Proceedings of the 11th RSI International Conference on Robotics and Mechatronics (ICRoM 2023), Dec. 19-21, 2023, Tehran, Iran

Practical Implementation of Multiple Faults in a Coupled-Tank System: Verified by Model-Based Fault Detection Methods

Hamid Reza Chavoshi
K.N. Toosi University of Technology
Tehran, Iran
hr.chavoshi@email.kntu.ac.ir

Ali Khoshlahjeh Sedgh
K.N. Toosi University of Technology
Tehran, Iran
khoshlahjehali@email.kntu.ac.ir

Mahdi Aliyari Shoorehdeli
Department of Mechatronics Engineering, Faculty of Electrical Engineering
K.N. Toosi University of Technology
Tehran, Iran
aliyari@kntu.ac.ir

Hamid Khaloozadeh
Industrial Control Center of Excellence, Systems and Control Engineering
K.N. Toosi University of Technology
Tehran, Iran
h_khaloozadeh@kntu.ac.ir

***Abstract*— In order to prevent system failures, fault detection and performance assessment are always essential factors. This article investigates the practical implementation of common faults in a two-tank networked control system, one of the most widely utilized liquid-level control systems in the food industries, petrochemicals, and many more. The system was modeled and controlled, and then, faults were conducted on the system, including leakage, sedimentation, sensor bias, and pump stuck, which covered a wide range of fault types. After that, model-based fault detection methods, such as unknown input observer (UIO), parity, and factorization methods, were employed to evaluate fault implementation. By thresholding residuals of fault detection methods, the factorization method shows the most effective performance concerning false alarm rates (FAR), missed alarm rates (MAR), and detection delays (DD). Using the UIO is a suitable substitute, and parity is only appropriate for announcing the fault start time.**

***Keywords*— Fault detection, Two-tank Liquid-level networked control system, unknown input observer, Parity, Factorization.**

I. Introduction

The Coupled-Tank Level Control System (CTLCS) is a crucial component of the food industry, papermaking, boiler processes, petrochemical industries, and many others [1]. Two-tank systems are designed to maintain a constant liquid level, but their nonlinear nature and faults pose control challenges [2].

It has been shown by [3] that the PID controller family can provide an appropriate response to CTLCS if the nominal values of the plant parameters are known in the frequency domain; therefore, knowing the system model is extremely useful. Based on existing physical relationships, the CTLCS mathematical model parameters will be determined similarly to the references [4, 5].

Different types of faults are classified according to their nature. Faults are generally divided into three categories: component, actuator, and sensor. Examples of CTLCS faults are listed below [6].

- Faults in components include leakage and sedimentation of pipes, stuck middle/output valves, and leakage tanks.
- Actuator faults usually result from stuck pumps and controller faults.
- Sensor faults are sensor bias, and sensor output stuck.

On the other hand, faults can also be classified based on their time-variant behavior, and three types can be distinguished: abrupt, incipient, and intermittent. This article tries to implement all these types of faults in the CTLCS.

As systems become larger and more complex, the necessity of fault detection increases to prevent system failures. Fault detection methods are divided into two main categories: signal-based and model-based methods. Since the CTLCS model is known, model-based fault detection methods are becoming more prevalent in CTLCSs. Model-based methods generally use the residual approach, in which a residual crossing a limit indicates a fault. Different ways can produce residuals sensitive to faults and robust to other input signals, such as reference input, disturbance, and noise. The three most effective methods are unknown input observer (UIO), parity, and factorization [7].

There are two significant points in model-based fault detection: fault detectability and decision-making based on the residual signal [7]. Fault detectability means a non-zero transfer function between residual and fault i ($[Grf_i(s)]\neq 0$) which indicates the existence of non-zero residual. On the other hand, $[Grf_i(0)]\neq 0$ means strong detectability and indicates the existence of non-zero residuals during all times that fault occurs.

Decision-making based on residual signals can be done using thresholds. There are two general approaches to thresholding: statistical and adaptive. Simple thresholding has a statistical basis and implements ζ-standard deviation by considering the normal distribution for the residual signal. Adaptive thresholding can be used in sliding windows to improve the accuracy of this method when the system has different operating points.

979-8-3503-0810-5/23/$31.00 ©2023 IEEE

The performance of fault detection systems can be measured by three indices: false alarm rate (FAR), missed alarm rate (MAR), and detection delay (DD) [8]. Detection methods should be chosen according to the degree of fault risk. The loss of a fault in an airport may result in an airplane crash, and MAR is more important there, whereas ignoring minor faults to increase cement production makes FAR more critical. Thus, it is necessary to consider FAR and MAR for choosing the appropriate fault detection method and have a tradeoff between them.

This article deals with the practical implementation of four common faults of a CTLCS, which are obtained based on the nature of the system and research done about this system. In order to detect these faults, UIO, parity, and factorization fault detection methods were used with statistical thresholding after modeling and controlling the system. Finally, three famous criteria are used to evaluate the implemented fault detection methods: FAR, MAR, and DD.

II. CTLCS System

A. System Description

Based on Fig. 1, this paper used the LD DIDACTIC brand system, which includes two tanks, one pump, and three valves. As a result of the significant difference between the liquid level of each tank, the outflow rates of each tank are proportional to the liquid level in each tank. Each tank has an independent outflow rate. The pump has a voltage range of 3.6V to 12V. The pump flow rate and tank output can also be adjusted manually using the adjustment valves. This system comprises a piezoelectric pressure sensor and a capacitive level sensor with linear voltage outputs. A data logger with two digital-to-analog converters (DAC) and four analog-to-digital converters (ADC) was used to apply commands and receive information, which can connect wirelessly to the computer through the Wi-Fi module.

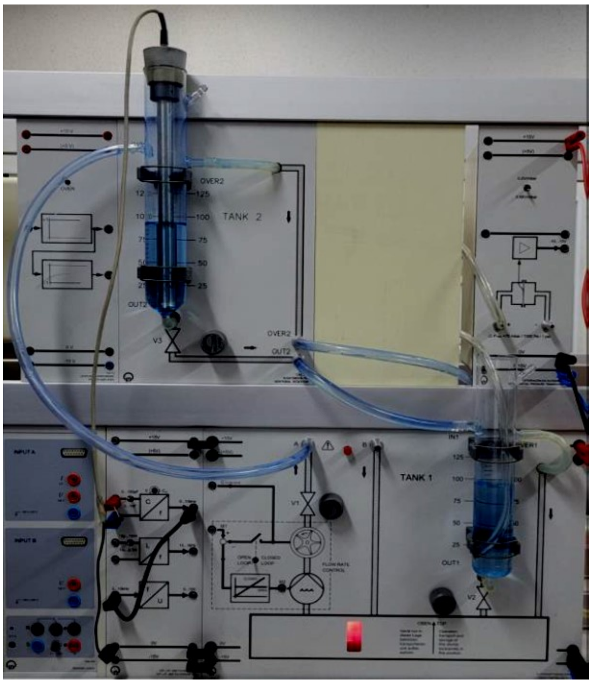

Fig. 1. Coupled-tank level control system.

B. System Model

Table I contains the description of system parameters.

TABLE I. Parameters Estimation.

Parameter	Description
P_i	Pressures at the bottoms of the tank i
R_{21}	Resistance of the pipe between the tanks
R_1	Resistance of the pipe attached to tank1 that allows water to flow out
C	The capacitance of the tanks
A	The cross-sectional area of the tanks
ρ	The mass density of water
g	Gravitational acceleration
μ	Absolute viscosity
l_{21}	Length of the pipe between the tanks
l_1	Pipe length of the tank1
d_{21}	The diameter of the pipe between the tanks
d_1	Pipe diameter of the tank1

An overview of the system and the names of its components is shown in Fig. 2.

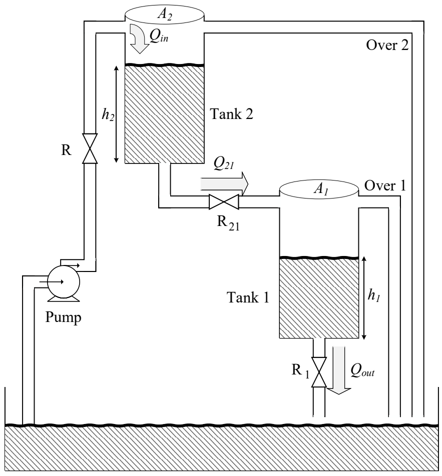

Fig. 2. Overview of the system and components.

System equations are as follows [4].

$$\left.\begin{array}{l}\begin{cases}\dot{V}_2 = A_2\dot{h}_2 = Q_{in} - Q_{21}\\ \dot{V}_1 = A_1\dot{h}_1 = Q_{21} - Q_{out}\end{cases}\\ Bernoulli:\begin{cases}Q_{21} = \beta_{21}\sqrt{h_2}\\ Q_{out} = \beta_{out}\sqrt{h_1}\end{cases}\end{array}\right\}\rightarrow\begin{cases}\dot{h}_2 = \dfrac{1}{A_2}Q_{in} - \dfrac{\beta_{21}}{A_2}\sqrt{h_2}\\ \dot{h}_1 = \dfrac{\beta_{21}}{A_1}\sqrt{h_2} - \dfrac{\beta_{out}}{A_1}\sqrt{h_1}\end{cases} \tag{1}$$

β are the coefficients related to outflow, and we are dealing with a nonlinear system in which linearization (around 3.9cm for the lower tank and around 2.55cm for the upper tank) can be written as follows based on [9].

$$\begin{aligned}\dot{P}_2 &= \frac{Q_{in} - Q_{21}}{C_2} = \frac{Q_{in} - \left(\dfrac{P_2}{R_{21}}\right)}{C_2} = -\frac{P_2}{R_{21}C_2} + \frac{Q_{in}}{C_2}\\ \dot{P}_1 &= \frac{Q_{21} - Q_{out}}{C_1} = \frac{\left(\dfrac{P_2}{R_{21}}\right) - \left(\dfrac{P_1}{R_1}\right)}{C_1} = \frac{P_2}{R_{21}C_1} - \frac{P_1}{R_1C_2}\end{aligned} \tag{2}$$

According to the similarity of the tanks ($C_1 = C_2 = C$), we have:

$$\begin{bmatrix} \dot{P}_2 \\ \dot{P}_1 \end{bmatrix} = \begin{bmatrix} -\frac{1}{R_{21}C} & 0 \\ \frac{1}{R_{21}C} & -\frac{1}{R_1C} \end{bmatrix} \begin{bmatrix} P_2 \\ P_1 \end{bmatrix} + \begin{bmatrix} \frac{1}{C} \\ 0 \end{bmatrix} Q_{in} \tag{3}$$

$$R_{21} = \frac{128 \times \mu \times l_{21}}{\pi \times d_{21}^4},\ R_1 = \frac{128 \times \mu \times l_1}{\pi \times d_1^4}, C = \frac{A}{\rho g}$$

Therefore, the transfer function of the single input multi output (SIMO) system is obtained as follows.

$$G(s) = \begin{bmatrix} \frac{0.0475}{s + 0.08368} \\ \frac{0.0026}{s^2 + 0.1076s + 0.0020} \end{bmatrix} \tag{4}$$

C. Controller Design

Since each tank has a similar dynamic to $\frac{1}{s}$, working in the open-loop conditions of the system is challenging because of the small size and quick filling of the tanks. Hence, the importance of controlling the system cannot be overstated. Since there is only one actuator (pump) on the system, and its indirect effect comes through to the lower tank level, it was decided to control the upper tank liquid level. The pump needs 3.6V of input to start working; this value was added directly to the control signal as a feedforward, and a PI with $k_p=0.1$ and $k_i=0.01$ was fine-tuned. The results of the upper tank system setpoint tracking, the lower tank behavior, and the control signal are shown in Fig. 3 and Fig. 4.

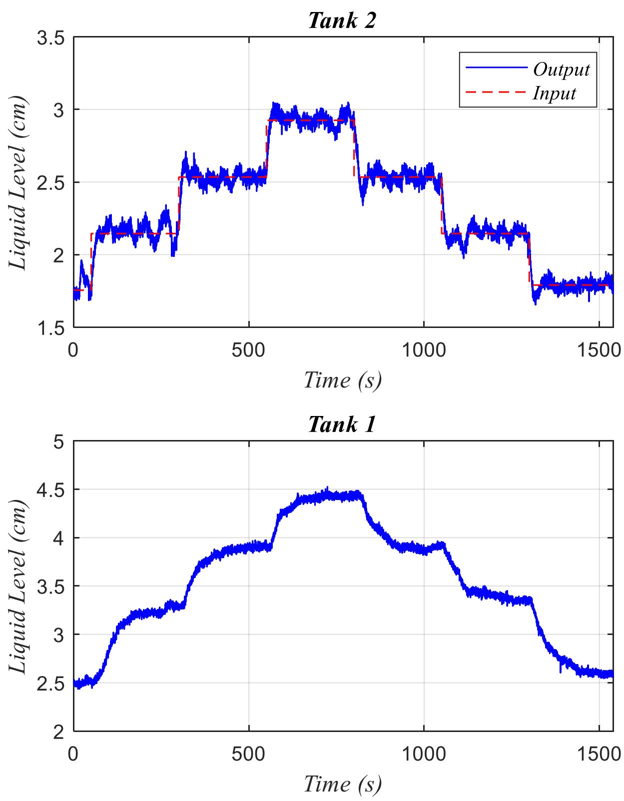

Fig. 3. System input and output in normal mode.

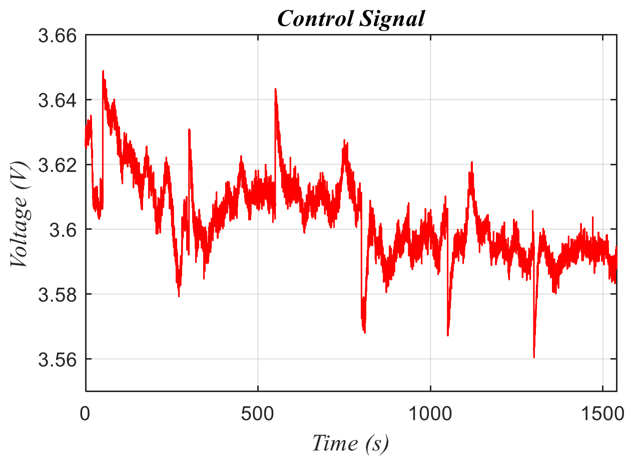

Fig. 4. Control signal in normal mode.

III. FAULT IMPLEMENTATION AND DETECTION

A. Fault Scenarios

Using linearized system equations requires the appropriate reference input to keep the system around the selected operating point. Reference inputs in this article consist of steps with amplitudes of 1.75, 2.15, 2.55, and 2.95 that go up and down. Every step lasts 250 seconds. The first 200 seconds of the data have been removed to remove disturbance effects caused by bubbles in the system pipes at system startup. Thus, based on the assumptions of working with the system, the introduction section, and a review of the references, including [10-12], different types of faults have been implemented in the system, as shown in Table II.

TABLE II. FAULT SCENARIOS.

Fault Category	Common Fault	Time Behavior	Implementation Description
Component	leakage	incipient	from the 450s until the end of the simulation and increased slightly every 150 seconds.
	sedimentation	abrupt	from the 450s until the end of the simulation.
Actuator	pump stuck	intermittent	has a random presence until the end of the simulation.
Sensor	lower tank -2 bias	abrupt	from the 450s until the end of the simulation.
	upper tank -1 bias	abrupt	from the 450s until the end of the simulation.

B. Model-Based Fault Detection

Once the system has been modeled and controlled and faults implemented, the next step is to utilize model-based methods for detecting those faults. These methods include UIO, parity, and factorization. Noise and disturbance are not considered in our designs for simplicity. Based on [12, 13], UIO, parity, and factorization methods are designed.

- UIO

It is possible to detect faults by using different types of UIO. Below are comprehensive equations for one type of UIO.

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) + B_f f(t) + B_d d(t) \\ y(t) = Cx(t) + D_f f(t) + D_d d(t) \end{cases}$$
$$\begin{gathered} z(t+1) = Fz(t) + Ju(t) + Ky(t) \\ r(t) = L_1 z(t) + L_2 u(t) \\ e(t) = Fz(t) - Tx(t) \end{gathered} \tag{5}$$

Without an unknown disturbance input, the above equations are simplified as follows.

$$\begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L\left(y(t) - \hat{y}(t)\right) + LD_f f(t) \\ \hat{y}(t) = C\hat{x}(t) + Du(t) \\ \dot{e}(t) = (A - LC)e(t) + LD_f f(t) \\ r(t) = e(t) - D_f f(t) \end{cases} \tag{6}$$

Consequently, after ensuring the observability of the open loop system and $D_f \neq 0$, the L matrix (which determines the error poles (eigenvalues of A-LC)) is designed to be stable and ten times faster than the system.

$$L = (1.0e+07) \times \begin{bmatrix} 1.0673 & 0 \\ 0.4129 & 5.8564 \end{bmatrix} \tag{7}$$

The observer outputs are subtracted from the system outputs to produce two residuals. The upper tank residual does not change with faults due to the robustness of the PI controller. In contrast, the lower tank residual is used for fault detection.

- Parity

Because the system is observable ($rank(O)=2$) and the matrix C has full rank ($rank(C)=2$), we can use the following inequality to calculate s.

$$\frac{rank(O)}{rank(C)} \leq s \leq rank(O) - rank(C) + 1 \tag{8}$$

According to $s=1$, $H_{o,s}$ and its null space are obtained as follows.

$$N(H_{o,s}) = \begin{bmatrix} -0.7011 & 0.0022 & 0.7130 & -0.0022 \\ -0.0061 & -0.7054 & -0.0016 & 0.7088 \end{bmatrix} \tag{9}$$

Disregarding the disturbance makes it possible to select each row of $H_{o,s}$ as v_s, and by choosing the first row, the last step is to calculate H_{us}.

$$H_{u,s} = \begin{bmatrix} D & 0 \\ CB & D \end{bmatrix} \tag{10}$$

Consequently, the parity method residual derived from (11) is used to detect faults.

$$\begin{gathered} r(k) = v_s \times \left(y_s(k) - H_{u,s} u_s(k)\right) \\ y_s(k) = \begin{bmatrix} y_s(k-s) \\ y_s(k-s+1) \\ \vdots \\ y_s(k) \end{bmatrix}, u_s(k) = \begin{bmatrix} u_s(k-s) \\ u_s(k-s+1) \\ \vdots \\ u_s(k) \end{bmatrix} \end{gathered} \tag{11}$$

- Factorization

Considering the stability and observability of the open loop system, the operation of left coprime factorization can be summarized as $M_u = I_{2\times2}$, $N_u = G$. On the other hand, the absence of disturbance makes it possible to consider R with any $\alpha I_{2\times2}$. By considering $\alpha=1$, two residuals are obtained from (12), similar to the reasoning at the end of the UIO design; we will use the lower tank residual.

$$r(t) = R \times \left(M_u y(t) - N_u u(t)\right) \tag{12}$$

It should be noted that the above methods have been implemented in Simulink/MATLAB.

C. Thresholding

Considering the normal distribution for the residuals and implementing ζ-standard deviation, thresholds can be obtained for the UIO, parity, and factorization methods.
By calculating the mean and variance of the normal data residual, 2σ was considered a suitable criterion in different methods.
Therefore, the upper and lower bands have been determined for the residuals of the methods so that normal data can be distinguished from faults.

IV. Experimental Result

This section discusses the results of practical fault implementation and detection methods. For the purpose of keeping the article short, a fault scenario was used to illustrate the process. The upper tank -1 bias scenario was the candidate, and its results are described in Fig. 5 and Fig. 6.

Fig. 5 and Fig. 6 reveal a sudden disruption in controller performance due to an abrupt fault after 450 seconds. A quick compensation was made due to the robustness of the PI controller. By adjusting the control signal to compensate for bias -1, the liquid level of the upper tank rises, the flow rate between the two tanks increases, and the level of the lower tank rises.

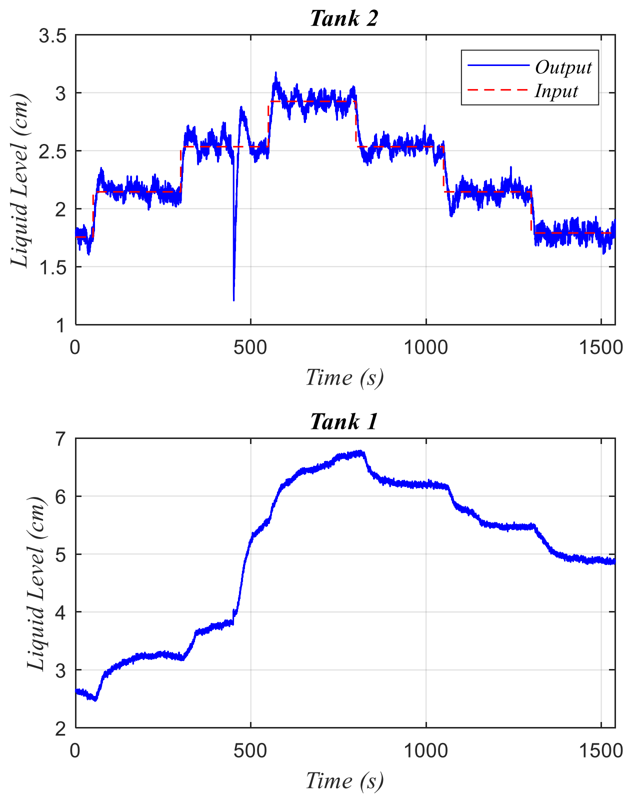

Fig. 5. System input and outputs in the presence of -1 bias fault in the upper tank sensor.

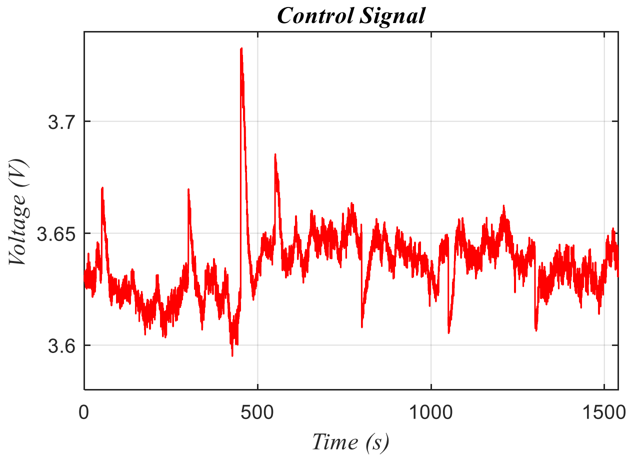

Fig. 6. Control signal in the presence of -1 bias fault in the upper tank sensor.

The results of implementing model-based fault detection methods and their residual thresholding are shown in Fig. 7, Fig. 8, and Fig. 9. Due to the sudden peak in the UIO residual, the beginning of the fault is visible, and the subsequent detection will improve over time. Due to the sudden peak at the beginning of the fault, the parity method also has a good DD, but it does not improve in detection over time. There is no strong detectability attribute to the parity method. Although the factorization method displays delayed fault detection performance, it can detect faults without error within moments after the fault is applied.

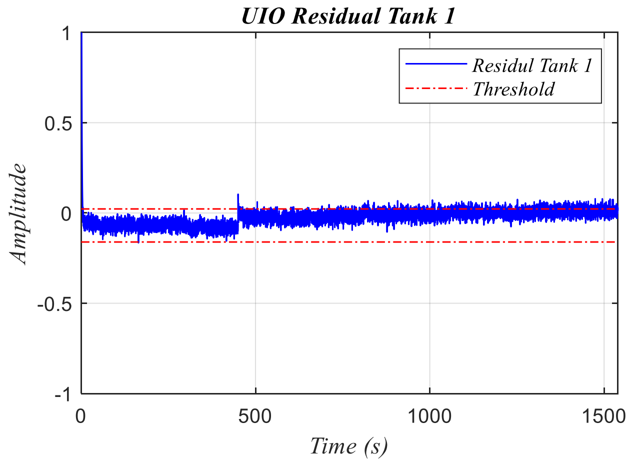

Fig. 7. Thresholding of the UIO residual in the presence of -1 bias fault in the upper tank sensor.

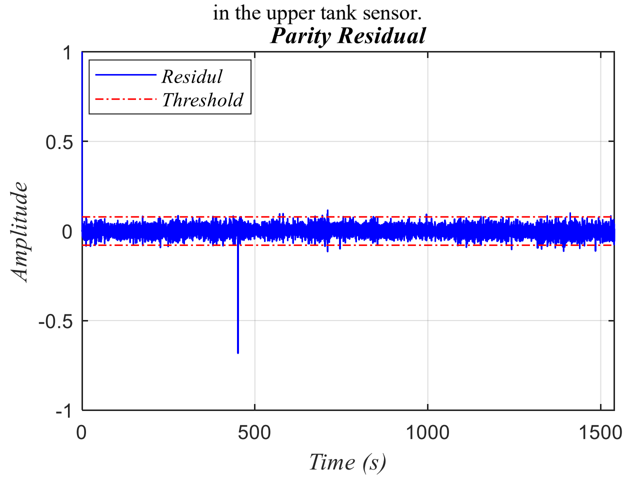

Fig. 8. Thresholding of the parity residual in the presence of -1 bias fault in the upper tank sensor.

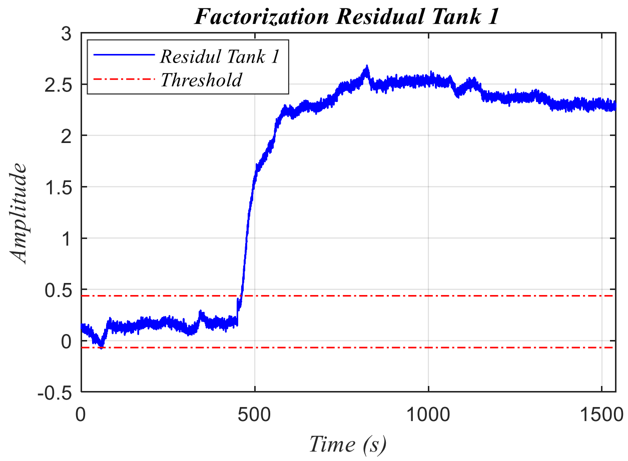

Fig. 9. Thresholding of the factorization residual in the presence of -1 bias fault in the upper tank sensor.

Finally, a detailed examination of fault detection methods can be done through numerical comparisons of FAR, MAR, and DD criteria with the help of Table III. The decision-maker considers factorization to be the best method based on the MAR criterion, and despite the poor performance of the other two methods, UIO is preferable to parity.

It is challenging to decide according to the FAR criteria, and all methods have performed well in this regard; however, in general, UIO can be considered the best method in this aspect, but in the bias scenario, factorization performance can be regarded better than UIO. In the scenarios of leakage and sedimentation, the same performance can be observed for factorization and parity (or, with great care, parity is better than factorization), and in the scenarios of lower tank-2 bias and upper tank -1 bias, the performance of parity is reported to be weak.

Finally, the DD values of the parity method are small, which makes the reaction of this method appropriate for fast fault detection, and the subsequent choices are UIO and factorization. In some faults where the DD values of all methods are small, we see the relative superiority of UIO and factorization. However, the proper performance of the parity method must be addressed.

TABLE III. PERFORMANCE ASSESSMENT OF THE MODEL-BASED FAULT DETECTION METHODS.

Fault	Performance Index		UIO	Parity	Factorization
leakage	*DD*	s	1.8	0.4	6.4
	MAR	%	83.75	93.04	0.59
	FAR	%	0.93	6.17	7.75
sedimentation	*DD*	s	80.8	1.6	131.6
	MAR	%	50.92	90.54	13.52
	FAR	%	1.04	12.91	13.43
pump stuck	*DD*	s	0.2	0.4	0.2
	MAR	%	25.57	82	0
	FAR	%	1.13	0.5	1.07
lower tank -2 bias	*DD*	s	0.2	5.6	0.2
	MAR	%	66.54	93.90	0
	FAR	%	1.06	6.90	0
upper tank -1 bias	*DD*	s	0.2	0.2	10.4
	MAR	%	85.47	99.32	1.01
	FAR	%	1.24	0.71	0.18

V. CONCLUSION

Implementing of UIO, parity, and factorization methods to detect the four common faults of the CTLCS system yielded three general results. First, considering the significant low values of MAR in the factorization method and the absence of significant differences in FAR and DD values between other methods, the best performance can be attributed to factorization. Second, in the absence of the factorization method, UIO can be considered as an alternative method for fault detection in non-sensitive industries because high MAR values increase the probability of missing fault detection. Finally, the weak results of the parity method in the FAR and MAR criteria and its acceptable results in the DD criterion indicate the presence of detectability in this method and the lack of strong detectability. As a result, the only use of this method is to announce the start of the fault.

There are two suggestions for future work. First, Consider the bubbles inside the pipes of this system as disturbances and model them to improve the performance of model-based methods. Also, entering the fluid into the tank from the bottom reduces the bubbles inside the pipes. Second, implementing cyber-attacks on this system, such as Denial of Service (DoS) attacks, Man in the Middle (MITM) attacks, False Data Injection (FDI) attacks, and many other cases, and separating them from faults is a practical issue necessary to prevent failure caused by malicious purposes.

REFERENCES

[1] D. S. Bhatt and S. N. Sharma, "State estimation of a benchmark two-tank problem: A Carleman linearization approach," *IFAC-PapersOnLine,* vol. 53, no. 2, pp. 2311-2316, 2020.

[2] M. Changela and A. Kumar, "Designing a controller for two tank interacting system," *International Journal of Science and Research,* vol. 4, no. 5, pp. 589-593, 2015.

[3] S. Ozcelik, R. Sosa, M. Faruqi, and M. Abudiab, "Design and Comparison of Various Controllers for a Two-Tank Liquid-Level System," in *2005 GSW*, 2022.

[4] H. R. Patel and V. A. Shah, "A passive fault-tolerant control strategy for a non-linear system: An application to the two tank conical non-interacting level control system," *Maskay,* vol. 9, no. 1, pp. 1-8, 2019.

[5] A. G. M. Mohammed, H. A. Sirag, and M. B. Amien, "Design of Fault Tolerant Control System (strategy) for Two Water Tanks," in *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, 2018: IEEE, pp. 1-6.

[6] E. Chanthery and A. Subias, "Diagnosis approaches for detection and isolation of cyber attacks and faults on a two-tank system," in *30th International Workshop on Principles of Diagnosis DX'19*, 2019.

[7] J. Chen and R. J. Patton, *Robust model-based fault diagnosis for dynamic systems*. Springer Science & Business Media, 2012.

[8] J. Xu, J. Wang, I. Izadi, and T. Chen, "Performance assessment and design for univariate alarm systems based on FAR, MAR, and AAD," *IEEE Transactions on Automation Science and Engineering,* vol. 9, no. 2, pp. 296-307, 2011.

[9] A. W. Werth, "Towards distinguishing between cyber-attacks and faults in cyber-physical systems," 2014.

[10] M. Amor, T. Ladhari, S. Hadj Said, and F. M'Sahli, "Fault-Tolerant Control for Three-Tank System in Case of Sensor Faults," *Mathematical Problems in Engineering,* vol. 2021, pp. 1-13, 2021.

[11] H. Patel and V. Shah, "Fault Tolerant Control Design Based on Takagi-Sugeno Fuzzy Logic: Application to a Three-Tank System," in *11th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2019)*, 2019: Atlantis Press, pp. 256-266.

[12] I. Hameed, E. I. El-Madbouly, and M. I. Abdo, "Sensor and actuator fault-hiding reconfigurable control design for a four-tank system benchmark," *International Journal of Innovative Computing, Information and Control,* vol. 11, no. 2, pp. 679-690, 2015.

[13] I. Izadi Najaf Abadi, "Fault diagnosis in sampled-data systems," 2006.