , the valve starts to response (point(b)). A slip jump of magnitude J is incurred when the valve starts to move at point (b) when the frictional force f_s converts to kinetic force f_d . From (c) to (d), the valve position varies linearly. The same scenario happens when the valve stops at point (d), and when the controller output changes direction.

Stiction in control valves can either be modeled via physics-based or data driven [4]. Due to the complex nature of the physics-based approach, data-driven modeling technique is highly favorable. In this paper, the widely acknowledged two parameter stiction model developed by Choudhury et al. [3] is used to model and describe the stiction nonlinearity. The two parameters involved in this model are S (stickband+deadband) and J (slip-jump) – see Fig. 2. The model needs only the input signal or the controller output (op) and the specifications of S and J . For more details on the two parameter stiction model, readers are referred to Choudhury et al. [3].

3. Neural Network

An artificial neural network (ANN) or commonly just neural network (NN) is an interconnected group of artificial neurons that uses a mathematical model or computational model for information processing based on a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network. In this paper, six types of NN for quantifying the control valve stiction are investigated.

3.1 Feedforward-backpropagation Neural Network

Feedforward backpropagation neural networks (FF networks) are the most popular and most widely used models in many practical applications [10]. They are known by many different names, such as "multi-layer

perceptrons." The following diagram illustrates a FF networks network with three layers:

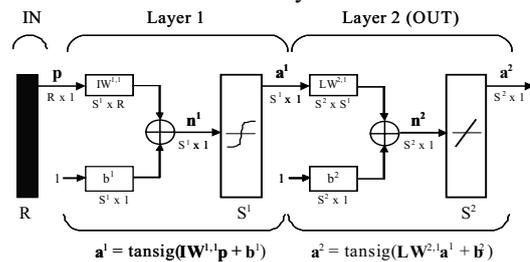


Fig. 3 Graphical representation of a BP network architecture.

Backpropagation (BP) network was created by generalizing the Widrow-Hoff learning rule to multiple-layer networks and nonlinear but differentiable transfer functions [11]. BP network with biases, a sigmoid ('tansig' or 'logsig') transfer functions at the hidden layers, and a linear transfer function at the output layer is capable of approximating any functions. BP networks architecture is slightly more complex than a single layer network. In addition to a single (hidden) layer consisting nodes with sigmoid transfer function, another layer called the output layer is required. The output layer is usually kept linear to produce output values in the similar range as the target values. However, the sigmoid transfer functions (either 'logsig' or 'tansig') are often used if the outputs need to be constrained to the range of $[0,1]$ or $[-1,1]$. The minimum architecture of BP networks is illustrated as layer diagram in Fig. 3. The $(R \times 1)$ inputs p are fed to Layer 1 (hidden layer) consisting of S^1 'tansig' nodes. The resulting outputs a^1 with 'linear' transfer function retain the same size ($S^2 \times 1$) as the net inputs n^2 to Layer 2 (output layer). With this architecture, the BP networks are capable of approximating any linear and nonlinear functions given adequate number of hidden nodes.

3.2 Cascade-forward Backpropagation Network

Feedforward networks have one-way connections from input to output layers. They are most commonly used for prediction, pattern recognition, and nonlinear function fitting. Supported feedforward networks include feedforward backpropagation and cascade-forward backpropagation. In CF network, each subsequent layer has weights coming from the input as well as from all previous layers.

Like FF networks, CF networks uses BP algorithm for updating of weights but the main symptoms of the network is that each layer neurons related to all previous layer neurons. In [12], several NN topologies were evaluated and it was found that the cascade forward NN with BP training provides the best performance in terms of convergence time, optimum network structure and recognition performance. The training of multi-layer perceptron (MLP) networks normally involves BP training as it provides high degrees of robustness and generalization [13].

3.3 Recurrent Neural Network

In Feedforward NN, the neurons in one layer receive inputs from the previous layer. Neurons in one layer deliver its output to the next layer; the connections are completely unidirectional; whereas in Recurrent NN, some connections are present from a layer to the previous layers. The next value of output is regressed on previous values of input signal (see Fig.4).

3.3.1. NARX Network

The nonlinear autoregressive network with exogenous inputs (NARX) is a recurrent dynamic network, with feedback connections enclosing several layers of the network.

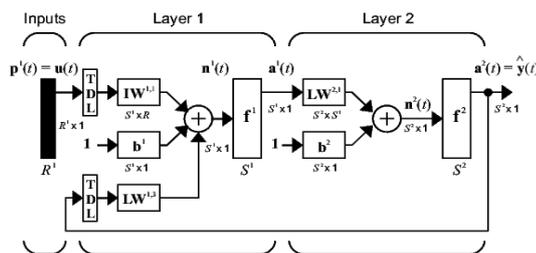


Fig. 4. Recurrent NARX NN structure.

The NARX model is based on the linear ARX model, which is commonly used in time-series modeling. The defining equation for the NARX model is shown in (1), where the next value of the dependent output signal $y(t)$ is regressed on previous values of the output signal and previous values of an independent (exogenous) input signal.

$$y(t) = f(y(t-1), y(t-2), \dots, y(t-n_y), u(t-1), u(t-2), \dots, u(t-n_u)) \quad (1)$$

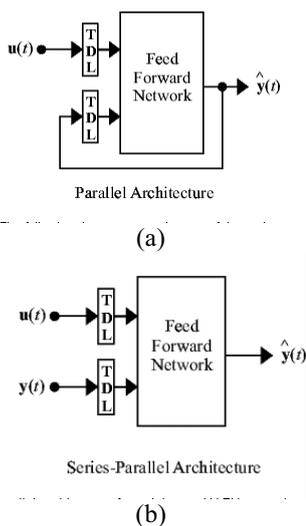


Fig. 5. NARX network architecture.

Standard NARX architecture is as shown in Fig. 5(a). It enables the output to be fed back to the input of the feedforward neural network. This is considered a feedforward BP network with feedback from output to

input. In series parallel architecture(NARXSP), Fig. 5(b), the true output which is available during the training of the network is used instead of feeding back the estimated output. The advantage is that the input to the feedforward network is more accurate. Besides, the resulting network has a purely feedforward architecture, and static BP can be used for training.

3.3.2. Simple Recurrent Network (SRN)

Simple Recurrent Network (SRN) is also known as Elman network. In Elman network, the input vector is similarly propagated through a weight layer but also combined with the previous state activation through an additional recurrent weight layer. A two-layer Elman network is shown as in Fig.6.

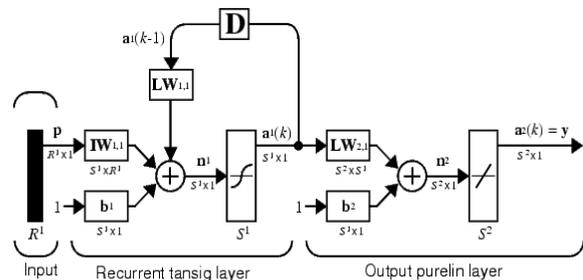


Fig. 6. Elman network structure.

The output of the network is determined by the state and a set of output weights, W ,

$$y_k(t) = f(net_k(t))$$

$$net_k(t) = \sum_j^m y_j(t)w_{kj} + \theta_k \quad (2)$$

Elman network has activation feedback which embodies short-term memory. A state layer is updated through the external input of the network as well as the activation from the previous forward propagation. The feedback is modified by a set of weights as to enable automatic adaption through learning (e.g. BP). Elman network differs from conventional two-layer networks in that the first layer has a recurrent connection. The delay in this connection stores values from the previous time step, which can be used in the current time step. Because the network can store information for future reference, it is able to learn temporal patterns as well as spatial patterns. The Elman network can be trained to respond to, and to generate, both kinds of patterns.

3.3.3. Layer-recurrent Network (LRN)

An earlier simplified version of this network was introduced by Elman. In the LRN, there is a feedback loop, with a single delay, around each layer of the network except for the last layer. The original Elman network had only two layers. The original Elman network was trained using an approximation to the BP algorithm. Fig. 7 illustrates a two-layer LRN.

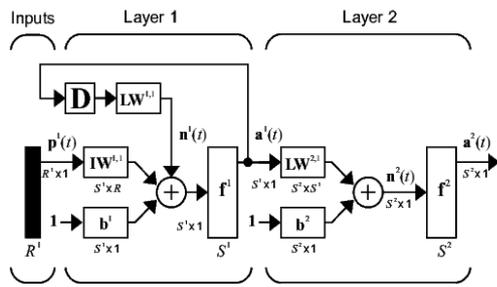


Fig. 7. Layer-recurrent neural network structure.

LRN generalizes the Elman network to have an arbitrary number of layers and to have arbitrary transfer functions in each layer. LRN is trained using exact versions of the gradient-based algorithms used in BP.

4. Quantification algorithm

4.1 Case study description

Case study used in [4] is used for simulating the proposed method as in Fig. 8:

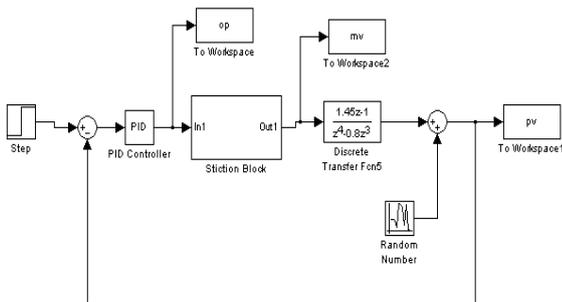


Fig. 8. Simulink block diagram used for generating stiction data adapted from [4].

The process model is:

$$G(z) = \frac{1.45z - 1}{z^4 - 0.8z^3} \tag{3}$$

The controller applied is as the following:

$$C(s) = K_c \left(1 + \frac{1}{\tau_i s} \right) \tag{4}$$

Three different sets of data are generated using $K_c=0.05$, $K_c=0.10$ and $K_c=0.15$. The integral parameter, τ_i is fixed at 1. The two parameter stiction model described in [3] is used, and the values of stiction parameters S and J are fixed at 3 and 1 respectively.

4.2 Process model prediction

In this section, the six types of NN are used to predict the process model and the prediction results are compared. In this analysis, we consider the case of stiction undershoot ($S > J$) is used with $K_c=0.05$. In this case, mv and pv data of $S=3$ and $J=1$ are generated. The model structures for

each of the NN types are initially analyzed and the optimized architecture is selected.

Figures 9 -14 show the results for the six stiction models. All NN are able to predict the process output satisfactorily. However, elman and layer recurrent NN failed to continuously track the actual process output when it travels to the top and bottom peaks. There is also a slight deviation at the top peaks of both NN when the signal is at steady state mode. The remaining four NN figures show comparable visual results. Therefore, statistical analysis is used to choose the best architecture. RMSE and CDC for all six types of NN are tabulated in Table 1.

From the table, RMSE for feedforward backpropagation, cascade forward backpropagation, NARX and NARXSP NN shows different values but close to each other. This is expected because of the close visual results of the four NNs. However, CDC values show greater deviation and are considered in the screening process. From the analysis, it is clear that NARXSP has the lowest RMSE value (0.044) and highest CDC value (44.1077). As a result, NARXSP is concluded as the best process model to be used in the estimation algorithm.

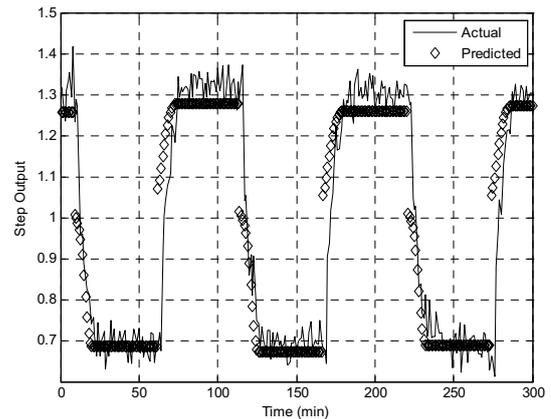


Fig. 9. Actual and predicted process output using layer NN.

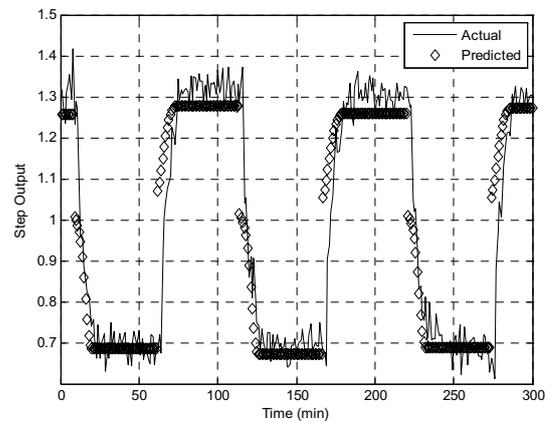


Fig. 10 Actual and predicted process output using elman NN.

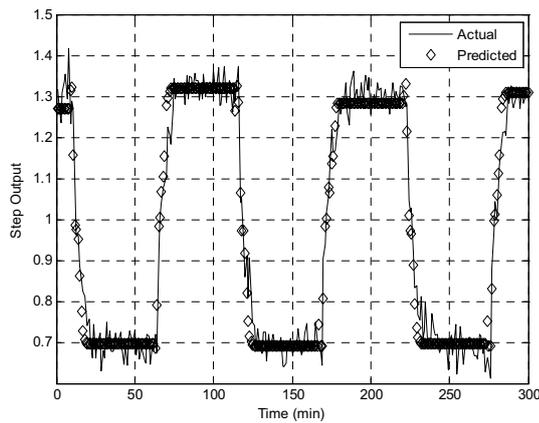


Fig. 11. Actual and predicted process output using feedforward BP NN.

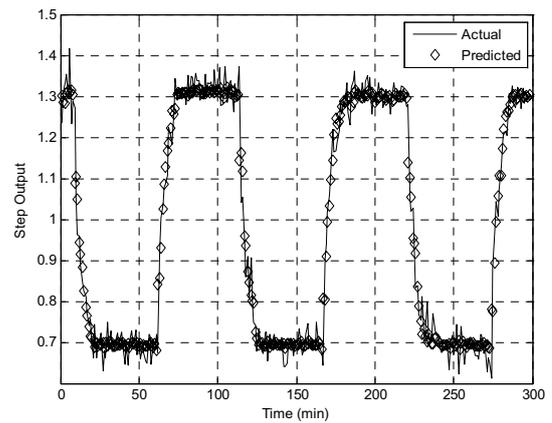


Fig. 14. Actual and predicted process output using NARXSP NN.

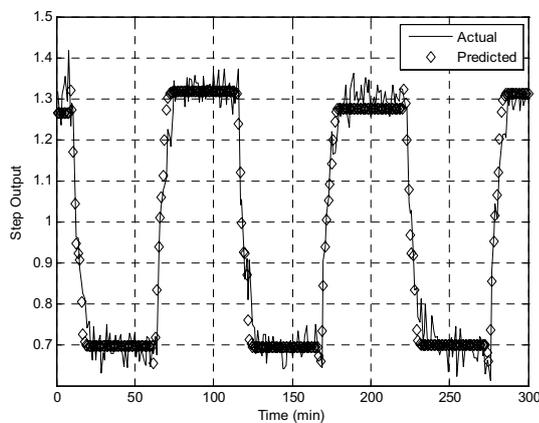


Fig. 12. Actual and predicted process output using cascade forward BP NN.

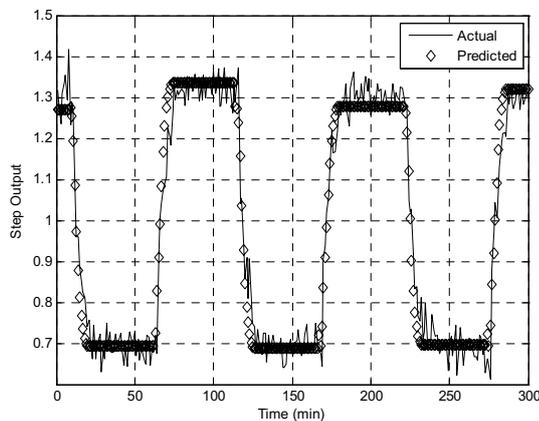


Fig. 13 Actual and predicted process output using NARX NN.

Table 1. Statistical analysis for NN

Neural Network Model	RMSE	CDC
Elman	0.1078	18.0602%
LRN	0.1078	18.0602%
FF	0.0454	22.0736%
CF	0.0466	30.1003%
NARX	0.0499	30.7692%
NARXSP	0.0440	44.1077%

4.3 Quantification of stiction: S and J estimation

Since the process model already being identified in the previous section, the next step is to estimate or quantify the actual amount of S (stickband+deadband) and J (slip-jump) present in a sticky valve. Figure 15 shows the flow chart of the procedure.

The controller output, op and process variable, pv are the problematic loop data. These data is typically available in the industrial plant. The next step is to guess initial value of S and J . In this case, S and J are assumed to be in the range of 0 to 10 ($0 < S_1 < 10$, $0 < J_1 < 10$). Then, control valve output, mv data is generated using Simulink block as in Figure 16, using two parameter stiction model of [3]. The mv and pv are the inputs to the NARXSP NN process model to predict the corresponding pv_{pred} . $RMSE_1$ is calculated for the difference between pv and pv_{pred} values. The next step is to choose S_2 (ie $S_2 < S_1$) with constant J and repeat the steps until $RMSE_2$ calculation. If $RMSE_2$ is greater than $RMSE_1$, all values of $S < S_1$ are discarded since they will give larger errors. Now, S value have been narrowed down to $S_1 < S < 10$. The same procedures are repeated until minimum RMSE is calculated. The same procedure is then applied to estimate J using S with the lowest RMSE. The final value S and J are report as stiction.

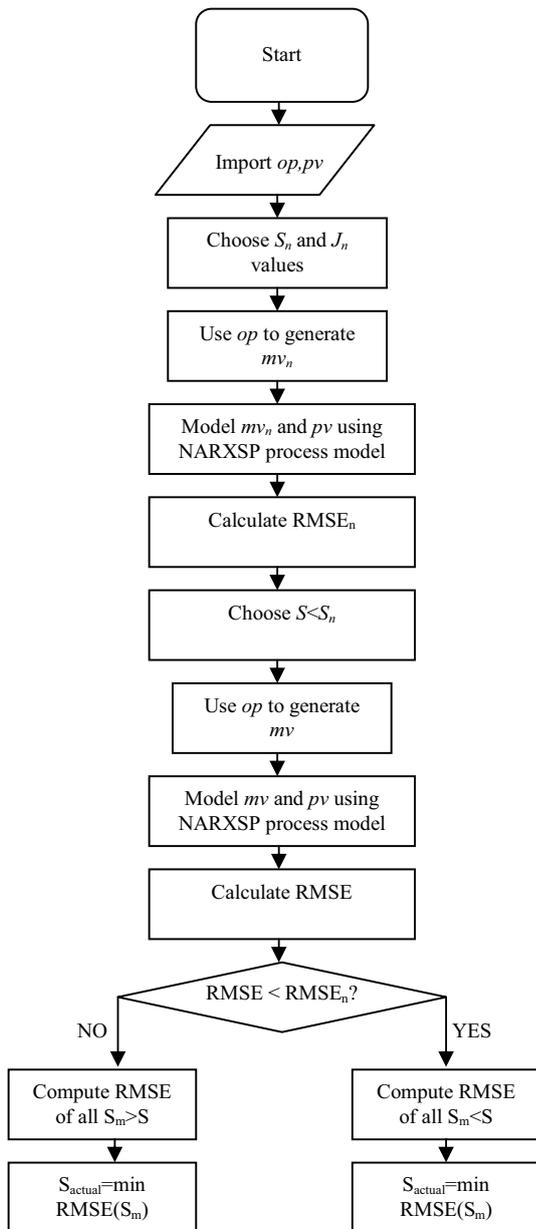


Fig. 15. Flow chart for S and J estimation.

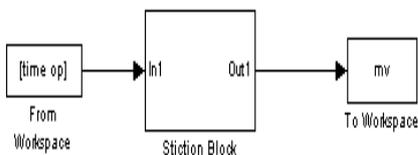


Fig. 16. Simulink block for generating mv data.

4.4 Numerical evaluations

Three sets of data (for $K_c=0.05$, $K_c=0.10$, and $K_c=0.15$, respectively) are generated using Figure 8 where the stiction parameters are fixed at $S=3$ and $J=1$. The different K_c values are imperative to evaluate the robustness of the estimation algorithm against varying operating conditions. Random noise with zero mean is also added to further corrupt the data. The op from these data is the input to generate mv using Figure 16. The mv generated together with pv are the inputs to the NARXSP NN process model. Using the estimation algorithm described in Fig. 15, the RMSE for every case is tabulated in Table 2. For all three cases of $K_c=0.05$, $K_c=0.10$, and $K_c=0.15$, the estimation algorithm using NARXSP NN correctly and efficiently quantified the amount of stiction that exists in the system. In all three cases, $S=3$ and $J=1$ are detected which are the right value for all three cases.

Table 2 (a). Statistical analysis for $K_c=0.05$.

1 st level			
Combination	S5J1	S2J1	S1J1
RMSE	0.0544	0.0479	0.0500
2 nd level			
Combination	S2J1	S3J1	S4J1
RMSE	0.0479	0.0409	0.0541
3 rd level			
Combination	S3J1	S3J2	
RMSE	0.0409	0.0410	

Table 2 (b). Statistical analysis for $K_c=0.10$.

1 st level			
Combination	S5J1	S2J1	S1J1
RMSE	0.1089	0.0978	0.0937
2 nd level			
Combination	S1J1	S3J1	S4J1
RMSE	0.0937	0.0912	0.1002
3 rd level			
Combination	S3J1	S3J2	
RMSE	0.0912	0.0921	

Table 2 (c). Statistical analysis for $K_c=0.15$.

1 st level			
Combination	S5J1	S2J1	S1J1
RMSE	0.6030	0.6019	0.5434
2 nd level			
Combination	S1J1	S3J1	S4J1
RMSE	0.5434	0.4888	0.6119
3 rd level			
Combination	S3J1	S3J2	
RMSE	0.4888	0.6053	

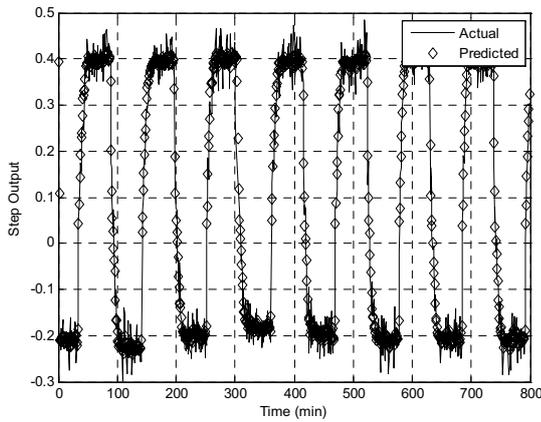


Fig. 17. Actual and predicted process output for $K_c=0.05$.

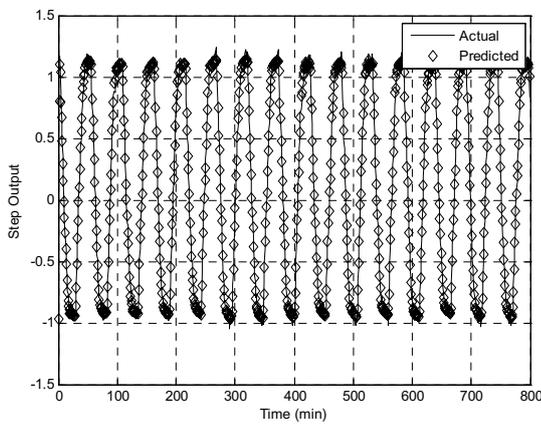


Fig. 18. Actual and predicted process output for $K_c=0.10$.

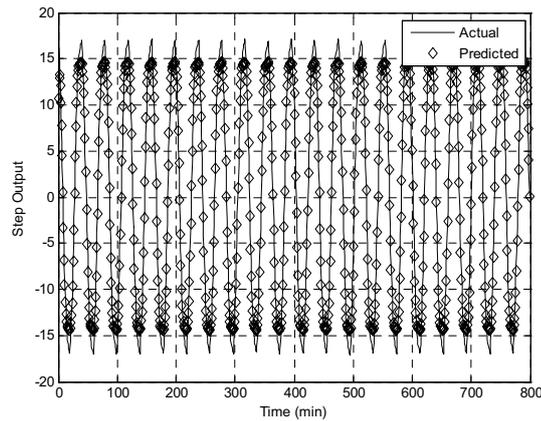


Fig. 19. Actual and predicted process output for $K_c=0.15$.

5. Conclusion

In this paper, a Neural-network based stiction quantification algorithm has been developed using routine operating data. Results show that the method performs

satisfactorily in quantifying the two parameter stiction model.

Acknowledgements

The authors would like to thank Universiti Teknologi PETRONAS for the funding provided for this work.

References

- [1] M. A. A. S. Choudhury, S. L. Shah, & N. F. Thornhill, Diagnosis of poor control-loop performance using higher-order statistics, *Automatica*, 40, 2004, 1719-1728.
- [2] Zabiri, H., and Samyudia, Y. (2006). A hybrid formulation and design of model predictive control for systems under actuator saturation and backlash. *Journal of Process Control*, 16, pp. 693-709.
- [3] Choudhury, M. A. A. S., Thornhill, N. F., and Shah, S. L. (2005). Modeling valve stiction. *Control Engineering Practice*, 13, pp. 641-658.
- [4] M.A.A. Shoukat Choudhury, Mridul Jain, Sirish L. Shah. (2008). Stiction-definition, modeling, detection and quantification. *Journal of Process Control*, 18, pp. 232-243.
- [5] R. Srivinasan, R. Rengaswamy, S. Narasimhan, R. Miller. (2005). Control loop performance assessment: Ii. Hammerstein model approach for stiction diagnosis. *Industrial & Engineering Chemistry Research*.
- [6] M.A.A.S. Choudhury, N.F. Thornhill, S.L. Shah, D.S. Shook. (2006). Automatic detection and quantification of stiction in control valves. *Control Engineering Practice* 14, 1395-1412.
- [7] Mohieddine Jelali. (2008). Estimation of valve stiction in control loops using separable least-squares and global search algorithms. *Journal of Process Control*, 18, pp. 632-642.
- [8] Kano, M., H., Kugemoto, H., and Shimizu, K. (2004). Practical model and detection algorithm for valve stiction. In *Proceedings of the Seventh IFAC-DYCOPS Symposium*, Boston, USA.
- [9] Choudhury, M A. A. S., Kariwala, V., Shah, S. L., Douke, H., Takada, H., and Thornhill, N. F. (2005). A simple test to confirm control valve stiction. *IFAC World Congress*, Praha.
- [10] Hagan M.T., Demuth H.B., Beale M.H. (1996). *Neural Network Design*, PWS Publishing Company, Boston, MA.
- [11] Radhakrishnan, V. R., Zabiri, H., and Thanh, D. V. (2006). Application of Multivariable Modeling in the Hydrocarbon Industry, *International Conference on Computer Process Control*, Lake Louise, Alberta Canada, January 10-16.
- [12] Qahwaji, R. and T. Colak, Neural Network-based Prediction of Solar Activities, in *CITSA2006: Orlando*. (2006).
- [13] Kim, J., Mowat, A., Poole, P., and Kasabov, N., "Linear And Non-Linear Pattern Recognition Models For Classification Of Fruit From Visible-Near Infrared Spectra", *Chemometrics And Intelligent Laboratory Systems*, 2000. 51: pp.,201-216.