NN-based Algorithm for Control Valve Stiction Quantification

H. ZABIRI, A. MAULUD, N. OMAR
Chemical Engineering Department, Universiti Teknologi PETRONAS
Bandar Sri Iskandar, Tronoh,
31750 Perak
MALAYSIA
haslindazabiri@petronas.com.my

Abstract: - Control valve stiction is the most commonly found valve problem in the process industry. Quantification of the actual amount of stiction present in a loop is an important step that may help in scheduling the optimum maintenance work for the valves. In this paper, a Neural-network based stiction quantification algorithm is developed. It is shown that the performance of the proposed quantification algorithm is comparable to other method whereby accurate estimation of the stiction amount can be achieved even in the presence of random noise. Its robustness towards external oscillating disturbances is also investigated.

Key-Words: - Control valve, stiction, neural network, quantification

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 are 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 hinder its deleterious effect before it propagates.

Detection and modeling of stiction nonlinearity in a control loop has been extensively reported in the literature, however quantification of the actual amount of stiction is still an open research area [4]. Srinivasan *et al.* [5] uses a Hammerstein model identification approach along with one parameter stiction model (stickband plus deadband estimation) to detect and quantify valve stiction. However this method does not capture the true stiction behavior [4]. On the other hand, Choudhury *et al.*[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, however we investigated the possibility of incorporating Neural-network to simultaneously identify/predict the unknown process model and quantify the stiction amount. In this NN-based quantification algorithm, the actual valve positioner data (*mv*) is not required and no *a-priori* knowledge of the process model is necessary. Only *pseudo-mv*, generated from the two parameter stiction model of [3], is used as part of the quantification step. 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 and its numerical results. Finally, the conclusions are drawn.

2 Background

2.1 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.

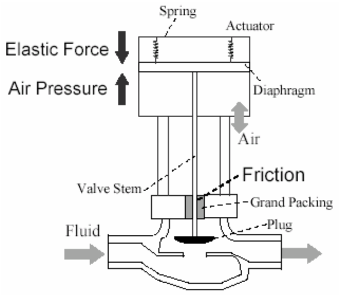


Fig. 1 Structure of pneumatic control valve adapted from [8].

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

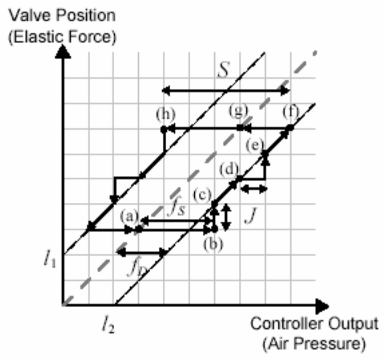


Fig. 2 Typical input-output behavior of a sticky valve adapted from [8].

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].

2.2 Neural Network

An artificial neural network (ANN) or commonly known as 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 [10,11,12]. To develop the final quantification algorithm for stiction, a good NN type needs to be identified upfront that can provide the most accurate identification and prediction for (any) unknown process model. This is particularly a very important step of the algorithm development. Even though any NN is well-known for its capability of predicting any process behavior, high level prediction accuracy is imperative here since any inaccurate prediction, given by the selected NN, may result in ambiguous prediction error and the determination of the actual amount of stiction present might be cumbersome.

For the purpose of the quantification algorithm development, the input data considered will be the *pseudo-mv* (*pseudo*-valve positioner data) and the actual process variable data, *pv*. In this paper, we consider the popular and well-known NN types, namely, Feedforward-Backpropagation NN and Recurrent NN, as well as Elman and Layer Recurrent Networks and Cascade Networks.

2.2.1 Feed-forward Backpropagation NN

Feedforward backpropagation neural networks (FF networks) are the most popular and most widely used models in many practical applications [9]. They are known by many different names, such as "multi-layer perceptrons." Fig. 3 illustrates a FF networks network with three layers.

FF network was created by generalizing the Widrow-Hoff learning rule to multiple-layer networks and nonlinear but differentiable transfer functions [13]. FF 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 function to an arbitrary accuracy.

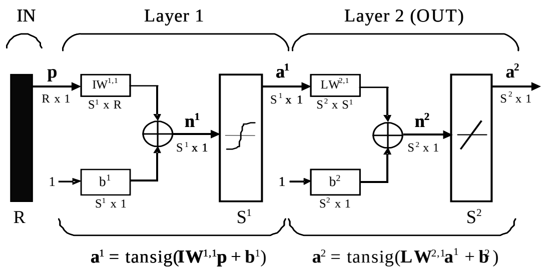


Fig. 3 Graphical representation of a BP network architecture.

FF network 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 FF network is illustrated as layer diagram in Fig. 3. The (R x 1) inputs p are fed to Layer 1 (hidden layer) consisting of S^1 'tansig' nodes. The resulting outputs a^2 with 'linear' transfer function retain the same size (S^2 x 1) as the net inputs n^2 to Layer 2 (output layer). With this architecture, the FF networks are capable of approximating any linear and nonlinear functions given adequate number of hidden nodes.

2.2.2 Recurrent NN

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).

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

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.

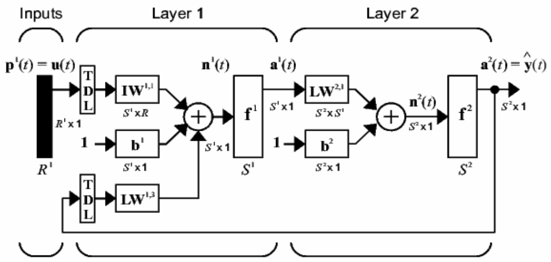
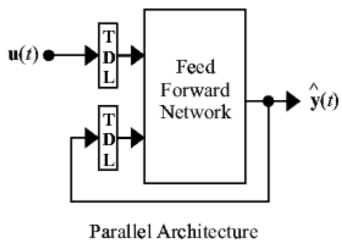
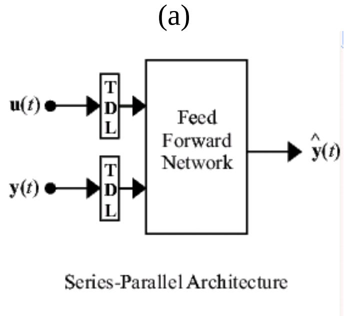


Fig. 4 Recurrent NARX NN structure.

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



(a)



(b)

Fig. 5 NARX network architecture.

2.2.3 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.

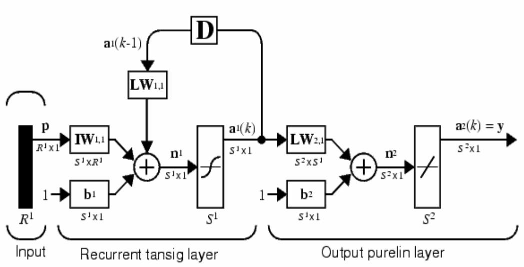


Fig. 6 Elman network structure.

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

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

$$net_k(t) = \sum_{j}^{m} y_j(t) w_{kj} + \theta_k$$

(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.

2.2.4 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.

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.

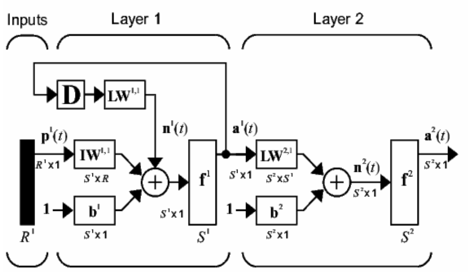


Fig. 7 Layer-recurrent neural network structure.

2.2.5 Cascade-forward backpropagation Network (CF)

Feedforward networks have one-way connection 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.

3 Quantification Algorithm

In this section, the development of quantification algorithm is presented. The best type of NN that can efficiently identify/predict the process model from the *pseudo-mv* and *pv* data is initially determined. This NN is then incorporated as part of the final quantification algorithm.

The performance of the NN-based quantification algorithm is then exhibited for estimating the unknown stiction amount presents in the case study selected. Analysis of the quantification algorithm performance under varying unknown stiction amount plus external oscillating disturbances is also presented.

3.2 Case Study Description

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

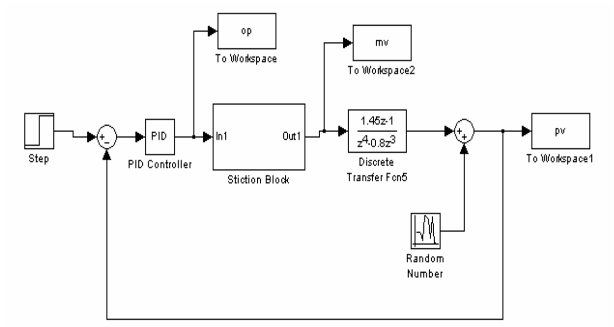


Fig.8. Simulink block diagram used for generating stiction data adapted from [4].

The process model is:

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

A proportional-integral (PI) controller is used.

3.3 NN selection analysis for process model identification

As mentioned in the earlier section, to develop the final quantification algorithm for stiction, a good NN type needs to be identified upfront that can provide the most accurate identification and prediction for (any) unknown process model. This is particularly a very important step of the algorithm development. Even though any NN is well-known for its capability of predicting any process behavior, high level prediction accuracy is imperative here since any inaccurate prediction, given by the selected NN, may result in ambiguous prediction error and the determination of the actual amount of stiction present might be cumbersome.

In this section, the capability of the six types of NN mentioned in the earlier section are investigated to identify an unknown process model from *pseudo-mv* (valve output) and *pv* (process variable) data, and the prediction error results are compared.

In this analysis, we consider the case of stiction undershoot (S>J) with K_c=*0.05* and 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.

The *pseudo-mv* and *pv* data of $S=3$ and $J=1$ are generated using the case study described in Fig. 8. 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, there is a slight deviation at the peaks of both NN when the signal is at steady state mode for feedforward BP NN.

To select the best NN type, statistical analysis is used to choose the best network type. Root Mean Squared Error (RMSE) and Correct Directional Change (CDC) for all types of NN are tabulated in Table 1.

From the table, RMSE for feedforward backpropagation and NARXSP NN shows different values but close to each other. This is expected because of the close visual results of both NNs. However, CDC values show greater deviation and are considered in the screening process.

TABLE 1.
STATISTICAL ANALYSIS FOR NN ARCHITECTURE

Neural Network Model	*RMSE*	*CDC*
FF	0.0454	22.07%
NARX	0.0499	30.76%
NARXSP	0.0440	44.10%
Elman	0.1078	18.06%
LRN	0.1078	18.06%
CF	0.0466	30.10%

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 NN type to be used in the process model identification/prediction stage for the quantification algorithm.

One disadvantage of NRAXSP NN that is widely acknowledged is that the model cannot be used independently from the plant, and only one-step ahead prediction can be achieved. Since the quantification algorithm is developed for offline analysis and no multi-step ahead prediction is necessary, the usage of NARXSP NN will not present any problem.

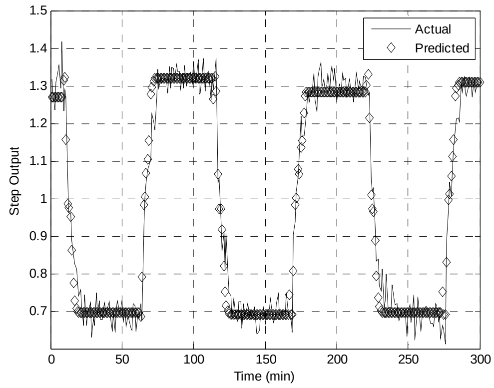


Fig. 9. Actual and predicted process output using feedforward BP NN.

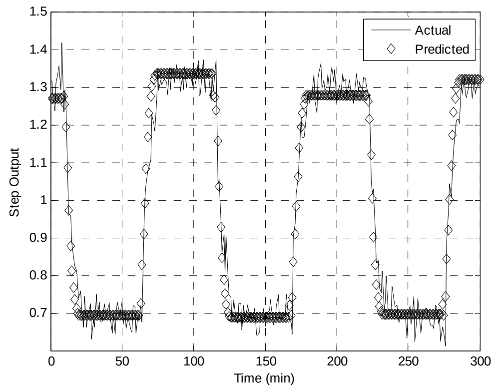


Fig. 10 Actual and predicted process output using NARX NN.

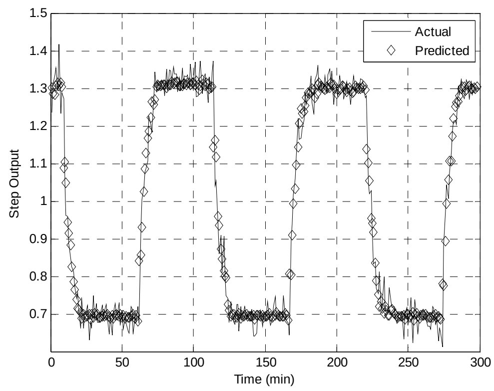


Fig. 11. Actual and predicted process output using NARXSP NN.

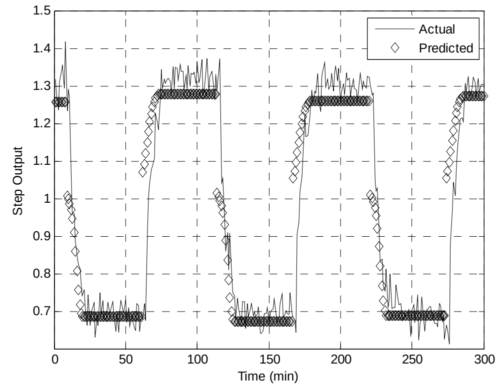


Fig. 12. Actual and predicted process output using layer NN.

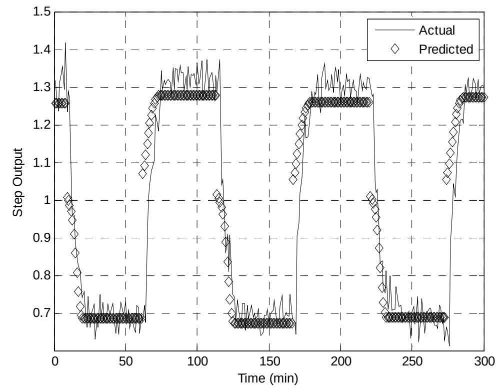


Fig. 13 Actual and predicted process output using elman NN.

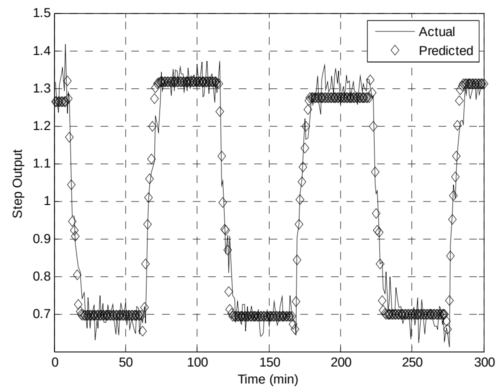


Fig. 14. Actual and predicted process output using cascade forward BP NN

3.4 Quantification of stiction: *S* and *J* estimation

Since the type of NN has been selected in the previous section for process model identification/prediction stage, the next step is to incorporate this into the quantification algorithm 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. This proposed procedure eliminates the need of the actual valve positioner data (*mv*).

The controller output, *op*, and process variable, *pv*, are the problematic loop data. These data are typically available in the industrial plant. The next step is to guess initial value of *S* and *J*.

Then, *pseudo*-valve positioner data, mv_n data is generated using Simulink block as in Figure 16, using two parameter stiction model of Choudhury et. al [3]. The pseudo-mv_n and *pv* are the inputs to the NARXSP NN for process model identification/prediction.

The resulting process model is then used 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 (i.e. $S_2<S_1$) 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. 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 reported as stiction.

3.5 Numerical Evaluation

For simulation purposes, 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 integral parameter, τ_i is fixed at 1. The two parameter stiction model described in [3] is again used, and the values of stiction parameters *S* and *J* are fixed at 3 and 1 respectively for all cases considered.

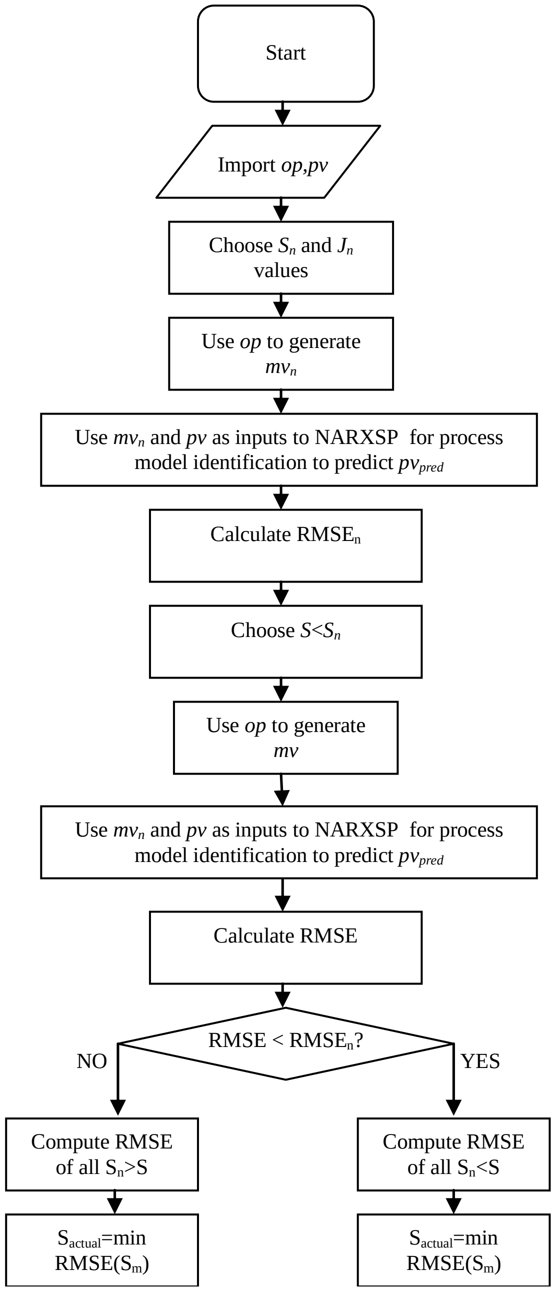


Fig. 15. Flow chart for *S* and *J* estimation.

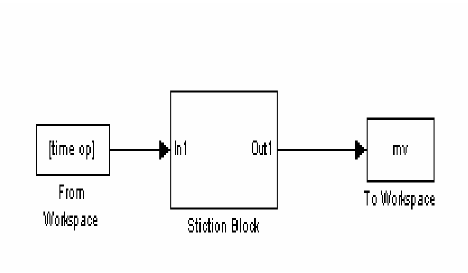


Fig. 16. Simulink block for generating *mv* data using Choudhury et. al [3] stiction.

Using the quantification 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 exist in the system. In all three cases, *S*=3 and *J*=1. The low number of iterations required however, depends strongly on the good initial guesses made on the *S* and *J* parameters.

TABLE 2 (A).
STATISTICAL ANALYSIS FOR K_c=*0.05*

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

TABLE 2 (B).
STATISTICAL ANALYSIS FOR K_c=*0.10*

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

TABLE 2 (C).
STATISTICAL ANALYSIS FOR K_c=*0.15*

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

3.5.1 Effect of varying stiction strengths to the NN-based stiction quantification algorithm

Table 3 shows the summary of the performance of the proposed method for varying values of the stiction parameters *S* and *J*. In all cases, K_c=*0.05* is used with all other parameters are kept constant. From the table, it can be clearly observed that the proposed method can correctly and efficiently quantified the amount of stiction present in each case.

TABLE 3.
SUMMARY OF STICTION ESTIMATION FOR VARYING *S* AND *J* VALUES

S		*J*	
Actual	Estimated	Actual	Estimated
1	1	0	0
1	1	1	1
4	4	2	2
6	6	4	5
8	8	8	8
8	8	10	10

3.5.2 Effect of external oscillating disturbance to the NN-based stiction quantification algorithm

The performance of the proposed stiction quantification algorithm is further tested in the presence of external oscillating disturbance acting on the process output. The external disturbance is taken as a sine wave with varying amplitude and frequency strengths. K_c=*0.05* is used with all other parameters are kept constant.

Table 4 shows the estimation results for sine wave with small amplitude and frequency, i.e. ampliture 0.5 and frequency 0.5 rad/sec. All other parameters are kept constant. For all cases, *S* and *J* are estimated correctly.

TABLE 4.
STICTION ESTIMATION FOR SYSTEM WITH EXTERNAL DISTURBANCE (AMPLITUDE = 0.5, FREQUENCY = 0.5 RAD/SEC)

S		*J*	
Actual	Estimated	Actual	Estimated
1	1	1	1
4	4	2	1
6	6	4	4
10	10	5	5
12	12	4	4

However, when the amplitude is increased to 1.0

whilst the frequency is lowered to 0.1 rad/sec, higher values of *J* show false estimation when *S* is greater than 10 – please see Table 5. The proposed method is not able to retain its accuracy in the presence of external oscillating disturbance with high amplitude when the amount of stiction present in the system is greater than 10% of the valve travel span. The unstable data probably reduced the capacity of the NARXSP NN to properly identify the process model.

TABLE 5.
STICTION ESTIMATION FOR SYSTEM WITH EXTERNAL DISTURBANCE (AMPLITUDE = 1.0, FREQUENCY = 0.1 RAD/SEC)

S		*J*	
Actual	Estimated	Actual	Estimated
1	1	1	1
4	4	2	2
6	6	4	4
10	11	5	1
12	13	4	1

Further analysis is made by increasing both the amplitude and frequency to 1.0, and Table 6 shows that even though all values of *S* are efficiently and correctly quantified, the performance of the proposed method deteriorated even further in terms of *J* values estimation. This indicates that the proposed method is fairly sensitive to external oscillating disturbance especially when the amplitude and frequency are high, and should be used with caution. Further improvements may be necessary to increase the robustness of the proposed stiction quantification algorithm in estimating the *J* parameters.

TABLE 6.
STICTION ESTIMATION FOR SYSTEM WITH EXTERNAL DISTURBANCE (AMPLITUDE = 1.0, FREQUENCY = 1.0 RAD/SEC)

S		*J*	
Actual	Estimated	Actual	Estimated
1	1	1	1
4	4	2	0
6	6	4	0
10	10	5	1
12	12	4	1

4 Conclusion

In this paper, a Neural-network based stiction quantification algorithm has been developed using routine operating data. The algorithm is simple and works efficiently in quantifying the actual amount of stiction present in the loop, even in the presence of corrupted data. The proposed method also works well for varying stiction strengths. However, the method is found to be sensitive especially in quantifying the correct *J* values in the presence of external oscillating disturbances. Further improvements have to be incorporated to increase the robustness of the proposed method in the presence of external oscillating disturbances.

References:

[1] M. A. A. S. Choudhury, S. L. Shah, & N. F. Thornhill (2004). Diagnosis of poor control-loop performance using higher-order statistics, *Automatica*, *40*, pp.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*, 44(17), pp. 6719-6728.

[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] Singh, J., Ganesh, A. (2008), Design and Analysis of GA based Neural/Fuzzy Optimum Adaptive Control, WSEAS Transactions on Systems and Control, Issue 5, Volume 3.

[11] Bayar, G. Konukseven, E. I., Koku, A. B. (2008), Control of a Differentially Driven Mobile Robot Using Radial Basis Function Based Neural Networks, WSEAS Transactions on Systems and Control, Issue 12, Volume 3.

[12] Dinh, B. H., Dunnigan, M. W. Donald Reay, S. (2008), A Practical Approach for Position Control of a Robotic Manipulator Using a Radial Basis Function Network and a Simple Vision System, WSEAS Transactions on Systems and Control, Issue 4, Volume 3
.

[13] Hagan M.T., Demuth H.B., Beale M.H. (1996). *Neural Network Design*, PWS Publishing Company, Boston, MA.