IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 14, 2018, accepted September 29, 2018, date of publication October 8, 2018,
date of current version November 14, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2874752

Sim3Tanks: A Benchmark Model Simulator for Process Control and Monitoring

ARLLEM O. FARIAS[1], GABRIEL ALISSON C. QUEIROZ[1], IURY V. BESSA[2,3], RENAN LANDAU P. MEDEIROS[3], LUCAS C. CORDEIRO[4], AND REINALDO M. PALHARES[5], (Member, IEEE)

[1]Graduate Program in Electrical Engineering, Federal University of Amazonas, Manaus 69067-005, Brazil
[2]Graduate Program in Electrical Engineering, Federal University of Minas Gerais, Belo Horizonte 31270-901, Brazil
[3]Department of Electricity, Federal University of Amazonas, Manaus 69067-005, Brazil
[4]School of Computer Science, University of Manchester, Manchester M13 9PL, U.K.
[5]Department of Electronics Engineering, Federal University of Minas Gerais, Belo Horizonte 31270-901, Brazil

Corresponding author: Arllem O. Farias (arllemfarias@ufam.edu.br) and Iury V. Bessa (iurybessa@ufam.edu.br)

This work was supported in part by the Brazilian agencies FAPEAM under Grant 062.00719/2016, in part by CNPq and FAPEMIG, in part by the PROPG-CAPES/FAPEAM Scholarship Program, and in part by the 111 Project under Grant B16014.

ABSTRACT This paper describes a simulator for the three-tank system process named Sim3Tanks. This process presents a hybrid and nonlinear behavior and it is subject to different kinds of perturbations, faults, and noises. Sim3Tanks was developed in the MATLAB/Simulink environment and can be used via graphical user interface, Simulink block diagram, and command-line. Sim3Tanks is suitable for studying and developing process control, fault detection and isolation, and fault tolerant control strategies for nonlinear multi-variable systems. In order to illustrate the potential of Sim3Tanks, four scenarios are discussed throughout this paper: PID control strategies for the level and flow rates; a fault detection algorithm based on unscented Kalman filter and generalized likelihood ratio; a fault isolation system based on Bayesian networks; and a control reconfiguration based on static virtual actuator and sensor. A video demonstration of Sim3Tanks can be found at https://github.com/e-controls/Sim3Tanks.

INDEX TERMS Virtual laboratory, three-tank system, fault detection and isolation, fault tolerant control.

I. INTRODUCTION

Hybrid systems have their dynamics described by a mix of continuous and discrete-event variables [1]–[3]. Such systems have become particularly important with the emerging of embedded and cyber-physical systems (ECPS), which are systems that always present as fundamental property the hybridness [4]. Most industrial processes involve a top layer with abstract protocols, activation and coordination algorithms, which typically handle discrete data, and a bottom layer with control algorithms of concrete devices, where the control laws are continuous or their resulting signals must be translated to continuous signals. In addition, these processes are generally multi-variable, nonlinear, and they are usually described by complex mathematical models [5], [6].

Process control systems have been deeply studied to ensure the stability and performance of industrial processes [7], [8]. These studies have developed strategies that allow achieving high performance and reliability of industrial plants and equipment, thus increasing productivity and income, in addition to improving operator and environment safety. Commonly, industrial process control involves variables as pressure, temperature, level, and flow rate. In this context, the three-tank system benchmark [9], which has level and flow rate as process variables, has been widely used to study and develop novel techniques and solutions for process control [10], [11], fault detection and isolation (FDI) [12], and hybrid systems modeling [13], since it provides flexible and complex dynamics that can be reconfigured to present different operation modes. Additionally, it offers different open challenges related to process control, FDI, and fault tolerant control (FTC) due to its parameter variations, nonlinear dynamics, and fault occurrences.

Due to the similarity of the three-tank system benchmark with some industrial processes (e.g., wastewater treatment, stirred-tank reactor, and refineries), it is important the development of test-bed emulators and simulators, which provide realistic experiences of operation, control, and monitoring for these kind of processes. In particular, the use of realistic

simulators can reduce costs and also provide some experience of how to manipulate real world processes, thus avoiding the risk of damaging high-cost equipment. These simulators also permit the computational analysis of process control and monitoring techniques before they are actually applied to real world processes.

The three-tank system used in this paper is based on the Heiming and Lunze's work [9], which is a benchmark model designed and implemented within the COSY programme of the European Science Foundation and aims to investigate different strategies of fault tolerant control. Their benchmark consists of three tanks connected side-by-side by means of pipes; valves to control the water flow through the pipes; level and flow sensors; and two identical pumps that provide the input water flow to the left and right tank of the system. In nominal case, only the left and middle tank are used; the right tank is not used and acts as redundant hardware. The main goal of this system is to provide a continuous water flow in the output pipe of the middle tank. For this purpose, a proportional-integral (PI) controller is used to regulate the input water flow to the left tank and an on/off controller is used to regulate the water flow supply to the middle tank.

In particular, the simulator described here is a suitable virtual environment for studying FDI [14], [15] and FTC [16], [17] techniques.

A. CONTRIBUTIONS

This paper presents Sim3Tanks,[1] which is a virtual laboratory software that can provide to students, engineers, and researchers some practical experience with nonlinear hybrid processes control and monitoring. Sim3Tanks was developed using the MATLAB/Simulink tool, version 9.2.0.538062 (R2017a), and it allows the behavior analysis of the three-tank system for different operation modes with occurrence of faults and disturbances. Specifically, this simulator allows the user to implement and simulate any control strategy as well as fault detection and isolation algorithms. The main features of Sim3Tanks are:

- intuitive graphical interface with easily manipulable components to define the simulation parameters and preferences as desired;
- adjustable physical structure, making possible the sizing according to the specifications of a real system;
- definition of different operation modes, thus allowing the creation of scenarios with 1, 2, or 3 coupled tanks;
- addition of process noise (disturbances in states) and measurement noise (disturbances in outputs);
- handling of several types of control strategies, e.g., feedforward, state and output feedback;
- inclusion of different types of faults (e.g., abrupt and incipient) in actuators, plant, and sensors of the system.

Accordingly, Sim3Tanks is applicable to the following purposes:

- investigation of the dynamic behavior of coupled tanks;
- testing and validation of linearization techniques;
- linear and nonlinear control strategies;
- fault detection and diagnosis algorithms;
- fault tolerant control strategies;
- system identification techniques;
- virtual laboratory for didactic purposes.

In order to show the features and applicability of Sim3Tanks, examples of process control, FDI, and FTC strategies are implemented in the simulator and their results are also described and discussed by using the facilities of analysis tools provided by Sim3Tanks. In particular, the illustrative examples consist of well-known and simple control strategies, i.e., Ziegler-Nichols method for level and flow rate proportional-integral-derivative (PID) controller tuning, unscented Kalman filter based diagnosis observer for residual generation, generalized likelihood ratio for residual evaluation, Bayesian networks for fault isolation, and a control reconfiguration based on static reconfiguration blocks. Although the illustrative examples use techniques already consolidated in the literature, they show the applicability of Sim3Tanks for validation and simulation of more sophisticated techniques, including the state-of-the-art in process control and monitoring.

B. RELATED WORK

The three-tank system used in Sim3Tanks has already been studied by prior initiatives that developed similar simulation tools. Hashim *et al.* [18] have developed an user interface system to facilitate the understanding of the process dynamics, thus allowing data analysis and behavior of the three-tank system. However, it is focused on performing experiments with controllers and does not permit changes in the system physical structure, e.g., change the tanks height and diameter. Vinagre *et al.* [19] present a different three-tank simulator, where they used a multi-agent approach with the goal to provide a distributed control with one controller for each tank, in such a way that the tanks can communicate themselves to exchange information and cooperate to achieve their main goals. This approach is robust against failures in communications between the tanks, because each agent is capable of taking actions on its own, attempting to keep the performance as reliable as possible. The simulators presented by Bisták and Huba [20], [21], Chevalier *et al.* [22], and Dormido *et al.* [23] are proposed as remote laboratories, where students can obtain access to physical applications and equipment using Wi-Fi connection or via Ethernet LAN. These remote laboratories provide practical experiments of modeling and control of dynamic systems in real systems without students being afraid of damaging the real laboratory equipment or needing to wait a long time for practical lectures due to the limited time for laboratory sessions and scarce financial resources of universities.

The proposed simulator can be used via three different interfaces in the MATLAB environment: graphical user interface development environment (GUIDE), block diagram

[1] Link to download: https://github.com/e-controls/Sim3Tanks

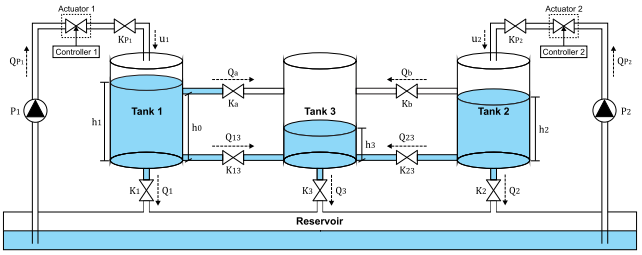


FIGURE 1. Three-tank system.

(i.e., via Simulink), and command-line (i.e., via MATLAB script). Sim3Tanks runs locally (without requiring internet connection) and allows the user to define scenarios with 1, 2, or 3 coupled tanks. In addition, it is able to simulate different faults in the system (e.g., blocking or clogging of valves, leakage in tanks, and malfunction of sensors and pumps), where each fault might occur with different behavior and magnitude. Thus, several strategies of process control (fault tolerant or not) and monitoring can be easily evaluated and compared separately or jointly.

C. OUTLINE

This paper is organized as follows: Section II describes the operation mode and faults that might occur in the three-tank system. Section III presents the Sim3Tanks features and its interfaces. Section IV describes the main control objectives, the implemented flow and level control strategies and their results. Section V presents the main FDI objectives, the implemented FDI strategy and its results. Section VI describes the main FTC objectives, the implemented FTC strategy and its results. Finally, Section VII concludes and describes future works.

II. THREE-TANK SYSTEM DESCRIPTION

The three-tank system consists of three cylindrical tanks connected by four pipes, which allow the fluid exchange between the lateral tanks (tanks 1 and 2) and the central tank (tank 3) in both directions, as illustrated in Fig. 1. At the top of each tank there is an ultrasonic sensor responsible for measuring the liquid level inside the tank. In all pipes of the system, there is a flow control valve (proportional or on/off) and a Hall effect flow sensor. As a result, the system has a total of twelve sensors, i.e., three level sensors (one per tank) and nine flow sensors (one per valve). In addition, the system has two servo-valves as actuators; the first actuator regulates the input flow to the tank 1 and is located before valve K_{P_1}; the second actuator regulates the input flow to the tank 2 and is located before valve K_{P_2}.

The upper pipes and valves that connect the tanks 1 and 2 to the tank 3 are located at the same height h_0 and are called transmission pipes and valves; the lower pipes and valves that connect the tanks 1 and 2 to the tank 3 are aligned with the base of the tanks and are called connection pipes and valves. At the bottom of each tank are the output pipes and valves; the dashed arrows illustrated in Fig. 1 indicate the reference direction of each flow.

TABLE 1. Values of Δh_a and Δh_b for the flows Q_a and Q_b.

Scenarios			Δh_a	Δh_b
$h_1 \le h_0$	$h_2 \le h_0$	$h_3 \le h_0$	0	0
$h_1 \le h_0$	$h_2 \le h_0$	$h_3 > h_0$	$h_0 - h_3$	$h_0 - h_3$
$h_1 \le h_0$	$h_2 > h_0$	$h_3 \le h_0$	0	$h_2 - h_3$
$h_1 \le h_0$	$h_2 > h_0$	$h_3 > h_0$	$h_0 - h_3$	$h_2 - h_3$
$h_1 > h_0$	$h_2 \le h_0$	$h_3 \le h_0$	$h_1 - h_0$	0
$h_1 > h_0$	$h_2 \le h_0$	$h_3 > h_0$	$h_1 - h_3$	$h_0 - h_3$
$h_1 > h_0$	$h_2 > h_0$	$h_3 \le h_0$	$h_1 - h_0$	$h_2 - h_0$
$h_1 > h_0$	$h_2 > h_0$	$h_3 > h_0$	$h_1 - h_3$	$h_2 - h_3$

The flow rates from the pumps P_1 and P_2 (i.e., Q_{P_1} and Q_{P_2}) are inputs defined by the user, whereas the flows through the transmission, connection, and output pipes can be determined, respectively, by using (1)–(3), where K_v, K_{i3}, and K_j correspond to the state of the flow control valves ($K_v, K_{i3}, K_j \in [0, 1]$) and $\beta = \mu S\sqrt{2\,g}$, where μ is the flow correction term, S is the pipe cross-sectional area, and g is the gravity acceleration constant. The difference of level Δh_v (for $v = a, b$) used in (1) depends on the levels h_1, h_2, and h_3 being above or below the transmission pipe height h_0; its values for each scenario are described in Table 1.

$$Q_v = K_v \beta \,\mathrm{sgn}(\Delta h_v)\sqrt{|\Delta h_v|}, \quad v = a, b. \tag{1}$$

$$Q_{i3} = K_{i3} \beta \,\mathrm{sgn}(h_i - h_3)\sqrt{|h_i - h_3|}, \quad i = 1, 2. \tag{2}$$

$$Q_j = K_j \beta \sqrt{h_j}, \quad j = 1, 2, 3. \tag{3}$$

The function sgn(·) indicates the flow direction inside the pipes and is given by:

$$\mathrm{sgn}(x) = \begin{cases} 1, & \text{if } x \ge 0 \\ -1, & \text{if } x < 0. \end{cases} \tag{4}$$

A. STATE EQUATIONS

The volume variation, $\dot{V}$, inside a cylindrical tank of cross-sectional area S_c, can be described by (5), where $\dot{h}$ is the level variation inside the tank, $\sum Q_{\text{in}}$ is the sum over all input flows into the tank, and $\sum Q_{\text{out}}$ is the sum over all output flows from the tank [9].

$$\dot{V} = S_c \times \dot{h} = \sum Q_{\text{in}} - \sum Q_{\text{out}}. \tag{5}$$

Assume that the input and state vectors are defined as $\boldsymbol{u} = [Q_{P_1}\ Q_{P_2}]^T$ and $\boldsymbol{x} = [h_1\ h_2\ h_3]^T$. Then, using (5) it is possible to describe the level in each tank (h_1, h_2, and h_3, cf. Fig. 1) as follows:

$$\dot{h}_1 = \frac{1}{S_c} K_{P_1} Q_{P_1} - \frac{1}{S_c}(Q_a + Q_{13} + Q_1), \tag{6}$$

$$\dot{h}_2 = \frac{1}{S_c} K_{P_2} Q_{P_2} - \frac{1}{S_c}(Q_b + Q_{23} + Q_2), \tag{7}$$

$$\dot{h}_3 = \frac{1}{S_c}(Q_{13} + Q_{23} + Q_a + Q_b) - \frac{1}{S_c} Q_3. \tag{8}$$

Therefore, the state equations described in (6)–(8) can be rewritten in the compact form as follows:

$$\dot{\boldsymbol{x}} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} m_1(\boldsymbol{x}) \\ m_2(\boldsymbol{x}) \\ m_3(\boldsymbol{x}) \end{bmatrix} + \begin{bmatrix} \frac{K_{\mathrm{P}_1}}{S_c} & 0 \\ 0 & \frac{K_{\mathrm{P}_2}}{S_c} \\ 0 & 0 \end{bmatrix} \boldsymbol{u}, \tag{9}$$

where

$$m_1(\boldsymbol{x}) = -\frac{Q_a}{S_c} - \frac{Q_{13}}{S_c} - \frac{Q_1}{S_c}, \tag{10}$$

$$m_2(\boldsymbol{x}) = -\frac{Q_b}{S_c} - \frac{Q_{23}}{S_c} - \frac{Q_2}{S_c}, \tag{11}$$

$$m_3(\boldsymbol{x}) = \frac{Q_a}{S_c} + \frac{Q_b}{S_c} + \frac{Q_{13}}{S_c} + \frac{Q_{23}}{S_c} - \frac{Q_3}{S_c}. \tag{12}$$

Note that the flows Q_a and Q_b are given by (1); Q_{13} and Q_{23} are given by (2); Q_1, Q_2, and Q_3 are given by (3).

B. OUTPUT EQUATIONS

The output vector contains all the system variables illustrated in Fig. 1 that can be measured via level and flow sensors. In particular, the output equation can be described as follows:

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 0 \\ 0 \\ Q_a \\ Q_b \\ Q_{13} \\ Q_{23} \\ Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ K_{\mathrm{P}_1} & 0 \\ 0 & K_{\mathrm{P}_2} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{u}. \tag{13}$$

Note that in (9) and (13) the state of the valves K_{P_1} and K_{P_2} are included in the system model (matrices B and D) and not directly coupled to the input flows u_1 and u_2, as in the other flows described in (1)–(3). This means that, regardless of the actuator type used in the system (e.g., servo-valves, servo-pumps, or any other flow control device), the input flow rates (i.e., u_1 and u_2) will always depend on the states of the valves K_{P_1} and K_{P_2}. This configuration is needed for keeping the benchmark characteristics during the simulation of actuator faults.

C. FAULT SCENARIOS

Faults might occur in the three-tank system due to mechanical or electrical damage, which normally appear due to wear of essential devices for the system operation, e.g., valves, pumps, level and flow sensors. Additionally, faults might occur due to physical damage in the system structure, which might cause leakage in tanks or clogging in pipes of the system.

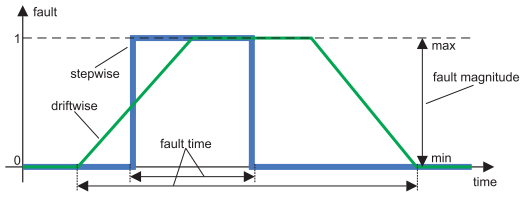


FIGURE 2. Types of faults.

In this study, two types of faults will be considered: abrupt and incipient. On the one hand, abrupt faults (called as stepwise) exhibit sudden changes in the system; on the other hand, incipient faults (called as driftwise) appear gradually in the system and their effects are more difficult to be noticed [24]. Fig. 2 illustrates both examples of stepwise and driftwise faults. Note that the driftwise fault signal is split into three stages and each stage consumes one-third of the fault time, i.e., the fault is introduced during the first stage with a constant growth rate, afterward the fault is active in the maximum magnitude during the second stage, and finally decreases with a constant rate during the third stage. Additionally, any other fault signal can be generated and used as custom test signal.

Each fault in the system is represented by $f_\gamma \in [0, 1]$, where γ is the identification number assigned to the fault and the value of f_γ corresponds to the fault magnitude; thus, $f_\gamma = 1$ indicates that the fault γ occurs with maximum magnitude and $f_\gamma = 0$ indicates that the fault γ does not occur in the system.

The faults that affect the three-tank system are described below and can be split into three major classes: actuator, plant, and sensor faults.

1) ACTUATOR FAULTS

The system actuators are the devices responsible for regulating the flow rate supply to the tanks 1 and 2 shown in Fig. 1. The actuator wear reduces its flow rate supply capacity, which can be represented by an attenuation factor $K \in [0, 1]$ that multiplies the actuator nominal output, as described in (14).

$$u_{\mathrm{real}} = K \times u_{\mathrm{nominal}}. \tag{14}$$

The system has two servo-valves that control the flows from the pumps P_1 and P_2, i.e., Q_{P_1} and Q_{P_2}. The actuator faults are denoted by f_1 and f_2, which represent the fault occurrence in the actuator of the tank 1 and 2, respectively. In a fault-free scenario, the values of f_1 and f_2 are expected to be zero. However, if an actuator fault occurs, then the value of f_1 or f_2 will be nonzero; if $f_i = 1$ $(i = 1, 2)$, then there was total loss of the actuator i, caused by blocking or clogging of its servo-valve.

The three-tank system benchmark permits the simulation of the faults f_1 and f_2 by changing the state of the valves K_{P_1} and K_{P_2}, respectively. For instance, if the user intends to simulate a blocking or clogging with intensity of 20% in actuator i, which characterizes a fault magnitude $f_i = 0.2$, then it is necessary to close 20% of the valve K_{P_i}.

2) PLANT FAULTS

The debris accumulation in the system might cause clogging in the pipes over time. Faults due to clogging usually have a slow evolution and, consequently, their negative effects take time to be detected. In addition, there is also the possibility of leakage from the system tanks, which might also lead to an extremely slow evolution. These faults are more difficult to be detected; if the system is in closed-loop configuration, the small variations caused by these faults will tend to be corrected by the controller. Thus, their effects will be camouflaged by the system control action and will pass to be noticed only when the actuator is saturated, i.e., when the maximum flow rate value that it can deliver to the system is achieved.

The three-tank system benchmark permits the simulation of plant faults by changing the defined state for the other valves of the system. As a result, there are seven possible plant faults, where each fault is associated to the operation mode of a valve, as described in Table 2; the acronyms TPi3 and CPi3 ($i = 1, 2$) correspond to the transmission and connection pipes from the tank i to the tank 3, respectively.

Note that the mathematical representation of plant fault (i.e., clogging or leakage) is related to the operation mode defined for the valve. For instance, if the operation mode of the valve K_1 is defined as normally open (i.e., $K_1 = 1$), then closing K_1 simulates a clogging in the output pipe of the tank 1; otherwise, if the operation mode of the valve K_1 is defined as normally closed (i.e., $K_1 = 0$), then opening K_1 simulates a leakage in tank 1. However, in both cases the fault only occurs when $f_7 > 0$. The same happens for the other plant faults described in Table 2.

3) SENSOR FAULTS

Faults in sensors normally occur due to mechanical damage that cause some displacement in ideal position of the sensors, thus providing incorrect measurements [25]. In addition, there is also the possibility of electrical damage that might occur inside the electronic circuit or at the power supply of the sensors. The effect of these damage can be represented by an attenuation factor that multiplies the actual value to be measured; if this attenuation factor is zero, then the sensor is totally lost. Additionally, faults that cause additive effects on sensor measurements might occur, i.e., offset faults; or yet, fixed value faults, where the sensor outputs are not updated with new values [25]. Here, only multiplicative faults in sensors will be considered, i.e., those whose measurements are results from the product between an attenuation factor and the real measured values.

As defined in Subsection II-B, the system output illustrated in Fig. 1 consists of all levels and flows that are measured by sensors. Thus, the system has three level sensors and nine flow sensors. The symbols for these faults are described in Table 2, where u_1 and u_2 correspond to the flow rate that the actuators are providing to the tanks 1 and 2, respectively.

TABLE 2. Faults in the three-tank system.

	Symbol	Description
Actuators	f_1	Blocking/Clogging in actuator 1, if $K_{P_1} = 1$
		Disturbance in tank 1, if $K_{P_1} = 0$
	f_2	Blocking/Clogging in actuator 2, if $K_{P_2} = 1$
		Disturbance in tank 2, if $K_{P_2} = 0$
Plant	f_3	Clogging in the TP13, if $K_a = 1$
		Leakage through the TP13, if $K_a = 0$
	f_4	Clogging in the TP23, if $K_b = 1$
		Leakage through the TP23, if $K_b = 0$
	f_5	Clogging in the CP13, if $K_{13} = 1$
		Leakage through the CP13, if $K_{13} = 0$
	f_6	Clogging in the CP23, if $K_{23} = 1$
		Leakage through the CP23, if $K_{23} = 0$
	f_7	Clogging in output pipe of the tank 1, if $K_1 = 1$
		Leakage in tank 1, if $K_1 = 0$
	f_8	Clogging in output pipe of the tank 2, if $K_2 = 1$
		Leakage in tank 2, if $K_2 = 0$
	f_9	Clogging in output pipe of the tank 3, if $K_3 = 1$
		Leakage in tank 3, if $K_3 = 0$
Sensors	f_{10}	Level sensor fault of the tank 1
	f_{11}	Level sensor fault of the tank 2
	f_{12}	Level sensor fault of the tank 3
	f_{13}	Flow sensor fault u_1 (input flow of the tank 1)
	f_{14}	Flow sensor fault u_2 (input flow of the tank 2)
	f_{15}	Flow sensor fault Q_a (flow through the TP13)
	f_{16}	Flow sensor fault Q_b (flow through the TP23)
	f_{17}	Flow sensor fault Q_{13} (flow through the CP13)
	f_{18}	Flow sensor fault Q_{23} (flow through the CP23)
	f_{19}	Flow sensor fault Q_1 (output flow of the tank 1)
	f_{20}	Flow sensor fault Q_2 (output flow of the tank 2)
	f_{21}	Flow sensor fault Q_3 (output flow of the tank 3)

D. FAULT MODEL

The mathematical model of the flows described in (1)–(3) can be extended to incorporate the fault effects, i.e., the model given by (9) and (13) can be rewritten as function of the fault vector $\boldsymbol{f} = [f_1\ f_2\ \ldots\ f_{21}]^T$. Thus, the fault model $\bar{\boldsymbol{g}}(\bar{\boldsymbol{x}}, \boldsymbol{u}, \boldsymbol{f})$ and $\bar{\boldsymbol{h}}(\bar{\boldsymbol{x}}, \boldsymbol{u}, \boldsymbol{f})$ can be obtained as follows:

Transmission flows: their dynamics depends on the operation mode of the transmission valves K_a and K_b. If $K_v (v = a, b)$ is normally open, then $K_v = 1$; otherwise, if K_v is normally closed, then $K_v = 0$.

$$\bar{Q}_a = \begin{cases} (1 - f_3)\beta \text{sgn}(\Delta h_a)\sqrt{|\Delta h_a|}, & \text{if } K_a = 1, \\ f_3 \beta \text{sgn}(\Delta h_a)\sqrt{|\Delta h_a|}, & \text{if } K_a = 0. \end{cases} \quad (15)$$

$$\bar{Q}_b = \begin{cases} (1 - f_4)\beta \text{sgn}(\Delta h_b)\sqrt{|\Delta h_b|}, & \text{if } K_b = 1, \\ f_4 \beta \text{sgn}(\Delta h_b)\sqrt{|\Delta h_b|}, & \text{if } K_b = 0. \end{cases} \quad (16)$$

Connection flows: their dynamics depends on the operation mode of the connection valves K_{13} and K_{23}. If $K_{i3} (i = 1, 2)$ is normally open, then $K_{i3} = 1$; otherwise,

$K_{i3} = 0$.

$$\bar{Q}_{13} = \begin{cases} (1-f_5)\beta \text{sgn}(h_1 - h_3)\sqrt{|h_1 - h_3|}, & \text{if } K_{13} = 1, \\ f_5\beta \text{sgn}(h_1 - h_3)\sqrt{|h_1 - h_3|}, & \text{if } K_{13} = 0. \end{cases} \tag{17}$$

$$\bar{Q}_{23} = \begin{cases} (1-f_6)\beta \text{sgn}(h_2 - h_3)\sqrt{|h_2 - h_3|}, & \text{if } K_{23} = 1, \\ f_6\beta \text{sgn}(h_2 - h_3)\sqrt{|h_2 - h_3|}, & \text{if } K_{23} = 0. \end{cases} \tag{18}$$

Output flows: their dynamics depends on the operation mode of the output valves K_1, K_2, and K_3. If $K_j (j = 1, 2, 3)$ is normally open, then $K_j = 1$; otherwise $K_j = 0$.

$$\bar{Q}_1 = \begin{cases} (1-f_7)\beta\sqrt{h_1}, & \text{if } K_1 = 1, \\ f_7\beta\sqrt{h_1}, & \text{if } K_1 = 0. \end{cases} \tag{19}$$

$$\bar{Q}_2 = \begin{cases} (1-f_8)\beta\sqrt{h_2}, & \text{if } K_2 = 1, \\ f_8\beta\sqrt{h_2}, & \text{if } K_2 = 0. \end{cases} \tag{20}$$

$$\bar{Q}_3 = \begin{cases} (1-f_9)\beta\sqrt{h_3}, & \text{if } K_3 = 1, \\ f_9\beta\sqrt{h_3}, & \text{if } K_3 = 0. \end{cases} \tag{21}$$

Finally, the fault model for the three-tank system illustrated in Fig. 1 can be described as follows:

$$\dot{\bar{x}} = \bar{g}(\bar{x}, \boldsymbol{u}, \boldsymbol{f}) = \begin{bmatrix} \bar{m}_1(\boldsymbol{x}) \\ \bar{m}_2(\boldsymbol{x}) \\ \bar{m}_3(\boldsymbol{x}) \end{bmatrix} + \begin{bmatrix} \frac{1-f_1}{S_c} & 0 \\ 0 & \frac{1-f_2}{S_c} \\ 0 & 0 \end{bmatrix} \boldsymbol{u}, \tag{22}$$

$$\bar{y} = \bar{h}(\bar{x}, \boldsymbol{u}, \boldsymbol{f}) = \begin{bmatrix} (1-f_{10})x_1 \\ (1-f_{11})x_2 \\ (1-f_{12})x_3 \\ 0 \\ 0 \\ (1-f_{15})\bar{Q}_a \\ (1-f_{16})\bar{Q}_b \\ (1-f_{17})\bar{Q}_{13} \\ (1-f_{18})\bar{Q}_{23} \\ (1-f_{19})\bar{Q}_1 \\ (1-f_{20})\bar{Q}_2 \\ (1-f_{21})\bar{Q}_3 \end{bmatrix} + \bar{\boldsymbol{D}}\boldsymbol{u}, \tag{23}$$

where

$$\bar{m}_1(\boldsymbol{x}) = -\frac{\bar{Q}_a}{S_c} - \frac{\bar{Q}_{13}}{S_c} - \frac{\bar{Q}_1}{S_c}, \tag{24}$$

$$\bar{m}_2(\boldsymbol{x}) = -\frac{\bar{Q}_b}{S_c} - \frac{\bar{Q}_{23}}{S_c} - \frac{\bar{Q}_2}{S_c}, \tag{25}$$

$$\bar{m}_3(\boldsymbol{x}) = \frac{\bar{Q}_a}{S_c} + \frac{\bar{Q}_b}{S_c} + \frac{\bar{Q}_{13}}{S_c} + \frac{\bar{Q}_{23}}{.} S_c - \frac{\bar{Q}_3}{S_c}, \tag{26}$$

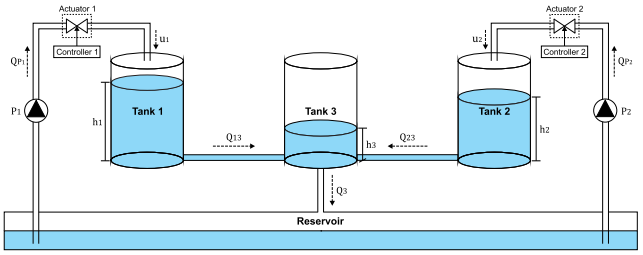


FIGURE 3. Default case study. In this scenario the faults f_3, f_4, f_{15}, f_{16}, f_{19}, and f_{20} do not exist and the measured signals are $\tilde{h}_1$, $\tilde{h}_2$, $\tilde{h}_3$, $\tilde{u}_1$, $\tilde{u}_2$, $\tilde{Q}_{13}$, $\tilde{Q}_{23}$, and $\tilde{Q}_3$.

$$\bar{\boldsymbol{D}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ (1-f_{13})(1-f_1) & 0 \\ 0 & (1-f_{14})(1-f_2) \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{27}$$

Note that for the fault model the flows $\bar{Q}_a$ and $\bar{Q}_b$ are given by (15)–(16); $\bar{Q}_{13}$ and $\bar{Q}_{23}$ are given by (17)–(18); $\bar{Q}_1$, $\bar{Q}_2$, and $\bar{Q}_3$ are given by (19)–(21).

III. THREE-TANK SYSTEM SIMULATOR

Sim3Tanks was developed in the MATLAB/Simulink environment and can be used via three different interfaces: graphical user interface (GUI), Simulink block diagram, and MATLAB script. It uses the `ode45` solver to solve numerically the ordinary differential equations (ODE) of the fault model described in (22). There are other solvers available in MATLAB, e.g., `ode23` and `ode113`, but `ode45` is more suitable for ODE with smooth solutions, since it is based on Dormand-Prince method [26] and evaluates six times the ODE, in order to calculate accurate solutions of fourth and fifth order.

The simulator allows the user to define an operation mode for the valves of the system (open or closed), thus allowing the definition of different case studies; for instance, closing the valves K_a, K_b, K_1, K_2 and leaving the other valves open, the case study illustrated in Fig. 3 is defined, where it is considered that the tanks 1 and 2 do not have output pipes and are connected to the tank 3 only by means of the connection pipes; this scenario is used by default to demonstrate Sim3Tanks. In addition, it is possible to simulate all 21 faults described in Table 2, where each fault can exhibit one of the behaviors illustrated in Fig. 2. Note that throughout this paper, the notation $\tilde{\bullet}$ denotes measured signals as, for instance, level $\tilde{h}$, control signal (flow rate) $\tilde{u}$, and flow $\tilde{Q}$.

Fig. 4 presents the software architecture of Sim3Tanks, which can be split into four blocks: interface, pre-processing,

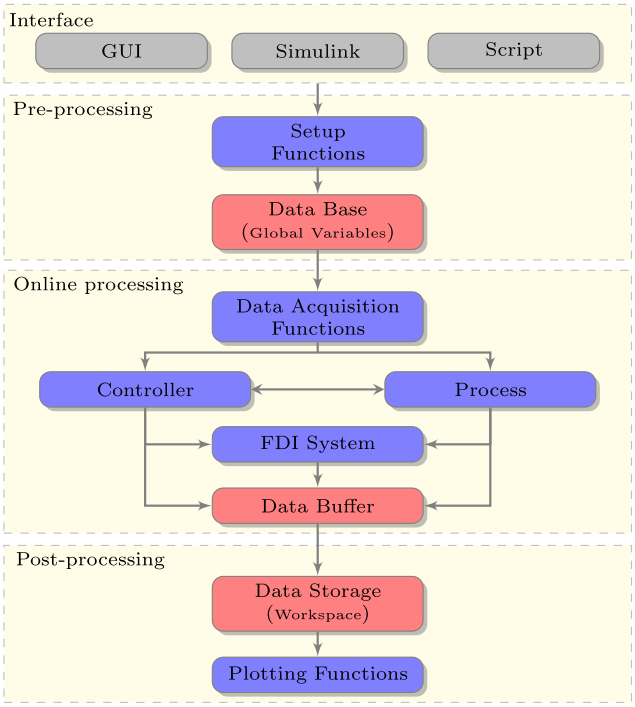


FIGURE 4. Software architecture of Sim3Tanks.

online processing, and post-processing. Firstly, the user should choose one of the three interfaces (i.e., GUI, Simulink, or Script) to interact with the simulator and then inform the simulation parameters and preferences. In the pre-processing steps, Sim3Tanks executes the setup functions to set the simulation time vector, system parameters (e.g., height and diameter of the tanks and pipes), operation mode of the valves (open or closed), maximum flow rate from the pumps (i.e., Q_{P_1} and Q_{P_2}), in addition to generating the fault signals; all generated data by the setup functions are stored in a data base via global variables. During the online processing steps, the required data by the controller and process are acquired from the data base by means of data acquisition functions; the controller receives feedback signals from the process and returns the control signals (i.e., u_1 and u_2). In order to perform the process monitoring, the FDI system receives the control and sensor signals; all output data generated by the process, controller, and FDI system are stored in the MATLAB buffer until the simulation is concluded. During post-processing steps, the simulation data are exported from buffer to the user's workspace; lastly, their results are graphically illustrated by means of plotting functions.

A. SIM3TANKS – GRAPHICAL USER INTERFACE

The graphical user interface of Sim3Tanks was created using MATLAB GUIDE (GUI Development Environment). Fig. 5 shows Sim3Tanks GUI, where the user needs to set the following simulation parameters and preferences: system parameters (e.g., R = 5 cm, H_{max} = 50 cm, r = 0.635 cm, h_0 = 30 cm, μ = 1, g = $981\frac{cm}{s^2}$), initial conditions (i.e., initial level in the tanks), maximum flow rate from the pumps P_1 and P_2 (e.g., $Q_{P_1} = Q_{P_2} = 80\frac{cm^3}{s}$), noise power (i.e., mean and standard deviation of white noise), simulation time in seconds, and operation mode of the valves (i.e., the user must define a case study such as the one presented in Fig. 3). When the simulation is initiated, the screen shown in Fig. 6 is used to graphically demonstrate the system behavior. Furthermore, the user can include either a digital control and an FDI system, which will be connected to Sim3Tanks as shown in Fig. 7.

1) INCLUDING A CONTROL SYSTEM

To add any digital controller to Sim3Tanks GUI just create a function with the mask:

```
function [cout1,cout2,cout3] = mycontroller(setpoint,Qp,y)
```

where the input arguments are:

1) setpoint: desired value,
2) Q_p: flow from the pumps ($Q_p = [Q_{P_1}, Q_{P_2}]^T$),
3) y: all measured outputs (see (13)),

and the output arguments are:

1) $cout_1$: must be the control signal ($cout_1 = [u_1, u_2]^T$),
2) $cout_2$: free variable (any desired parameter),
3) $cout_3$: free variable (any desired parameter).

The user should add the function `mycontroller` and all its sub-functions in the MATLAB current folder; back on the main screen (see Fig. 5), the user should mark the checkbox *yes* in the panel *Control system*; in the textbox, the user should also type the function name of the designed controller (i.e., `mycontroller`), and then enter the desired setpoint and sample time. After that, the simulation can be initiated; once it is finished, the controller output arguments, i.e., $cout_1$, $cout_2$, and $cout_3$ will be exported to the user's workspace.

2) INCLUDING AN FDI SYSTEM

To add any FDI system to Sim3Tanks GUI just create a function with the mask:

```
function [fout1,fout2,fout3,fout4,fout5]= myfdisystem(u,y)
```

where the input arguments are:

1) u: control signals ($u = [u_1, u_2]^T$),
2) y: all measured outputs (see (13)),

and the output arguments are:

1) $fout_1$: free variable (any desired parameter),
2) $fout_2$: free variable (any desired parameter),
3) $fout_3$: free variable (any desired parameter),
4) $fout_4$: free variable (any desired parameter),
5) $fout_5$: free variable (any desired parameter).

The user should add the function `myfdisystem` and all its sub-functions in the MATLAB current folder; back on the main screen (see Fig. 5), the user should mark the checkbox *yes* in the panel *FDI system*; in the textbox, the user should type the function name of the designed FDI system

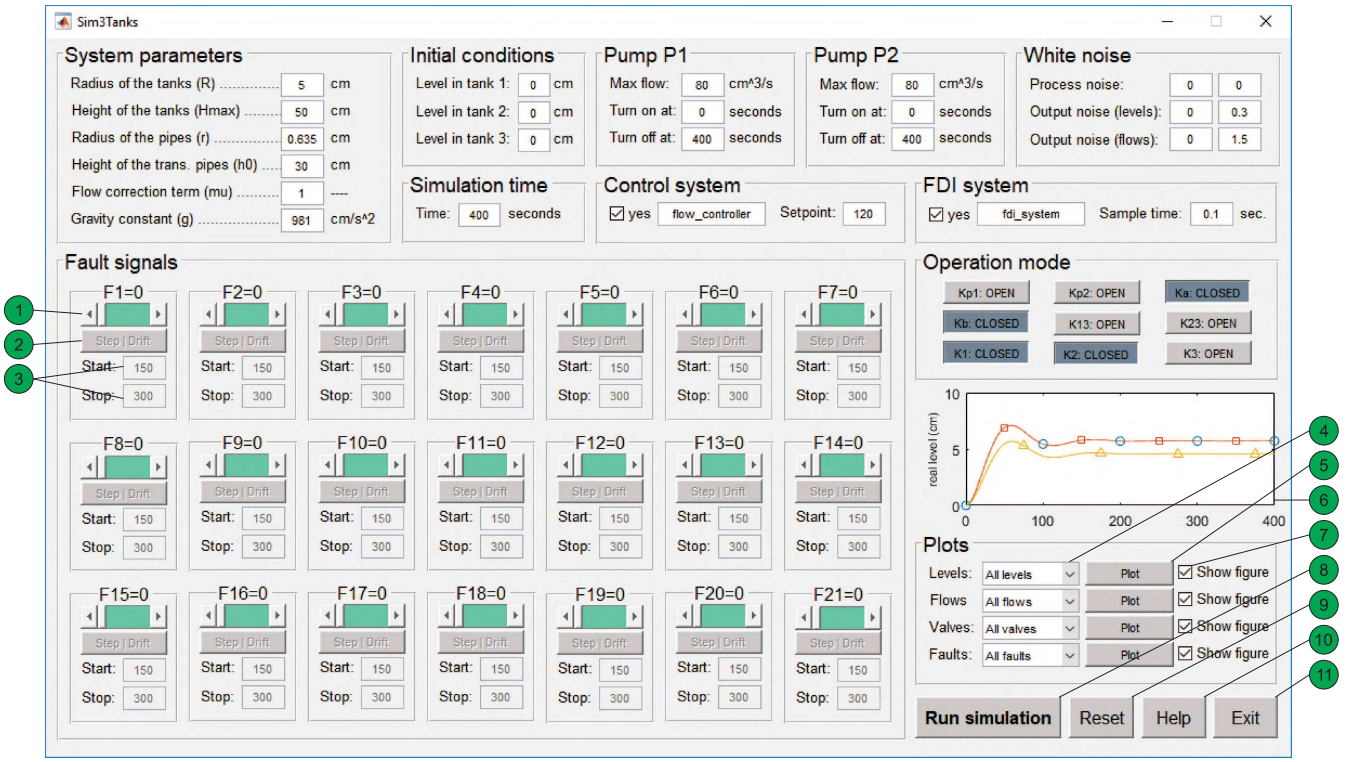


FIGURE 5. Main screen of Sim3Tanks GUI. 1: sets the fault magnitude. 2: sets the fault type. 3: sets the fault time. 4: selects the signal to be plotted. 5: plots the selected signal. 6: displays the plotted signal. 7: displays the plotted signal in an independent figure. 8: starts the simulation. 9: resets the simulation preferences and parameters to the default scenario. 10: displays the help screen. 11: closes Sim3Tanks GUI and all its screens and figures.

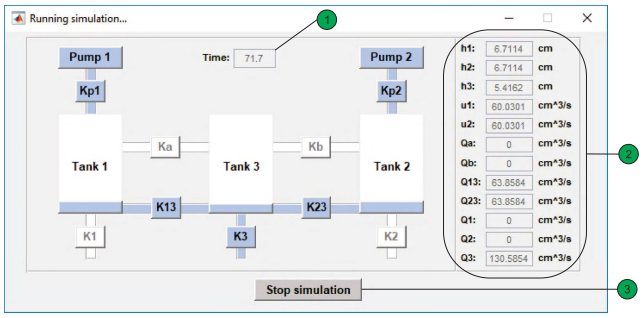


FIGURE 6. Real-time animation screen. 1: displays the current simulation time. 2: displays the current real values of the system variables. 3: stops the simulation and cleans all data stored in the MATLAB buffer.

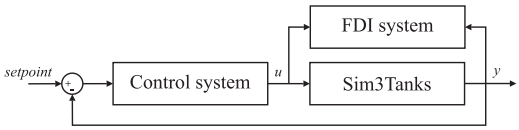


FIGURE 7. Connection diagram among Sim3Tanks, control and FDI system. Sim3Tanks is in closed-loop only when a control system is included; otherwise, Sim3Tanks is in open-loop and the control signal is $u = [Q_{P_1}\ Q_{P_2}]^T$.

(i.e., `myfdisystem`), and then enter the desired sample time. After that, the simulation can be initiated; once it is finished, the FDI system outputs, i.e., fout_1, fout_2, fout_3, fout_4, and fout_5 will be exported to the user's workspace.

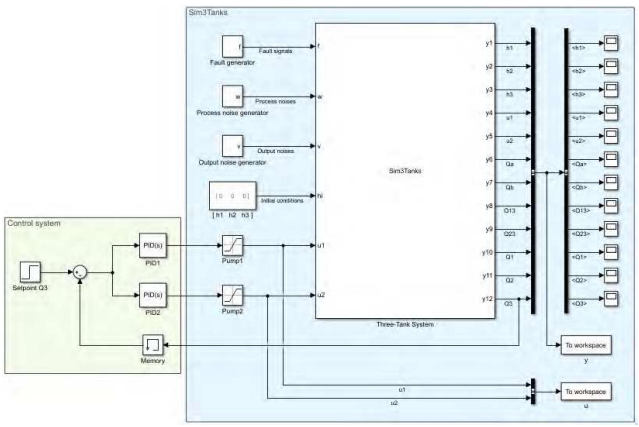


FIGURE 8. Sim3Tanks in Simulink.

B. SIM3TANKS – SIMULINK

Fig. 8 shows Sim3Tanks in closed-loop with an analog control strategy to flow Q_3, where a memory block is used to avoid errors due to algebraic loops; this procedure must be carried out in all system feedback signals. The outputs of Sim3Tanks block are the measured signals by the level and flow sensors, as described in (13); the inputs are the fault signals, process and output noises, initial conditions, and control signals. To set the simulation parameters and preferences, the user should just click twice on Sim3Tanks block and the setup menu will be shown. In contrast to the graphical interface, this

version of Sim3Tanks allows the user to simulate different fault signals, in addition to those illustrated in Fig. 2.

C. SIM3TANKS – SCRIPT

In this version of Sim3Tanks, the simulation parameters and preferences are defined by the following setup functions: `set_time`, `set_parameters`, `set_operation_mode`, `set_pumps`, and `set_faults`. In the online processing steps, the data acquisition is performed by `get_pumps` and `get_faults`, the process and output noises are generated by `get_process_noise` and `get_output_noise`; all these data are sent to function `three_tank_system_simulator`, where they are processed to obtain the system outputs (i.e., the real and measured level and flows of the three-tank system), which will be exported to the user's workspace after the simulation is concluded.

IV. CONTROL OBJECTIVES AND STRATEGIES

This section briefly describes the potential control objectives and shows examples of flow and level control strategies for the default case study illustrated in Fig. 3.

A. CONTROL OBJECTIVES

The potential objectives for process control can be summarized as follows:

- Maintain the process variables at desired values (setpoints);
- Recover quickly and smoothly from disturbances and setpoint changes;
- Keep the process operating conditions within equipment constraints.

B. CONTROL STRATEGY EXAMPLES

In order to describe the implementation and evaluation of a control strategy for the three-tank system illustrated in Fig 3, flow and level control strategies were implemented and tested in Sim3Tanks. Each strategy consists of two digital PID controllers (one per pump). In the flow control strategy, both PIDs work together to maintain the output flow Q_3 at a given setpoint. In the level control strategy, initially each PID works individually to maintain the levels h_1 and h_2 at their respective setpoints; later, the same PIDs work together to maintain the level h_3 at a given setpoint. The PIDs of each strategy were tuned using the closed-loop Ziegler-Nichols (ZN) tuning rules [27] and discretized by Tustin approximation [28] with sampling time $T = 0.1$s.

1) FLOW CONTROL STRATEGY

The purpose of this control strategy is to maintain the output flow Q_3 at a given setpoint. The system overview with the control strategy is illustrated in Fig. 9, where C_1 and C_2 are identical PID controllers, i.e., they have the same transfer function, and s_0 is the desired flow rate value (setpoint).

Closed-loop ZN tuning rules are applied to the control architecture illustrated in Fig. 9, the gains obtained after some slight adjustments on ZN gains are described in Table 3, where K_p is the proportional gain, T_i is the integral time, and T_d is the derivative time.

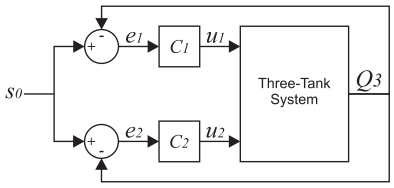


FIGURE 9. Flow control strategy.

TABLE 3. Flow controller parameters.

K_p	T_i	T_d
0.004	0.1	0.025

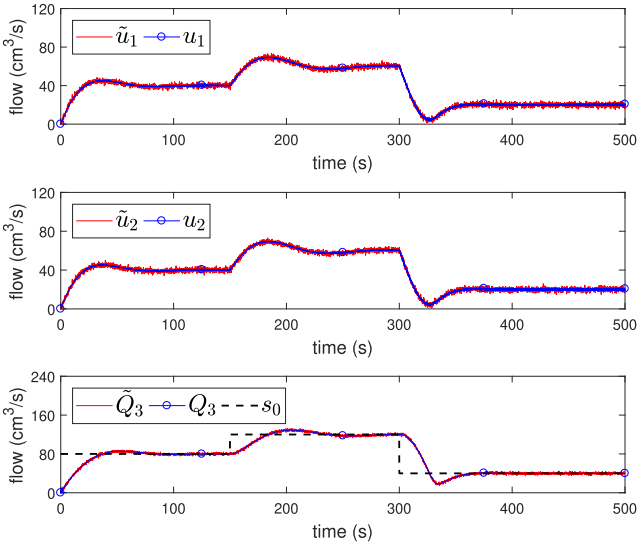


FIGURE 10. System response with the flow control strategy.

a: CLOSED-LOOP RESPONSE

Fig. 10 depicts the system response if this flow control strategy is applied to Sim3Tanks with the following setpoint changes: $s_0 = 80\frac{\text{cm}^3}{\text{s}}$ at $t = 0$s, $s_0 = 120\frac{\text{cm}^3}{\text{s}}$ at $t = 150$s, and $s_0 = 40\frac{\text{cm}^3}{\text{s}}$ at $t = 300$s. All measured signals ($\tilde{u}_1$, $\tilde{u}_2$, and $\tilde{Q}_3$) are corrupted by process and output noises, but it is clearly noticeable that the controllers act synchronously to maintain the flow Q_3 at the desired setpoint.

b: FAULTY-CASE RESPONSE

In order to demonstrate the closed-loop system response in faulty cases, two scenarios with different faults were simulated; these scenarios are described in Table 4 and their fault signals over time are illustrated in Fig. 11.

- *Scenario #1*: in this scenario, the control system needs to maintain the flow Q_3 at $80\frac{\text{cm}^3}{\text{s}}$; the fault signal shown in Fig. 11a is applied to Sim3Tanks to simulate a clogging in actuator 1 starting at $t = 100$s. The system response is

TABLE 4. Fault scenarios for the flow control strategy.

Scenario	Fault	Type	Time
#1	$f_1 = 1$	driftwise	$[100\text{s}, 400\text{s}]$
#2	$f_7 = 0.8$	stepwise	$[200\text{s}, 500\text{s}]$

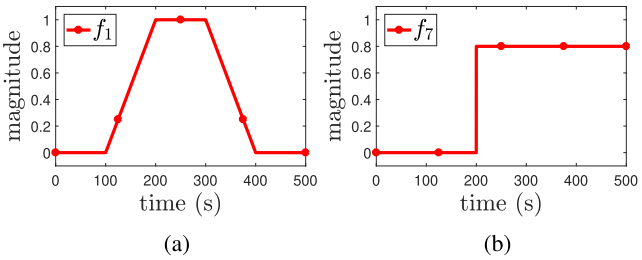


FIGURE 11. Fault signals over time for the flow control. (a) Scenario #1. (b) Scenario #2.

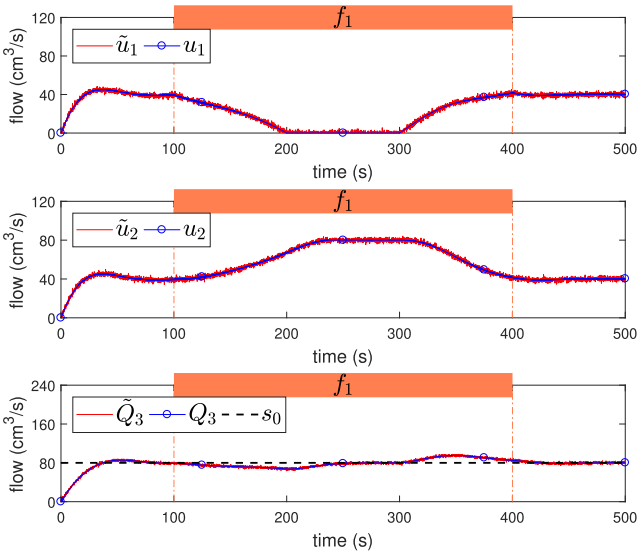


FIGURE 12. Flow control response for the fault scenario #1.

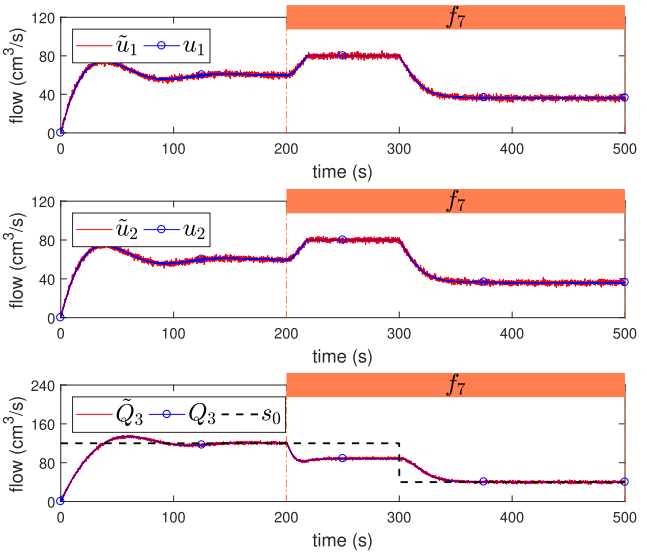


FIGURE 13. Flow control response for the fault scenario #2.

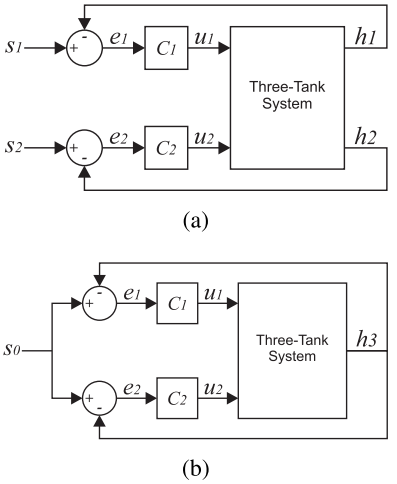


FIGURE 14. Level control strategies. (a) Strategy for levels h_1 and h_2. (b) Strategy for level h_3.

shown in Fig. 12, where it is noticed that the actuator 2 is able to compensate the loss of flow (caused by f_1) and maintain the flow Q_3 at the desired setpoint, although the regulation ability is harmed during the fault signal transients.

• *Scenario #2*: in this scenario, the system is subject to a setpoint change from $120\frac{\text{cm}^3}{\text{s}}$ to $40\frac{\text{cm}^3}{\text{s}}$ (at $t = 300\text{s}$) during a leakage fault in tank 1, which starts at $t = 200\text{s}$; Fig. 11b illustrates the fault signal over time. The system response is shown in Fig. 13, where it is possible to observe that just after f_7 starts, the flow Q_3 cannot achieve the desired value (i.e., $120\frac{\text{cm}^3}{\text{s}}$) due to the saturation of the pumps P_1 and P_2; however, when the reference is reduced to $40\frac{\text{cm}^3}{\text{s}}$, the saturation no longer occurs and the flow Q_3 achieves the desired value, even with the leakage presence.

2) LEVEL CONTROL STRATEGY

The level control is split into two strategies: a strategy for levels h_1 and h_2, and another strategy for level h_3, as illustrated in Fig. 14. The first strategy is illustrated in Fig. 14a, where each PID works individually to maintain the levels h_1 and h_2 at their respective setpoints (i.e., s_1 and s_2); Fig. 14b illustrates the second strategy, where the PIDs work together with the goal to maintain the level h_3 at its setpoint (s_0). In both strategies the controllers C_1 and C_2 have the same transfer function (due to the system symmetry).

Closed-loop ZN tuning rules are used to compute the gains of the controllers C_1 and C_2, as illustrated in Fig. 14. The gains described in Table 5 are obtained after some slight adjustments on ZN gains.

TABLE 5. Level controller parameters.

K_p	T_i	T_d
1	0.1	0.004

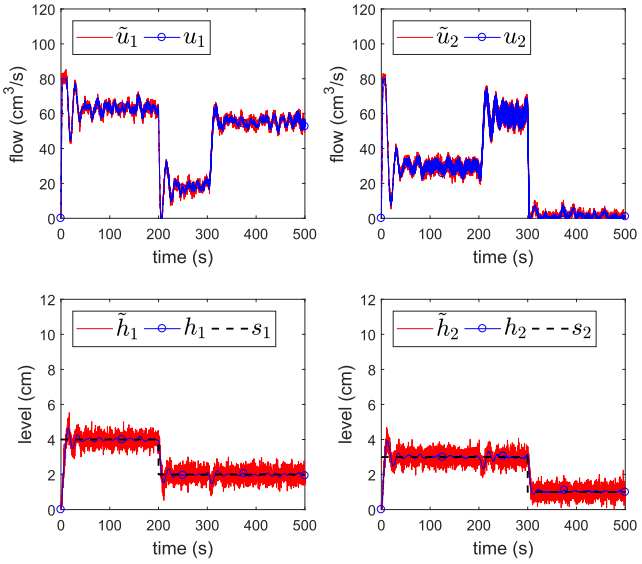


FIGURE 15. System response with the level control strategy.

TABLE 6. Fault scenarios for the level control strategies.

Scenario	Fault	Type	Time	Level
#1	$f_{11} = 1$	stepwise	$[200\text{s}, 500\text{s}]$	h_1, h_2
#2	$f_6 = 0.8$	driftwise	$[100\text{s}, 400\text{s}]$	h_1, h_2
#3	$f_6 = 0.8$	driftwise	$[100\text{s}, 400\text{s}]$	h_3

a: CLOSED-LOOP RESPONSE

Fig. 15 depicts the system response if the level control strategy illustrated in Fig. 14a is applied to Sim3Tanks. For level h_1, the setpoint is $s_1 = 4$ cm at $t = 0$s and changes to $s_1 = 2$ cm at $t = 200$s. For level h_2, the setpoint is $s_2 = 3$ cm at $t = 0$s and changes to $s_2 = 1$ cm at $t = 300$s. Note that if a level reduction occurs in tank 1 (at $t = 200$s), instantly, the controller C_2 needs to compensate the flow reduction (caused by C_1) to seek the level maintenance in tank 2. The same effect happens when a level reduction occurs in tank 2 (at $t = 300$s).

b: FAULTY-CASE RESPONSE

Three different scenarios were simulated to demonstrate the system behavior in faulty cases. The levels that are controlled and the faults that occur in each scenario are described in Table 6; the fault signals over time are illustrated in Fig. 16.

- *Scenario #1*: in this scenario, the fault signal shown in Fig. 16a is applied to Sim3Tanks and the system response is shown in Fig. 17, where the level sensor h_2 stops working correctly at $t = 200$s. Note that the measured level $\tilde{h}_2$ goes to 0 cm after the fault; from this moment, the controller C_2 acts based on wrong measurements and thus increases the input flow rate u_2, thus aiming to recover the setpoint tracking. However, C_2 saturates the pump P_2 and raises the level h_2 to a value above the desired setpoint. Consequently, the controller C_1 is induced to reduce u_1 to maintain h_1 at the desired setpoint.

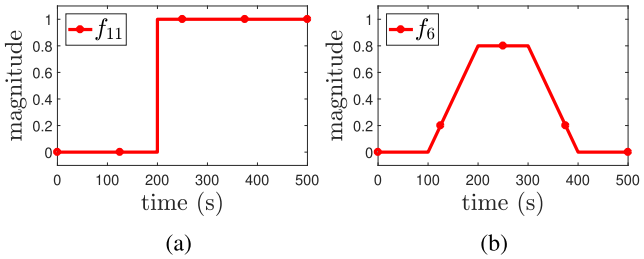


FIGURE 16. Fault signals over time for the level control. (a) Scenario #1. (b) Scenarios #2 and #3.

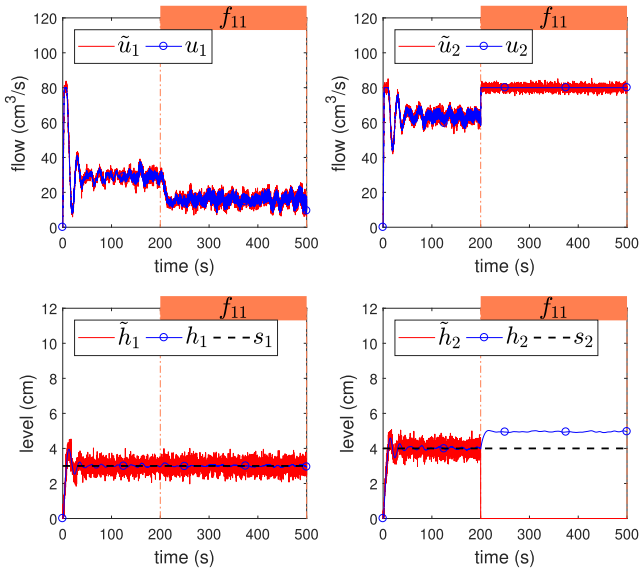


FIGURE 17. Level control response for the fault scenario #1.

- *Scenario #2*: in this scenario, the fault signal shown in Fig. 16b is applied to Sim3Tanks to simulate an incipient clogging in CP23. The system response is shown in Fig. 18; at the peak of f_6, the flow through CP23 (i.e., Q_{23}) is reduced by 80%; in order to compensate the liquid retention in tank 2 (caused by f_6) and maintain h_2 at the desired setpoint, the controller C_2 is forced to reduce the input flow rate in tank 2 (i.e., u_2). Since it is a coupled system, the controller C_1 increases the input flow rate in tank 1 (i.e., u_1), thus aiming to maintain h_1 at its setpoint. When s_1 changes to 8 cm (at $t = 200$s) during the fault, the controller C_1 quickly increases u_1 to achieve the new setpoint, but it is not possible due to the saturation of the pump P_1. The level h_2 is not affected by saturation of P_1 because, according to the strategy illustrated in Fig. 14a, the goal of C_2 is just to maintain the level h_2 at the desired setpoint, which already happened before P_1 saturates.

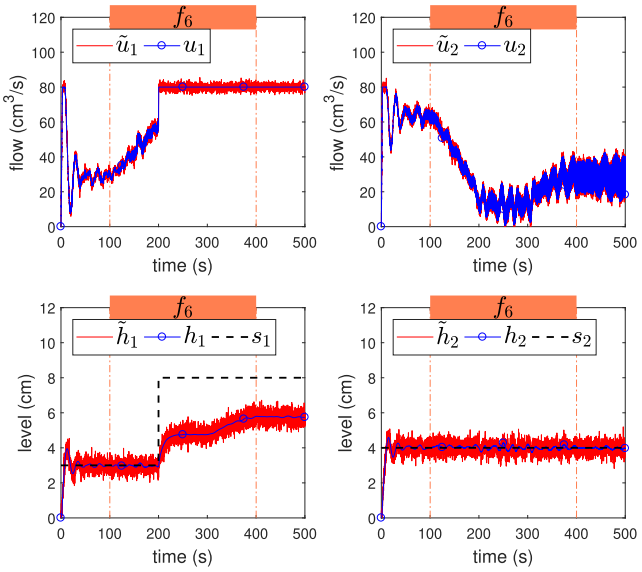
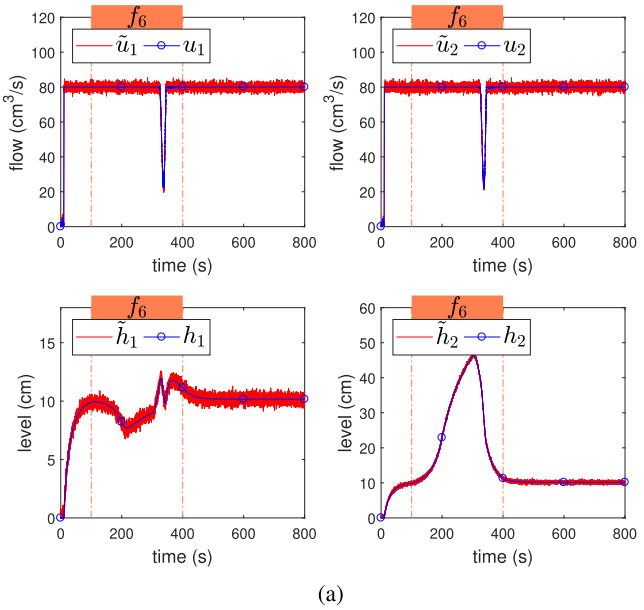
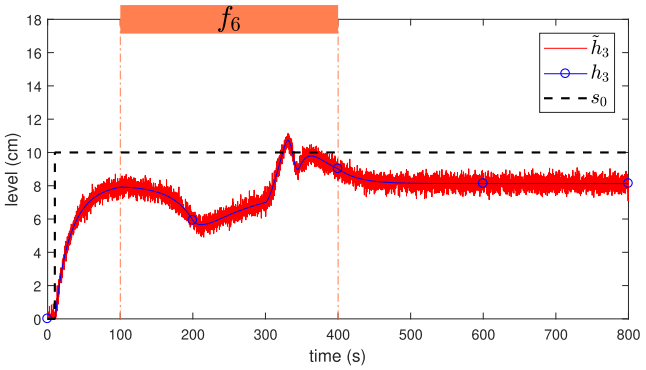


FIGURE 18. Level control response for the fault scenario #2.



(a)



(b)

FIGURE 19. Level control response for the fault scenario #3. (a) Control signals and level in tanks 1 and 2. (b) Level in tank 3.

Note that Fig. 17 and 18 show that the system illustrated in Fig. 3 is tightly coupled; thus, the level dynamics of the tank 1 affects the level in tank 2 and vice versa. That coupling explains the disturbances in tank 1 when a fault occurs in tank 2, even when a decentralized control strategy is used (see Fig. 14a). However, the coupling effect can be used as redundancy; thus allowing joint control of the system variables, improving the robustness and fault tolerance. In order to illustrate this idea, the second level control strategy depicted in Fig. 14b is implemented in Sim3Tanks. The controllers of this new strategy have the same parameters described in Table 5 and act together to maintain h_3 at a given desired setpoint.

- *Scenario #3*: this scenario is proposed to investigate the behavior of faulty level control system when the control loops are redundant, as illustrated in Fig. 14b; in this scenario, the same fault signal shown in Fig. 16b is applied to Sim3Tanks. Fig. 19 shows the system response; the idea of coupling is illustrated again in Fig. 19a, where the level in tank 1 decreases just after f_6 starts (due to the flow retention in CP23, which raises the level in tank 2) and increases quickly when the fault is near the end. Fig. 19b shows the performance of the control system during a clogging in CP23; when the level h_3 overcomes the setpoint (around $t = 325$s), both controllers quickly decrease the input flow rate in the tanks, thus aiming to maintain the level h_3 at the desired value. When the fault ends, the level h_3 achieves the steady state, but maintains its value below of the desired setpoint due to saturation of the pumps P_1 and P_2, which cannot exceed $80\frac{\text{cm}^3}{\text{s}}$. Note that the maximum value of Q_3 occurs when the pumps are saturated and the system is fault-free, in this conditions its value is given by:

$$Q_{3_{\max}} = Q_{P_1} + Q_{P_2},$$

but it also can be described by (3), so the maximum level in tank 3 can be determined by:

$$h_{3_{\max}} = \left(\frac{Q_{P_1} + Q_{P_2}}{\beta}\right)^2,$$

then for the system parameters described in Subsection III-A, the maximum level in tank 3 is:

$$h_{3_{\max}} = 8.131 \text{ cm}.$$

V. FAULT DETECTION AND ISOLATION OBJECTIVES AND STRATEGIES

This section briefly describes the potential FDI objectives and presents an example of FDI strategy to monitor the default case study (see Fig. 3) in closed-loop with the flow control strategy designed and illustrated in Fig. 9.

A. FDI OBJECTIVES

The potential FDI objectives for process monitoring can be summarized as follows:

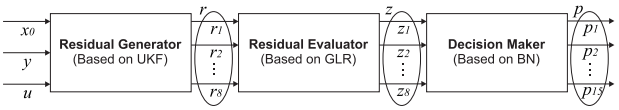


FIGURE 20. Architecture of the FDI strategy.

TABLE 7. Fault scenarios for the FDI strategy.

Scenario	Fault	Type	Time
#1	$f_1 = 0.8$	stepwise	$[100s, 300s]$
	$f_5 = 0.5$	driftwise	$[400s, 700s]$
	$f_{21} = 1$	stepwise	$[800s, 900s]$
#2	$f_2 = 1$	driftwise	$[100s, 400s]$
	$f_7 = 0.6$	stepwise	$[500s, 600s]$
	$f_{11} = 1$	driftwise	$[700s, 1000s]$

- Determine the occurrence and detection time of faults in the process [24]. To evaluate the detection efficiency, the following figures of merit are usually employed: fault detection rate (FDR), which corresponds to the probability of a fault detection, given that a fault really occurs, i.e., true positives; and false alarm rate (FAR), which corresponds to the probability of wrongly fault detection, i.e., false positives [29];
- Localize where the fault has occurred in the process [30]. The isolation ability of an FDI strategy can be evaluated by means of confusion matrices, which indicate the probability of each fault being isolated given its occurrence [31].

B. FDI STRATEGY EXAMPLE

The FDI strategy used in this example is based on model [32], [33] and consists of a residual generator based on unscented Kalman filter (UKF), a residual evaluator based on generalized likelihood ratio (GLR), and a decision maker based on Bayesian network (BN). As illustrated in Fig. 20, the residual generator receives the following input vectors: initial conditions of the UKF (x_0), output signals from the system (y), control signals (u), and computes the residual vector (r), i.e., the fault indicator signals [24]. The residual evaluator receives the vector r and determines the threshold crossing vector (z), which is used by a decision maker to compute the fault probability vector (p).

In order to demonstrate the operation of the FDI strategy, the fault scenarios described in Table 7 are applied to Sim3Tanks and their results are shown and briefly discussed.

1) UKF-BASED RESIDUAL GENERATION

The UKF uses the nonlinear model of the system illustrated in Fig. 3 (discretized by the Euler method [28] with sampling time $T = 0.1$s) for estimating the system outputs and thus computing the residual signal by the difference between the measured outputs by the sensors ($\tilde{y}$) and the estimated outputs

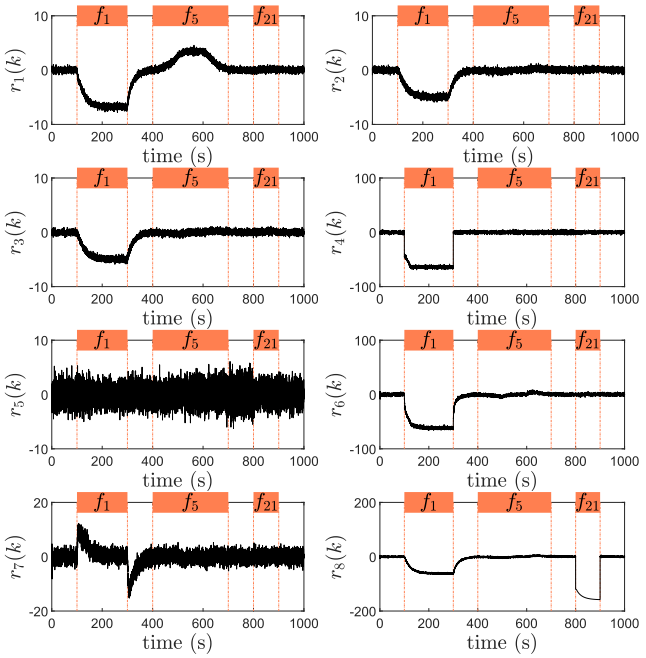


FIGURE 21. Residual signals for the fault scenario #1.

by the filter ($\hat{y}$). Fig. 21 illustrates the residual signals if the fault scenario #1 (described in Table 7) is applied to Sim3Tanks; as a result, it is observed that f_1 sensitizes most of the fault indicators (except r_5); f_5 significantly affects r_1 and causes small variations on r_6; and f_{21} affects only r_8. The fault indicators for the scenario #2 are shown in Fig. 22, where it is possible to observe that none of the simulated faults (f_2, f_7, and f_{11}) change the behavior of r_4; however, f_2 affects most of the residual signals and causes small variations on r_6; f_7 does not affect r_5; and f_{11} only sensitizes the fault indicator r_2.

2) GLR-BASED RESIDUAL EVALUATION

Due to the nature of some faults, i.e., their magnitude and temporal behavior, it is possible that their occurrence leaves imperceptible effects on the residual signals, so it becomes important to use some residual evaluation technique to detect these effects during the fault diagnosis process. In this case, the GLR algorithm [34] is used to detect changes in the mean of the residual signals r by computing the GLR decision function $\delta(k)$ for each residual sample at discrete time k:

$$\delta(k) = \frac{1}{2\sigma_0^2} \max_{k-M+1 \le j \le k} \frac{1}{k-j+1} \left[\sum_{i=j}^{k} (r(i) - \mu_0) \right]^2, \quad (28)$$

where μ_0 and σ_0 are the mean and standard deviation of r in a fault-free case, and M is the data window length. A fault is detected when one of the values of $\delta(k)$ crosses a certain threshold J_{th}. The i-th element of the vector z is 1 when $\delta_i(k) > J_{th}$ (fault detected), and is 0 otherwise. The value of each J_{th} was obtained by performing a fault-free simulation with 10^5 samples; for each residual signal, the maximum value of $\delta(k)$ was computed, multiplied by 2 and

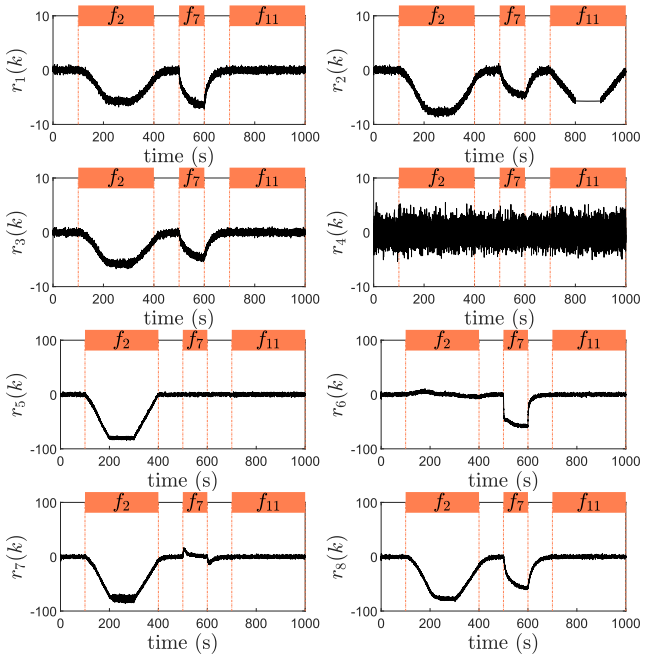


FIGURE 22. Residual signals for the fault scenario #2.

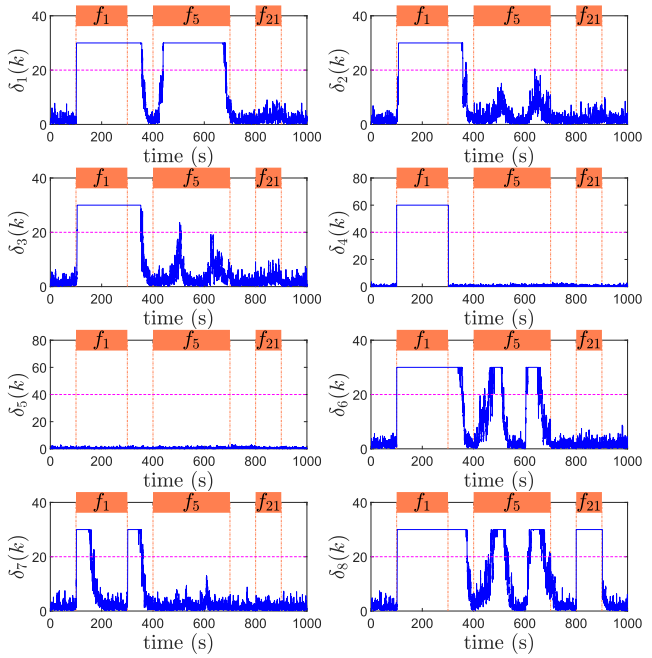


FIGURE 23. GLR functions for the fault scenario #1.

TABLE 8. Threshold values.

	$\delta_1(k)$	$\delta_2(k)$	$\delta_3(k)$	$\delta_4(k)$	$\delta_5(k)$	$\delta_6(k)$	$\delta_7(k)$	$\delta_8(k)$
J_{th}	20	20	20	40	40	20	20	20

rounded to the nearest multiple of 10; the obtained values are described in Table 8. Note that Sim3Tanks allows the user to implement more powerful techniques for residual generation and evaluation (e.g., adaptive threshold) to decide if a fault is detected, as well as any other state-of-the-art FDI model-driven or data-driven technique, which are not discussed here for the sake of brevity. The goal of this example is constrained to show the potential of Sim3Tanks to develop and evaluate control and FDI techniques as well as their interconnections.

The behavior of the GLR decision functions for the residual signals shown in Fig. 21 and 22 are illustrated, respectively, in Fig. 23 and 24, where the values of $\delta_i(k)$ were saturated at 50% above their threshold values. Note that the functions $\delta_i(k)$ emphasize changes on the residual signals, making visible details that were visually indistinguishable; for instance, the residue r_8 shown in Fig. 21 does not present visible changes when f_5 occurs, but after being processed by the GLR algorithm, changes are detected and emphasized to the user, as illustrated in the function $\delta_8(k)$ of Fig. 23.

3) BN-BASED DECISION MAKING

Different faults can sensitize different GLR decision functions; this feature makes it possible to classify different faults according to the fault detection system response. In this case, the fault isolation is performed by a decision-making algorithm based on BN [35]. The binary evidence vector z is used in (29) to compute the probability of each fault having

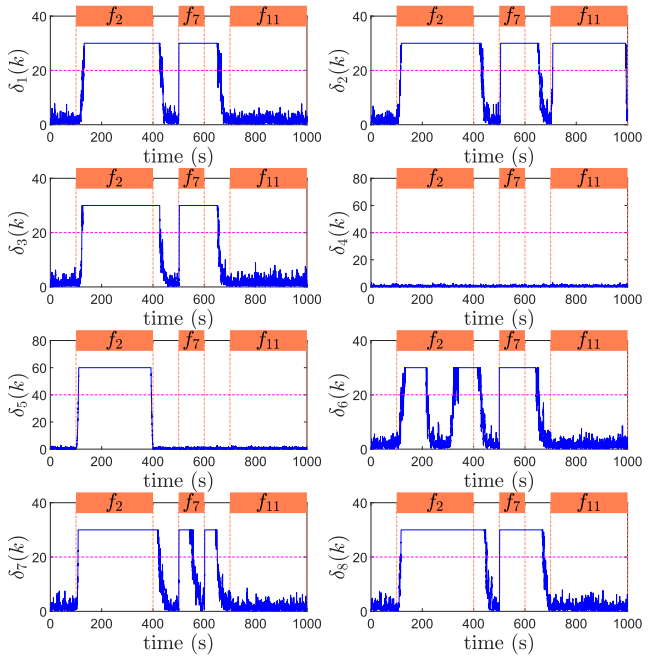


FIGURE 24. GLR functions for the fault scenario #2.

occurred in the system.

$$p(f_\gamma|z_1,\cdots,z_8)=\frac{p(z_1,\cdots,z_8|f_\gamma)\cdot p(f_\gamma)}{p(z_1,\cdots,z_8)}. \tag{29}$$

Note that the Bayesian network is a data-driven approach, i.e., the user must have a large amount of data input and output of the system in the most diverse scenarios to train the BN. The training data were obtained by creating a scenario with stepwise and driftwise faults with different magnitudes and

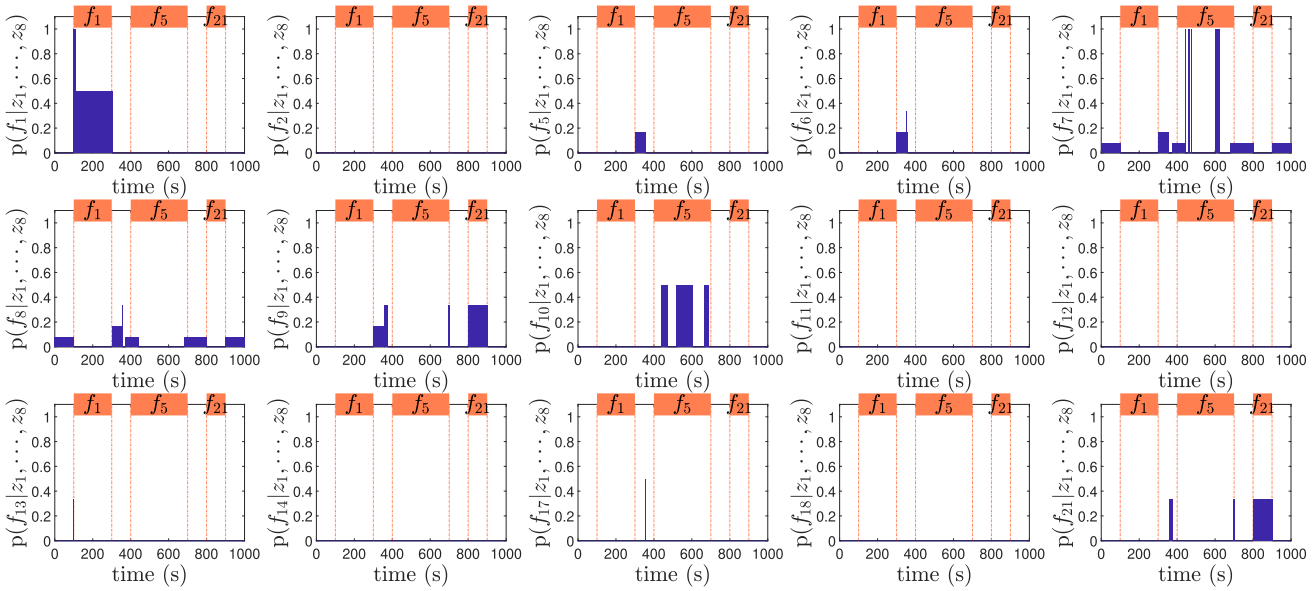


FIGURE 25. Fault probabilities computed by the Bayesian network for the fault scenario #1.

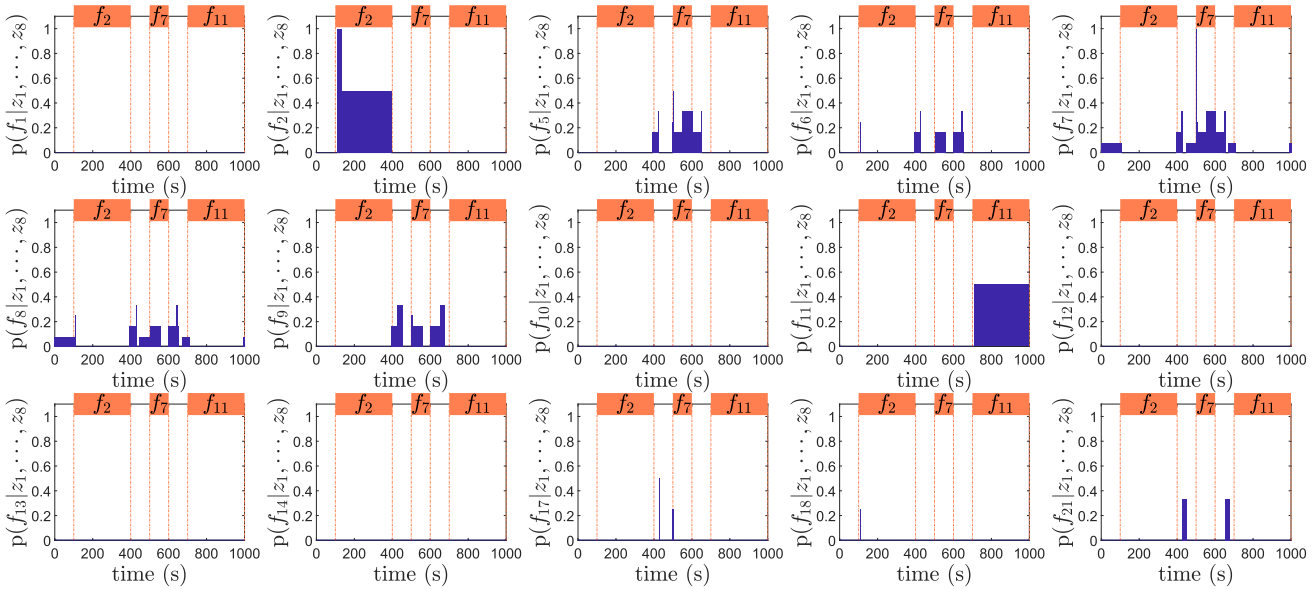


FIGURE 26. Fault probabilities computed by the Bayesian network for the fault scenario #2.

fault time; in addition to setpoint changes. The generated data by Sim3Tanks for the created scenario are exported to the MATLAB workspace, where they can be manipulated to be used in the training methodology.

Fig. 25 illustrates the fault probabilities for the vector z generated from the functions $\delta(k)$ shown in Fig. 23. For the sequence of faults described in the scenario #1, the FDI strategy correctly isolates only f_1, because at fault time of f_1, the highest probability calculated by BN is $p(f_1)$. At fault time of f_5, the highest probability is $p(f_{10})$ instead of being $p(f_5)$. Already for f_{21}, the BN computes the same probability for $p(f_9)$ and $p(f_{21})$, which means that the fault isolation algorithm does not present a decision about which fault has occurred in the system (f_9 or f_{21}). Fig. 26 shows the fault probabilities for the fault scenario #2, where the FDI strategy isolates f_2 and f_{11}, but does not isolate f_7.

C. EVALUATION OF THE FDI STRATEGY

In order to evaluate the FDI strategy in terms of detection efficiency and isolation ability for stepwise and driftwise faults, two sets of tests were performed for each type of fault (see Fig. 2): a set for stepwise and another one for driftwise faults. It should be noted that the FDI strategy was developed for the default case study illustrated

TABLE 9. Values of FDR and FAR.

Scenario	Fault	Driftwise		Stepwise	
		FDR	FAR	FDR	FAR
#1	f_1	1		0.97	
#2	f_2	1		0.97	
#3	f_5	1		0.93	
#4	f_6	1		0.93	
#5	f_7	0.99		0.98	
#6	f_8	0.99		0.98	
#7	f_9	1		0.99	
#8	f_{10}	1	0.05	0.95	0.03
#9	f_{11}	1		0.96	
#10	f_{12}	1		0.93	
#11	f_{13}	1		0.94	
#12	f_{14}	1		0.94	
#13	f_{17}	1		0.97	
#14	f_{18}	1		0.97	
#15	f_{21}	1		0.99	

in Fig. 3, where it is possible to occur 15 faults among the 21 described in Table 2. Thus, each set of test was composed of 15 scenarios, i.e., one per fault; and in each scenario, the simulation time was 500s with sampling time $T = 0.1$s, and each fault occurred between $[100\text{s}, 400\text{s}]$ with maximum magnitude.

1) FAULT DETECTION EFFICIENCY

As shown in Table 9, the fault detection algorithm presents high values of FDR, indicating that the occurrence of f_γ has a high probability of being detected. In addition, it presents a low value of FAR, indicating a low probability of alerting a false detection. It is also noticed that driftwise faults have lower values of FDR than stepwise faults, this happens because their evolution is more slower and thus their detection takes longer to be confirmed by the FDI strategy, i.e., the GLR decision functions (see (28)) must process more residual samples until at least one of them crosses its respective threshold (see Table 8).

2) FAULT ISOLATION ABILITY

Tables 10 and 11 describe the confusion matrix for stepwise and driftwise faults, where the first column represents the occurred faults (f_γ), the first row represents the isolated faults ($\hat{f}_\gamma$) by the FDI strategy, and the data represent the probability of each fault being isolated given its occurrence. These tables show that the FDI strategy presents high probability to correctly isolate the following driftwise and stepwise faults: f_1, f_2, f_{10}, f_{11}, and f_{12}. For the other faults, the FDI strategy presents high uncertainties about which fault has occurred in the system.

VI. FAULT TOLERANCE OBJECTIVES AND STRATEGIES

This section discusses the fault tolerance requirements and provides an example of a well-known FTC technique called fault hiding. The closed-loop system performance with FTC is analyzed and compared to the flow control strategy presented in Subsection IV-B1 by means of the Sim3Tanks features.

A. FTC OBJECTIVES

Generally, the main control objectives and constraints related to fault tolerance are described as follows:

- The closed-loop system must remain stable after a fault occurrence;
- The FTC system should recover the nominal tracking ability and performance after a fault occurrence;
- The FTC system must be implemented online and it must work autonomously, i.e., the FTC must be able to automatically accommodate the faults or reconfigure the control loop fast enough to respect the real-time constraints;
- The FTC system should be robust to disturbances, parametric uncertainties, and flaws or delays on fault detection and diagnosis.

B. FTC STRATEGY EXAMPLE

As an example, this work shows the obtained results by using the simplest form of an FTC technique known as fault hiding [30], [36]. The fault hiding consists of adding some reconfiguration blocks to the control loop, thus aiming to compensate the faults without controller redesign. A reconfiguration block named virtual sensor (VS) is employed to compensate sensor faults by providing correct estimates of the measured variables to the nominal controller from the faulty sensor signals. Similarly, a reconfiguration block named virtual actuator (VA) is employed when an actuator fault occurs. The VA receives (faulty) control signals (u_f) from the nominal controller and computes a compensated control signal (u_r) capable of producing the same dynamical effect through faulty actuators.

For the sake of simplicity, this work implements and evaluates the simplest kind of VS and VA, which are called static VS and static VA. More sophisticated structures and design methodology for reconfiguration blocks can be found in literature [30], [36] to solve a wide range of FTC problems. However, in this work, we are interested in demonstrating how the Sim3Tanks features allow us to analyze and evaluate an FTC technique and its synergy with an FDI strategy and the rest of a control system.

1) STATIC RECONFIGURATION BLOCKS DESIGN

A static VS is the reconfiguration block that solves directly the reconfiguration problem by using the pseudo-inverse method based on the linear model. In order to obtain the static VS model for the three-tank system, the dynamics and output equations are linearized around a desired operation

TABLE 10. Confusion matrix for stepwise faults.

	$\hat{f}_1$	$\hat{f}_2$	$\hat{f}_5$	$\hat{f}_6$	$\hat{f}_7$	$\hat{f}_8$	$\hat{f}_9$	$\hat{f}_{10}$	$\hat{f}_{11}$	$\hat{f}_{12}$	$\hat{f}_{13}$	$\hat{f}_{14}$	$\hat{f}_{17}$	$\hat{f}_{18}$	$\hat{f}_{21}$
f_1	0.51	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_2	0	0.51	0	0	0	0	0	0	0	0	0	0	0	0	0
f_5	0	0	0.30	0.03	0.30	0.03	0.03	0	0	0	0	0	0	0	0
f_6	0	0	0.03	0.30	0.03	0.30	0.03	0	0	0	0	0	0	0	0
f_7	0	0	0.30	0.03	0.30	0.03	0.03	0	0	0	0	0	0	0	0
f_8	0	0	0.03	0.30	0.03	0.30	0.03	0	0	0	0	0	0	0	0
f_9	0	0	0.17	0.17	0.17	0.17	0.17	0	0	0	0	0	0	0	0
f_{10}	0	0	0	0	0	0	0	0.50	0	0	0	0	0	0	0
f_{11}	0	0	0	0	0	0	0	0	0.50	0	0	0	0	0	0
f_{12}	0	0	0	0	0	0	0	0	0	0.50	0	0	0	0	0
f_{13}	0.33	0	0	0	0	0	0	0	0	0	0.33	0	0	0	0
f_{14}	0	0.33	0	0	0	0	0	0	0	0	0	0.33	0	0	0
f_{17}	0	0	0.24	0	0.24	0	0	0	0	0	0	0	0.26	0	0
f_{18}	0	0	0	0.25	0	0.25	0	0	0	0	0	0	0	0.25	0
f_{21}	0	0	0	0	0	0	0.33	0	0	0	0	0	0	0	0.33

TABLE 11. Confusion matrix for driftwise faults.

	$\hat{f}_1$	$\hat{f}_2$	$\hat{f}_5$	$\hat{f}_6$	$\hat{f}_7$	$\hat{f}_8$	$\hat{f}_9$	$\hat{f}_{10}$	$\hat{f}_{11}$	$\hat{f}_{12}$	$\hat{f}_{13}$	$\hat{f}_{14}$	$\hat{f}_{17}$	$\hat{f}_{18}$	$\hat{f}_{21}$
f_1	0.48	0	0.01	0	0.01	0.01	0	0	0	0	0	0	0	0	0
f_2	0	0.48	0	0.01	0.01	0.01	0	0	0	0	0	0	0	0	0
f_5	0	0	0.19	0.08	0.21	0.09	0.09	0	0	0	0	0	0.01	0	0
f_6	0	0	0.08	0.18	0.09	0.21	0.08	0	0	0	0	0	0	0.01	0
f_7	0	0	0.24	0.08	0.25	0.08	0.08	0	0	0	0	0	0	0	0
f_8	0	0	0.08	0.25	0.08	0.25	0.08	0	0	0	0	0	0	0	0
f_9	0	0	0.16	0.16	0.16	0.16	0.18	0	0	0	0	0	0	0	0.01
f_{10}	0	0	0	0	0	0	0	0.47	0	0	0	0	0	0	0
f_{11}	0	0	0	0	0	0	0	0	0.48	0	0	0	0	0	0
f_{12}	0	0	0	0	0.01	0.01	0	0	0	0.46	0	0	0	0	0
f_{13}	0.31	0	0	0	0	0	0	0	0	0	0.31	0	0	0	0
f_{14}	0	0.31	0	0	0	0	0	0	0	0	0	0.31	0	0	0
f_{17}	0	0	0.24	0	0.24	0	0	0	0	0	0	0	0.25	0	0
f_{18}	0	0	0	0.24	0	0.24	0	0	0	0	0	0	0	0.24	0
f_{21}	0	0	0	0	0	0	0.33	0	0	0	0	0	0	0	0.33

point $\boldsymbol{x}^{\mathrm{o}} = [h_1^{\mathrm{o}}\ h_2^{\mathrm{o}}\ h_3^{\mathrm{o}}]$, thus obtaining the following nominal linearized model:

$$\begin{aligned} \Delta\dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \\ \Delta\boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u}, \end{aligned} \tag{30}$$

where $\boldsymbol{A} = \left.\frac{\partial \boldsymbol{g}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=x^{\mathrm{o}}}$, $\boldsymbol{B} = \left.\frac{\partial \boldsymbol{g}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}=x^{\mathrm{o}}}$, $\boldsymbol{C} = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=x^{\mathrm{o}}}$, $\boldsymbol{D} = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}=x^{\mathrm{o}}}$, and $\boldsymbol{g}(\boldsymbol{x},\boldsymbol{u})$ and $\boldsymbol{h}(\boldsymbol{x},\boldsymbol{u})$ are given by (9) and (13). Similarly, the following fault linearized model can be obtained:

$$\begin{aligned} \Delta\dot{\bar{\boldsymbol{x}}} &= \bar{\boldsymbol{A}}\bar{\boldsymbol{x}} + \bar{\boldsymbol{B}}\boldsymbol{u}, \\ \Delta\bar{\boldsymbol{y}} &= \bar{\boldsymbol{C}}\bar{\boldsymbol{x}} + \bar{\boldsymbol{D}}\boldsymbol{u}, \end{aligned} \tag{31}$$

where $\bar{\boldsymbol{A}} = \left.\frac{\partial \bar{\boldsymbol{g}}(\bar{\boldsymbol{x}},\boldsymbol{u},\boldsymbol{f})}{\partial \bar{\boldsymbol{x}}}\right|_{\bar{\boldsymbol{x}}=x^{\mathrm{o}}}$, $\bar{\boldsymbol{B}} = \left.\frac{\partial \bar{\boldsymbol{g}}(\bar{\boldsymbol{x}},\boldsymbol{u},\boldsymbol{f})}{\partial \boldsymbol{u}}\right|_{\bar{\boldsymbol{x}}=x^{\mathrm{o}}}$, $\bar{\boldsymbol{C}} = \left.\frac{\partial \bar{\boldsymbol{h}}(\bar{\boldsymbol{x}},\boldsymbol{u},\boldsymbol{f})}{\partial \bar{\boldsymbol{x}}}\right|_{\bar{\boldsymbol{x}}=x^{\mathrm{o}}}$, $\bar{\boldsymbol{D}} = \left.\frac{\partial \bar{\boldsymbol{h}}(\bar{\boldsymbol{x}},\boldsymbol{u},\boldsymbol{f})}{\partial \boldsymbol{u}}\right|_{\bar{\boldsymbol{x}}=x^{\mathrm{o}}}$, and $\bar{\boldsymbol{g}}(\bar{\boldsymbol{x}},\boldsymbol{u},\boldsymbol{f})$ and $\bar{\boldsymbol{h}}(\bar{\boldsymbol{x}},\boldsymbol{u},\boldsymbol{f})$ are given by (22) and (23).

The static VS is a matrix gain $\boldsymbol{S}_1$ such that $\boldsymbol{y} = \boldsymbol{S}_1 \cdot \bar{\boldsymbol{y}}$. For the sake of simplicity, it is designed a VS that is unable to compensate faulty sensors that provide the measurements $\tilde{u}_1$ and $\tilde{u}_2$, since these flow rates can be obtained directly from the controllers. This simplification avoids the computation of matrix $\bar{\boldsymbol{D}}$ for VS design purposes. Notice that $\boldsymbol{S}_1$ is computed from an estimate of $\boldsymbol{f}$, denoted $\hat{\boldsymbol{f}}$ as follows:

$$\hat{\boldsymbol{S}}_1 = \boldsymbol{C} \cdot \bar{\boldsymbol{C}}^{\dagger}, \tag{32}$$

where $\bar{\boldsymbol{C}}^{\dagger}$ is the pseudo-inverse of $\bar{\boldsymbol{C}}$.

Similarly, the static VA is a matrix gain $\boldsymbol{S}_2$ such that $\boldsymbol{u} = \boldsymbol{S}_2 \cdot \bar{\boldsymbol{u}}$. Thus, $\hat{\boldsymbol{S}}_2$ is computed as follows:

$$\hat{\boldsymbol{S}}_2 = \bar{\boldsymbol{B}}^{\dagger} \cdot \boldsymbol{B}. \tag{33}$$

C. EVALUATION OF THE FTC STRATEGY

Sim3Tanks is used to evaluate the VA and VS, whose design is described in the previous subsection. For this purpose,

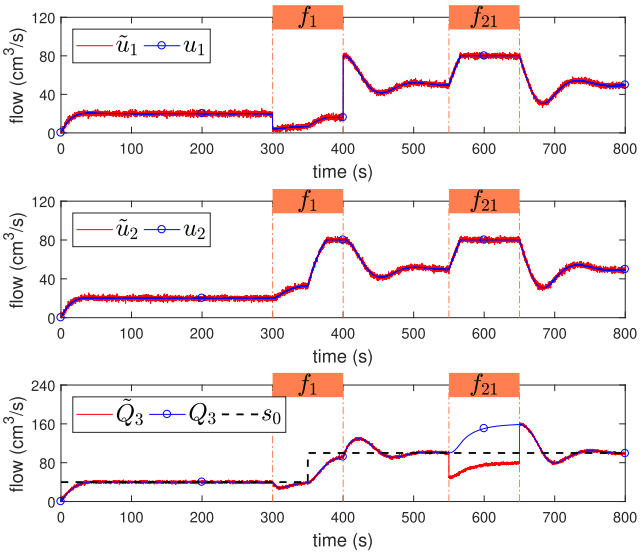


FIGURE 27. Flow control response without reconfiguration blocks.

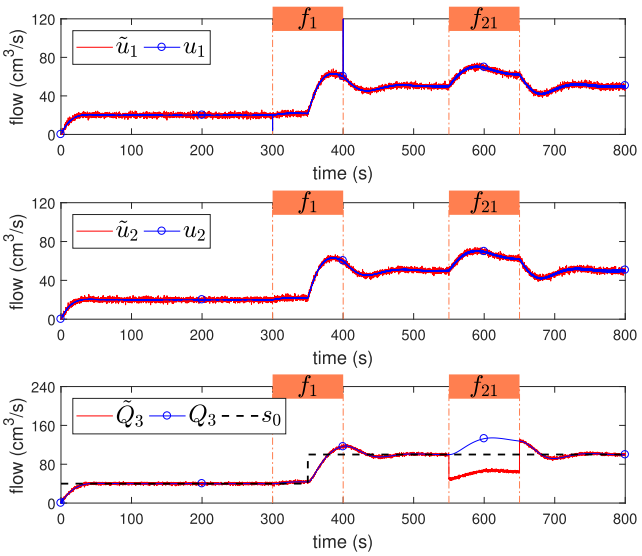


FIGURE 28. Flow control response with reconfiguration blocks fed by an ideal fault diagnosis.

simulations are performed with the flow control strategy described in Subsection IV-B1, where the flow Q_3 is controlled. The fault hiding strategy allows to maintain the same nominal controller, since the reconfiguration is performed through the reconfiguration blocks. Two faults are considered during the simulations: an actuator fault reduces by 80% the capacity of pump P_1 ($f_1 = 0.8$) between [300s, 400s]; and a sensor fault attenuates by 0.5 the flow measurements Q_3 (i.e., $f_{21} = 0.5$) between [550s, 650s], which is crucial for the flow control strategy depicted in Fig. 9. Furthermore, the control reference for Q_3 is changed from $40\frac{\text{cm}^3}{\text{s}}$ to $100\frac{\text{cm}^3}{\text{s}}$ (during an actuator fault).

Three simulations are performed: in the first scenario, the FTC is not implemented, thus the nominal control system is evaluated; the second scenario includes the static

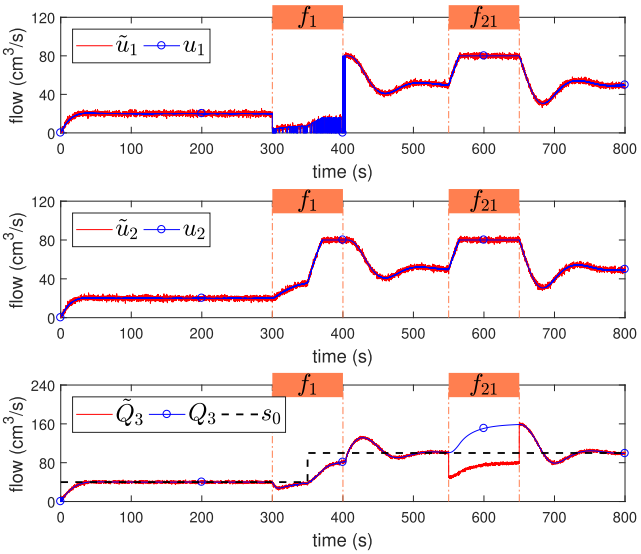


FIGURE 29. Flow control response with reconfiguration blocks integrated to the FDI system.

VA and VS in the control loop, but these reconfiguration blocks receive directly the fault signals, i.e., the fault detection and diagnosis system is assumed to be ideal (i.e., without delay or misdetection); finally, in the third scenario, the VA and VS are implemented and integrated to the FDI system presented in Subsection V-B.

Fig. 27 to 29 show the simulation results. By analyzing Fig. 27 and 28, it is noticed that the presence of VS and VA can improve the system response during the occurrence of faults. Fig. 28 shows that the system with VA is almost not sensitized when f_1 occurs, thus maintaining the system performance and reference tracking, even during a setpoint change. By contrast, the system response without VA (shown in Fig. 27) presents a perturbation in flow Q_3 and a slow reaction to the step reference change. Note also that in the system with VA the resulting control signal u_1 is immediately corrected after a short jump when the fault f_1 starts, but it remains incorrect in the system without VA. During the fault f_{21}, both simulations show that the output behavior is affected, but the fault effects are more smooth on the system with VS (see Fig. 28) than on the system without VS (see Fig. 27). The results of Sim3Tanks indicate that the fault f_{21} causes saturation of both actuators when there is no FTC strategy; this problem does not occur when the reconfiguration blocks are added to the system.

The obtained results for the third scenario are illustrated in Fig. 29, where it is shown that the integration between the fault hiding strategy and a non-ideal FDI system presents a result far below the ideal case shown in Fig. 28. Fig.30 shows the fault probabilities provided by the FDI system when the reconfiguration blocks are employed and Fig. 31 shows the probabilities without reconfiguration blocks. Fig. 29 to 31 allow concluding that the integration between FDI and FTC, which must be considered in real-world active FTC applications, reduces the performance of both subsystems

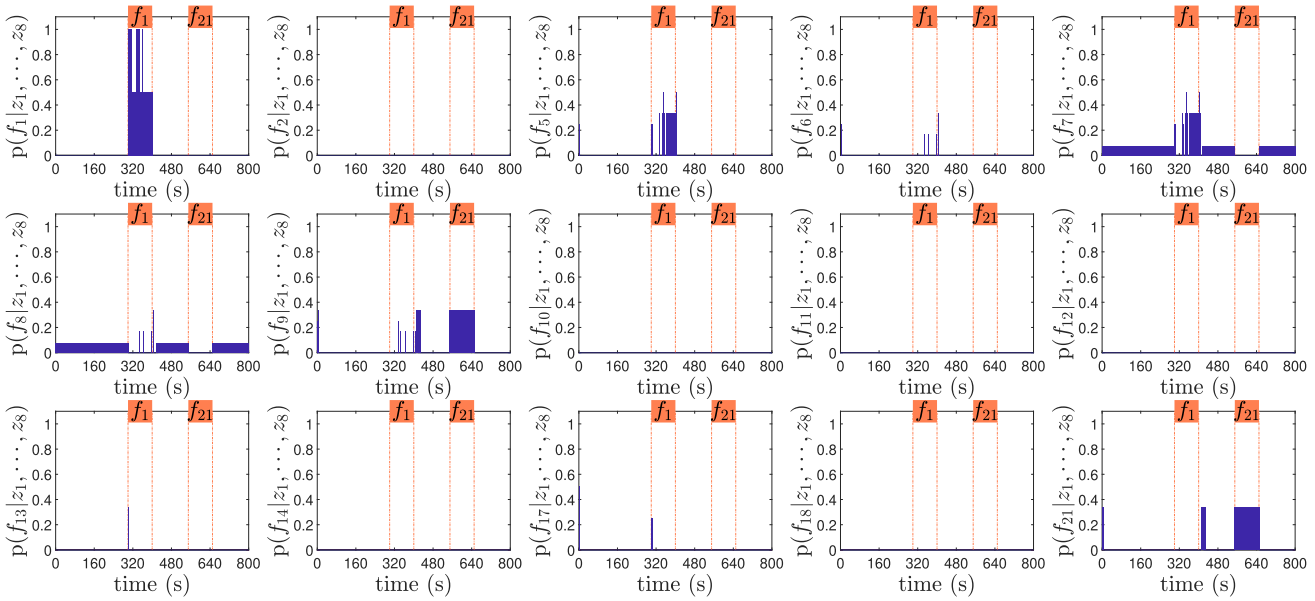


FIGURE 30. Fault probabilities computed by the FDI system with reconfiguration blocks.

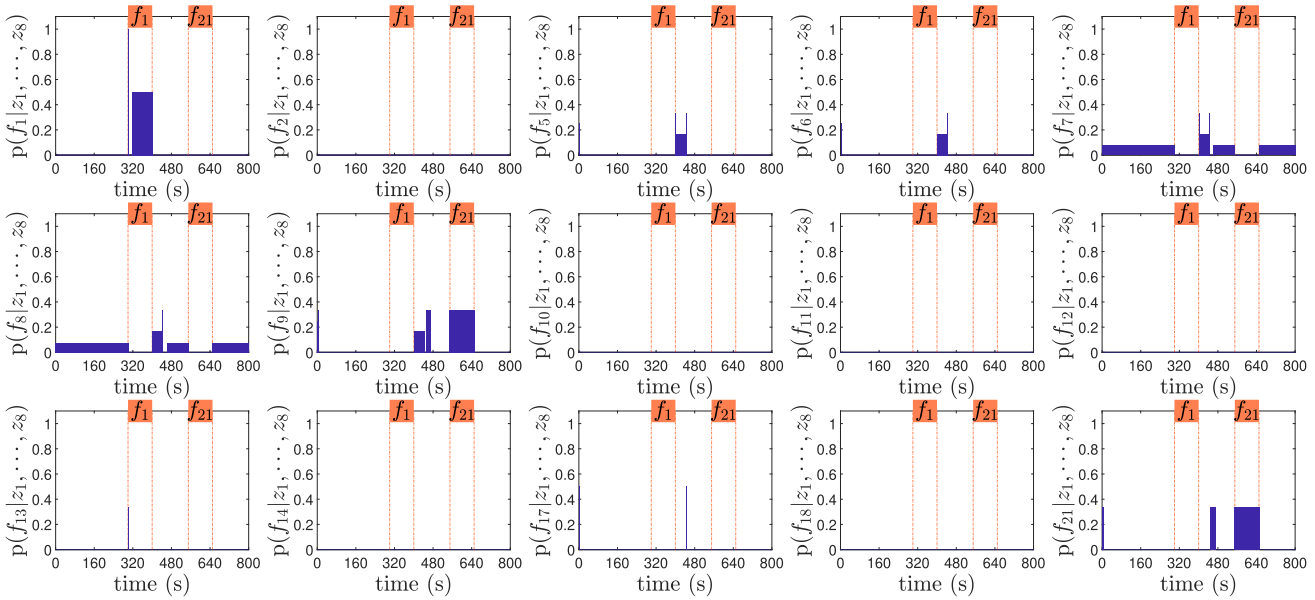


FIGURE 31. Fault probabilities computed by the FDI system without reconfiguration blocks.

(FDI and FTC). Clearly, the reconfiguration due to actuator fault of the tank 1 induces flaws in the FDI system that lead to an uncontrolled switching of the results from FTC and FDI that block the pump P_1. Consequently, the pump P_2 acts saturated due to the fault f_1 and the coupling effect of the system. Comparing the FDI results with FTC integration (see Fig. 30) and without integration (see Fig. 31) it is possible to observe that the persistent oscillation in $p(f_1)$ does not occur when there are no reconfiguration blocks. In both cases, the fault f_{21} is not isolated, since there are dubious indication for fault f_9 and f_{21}. For the ideal scenario, disregarding the FDI imperfections (see Fig. 28), the result is clearly better than the scenario with a non-ideal FDI system (see Fig. 29), which presented a result similar to the system that does not have any reconfiguration block (see Fig. 27), thus indicating that there is no reconfiguration when the FTC system receive the fault f_{21} information from the FDI system.

Note that Sim3Tanks facilitates the analysis of FTC techniques and their integration with FDI strategies, showing the effects of delays and misdetection, as well as the effects caused by actuator saturation. Such effects are often disregarded in FDI and FTC studies, but the experiments with

Sim3Tanks allowed one to conclude that they influence the results in closed-loop.

VII. CONCLUSIONS

This paper presented Sim3Tanks, which is a benchmark model simulator of a three-tank system. The simulator allows the user to test process control, fault detection and isolation, as well as fault tolerant control techniques. The three-tank system implemented in Sim3Tanks core presents some particularly interesting and challenging dynamic aspects, such as nonlinearities, actuator limitations, noises, and hybrid behavior (i.e., its dynamic description can vary greatly depending on the states of the valves and levels inside the tanks). The use of Sim3Tanks can help students to acquire some experience and knowledge in similar industrial processes (such as wastewater treatment, stirred-tank reactor, and refineries), in addition to help other engineers and researchers in the development of new industrial process monitoring and control techniques.

The three-tank system benchmark is a classic problem of nonlinear hybrid process control, which was previously described by Heiming and Lunze [9] and revisited other times [10]. The Sim3Tanks tool described here is a simulator of this benchmark problem, which can be used for research and didactic purposes in a more flexible and simple way. Sim3Tanks allows the user to define scenarios with 1, 2, or 3 coupled tanks, in addition to include different fault signals (e.g., abrupt and incipient faults). The resources and multi-interface environment facilitate the evaluation and comparison of control techniques and process monitoring strategies. Sim3Tanks also offers an intuitive way of response analysis by showing in real-time the dynamic behavior of the system. This kind of analysis can turn a difficult problem into a more attractive problem, thus bringing more knowledge and attention of students and researchers in this simulator.

Sim3Tanks can be used for simulations in three different ways in the MATLAB environment: through graphical interface (via GUIDE), through block diagram (via Simulink), and through command-line (via script). It is possible to simulate different kinds of faults (e.g., blocking or clogging of valves, leakage in tanks, and malfunction of sensors and pumps), thus allowing the user to evaluate and compare different fault detection and diagnosis and fault tolerant control techniques. In order to illustrate the potential of the tool, this paper presented examples of flow and level control strategies, an FDI strategy, and an FTC technique; showing and discussing their results separately and integrated with each other, thus providing evidence of real-world issues through simulation. Although the control and process monitoring examples presented in this study are simple, Sim3Tanks is ready to be employed for testing any state-of-the-art in process control (fault tolerant or not) and monitoring techniques.

REFERENCES

[1] K. Hariprasad and S. Bhartiya, "A computationally efficient robust tube based MPC for linear switched systems," *Nonlinear Anal., Hybrid Syst.*, vol. 19, pp. 60–76, Feb. 2016.

[2] C. Joseph, V. I. George, N. Narayanan, and T. S. Saranya, "Design of hybrid model predictive controller for a hybrid three tank system," *Int. J. Control Theory Appl.*, vol. 8, no. 3, pp. 1235–1242, 2015.

[3] N. N. Nandola and S. Bhartiya, "A multiple model approach for predictive control of nonlinear hybrid systems," *J. Process Control*, vol. 18, no. 2, pp. 131–148, 2008.

[4] E. A. Lee, "The past, present and future of cyber-physical systems: A focus on models," *Sensors*, vol. 15, no. 3, pp. 4837–4869, 2015.

[5] X. Su, X. Liu, P. Shi, and Y.-D. Song, "Sliding mode control of hybrid switched systems via an event-triggered mechanism," *Automatica*, vol. 90, pp. 294–303, Apr. 2018.

[6] C. Yuan and F. Wu, "Hybrid control for switched linear systems with average dwell time," *IEEE Trans. Autom. Control*, vol. 60, no. 1, pp. 240–245, Jan. 2015.

[7] Y. Qian, Y. Fang, and B. Lu, "Adaptive robust tracking control for an offshore ship-mounted crane subject to unmatched sea wave disturbances," *Mech. Syst. Signal Process.*, vol. 114, pp. 556–570, Jan. 2019.

[8] W. Lv, F. Wang, and Y. Li, "Adaptive finite-time tracking control for nonlinear systems with unmodeled dynamics using neural networks," *Adv. Difference Equ.*, vol. 2018, no. 1, pp. 1–17, 2018.

[9] B. Heiming and J. Lunze, "Definition of the three-tank benchmark problem for controller reconfiguration," in *Proc. Eur. Control Conf.*, Aug. 1999, pp. 4030–4034.

[10] M. Mahmoodabadi, M. Taherkhorsandi, and M. Talebipour, "Adaptive robust PID sliding control of a liquid level system based on multi-objective genetic algorithm optimization," *Control Cybern.*, vol. 46, no. 3, pp. 227–246, 2017.

[11] M. Sarailoo, Z. Rahmani, and B. Rezaie, "A novel model predictive control scheme based on bees algorithm in a class of nonlinear systems: Application to a three tank system," *Neurocomputing*, vol. 152, pp. 294–304, Mar. 2015.

[12] S. Zhao, B. Huang, and F. Liu, "Detection and diagnosis of multiple faults with uncertain modeling parameters," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 5, pp. 1873–1881, Sep. 2017.

[13] M. Sarailoo, Z. Rahmani, and B. Rezaie, "Modeling of three-tank system with nonlinear valves based on hybrid system approach," *J. Control Eng. Technol.*, vol. 3, no. 1, pp. 20–23, 2013.

[14] H. A. Nozari, S. Nazeri, H. D. Banadaki, and P. Castaldi, "Model-free fault detection and isolation of a benchmark process control system based on multiple classifiers techniques—A comparative study," *Control Eng. Pract.*, vol. 73, pp. 134–148, Apr. 2018.

[15] K. Villez, B. Srinivasan, R. Rengaswamy, S. Narasimhan, and V. Venkatasubramanian, "Kalman-based strategies for fault detection and identification (FDI): Extensions and critical evaluation for a buffer tank system," *Comput. Chem. Eng.*, vol. 35, no. 5, pp. 806–816, 2011.

[16] X. He, Z. Wang, Y. Liu, L. Qin, and D. Zhou, "Fault-tolerant control for an Internet-based three-tank system: Accommodation to sensor bias faults," *IEEE Trans. Ind. Electron.*, vol. 64, no. 3, pp. 2266–2275, Mar. 2017.

[17] X. He, Z. Wang, L. Qin, and D. Zhou, "Active fault-tolerant control for an Internet-based networked three-tank system," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 6, pp. 2150–2157, Nov. 2016.

[18] A. S. Hashim, Z. Janin, and N. Hambali, "Development of user interface system for control laboratory experiment," in *Proc. 5th Conf. Eng. Educ.*, Dec. 2013, pp. 149–153.

[19] G. Vinagre, D. Valério, P. Beirão, and J. S. da Costa, "Laboratory software for the three-tank benchmark system: From PID to multi-agent fault-tolerant fractional control," *Procedia-Social Behav. Sci.*, vol. 46, pp. 1919–1923, Jan. 2012.

[20] P. Bisták and M. Huba, "Three-tank virtual laboratory for dynamical feedforward control based on MATLAB," in *Proc. 19th Int. Conf. Elect. Drives Power Electron. (EDPE)*, Oct. 2017, pp. 318–323.

[21] P. Bisták and M. Huba, "Three-tank virtual laboratory for input saturation control based on matlab," *IFAC-PapersOnLine*, vol. 49, no. 6, pp. 207–212, 2016.

[22] A. Chevalier, C. Copot, C. Ionescu, and R. De Keyser, "A three-year feedback study of a remote laboratory used in control engineering studies," *IEEE Trans. Edu.*, vol. 60, no. 2, pp. 127–133, May 2017.

[23] R. Dormido *et al.*, "Development of a Web-based control laboratory for automation technicians: The three-tank system," *IEEE Trans. Educ.*, vol. 51, no. 1, pp. 35–44, Feb. 2008.

[24] R. Isermann, *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Heidelberg, Germany: Springer, 2006.

[25] P. F. Odgaard, J. Stoustrup, and M. Kinnaert, "Fault-tolerant control of wind turbines: A benchmark model," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1168–1182, Jul. 2013.
[26] J. R. Dormand and P. J. Prince, "A family of embedded Runge-Kutta formulae," *J. Comput. Appl. Math.*, vol. 6, no. 1, pp. 19–26, 1980.
[27] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Trans. Amer. Soc. Mech. Eng.*, vol. 64, no. 11, pp. 759–768, 1942.
[28] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems: Theory and Design*, 3rd ed. New York, NY, USA: Dover, 2011.
[29] S. X. Ding, *Model-based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*. Heidelberg, Germany: Springer, 2008.
[30] T. Steffen, *Control Reconfiguration of Dynamical Systems*, vol. 320. Heidelberg, Germany: Springer, 2005.
[31] D. Simon and D. L. Simon, "Analytic confusion matrix bounds for fault detection and isolation using a sum-of-squared-residuals approach," *IEEE Trans. Rel.*, vol. 59, no. 2, pp. 287–296, Jun. 2010.
[32] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3757–3767, Jun. 2015.
[33] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part II: Fault diagnosis with knowledge-based and hybrid/active approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3768–3774, Jun. 2015.
[34] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*, 2nd ed. Heidelberg, Germany: Springer, 2006.
[35] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural Network Design*, 2nd ed. 2014. [Online]. Available: http://hagan.okstate.edu/NNDesign.pdf
[36] J. H. Richter, *Reconfigurable Control of Nonlinear Dynamical Systems: A fault-hiding Approach*. Berlin, Germany: Springer, 2011.

ARLLEM O. FARIAS received the bachelor's degree in computer engineering from the Federal University of Amazonas, Brazil, in 2017, where he is currently pursuing the M.Sc. degree with the Graduate Program in Electrical Engineering. His main areas of interest are related to modeling and control of dynamic systems, fault detection and diagnosis, and fault tolerant control.

GABRIEL ALISSON C. QUEIROZ received the bachelor's degree in computer engineering from the Federal University of Amazonas, Brazil, in 2017. He is currently pursuing the M.Sc. degree with the Graduate Program in Electrical Engineering, Federal University of Amazonas. He is also a Software Developer at the Samsung Development Institute for Informatics of Amazônia, Manaus, Brazil. His areas of interest include control systems, embedded systems, distributed systems, and mobile development.

IURY V. BESSA received the B.Sc. and M.Sc. degrees in electrical engineering from the Federal University of Amazonas, Brazil, in 2014 and 2015, respectively. He is currently pursuing the Ph.D. degree at the Federal University of Minas Gerais. Since 2015, he has been an Assistant Professor at the Department of Electricity, Federal University of Amazonas. His areas of interest include computational intelligence, fault detection and diagnosis, fault tolerant control, formal verification, and synthesis of control systems.

RENAN LANDAU P. MEDEIROS received the B.E., M.Sc., and Ph.D. degrees in electrical engineering from the Federal University of Pará, Brazil, in 2013, 2014, and 2018, respectively. He is currently an Associate Professor at the Department of Electricity, Federal University of Amazonas, Brazil. He has experience in electrical engineering with an emphasis on automation and control of industrial and electrical power systems, multi-variable robust control and application. Since 2017, he has been a Full Time Researcher of the e-CONTROLS, a research group whose topics of interests include dynamic and control systems with an emphasis on electric power system control. His main areas of expertise and research interests are related to nonlinear control, multivariable robust control, modeling and design of robust control for electrical power systems.

LUCAS C. CORDEIRO received the Ph.D. degree in computer science from the University of Southampton, U.K., in 2011. He is currently a Senior Lecturer with the School of Computer Science, University of Manchester, U.K. He is also a Collaborator with the Postgraduate Program in Electrical Engineering and Informatics, Federal University of Amazonas, Brazil. His work focuses on software model checking, automated testing, program synthesis, and embedded and cyber-physical systems.

REINALDO M. PALHARES (M'14) received the Ph.D. degree in electrical engineering from the University of Campinas, Brazil, in 1998. He is currently a Full professor at the Department of Electronics Engineering, Federal University of Minas Gerais, Brazil. His main research interests include robust control; fault detection, diagnosis and prognostic; and soft computing. He has been serving as an Associate Editor for the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS and a Guest Editor for *The Journal of The Franklin Institute*-Special Section on Recent Advances on Control and Diagnosis via Process Measurements, the IEEE/ASME TRANSACTIONS ON MECHATRONICS for the Focused Section on Health Monitoring, Management and Control of Complex Mechatronic Systems, and the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS-Special Section on Artificial Intelligence in Industrial Systems.