https://doi.org/10.1051/e3sconf/202128703012

Control valve stiction detection by use of AlexNet and transfer learning

Y. Y. S. Henry[1], *C.* Aldrich*[3], and *H.* Zabiri[1,2]

[1]Department of Chemical Engineering, Universiti Teknologi PETRONAS, Bandar Seri Iskandar, 32610 Perak, Malaysia
[2]CO2RES, Institute of Contaminant Management, Universiti Teknologi PETRONAS, Bandar Seri Iskandar, 32610 Perak, Malaysia
[3]Western Australian School of Mines, Curtin University, GPO Box U1987, 6845 WA, Australia

Abstract. Control valve stiction is a common problem faced by the process industries, which can have a strong adverse effect on the profitable operation of plants. Although various stiction detection methods based on neural networks have been proposed, few of these studies have considered the performance of stiction detection based on the use of 2D representations of the process signals. In this paper, such an approach is proposed, based on the use of a pretrained convolutional neural network, AlexNet. The proposed convolutional neural network stiction detection (CNN-SD) method showed highly satisfactory performance, which can be further applied on real industrial data.

1 Introduction

A typical process plant consists of hundreds of control loops that are operating regularly [1]. Continuous mechanical movement of the control valve causes it to deteriorate owing to wear and tear. Consequently, nonlinear behaviors of control valves occur, such as stiction, deadband, backlash, and dead zone phenomena. Among all these oscillatory problems, stiction is the most common and long-standing problem in the process industries [1]. The presence of oscillation in process variables may cause high rejection rates and inferior product quality, which significantly reduces plant productivity [2]. As reported by [3], 30% of industrial oscillatory loops are caused by control valve problems. Hence, prior to breakdown maintenance of the plant, prompt detection of control valve stiction is crucial for mitigation action to be taken. As a result, financial loss owing to shutdown maintenance of the plant can be prevented.

Considerable research has been done with regard to control valve stiction detection. These approaches can be categorized as cross-correlation function-based, limit cycle pattern based, nonlinearity detection based, and waveform shape based [4]. Most of the detection methods proposed are based on the relationship between process variables (PV) and controller outputs (OP) due to the difficulties in observing manipulated variable (MV). Various detection methods based on neural networks have been proposed, such as NLPCA [5], ANN [6], NLPCA-AC [7], and SDN [8]. These methods consider the time series input directly as 1D signal. More recently, methods considering the time series input in 2D have been proposed [9]. Signals in 2D can take advantage of a large and powerful library of feature extraction methods. In addition to this, these approaches have been driven by advances in computer vision, particularly related to deep learning.

To date, few of these approaches have been applied to the problem of stiction detection. Recently, [10] have proposed a method that uses the simulated MV-OP plots as images to train the CNN for stiction detection. However, this method shows inconsistent stiction detection rate ranging from 68.3% to 96.8%. In addition, this method was only tested on stiction and oscillation cases, while the other deadband, no offset, well-tuned, and excessive integral were not tested. Hence, the other cases are included in this current study.

In this paper, the proposed convolutional neural network stiction detection (CNN-SD) method used simulated data from seven different cases to retrain a pretrained CNN, AlexNet. A pretrained convolutional neural network, AlexNet, was used as a feature extraction tool and classification tool for this purpose. Since MVs are frequently unmeasured and difficult to obtain [4], the method proposed here provides an alternative for using PV and OP data to generate images for stiction detection. In addition, the differences between features extracted directly from a pretrained CNN and from a retrained CNN are compared as well.

*Corresponding author: chris.aldrich@curtin.edu.au

2 Control valve stiction

As shown in Figure 1, a typical input-output behaviour of sticky valve consists of four main components: deadband, stickband, slip-jump, and moving phase. The occurrence of the stiction as depicted in Figure 1 can be described as follows:

- (A-C) - The MV remains constant even when the OP increases as it could not overcome the static friction, which corresponds to deadband and stickband.
- (C-D) - Once it overcomes the static friction, a sudden jump of the valve occurs, which corresponds to slip jump.
- (D-E) - The valve position increases linearly until it stops and sticks again.
- (E-F-G-A) - Similar behaviour can be observed when the controller output changes its direction.

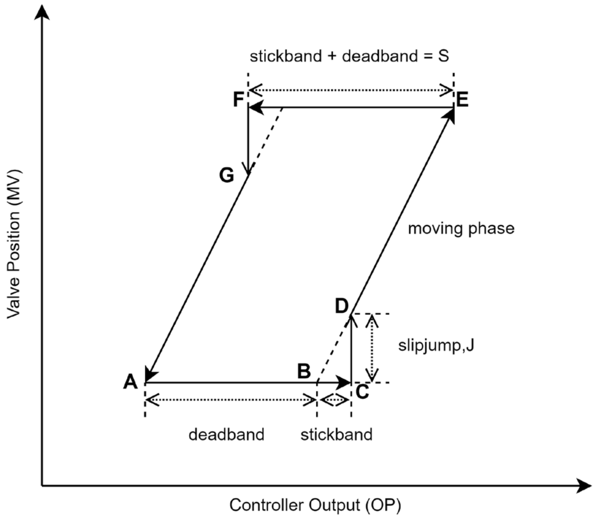


Fig. 1. Typical input-output behaviour of sticky valve adapted from [5].

3 Methodology

The structure of the proposed CNN-SD framework is presented in Fig. 2. Firstly, the time series data (PV-OP) from control loops are preprocessed by normalizing them into zero mean and unit variance. A fixed window of specific length length l and step size b is moved over the PV-OP data to cut them into batches. The segmented time series data are subsequently transformed into matrix plots (or known as images) using unthresholded reccurrence plots (URP). These images formed will then be fed into a partially retrained CNN (AlexNet) for either feature extraction or classification purpose. It should be noted that the AlexNet is partially retrained with simulated data before serving as feature extraction tool and classification tool. Prior to applying AlexNet, each images must be resized to 227x227x3. Consequently, a classification output of either stiction or non-stiction will be produced for each input image. Else, feature vectors can be extracted from trained AlexNet which would serve as predictors for classifier development.

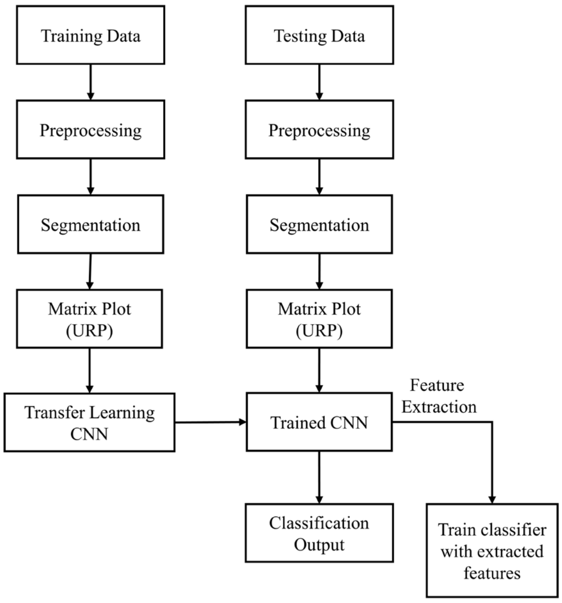


Fig. 2. Workflow of the proposed CNN-SD framework.

3.1 Preprocessing

The first step of the proposed CNN-SD is to acquire data and preprocess them accordingly. Appropriate preprocessing steps can help to improve the quality of the data, as well as to enhance accuracy of classification framework. In this paper, the raw time series data (PV-OP) acquired are preprocessed through z-normalization. For example, given a time series X = $\{x_1, x_2, \dots, x_n\}$, X can be normalized to zero mean and unit variance through equation (1).

$$x_n = \frac{x_n - \mu_X}{\sigma_X} \tag{1}$$

where μ_X is the mean of X and σ_X is the standard deviation of X.

3.2 Segmentation

A moving window segmentation is often required when deal with a huge time series dataset in order to process the data in batches. The time series is initially segmented into m windows with window length, l respectively. A step size, b, is then specified to determine the movement of the window across the time series. The selection of an optimal window size is important. Too small a window size will cause loss of important information, while too large a window size will delay the process of detecting anomalies [11] and may also decrease the sensitivity of the method to transient changes of short duration. Hence, finding an optimal window size is crucial so that the dynamic behavior of the time series can be presented with shortest possible window length, l_{min}. In general, the optimal window size can be selected using the autocorrelation of the time series or other empirical methods as a guide.

In this paper, a non-overlapping, contiguous window segmentation method was applied, where window length *l*, and step size *b* were equal. The optimal window size was determined empirically by trial and error.

3.3 Matrix plot

An image is formed by pixels which can be presented in a form of a matrix plot. In this paper, unthresholded recurrence plots (URP) were used to transform time series into images. URP is an image encoding method derived from recurrence plots (RP) without the use of thresholding [12], since the main goal is to extract as much dynamic information as possible. Generally, URPs are regarded as a direct representation of a distance matrix that measures the pairwise distance between observations. URP can be expressed as

$$URP_{ij} = ||\, x_i \; - x_j ||, \qquad i, j = 1,2, \dots, N \qquad (2)$$

where $||\cdot||$ represents the Euclidean norm in phase space.

3.4 Training of AlexNet

AlexNet is a convolutional neural network (CNN) developed by [13]. It consists of 5 convolutional layers, 3 max pooling layers and 3 fully connected layers. For each input images with size of 227x227x3, an output of 4096 feature vectors will be produced from layer FC2 as shown in Fig. 3.

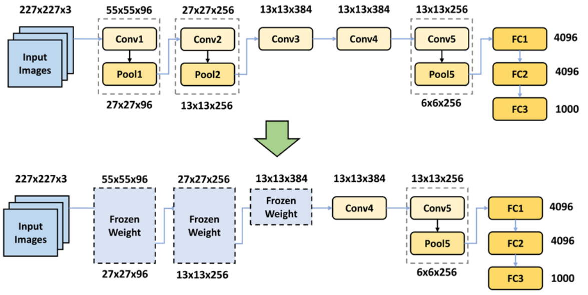


Fig. 3. Simplified schematic diagram of AlexNet architecture (top: original architecture, bottom: transfer learning architecture).

In general, CNN consists of large number of learnable parameters. Training CNN from scratch is computationally expensive and requires large amount of training images to be efficient. Hence, partially retraining a pretrained CNN can be an alternative to solve these problems. This approach is commonly known as transfer learning, where the knowledge learned from a previous task is transferred to a new task. The advantages of transfer learning are lower computational cost, faster training, and critically, the ability to process relatively small data sets, compared to the data required to train the CNN ab initio.

In this study, simulated data were used to partially retrain AlexNet. The training data set consisted of 70% randomly chosen images generated from simulated data, while the other 30% were used as test data to validate the model. The output classes were set to be either stiction or non-stiction.

For transfer learning, the weights of the first three convolutional blocks depicted in Fig. 3 were frozen, while the rest of the layers were fine-tuned with training images. The reason is that these earlier layers were used to learn low level features that are generally applicable across images with various sources [14], while the later layers were used to identified specific features from images that would help to differentiate the images into specific classes. Hence, the weights learned previously in the earlier layers remained unchanged. Training was done with stochastic gradient descent with a momentum (SGDM) optimizer, a learning rate of 0.0001, and a mini batch size of 10.

3.5 SVM classifier

The image features extracted from layer FC2 were used as predictors to build a support vector machine (SVM) classifier. SVM is a supervised machine learning technique that finds an optimal hyperplane with a maximum margin to separate classes of data in high dimensional space [15]. In this paper, five-fold cross validation was used to build the linear SVM model, while ten-fold cross validation was used to obtain generalization performance of the trained SVM. Extracted features were divided randomly into 60% training data, 20% validation data, and 20% testing data.

4 Generation of simulated data

In this study, the Choudhury stiction model developed by [16] was applied. The simulated data were generated using a simple single input single output (SISO) first order transfer function of the feedback control system adapted from [5]. A total number of 10,000 samples of PV and OP data were collected from each case at a sampling rate of 1 s. The initial 500 data points were discarded to ensure the time series had stabilized; leaving 9,500 samples for training. The transfer function is represented by:

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

Seven cases of simulated data are generated in this study, which include 3 non-stiction cases and 4 stiction cases. The control system parameters for each case are set accordingly. The three non-stiction cases are well-tune ($\boldsymbol{P} = K_c = 0.15$, $\boldsymbol{I} = K_c/\tau_i = 0.15\, s^{-1}$), excessive integral ($\boldsymbol{P} = K_c = 0.15$, $\boldsymbol{I} = K_c/\tau_i = 0.27\, s^{-1}$), and oscillatory ($\boldsymbol{P} = K_c = 0.15$, $\boldsymbol{I} = K_c/\tau_i = 0.15\, s^{-1}$). Note that that the oscillatory case is generated with the additional settings of sinusodial disturbance with an amplitude of 2 and frequency of 0.01 rad/s. On the other hand, the four stiction cases are undershoot stiction (***S***=3, ***J***=1), stiction with no offset (***S***=3, ***J***=3), overshoot stiction (***S***=1, ***J***=3), and stiction with deadband (***S***=3, ***J***=0). For all the stiction cases, the proportional ***P*** and integral ***I*** settings are same as well-tuned case.

5 Results and discussion

The simulated data were divided into batches using a window size of 100, as shown in Fig. 4. As can be seen from this figure, in the simulated data, it is difficult to distinguish between stiction and non-stiction conditions by visual inspection. These data were subsequently transformed into images through URP, as shown in Fig. 5. These images were used to retrain AlexNet, with the first three convolutional blocks remain frozen, as explained previously. The trained AlexNet was subsequently used as feature extraction tool and classification tool.

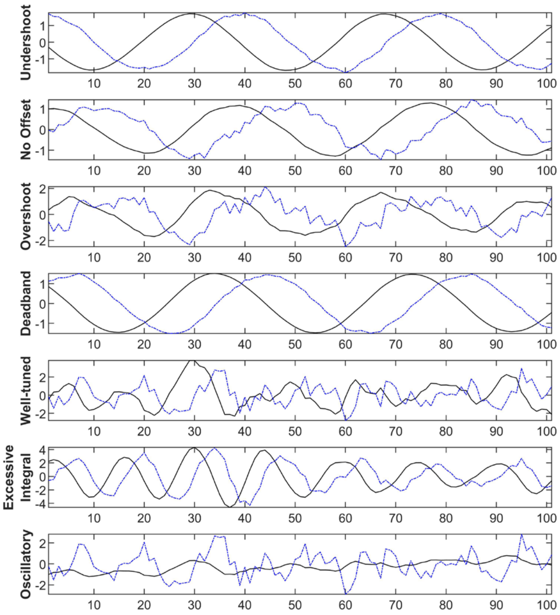


Fig. 4. Examples of simulated data PV (blue broken line) and OP (solid black line) generated for seven different cases.

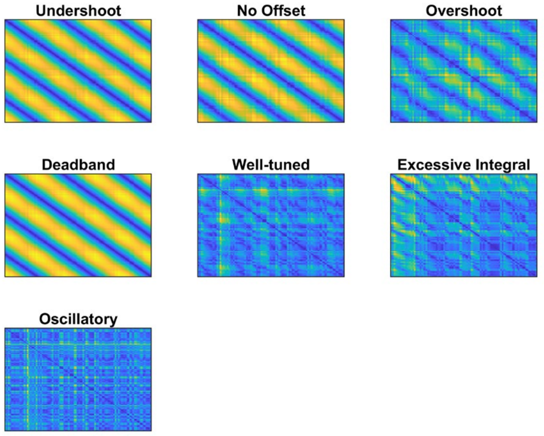
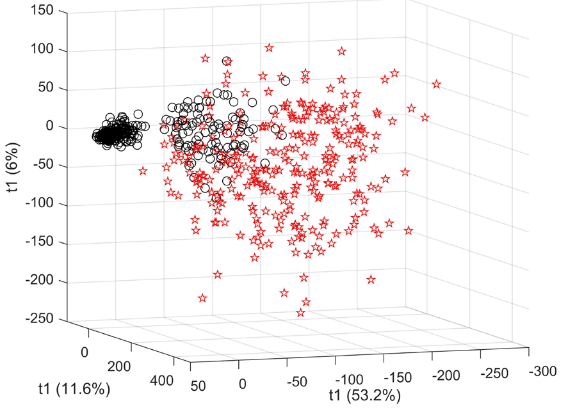


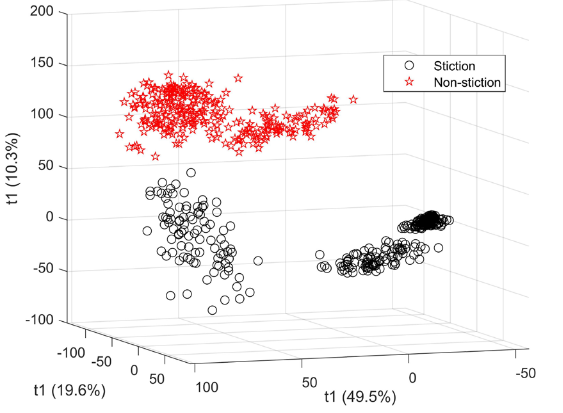
Fig. 5. Examples of 'imaged' URP generated for seven different cases.

5.1 Pretrained AlexNet vs partially retrained AlexNet

The image features extracted from the pretrained AlexNet and the partially retrained AlexNet can be visualized through a three-dimensional principal component scores plot, as shown in Fig. 6. As mentioned previously, these features were extracted from the layer FC2. From Fig. 6, it can be seen that the features from the partially retrained AlexNet provide better separation of the two classes than features obtained from the pretrained AlexNet. This can be further proven quantitatively using the SVM classifier. Two cases are considered, i.e. classification using the first two features only as predictors and classfication using all 4096 features generated by AlexNet.



(a)



(b)

Fig. 6. Principal component scores plots of features extracted from layer FC2 in AlexNet with a window size of 100, snowing stiction (black circles) and non-stiction (red stars). Descriptions: (a) features from pretrained AlexNet and (b) features from a partially retrained AlexNet.

As seen from Fig. 7, by just using the first two features from the partially retrained AlexNet, the SVM was able to achieve a classification accuracy of approximately 97.2%, compared with 84.4% obtained with the pretrained AlexNet. When all 4096 features are used as predictors, both the pretrained and partially retrained versions of AlexNet performed similarly, yielding near perfect classification.

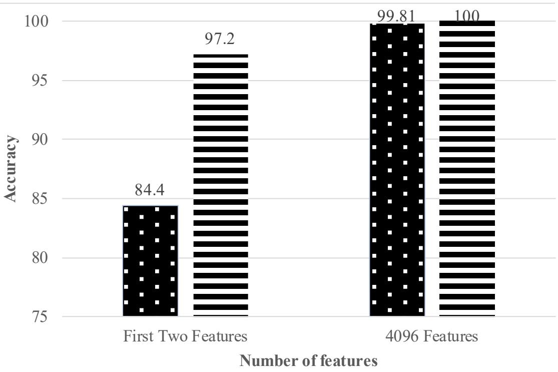


Fig. 7. Bar charts of SVM classification performance. Pretrained AlexNet (dotted bars) and partially retrained AlexNet (stripes bars).

5.2 Effect of window size on the classification performance

As discussed previously, window size is an important hyperparameter in the proposed method that needs to be optimized. This was done based on a line search of the windows size, while training of AlexNet was repeatedly 10 times for each window size. As can be seen from Fig. 8, the classification accuracy increased as the window size became larger. This is mainly due to the ability of larger window to capture the implicit dynamic behaviour of time series. Since the main objective is to detect the presence of stiction as soon as possible, the optimal window size is selected as the smallest size giving satisfactory classification i.e. a size of 30, which yields a classifier that is approxiately 90% accurate, with a detection rate of 30 s. This is significantly smaller than the window size of 500 used by recent pubslihed SDN method [8].

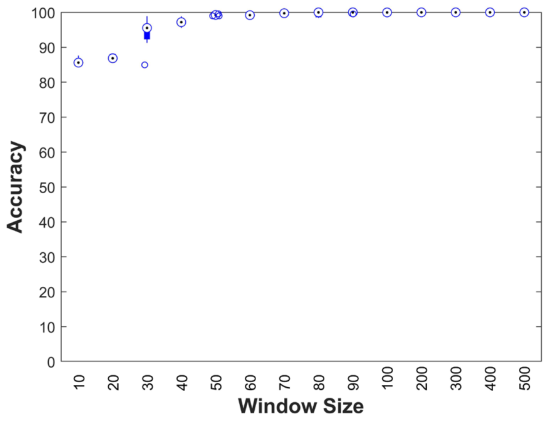


Fig. 8. Boxplots of classification accuracies of the partially retrained AlexNet for different window sizes.

As observed from Fig. 8, the classification accuracies are better and more consistent than those obtained with another CNN detection method proposed by [10].

5 Conclusion

A novel framework of stiction detection by use of the AlexNet convolutional neural network and transfer learning is proposed. The proposed methodology, convolutional neural network stiction detection (CNN-SD), has achieved a highly satisfactory performance, which suggests that the proposed method could also be successfully applied to real industrial data.

Finally, it should be noted that when applied to industrial data, further optimisation of the method would be possible, among other by considering different approaches to the 2D representation of the process signals. I would also be possible to use other, more recent convolutional neural networks that generally tend to outperform AlexNet on image clasification tasks.

The authors would like to thank Universiti Teknologi PETRONAS (UTP), Malaysia and Curtin University, Australia for the facility and technical support provided to complete this work.

References

[1] M. Choudhury, S. L. Shah, N. F. Thornhill, *"Detection and quantification of control valve stiction," the proceedings of DYCOPS 2004,* (2004).

[2] M. S. Choudhury, M. Jain, S. L. Shah, J. Process control*,* **18**, 232 (2008).

[3] C. Garcia, Control Eng. Pract.*,* **16**, 1231 (2008).

[4] R. B. di Capaci, C. Scali, Chem. Eng. Res. Des.*,* **130**, 230 (2018).

[5] H. Zabiri, M. Ramasamy, J. Process Control*,* **19**, 1368 (2009).

[6] A. R. Venceslau, L. A. Guedes, D. R. Silva, *"Artificial neural network approach for detection and diagnosis of valve stiction," in Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012)*, 1 (2012).

[7] W. K. Teh, H. Zabiri, Y. Samyudia, S. S. Jeremiah, B. Kamaruddin, A. A. A. M. Amiruddin, N. Ramli, Ind. Eng. Chem. Res.*,* **57**, 11350 (2018).

[8] A. A. A. M. Amiruddin, H. Zabiri, S. S. Jeremiah, W. K. Teh, B. Kamaruddin, Control Eng. Pract.*,* **90**, 63 (2019).

[9] N. Hatami, Y. Gavet, J. Debayle, *"Classification of time-series images using deep convolutional neural networks," in Tenth International Conference on Machine Vision (ICMV 2017),* **10696**, 106960Y (2018).

[10] N. R. Vazquez, D. P. Fernandes, D. H. Chen, IJCEA*,* **10** (2019).

[11] C. Aldrich, *"Process Fault Diagnosis for Continuous Dynamic Systems Over Multivariate Time Series," in Time Series Analysis: IntechOpen* (2019).

[12] A. Sipers, P. Borm, R. Peeters, Phys. Lett. A, **375**, 2309 (2011).

[13] A. Krizhevsky, I. Sutskever, G. E. Hinton, *Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems*, 1097 (2012).

[14] Y. Fu and C. Aldrich, Minerals Engineering*,* **132**, 183 (2019).

[15] C. Aldrich and L. Auret, *Unsupervised process monitoring and fault diagnosis with machine learning methods*: Springer (2013).

[16] M. A. A. S. Choudhury, N. F. Thornhillb, S. L. Shah, Control Eng. Pract., **13**, 641 (2005).