Implementation of Data-Driven Fault Detection Approach for a Three-Tank System: An Educational Perspective

N. M. M. Sobran[1*], M. D. M. Yusof[1], N. M. Ali[1], M. Zulkifli[2], M. M. Ghazaly[1]
[1]Fakulti Teknologi dan Kejuruteraan Elektrik, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia
[2]Department of Electrical Engineering Technology, Faculty of Engineering Technology, Universiti Tun Hussein Onn Malaysia, 84600 Pagoh, Johor, MALAYSIA
[*]corresponding author's email: nurmaisarah@utem.edu.my

Abstract –*Precise and efficient fault management are essential in today's industrial environment to avoid downtime and monetary losses. Any faulty events could affect factory process flow as well as product quality. Recently, due to the advancement in signal monitoring with cloud data storage, fault detection approach has changed from a model-based into to a data-driven approach. The main goal of this study is to provide an initial insight of the data-driven fault detection approach focusing on undergraduate or postgraduate students, including practitioners that are just new in this area. At the first stage, the data generation and collection approach are described with one process plant example, namely; the three-tank system. Later, detailed principal component analysis (PCA) and threshold determination are explained using normal datasets. After that, the faulty dataset is used to determine the performance of the PCA approach. It is found that the PCA approach is only able to detect six out of eleven faulty conditions, which results in 54.4% of the fault detection performance.*

***Keywords**: Fault detection, Principal Component Analysis, Three-tank water system, Upper and lower control limit.*

Article History
Received 4 September 2025
Received in revised form 28 September 2025
Accepted 28 October 2025

I. Introduction

Efficient system operation is crucial for optimizing production efficiency and avoiding financial losses in the field of process engineering. A fault detection (FD) system is essential to ensure the continuous and reliable operation at the industrial plant. For example, sensor failures unable to anticipate disruptions can have a domino effect, such as resulting machine breakdown, interrupted process flow, and consequently increased corrective costs to fix the damaged components. The situation can escalate until jeopardizing personnel safety and compromising the surrounding environment [1]. Thus, adopting FD approach in monitoring process plants becomes a noteworthy aspect in large-scale manufacturing systems [2].

Fault detection entails identifying abnormalities or deviations from normal system behavior. These tasks are the central focus in the preventive maintenance paradigm [3], which allows proactive interventions to correct problems before they become serious. In other words, FD has the ability in early detection of faults while the plant is still operating in a controllable region, which can help to avoid abnormal event progression.

Moving towards the digitalization era, the data-driven technique has been seen emerging in the study of FD. Statistical or machine learning methods such as PCA, Independent Component Analysis (ICA), Partial Least Square (PLS) and Support Vector Machine (SVM) were used in monitoring the process plant system [4]-[5] . At the same time, there are also artificial intelligence approaches such as autoencoder (AE) [6]-[7], long short-term memory (LSTM)[8]-[9], and convolutional Neural Networks (CNN) [10]-[11] that utilize the historical data in predicting faulty events in manufacturing systems.

Despite the importance of fault detection in the industrial systems, as well as the vast the data-driven methods, the concept behind data-driven approach in FD is less exposed in both undergraduate-level and postgraduate studies. Thus, the main goal of this paper is to provide theoretical insight with a simulation-based validation of the FD concept in engineering systems using a data-driven approach as an introduction to those who just started in the data-driven FD area of study. The framework of the data-driven approach will be discussed, and the concept of FD using PCA will be further explained throughout this paper.

II. Methodology

A. Basic overview of data-driven fault detection approach in engineering system

The data-driven approach starts with data collection of normal and faulty conditions from the industrial system. Recently, this stage has become handy due to the Internet of Things (IOT) technology, SCADA, and cloud computing that enable continuous process monitoring and online data storage [12]-[14]. After that, data will go through data preprocessing such as normalization, standardization, missing data, or noise filtering. Next, the data will be input to the selected data-driven model as the representation of normal events and testing with faulty events. After that, determination of upper and lower thresholds will be calculated based on the Gaussian distribution assumption with a decision boundary of μ±3σ [13]. Fig. 1 shows the fault detection data-driven approach in general.

B. Data collection stage: A Three-Tank System

The three-tank model that we used was referred to [15] as shown in Fig. 2. We extend the demonstration in [15] that focuses on the model-based FD method into a data-driven approach. We also further extend from the previous study by adding the noise signal into the system to ensure the collected data closely follow real-time industrial plant conditions. The three-tank system can be considered as a non-linear, multivariable, and time-series system that aligns with present industrial process plants. The normal and faulty dataset of the three-tank water level signal was generated based on the mathematical model of the three-tank system equations (6), (7), and (8) in the MATLAB environment as a representation of the real-time tank system in the manufacturing sector.

The linear control system state variables:

$$\dot{X}(t) = AX(t) + BU(t) \tag{1}$$

$$Y(t) = CX(t) \tag{2}$$

The input variables, U:

$$U^T = [u_1 \quad u_2]^T = [q_{i1} \quad q_{i2}]^T \tag{3}$$

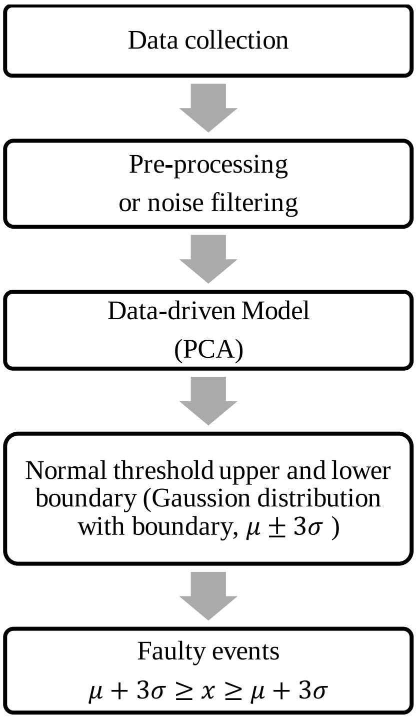
Data collection
Pre-processing or noise filtering
Data-driven Model (PCA)
Normal threshold upper and lower boundary (Gaussion distribution with boundary, $\mu \pm 3\sigma$)
Faulty events $\mu + 3\sigma \geq x \geq \mu + 3\sigma$

Fig. 1. Overview of the data-driven fault detection approach.

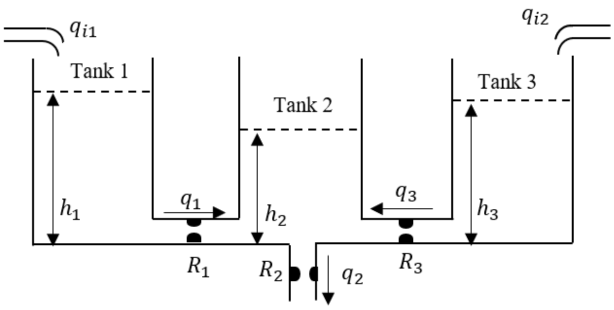

Fig. 2. Three-tank system

State variables and observed variables:

$$X^T = [x_1 \quad x_2 \quad x_3]^T = [h_1 \quad h_2 \quad h_3]^T \tag{4}$$

$$Y^T = [y_1 \quad y_2 \quad y_3]^T = [h_1 \quad h_2 \quad h_3]^T \tag{5}$$

Matrix A, B, and C are

$$A = \begin{bmatrix} -\frac{1}{S_1R_1} & \frac{1}{S_1R_1} & 0 \\ \frac{1}{S_2R_1} & -\frac{1}{S_1}\left(\frac{1}{R_1}+\frac{1}{R_2}+\frac{1}{R_3}\right) & \frac{1}{S_2R_3} \\ 0 & -\frac{1}{S_3R_3} & -\frac{1}{S_3R_3} \end{bmatrix} \tag{6}$$

$$B = \begin{bmatrix} \frac{1}{S_1} & 0 \\ 0 & 0 \\ 0 & \frac{1}{S_1} \end{bmatrix} \tag{7}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

where S_i (i = 1, 2, 3) denotes the cross-section areas of the three-tank. The parameters value in the model as per listed in Table III [15] at appendix.

Fig. 3 shows the Simulink MATLAB model based on the mathematical model of three-tank system. There are also eleven (11) switches labelled as P1 to P3 for abrupt disturbances, L1 to L3 for leakages, S1 to S3 for sensor faults, and B1 and B2 for pipe blockages at the respective Tank 1, Tank 2 and Tank 3. The switches were used to change from normal to faulty state. Five hundred data (n=500) were generated for each normal and faulty condition for the system. The parameters value used for noise, signal generators and gains as listed in Table IV at appendix.

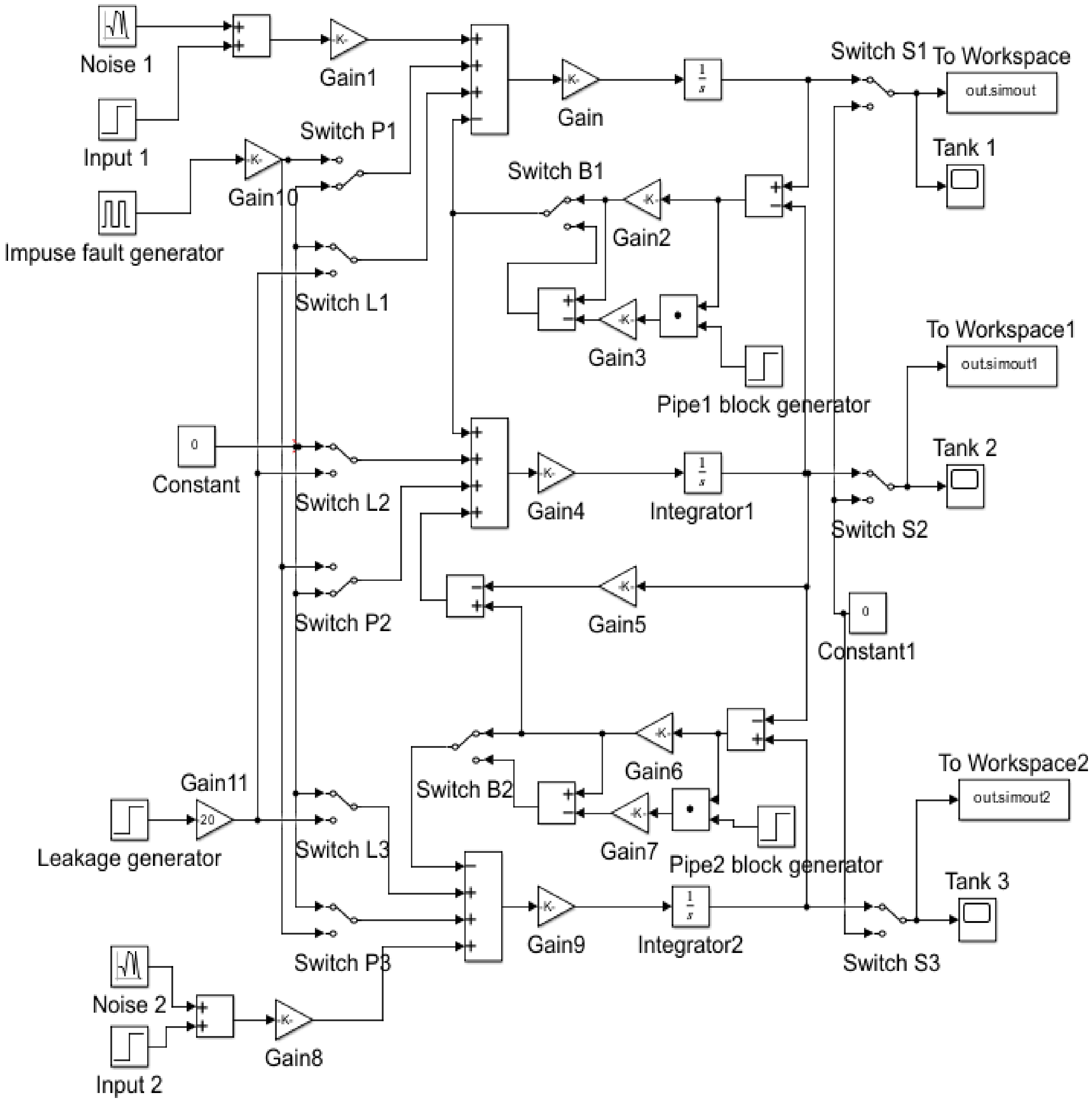

Fig. 3. Three-tank system mathematical modelling representation using Simulink MATLAB

C. Determination of upper and lower boundary using PCA approach using normal condition dataset.

Principal Component Analysis (PCA) is a well-known multivariate statistics method for identifying patterns in datasets as well as for dimensional reductions [16]. The PCA is able to transform a large dataset to a smaller set of variables in a linear combination of the original while retaining as much data variability as possible. This situation is suitable to be implemented in learning the normal dataset distribution and measuring the decision boundary as a threshold range in detecting the faulty signal.

The collected data, **X** consists of **n** number of observations and **p** variables with, **n×p** data matrix. At the initial stage of implementation PCA approach, the normal condition data will go through data centering and scaling as per stated at equation (9) with μ is the mean and σ is the standard deviation of the dataset. The MATLAB code as per shown in Fig. 4.

$$x = \frac{X-\mu}{\sigma} \tag{9}$$

MATLAB code:

```
% Calculate mean for each tank
tank_mean = mean(normaldata);
tank_std = std(normaldata);

% Center the data and scale
tank_center = (normaldata -
repmat(tank_mean,[m,1])) ./
repmat(tank_std,[m,1]);
```

Fig. 4. MATLAB code for centering and scaling

After that compute the covariance matrix, S where m is mean vector in d-dimension. See equation (10) with Fig. 5 MATLAB code.

$$S = \sum_{k=1}^{n}(x_k - m)(x_k - m)^T \tag{10}$$

MATLAB code:

```
% Compute covariance matrix for each tank
tank_cov = cov(tank_center);
```

Fig. 5. MATLAB code for covariance matrix

Followed by eigen vectors or also called principal components, $P = e_1, e_2, \ldots, e_d$ and corresponding eigen values, $\lambda_1, \lambda_2, \ldots., \lambda_d$ from the covariance matrix. The PCA model were evaluated by using data set x and eigen vectors matrix P [17] as in equation (11) and the MATLAB code as in Fig. 6.

$$\hat{x} = TP^T \tag{11}$$

MATLAB code:

```
% Calculate eigenvectors and eigenvalues for
each tank
[tank_eigvecs, tank_eigvals] = eig(tank_cov);

% Project centered data onto PCA space
tank_projected_data = tank_center *
tank_eigvecs;
```

Fig. 6. MATLAB code for projected data

After that calculate the residual between the scaling data set x and the PCA projected dataset, $\hat{x}$ followed by the squared prediction error (SPE) value [18] as stated at equation (12) with MATLAB code at Fig. 7.

$$SPE = \|X_i - \hat{x_i}\|^2 \tag{12}$$

MATLAB code:

```
% Calculate Residual between x and xbar
res = tank_center - tank_projected_data;

% Calculate squared prediction error (SPE) for
each tank
spe= abs(res.^2);
```

Fig. 7. MATLAB code for SPE

Lastly, to determine the upper and lower threshold value with $\mu \pm 3\sigma$ threshold cutoff under the assumption of Gaussian data distribution. The μ and σ represent the mean and standard deviation of the distributions and the MATLAB code as in Fig. 8.

MATLAB code:

```
% Calculate upper and lower control limit
ucl_tank = mean(spe) + 3*std(spe)
lcl_tank = mean(spe) - 3*std(spe)
```

Fig. 8. MATLAB code for control limit

In detecting the faulty events, the collected faulty dataset will go through similar preprocessing approach which is data centering and scaling based on the mean, μ and standard deviation,σ of the normal dataset. After that, the dataset will go through similar step in PCA which is determination of covariance, eigen values, data projection until the calculation of SPE. The calculated upper and lower control limit in normal condition will be used to determine whether the signal in normal or faulty condition.

III. Results & discussion

A. Simulation results for normal conditions from the Simulink MATLAB

Fig. 9 shows the response from the water tank system state space mathematical expression that is being simulated using Simulink MATLAB blocks. The input water level given to the system is five (5) in unit step with noise. From the graph, the water level for Tank 1 and Tank 3 is similar and followed the input value with an average of 5, whereas Tank 2 is below the input value with an average water tank level of 3.89. Fig. 10, on another hand, shows the SPE value after PCA implementation with residual and threshold determination. For PCA implementation, we select the data from sample 50 until 500 to include only the steady state response. From the figure, it is observable that all the SPE values in the normal dataset lie within the boundary. The value of the upper and lower threshold for each level is shown in Table 1, and the line were highlighted using red color in the figure.

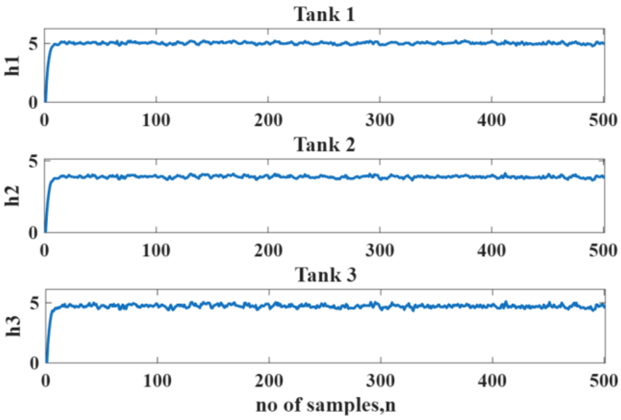

Fig. 9. Water level response for three tank system

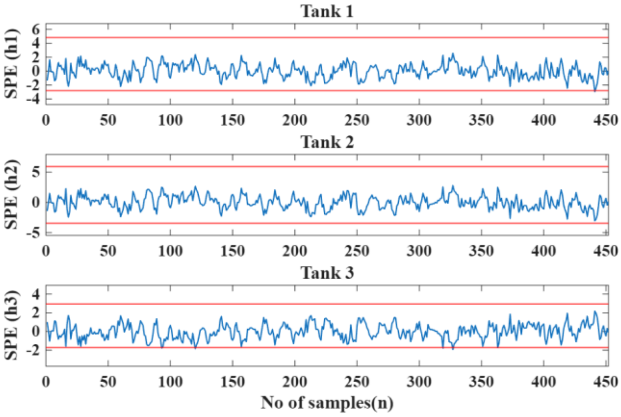

Fig. 10. SPE value after PCA implementation with upper and lower threshold.

TABLE I
UPPER & LOWER THRESHOLD FOR THE THREE TANK SYSTEM

Tank	Upper Control Limit (m)	Lower Control Limit (m)
1	4.8	-2.8
2	5.9	-3.5
3	2.9	-1.7

B. Abrupt disturbance

Abrupt disturbance represents a sudden problem while adding water to the tanks. The water pipe is in an on-and-off condition, affecting the water level condition. To simulate this condition, a pulse generator was used in the Simulink MATLAB, and switches P1, P2, and P3 were used to enable the situations. Fig. 11 shows the response of the abrupt changes onto the water level when the switch P1 was turned on, and Fig. 12 shows the SPE value for this event. To cover only the steady-state area, all the graphs for faulty events will consider the samples from 50 until 500. From Fig. 12 it is observable that the PCA is able to detect the abrupt faulty signal, which at the period of 50 to 100, 150 to 200, 250 to 300, and 350 to 400. We also found that, even with the switch P1 focusing on impact towards Tank 1, all the three water tank levels were affected.

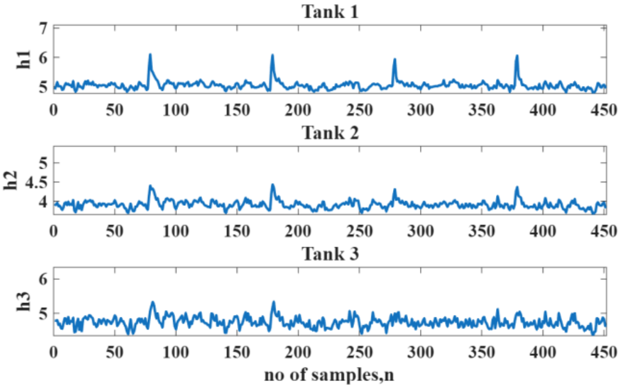

Fig. 11. Water level response after switch P1 was on.

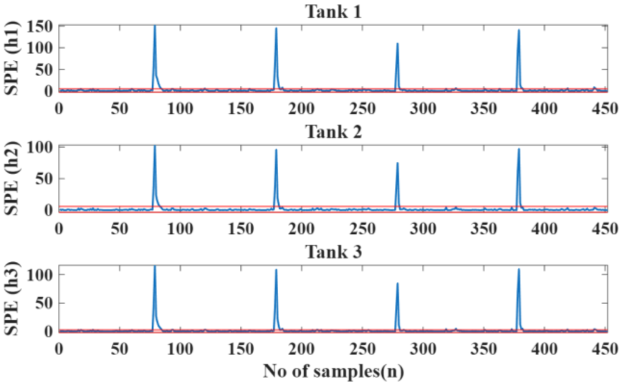

Fig. 12. SPE value for abrupt changes in tank system.

C. Leakage at water tank system

Leakage at the water tank is important to detect earlier due to prolonged leakages at tanks, will temper the water steady-state level as well as waste water resources. This study replicates the situation by using a signal generation block in the Simulink MATLAB environment with switches L1, L2, and L3. Under this section, the discussion will focus when the switch L1 was turned on and the response of the water tank level and the SPE value were illustrated in Fig. 13 and Fig. 14. From Fig. 13 it is found that all of the tank levels will be affected when leakage happens at Tank 1 (Switch L1 on). The water level starts to reduce, from sample n=50 and onwards, showing the symptom of leakage. This is aligned with findings at Fig. 14 where the SPE value raised above the upper control limit at the same sample number, indicating that faulty event occurred at this period of time. This situation also reflects the PCA's ability to distinguish between normal and faulty conditions under leakages events.

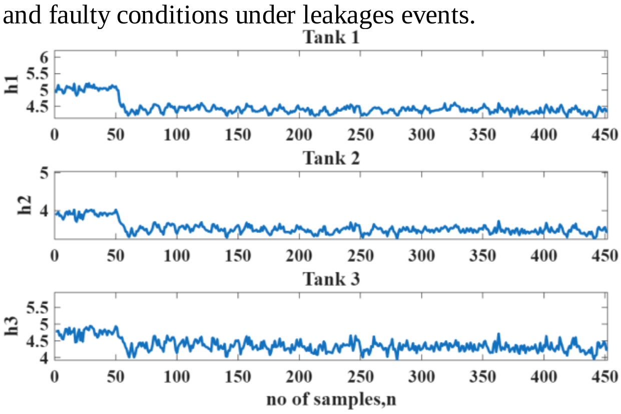

Fig. 13. Water level response when switch L1 was on.

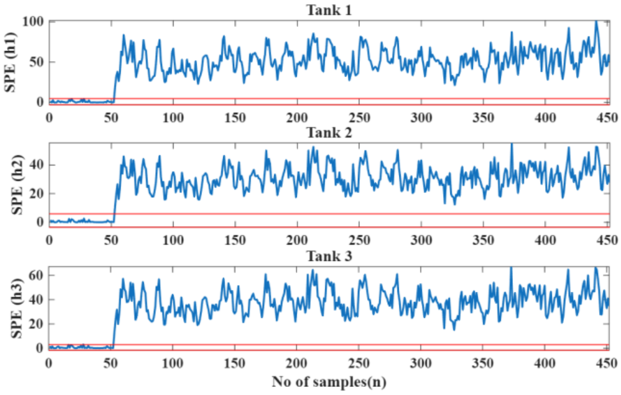

Fig. 14. SPE value for leakage events at the water tank system.

D. Sensor failure at water tank system

When the level sensor failed to operate as needed, no level signal was available for the feedback control mechanism to ensure the water level followed the initial setting. Furthermore, with a damaged sensor, the person in charge is unable to monitor the current situation of the tank system. Thus, it is crucial to detect this faulty condition in the process plant system. Sensor faults were presented with switches S1, S2, and S3 in on and off mode. Fig. 15 and Fig. 16 show the water level signal response as well as the SPE value when we switch off switch S1. From Fig. 15 it is observed that when the sensor was damaged, no signal response was able to be captured at Tank 1, whereas the other two tanks were not affected by this failure. When evaluating the SPE value at Fig. 16, since the signal value is zero at Tank 1, the signal was considered as normal signal since it's within the boundary. The misdetection happens under this situation where the PCA approach is unable to detect this faulty event. In contrast, results at Tank 2 and Tank 3, the SPE value shows a true interpretation where the signal also remains within the boundary as known previously; sensor failure at Tank 1 will not affect Tank 2 and Tank 3 water level signal.

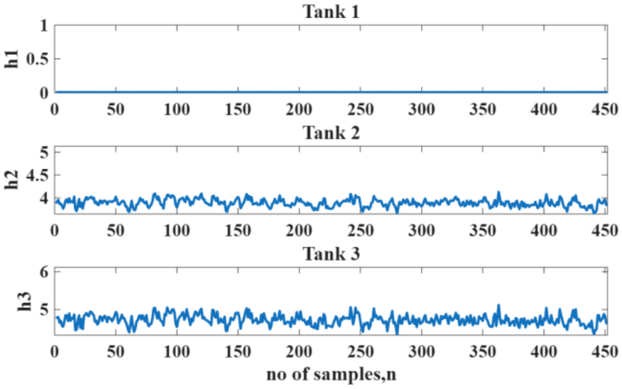

Fig. 15. Water level response when switch S1 was off.

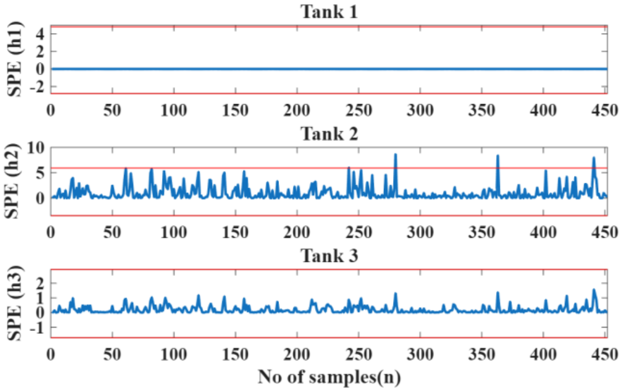

Fig. 16. SPE value for sensor failure events at the water tank system.

E. Pipe blockage at water tank system

Pipe blockage events will interrupt the water flow of the three-tank system. This faulty event was simulated using a signal generator as well as switches B1 and B2. Fig. 17 and Fig. 18 illustrate the signal condition when the switch B1 was turned on. From Fig. 17, it is found that, when a blockage situation happens at Tank 1, the water level of Tank 1 increases significantly while Tank 2 and Tank 3 remain unaffected by the situation. Compared with the SPE value at Fig. 18, Tank 1 has a true fault detection at sample time of 50 and onwards. However, Tank 2 and Tank 3 have incorrect fault detection since these two tanks should remain in normal condition based on Fig. 12. This is another event where PCA was unable to correctly distinguish between normal and faulty conditions.

Further analysis was done with the remaining faulty switch condition and is being interpreted as in Table II. From the table, it is found that only all leakage faulty events are correctly detected by the PCA approach. In contrast, all of the sensor failure events were totally unable to be detected by the PCA. The other faulty events such as abrupt fault at P3 condition has a wrong detection as well as for pipe blockage B1. With this, a total of six (6) faulty events were successfully detected. while the other five (5) were wrong detections, making the total of 54.4% accuracy of fault detection using the PCA approach. The reason behind the low accuracy of the PCA approach is due to the PCA projection being in linear condition only (see equation (11)), and the time series or temporal learning was not included when applying this method [19]-[20].

TABLE II
FAULT DETECTION PERFORMANCE

Faulty Condition	All faults are correctly detected using PCA approach.
Abrupt, P1	True
Abrupt, P2	True
Abrupt, P3	False
Leakage, L1	True
Leakage, L2	True
Leakage, L3	True
Sensor Failure, S1	False
Sensor Failure, S2	False
Sensor Failure, S3	False
Pipe Blockage, B1	False
Pipe Blockage, B2	True
Total of 'True' detections	6/11 (54.5%)

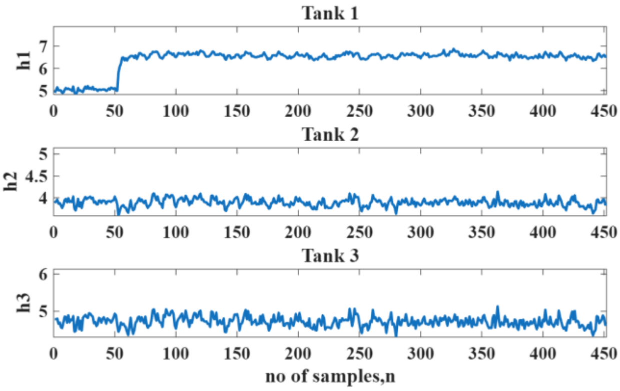

Fig. 17. Water level response when switch B1 was on.

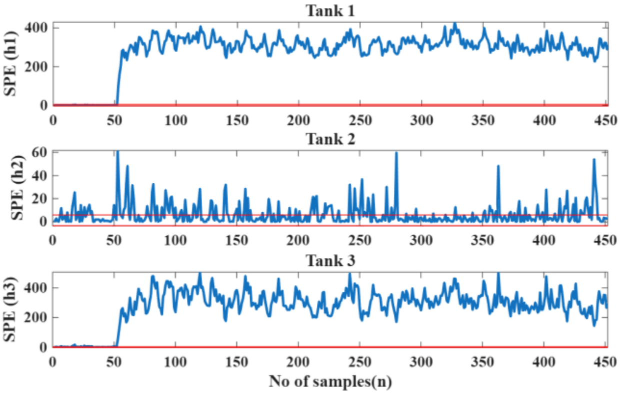

Fig. 18.SPE value for pipe blockage faulty events at the water tank system.

IV. Conclusion

This study focuses on the implementation of a data-driven fault detection approach using Simulink & MATLAB with the motivation to provide an initial insight for the undergraduate and postgraduate students. The three-tank system is used as an example of the real process plant signal conditions where eleven switches are used to represent eleven types of faulty conditions.

In the first stage, the data collection was conducted based on the three-tank system state-space representation in the SIMULINK environment. Later, the normal dataset will go through PCA method to calculate the upper and lower control limits. After that, based on the faulty dataset, the signal condition was observed, and the performance of the PCA fault detection approach was calculated. From the observation, we conclude that the PCA only able to detect six out of eleven faulty condition which makes the detection performance 54.4%. Further study is needed to improve the detection process, either in more reliable control limit determination or the data-driven method itself.

Appendix

TABLE III
THREE-TANK SYSTEM PARAMETER

$S_1(m^2)$	$S_2(m^2)$	$S_3(m^2)$
$7.07e^{-4}$	$1.3e^{-3}$	$4.91e^{-4}$
R_1 (sec/m^2)	R_2 (sec/m^2)	R_3 (sec/m^2)
$1.74e^4$	$1.0e^4$	$2.33e^4$

TABLE IV
SIMULINK MATLAB PARAMETER SETTING

Noise 1 & 2	Impulse Fault Generator	Leakage Generator
Variance: 0.5, Sample time: 0.01s	Amplitude: 100, Period:20s, Pulse width: 2%, Phase delay: 5s	Step time: 20 Final value: 200
Gain	Gain 1	Gain 2

7.07*10^-4	3900	1.724*10^4
Gain 4	Gain 5	Gain 6
1.3*10^-3	10^4	2.3351*10^4
Gain 8	Gain 9	Gain 10
3900	4.91*10^-4	100

Conflict of Interest

The authors declare no conflict of interest in the publication process of the research article.

Author Contributions

N.M.M. Sobran supervised the study, analyse the results and writing the overall paper. M.D.M Yusoff planned the experimentation simulation and data collections. N.M. Ali, M. Zulkiflie and M. M. Ghazaly draft review and advised correction and editing.

References

[1] L. Jia, Q. Gao, Z. Liu, H. Tan, and L. Zhou, “Multidisciplinary fault diagnosis of complex engineering systems: A case study of nuclear power plants,” *International Journal of Industrial Ergonomics*, vol. 80, p. 103060, Nov. 2020, doi: 10.1016/j.ergon.2020.103060.

[2] S. W. Kim, J. H. Kong, S. W. Lee, and S. Lee, “Recent Advances of Artificial Intelligence in Manufacturing Industrial Sectors: A Review,” *Int. J. Precis. Eng. Manuf.*, vol. 23, no. 1, pp. 111–129, Jan. 2022, doi: 10.1007/s12541-021-00600-3.

[3] C. Wijaya, J.-Z. Liu, I.-J. Wang, J.-K. King, and C.-T. Yang, “Toward Predictive Maintenance for Aerospace Hot Press Furnace and the Current Monitoring With Machine Learning,” *IEEE Access*, vol. 13, pp. 110689–110708, 2025, doi: 10.1109/ACCESS.2025.3580729.

[4] F. Gao, Y. Li, Y. Zhao, X. Yang, J. Huang, and J. Gao, “Causal analysis enhanced stacked denoising autoencoder for closed-loop industrial process monitoring,” in *2025 IEEE 14th Data Driven Control and Learning Systems (DDCLS)*, May 2025, pp. 1448–1453. doi: 10.1109/DDCLS66240.2025.11065474.

[5] K. Yan, Z. Ji, and W. Shen, “Online fault detection methods for chillers combining extended kalman filter and recursive one-class SVM,” *Neurocomputing*, vol. 228, pp. 205–212, Mar. 2017, doi: 10.1016/j.neucom.2016.09.076.

[6] Z. Wang, J. Sun, X. Lu, Y. Hu, and D. Zhang, “SAE-CCA-based Fault Detection of Tandem Cold Rolling,” in *2021 33rd Chinese Control and Decision Conference (CCDC)*, May 2021, pp. 6773–6778. doi: 10.1109/CCDC52312.2021.9601870.

[7] X. Bi and J. Zhao, “A novel orthogonal self-attentive variational autoencoder method for interpretable chemical process fault detection and identification,” *Process Safety and Environmental Protection*, vol. 156, pp. 581–597, Dec. 2021, doi: 10.1016/j.psep.2021.10.036.

[8] S. Belagoune, N. Bali, A. Bakdi, B. Baadji, and K. Atif, “Deep learning through LSTM classification and regression for transmission line fault detection, diagnosis and location in large-scale multi-machine power systems,” *Measurement*, vol. 177, p. 109330, June 2021, doi: 10.1016/j.measurement.2021.109330.

[9] I. Lomov, M. Lyubimov, I. Makarov, and L. E. Zhukov, “Fault detection in Tennessee Eastman process with temporal deep learning models,” *Journal of Industrial Information Integration*, vol. 23, p. 100216, Sept. 2021, doi: 10.1016/j.jii.2021.100216.

[10] M. Jalayer, C. Orsenigo, and C. Vercellis, “Fault detection and diagnosis for rotating machinery: A model based on convolutional LSTM, Fast Fourier and continuous wavelet transforms,” *Computers in Industry*, vol. 125, p. 103378, Feb. 2021, doi: 10.1016/j.compind.2020.103378.

[11] K. B. Lee, S. Cheon, and C. O. Kim, “A Convolutional Neural Network for Fault Classification and Diagnosis in Semiconductor Manufacturing Processes,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 30, no. 2, pp. 135–142, May 2017, doi: 10.1109/TSM.2017.2676245.

[12] Z. Chang, A.-J. Zhang, H. Wang, J. Xu, and T. Han, “Photovoltaic Cell Anomaly Detection Enabled by Scale Distribution Alignment Learning and Multiscale Linear Attention Framework,” *IEEE Internet of Things Journal*, vol. 11, no. 16, pp. 27816–27827, Aug. 2024, doi: 10.1109/JIOT.2024.3403711.

[13] D. P. Yedurkar, T. Schlech, and M. G. R. Sause, “A Systematic Review on Smart and Predictive Maintenance in Tool Condition Monitoring,” *IEEE Access*, vol. 13, pp. 106246–106286, 2025, doi: 10.1109/ACCESS.2025.3579204.

[14] X. Allka, P. Ferrer-Cid, J. M. Barcelo-Ordinas, and J. Garcia-Vidal, “Pattern-Based Attention Recurrent Autoencoder for Anomaly Detection in Air Quality Sensor Networks,” *IEEE Transactions on Network Science and Engineering*, vol. 11, no. 6, pp. 6372–6381, Nov. 2024, doi: 10.1109/TNSE.2024.3454459.

[15] J. Z. Shi, A. Elshanti, F. Gu, and A. Ball, “A Desk-top tutorial Demonstration of Model-based Fault Detection and Diagnosis,” p. 20.

[16] Y. Zhao, X. He, M. G. Pecht, J. Zhang, and D. Zhou, “Detection and detectability of intermittent faults based on moving average T2 control charts with multiple window lengths,” *Journal of Process Control*, vol. 92, pp. 296–309, Aug. 2020, doi: 10.1016/j.jprocont.2020.07.002.

[17] I. Jaffel, O. Taouali, M. F. Harkat, and H. Messaoud, “A Fault Detection Index Using Principal Component Analysis And Mahalanobis Distance,” *IFAC-PapersOnLine*, vol. 48, no. 21, pp. 1397–1401, 2015, doi: 10.1016/j.ifacol.2015.09.720.

[18] T. J. Rato and M. S. Reis, “Fault detection in the Tennessee Eastman benchmark process using dynamic principal components analysis based on decorrelated residuals (DPCA-DR),” *Chemometrics and Intelligent Laboratory Systems*, vol. 125, no. Supplement C, pp. 101–108, June 2013, doi: 10.1016/j.chemolab.2013.04.002.

[19] D. E. Spina *et al.*, “Comparison of autoencoder architectures for fault detection in industrial processes,” *Digital Chemical Engineering*, vol. 12, p. 100162, Sept. 2024, doi: 10.1016/j.dche.2024.100162.

[20] J. Yu, X. Liu, and L. Ye, “Convolutional Long Short-Term Memory Autoencoder-Based Feature Learning for Fault Detection in Industrial Processes,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–15, 2021, doi: 10.1109/TIM.2020.3039614.