Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Teaching Model-based Fault Detection and Isolation using a Virtual Laboratory Environment

Javier Sotomayor-Moriano*. Gustavo Pérez-Zúñiga. Mario Soto***. Luis Enciso******
Engineering Department, Pontifical Catholic University of Peru, PUCP,

(e-mail: *jsotom@pucp.edu.pe, **gustavo.perez@pucp.pe, ***mario.soto@pucp.pe, ****lenciso@pucp.pe)

Abstract: Fault detection and isolation (FDI) systems play a key role to provide efficiency, reliability and safety in today's industrial processes. The teaching of FDI systems is facilitated if it is carried out not only with theoretical lectures but also with practical experiences. This paper proposes a virtual laboratory environment (VLE) to carry out online practical experiences with FDI systems for a benchmark process. Thanks to this VLE, students can set up faults in sensors, actuators or in the process itself, program model-based FDI algorithms and test FDI system performance. The use of this environment is illustrated by testing the performance of FDI systems for the quadruple-tank process (4TP) under different fault scenarios. Finally, the procedure of using this proposal for practical experience with two model-based FDI design methods is shown.

Keywords: Fault detection and isolation, virtual lab, model-based FDI, control education.

1. INTRODUCTION

In control of industrial processes, faults in components can lead to critical operation problems. Faults in sensors, actuators or in the process itself, change the performance of the closed-loop control, which further results in a malfunction of the loop. In order to avoid production deteriorations or damage to machines and humans, faults have to be found as quickly as possible (Blanke, 2016); therefore, to fulfil the growing demand for efficiency, reliability and safety of process control, FDI systems are used. Here, fault detection: decide whether a fault has occurred; and fault isolation: find in which component a fault has occurred (Blanke, 2016). Currently, FDI systems play a key role in industrial processes and; therefore, it is necessary to develop resources for teaching engineering students. Teaching FDI systems is facilitated when it is carried out not only with theoretical lectures but also with practical experiences. In control education, lab experimentation plays a key role to connect theory and practice (Heradio, 2016). However, traditional hands-on labs involve high costs associated with equipment, space, and maintenance staff (Gomes, 2009). On the other hand, the development of new technologies that take advantage of the web-based resources have reduced laboratory costs, replacing traditional laboratories with on-line laboratories; the use of virtual and remote laboratories for control education has been expanding for the last two decades. Therefore, the use of an interactive online lab is an excellent option to carry out practical experiences with FDI systems.

Since its appearance (early 1970s) to date, the model-based FDI technique has demonstrated its efficiency in detecting faults by a great number of successful applications in industrial processes (Ding, 2013), also it has been the most widespread in the automatic control community due to its close relationship with modern control theory. Model-based FDI utilize a model of the monitored process, based on which algorithms of FDI system are implemented. The model-based methods include observer based (Chen and Patton, 1999), parity equation based (Getler, 1998) and parameter estimation based (Isermann, 2006) methods.

This proposal provides resources for programming and testing model-based FDI algorithms. Here, the VLE contains a virtual benchmark process with several configurations, and compatibility to set previously designed model-based FDI algorithms based on symbolic or algebraic equations. Presentation of data and results is also considered for states, measurements, residuals and signature matrices among other functionalities that are presented in this paper. This VLE use a Web server app (Sotomayor, 2019) to manage students' assignments and reports. We keep consistency in the main application being implemented in Python, which remains ranked the #1 ranked programming language in 2019 (Cass, 2019), and has compatibility to work with Python and Matlab scripts, furthermore avoiding security issues through the automatic generation of reports that do not include code files in the uploaded files.

This article is structured in the following way: in Section 2 the benchmark process is introduced. Section 3 describes the VLE resources. In Section 4, an overview of the model-based FDI system design is presented; and in section 5, the procedure of using this proposal for practical experience is shown.

2. BENCHMARK PROCESS DESCRIPTION

The quadruple tank process (4TP) is a benchmark process, which is useful for teaching and research purposes. It allows different operating configurations, also it is possible to represent this system as nonlinear or linear, exposing students to a broader practical issues. Fig. 1 shows the basic configuration of 4TP (Johansson, 2000), (Alvarado, 2006).

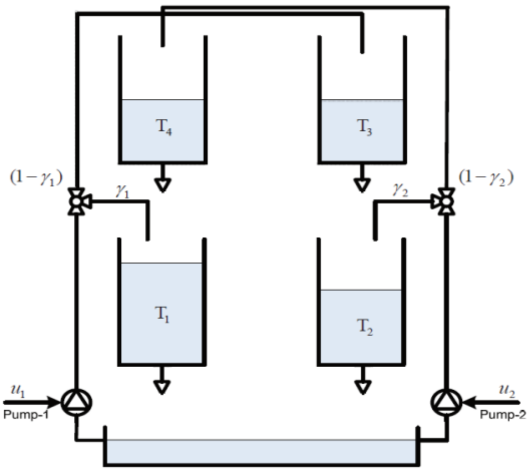

Fig. 1. Basic configuration of the 4TP (Alvarado, 2006).

Fig. 2. Quadruple tanks process. Advanced Control Lab (PUCP)

The flow balance is made for each tank and a non-linear model is obtained (Alvarado, 2006), (Dormido, 2008):

$$\frac{dh_1}{dt} = -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_4}{A_1}\sqrt{2gh_4} + \frac{\gamma_1 k_1}{A_1}u_1 \quad (1)$$

$$\frac{dh_2}{dt} = -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_3}{A_2}\sqrt{2gh_3} + \frac{\gamma_2 k_2}{A_2}u_2 \quad (2)$$

$$\frac{dh_3}{dt} = -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{(1-\gamma_1)k_1}{A_3}u_1 \quad (3)$$

$$\frac{dh_4}{dt} = -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_2)k_2}{A_4}u_2 \quad (4)$$

Where, $\mathrm{A_i}$: Cross section of the tank i ($\mathrm{cm^2}$); $\mathrm{a_i}$: Cross section of the tank outlet i ($\mathrm{cm^2}$); $\mathrm{h_i}$: Water level in the tank in i (m); u_i: voltage applied to Pump i and the corresponding flow is $k_i\mathrm{u_i}$; g: Acceleration of gravity (cm/s^2); $\mathrm{q_i}$: Input flow to the tank (cm^3/s); γ_i: Opening parameter of the 3-way valve.
k_i: Voltage parameter (cm^3/Vs).

Virtual quadruple tank process

The VLE consists of a virtual 4TP. In the virtual 4TP, has been considered the specifications of physical 4TP located in the Advanced Control Lab of PUCP (Fig. 2), in which:
• Maximum flow delivered by the pumps: 266.7cm3/s.
• Pipe diameter: 1.27cm.
• Maximum height in tanks: 40cm.

Validation works allowed to verify virtual 4TP model matched with physical 4TP behavior. In this way, data obtained when working with virtual 4TP are close to those of the physical process. This proposal considers that, after theoretical design of FDI systems using some of the model based methods, students will be able to carry out practical experiences with this VLE to evaluate their solutions.

In the virtual 4TP, it is possible:

- Close and/or open some of the valves, and change the connection of various pipes to obtain other process configurations.
- Set up some typical faults in sensors, actuators and the process itself.
- Include process disturbances and model uncertainties.

3. DEVELOPMENT OF THE VIRTUAL LABORATORY ENVIRONMENT FOR FDI

The proposed VLE for FDI is a novel tool that features a local application called "Desktop app", which is complemented with previously deployed "Server app" (Sotomayor, 2019) that manages assignments and works of students. In summary, VLE's architecture is shown in Fig. 3, and in the following it is presented the features of the Desktop app.

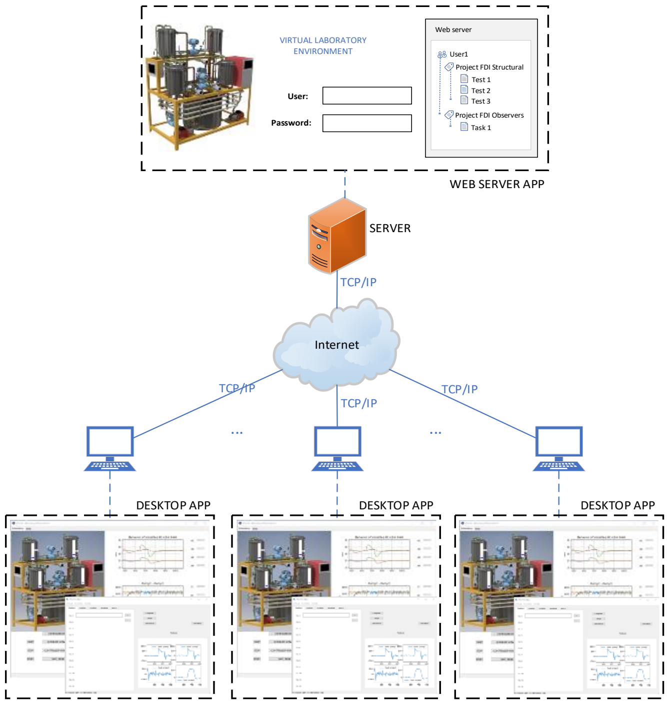

Fig 3. Web server and Desktop apps architecture.

3.1 Desktop app

Desktop app is the application that runs the interactive virtual 4TP that represents the dynamics of our physical 4TP laboratory plant. This simulator can be configured accordingly before and during the simulation to represent different configurations of the 4TP and to input commands. All faults, such as sensor, actuator or process faults, can be emulated accordingly, as well as other additional process conditions (such as disturbances or noises). Additionally, the Desktop app has tools that allow students to implement their own FDI algorithms, using symbolic equations or code, allowing to test the integral solution and to evaluate them using, for example, signature matrices. These algorithms can be updated as well during real-time execution. FDI tools inside the Desktop app are adapted to work with Matlab and Python interchangeably for implementations of model-based algorithms of residual processing and decision logic. Also, the Desktop app is well suited to generate custom charts for every variable of the process.

3.2. Communication between the Desktop app and scripts

The communication between the Desktop app and the scripts, is done in two ways. When Python is used, communication is done by modular implementation of the FDI algorithm as an object that can be called from the Desktop app, a template is provided for such cases. Meanwhile, for Matlab scripts, the communication is done using sockets, thus ensuring a nearly real-time bidirectional communication between them (Fig. 4).

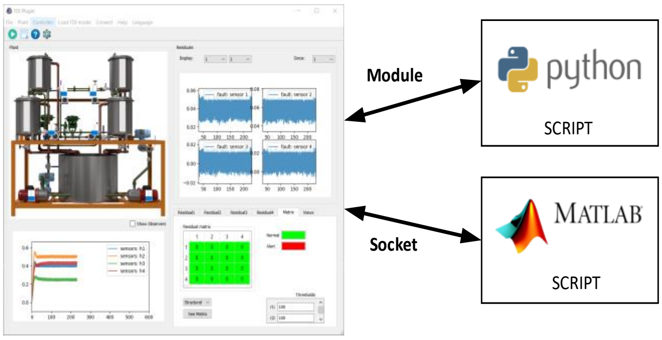

Fig 4. Communication between "Desktop app" and scripts.

3.3. Usage

In the following, we describe the use of the VLE:

1. The student enters the local application installed in a PC of a computer lab, he must access with an account assigned.

2. Once the student performs this authentication, the local application connects with the web application and the desktop app will be ready to work by an authorized period of time.

3. Following, when the student decides to start the practice, the desktop app, which contains the virtual 4TP process (benchmark 4TP), will communicate with Matlab and/or Python, student will be alerted automatically if this communication presents some anomaly conditions.

4. Within Matlab or Python, the student will be able to upload or modify his algorithms of model-based FDI techniques, and run the simulations during the authorized time.

5. When the student finishes his practice, he will need to save and send his work. This operation will be automatically managed by the Desktop app, and a report with all the work (charts and algorithms) will be uploaded into the Web server application, for further evaluation by the instructor.

6. Finally, the student can continue with another practice or close session.

4. OVERVIEW OF MODEL-BASED FDI SYSTEM DESIGN

FDI system performs two tasks fault detection and fault isolation. Here, the aim of fault detection is to generate several features, which indicate no faulty or faulty status, likewise, based on different features, fault isolation is to apply some strategy to determine the location of the fault. Model-based FDI technique stands on the comparison between the process measured outputs and the predicted value from the process model. Fig. 5 shows model-based FDI scheme. Therefore, the design of the model-based FDI system involves: construction of a residual generator, setting of a suitable residual evaluation function and determination of a threshold (Ding, 2013).

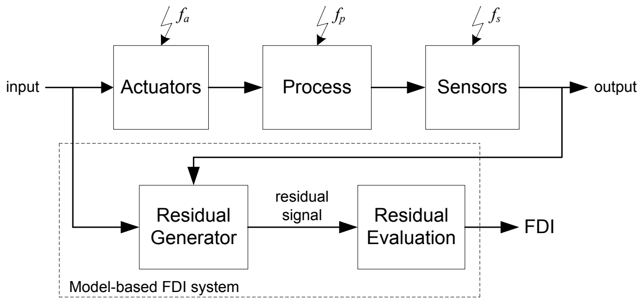

Fig. 5. Description of model-based FDI system; actuator fault f_a, process fault f_p, and sensor fault f_s.

The residual signal is the difference between the measured and estimated process output, this signal is zero in fault free case. However, residual signal becomes non zero not only due to the presence of faults, but also due to model uncertainties and unknown disturbances. Thus, the task of the residual generator is the generation of residuals, which are independent of the model uncertainties and disturbances. The residuals can be generated by: fault detection filter, diagnostic observer, parity relation or structural analysis. Residual evaluation serves to extract the useful information about the faults (residual feature) by processing of the residual signal, in order to arrive at detection and isolation decisions. The threshold logic is used to decide of fault occurrence by comparison between the residual feature and the threshold value (Fault Detection). Thresholds are obtained empirically or by theoretical considerations (Gertler, 1998). While a single residual may be sufficient for fault detection, the isolation of faults requires a set of residuals (Gertler, 1998). Fault isolation task could rely only on residual generators, in this case these are designed for isolation enhanced residuals, exhibiting structural or directional properties. However, in most cases, it is also necessary the use of some strategies in the residual evaluation unit to achieve fault isolation.

In order to achieve the development of skills in model-based FDI technique, the proposed VLE is use to:

- Explain how faults in components influence the process operation.
- Apply algorithms of the model-based FDI system.
- Check the performance of the designed FDI system.
- Practice with benchmark process under different fault scenarios.

5. USING THE VLE

Teaching FDI design with proposed VLE considers:

- Homework on theoretical design solution, which include obtain FDI system algorithms.
- Practical experience with VLE: upload the FDI system algorithms, set up fault scenario, evaluation of designed FDI system. Report generation

In this work, the use of VLE is show with two model-based FDI design methods.

5.1 FDI system with dedicated observer scheme (DOS).

In this case, DOS (Clark, 1979) is to design a FDI system of sensors faults in the benchmark process, for this purpose, it is need a linear model of the process.

a) Theoretical design

In order to design a set of observers, the model is linearized around an operating point. Here is considering,

$$A = \begin{bmatrix} -1/T_1 & 0 & 0 & A_4/(A_1T_4) \\ 0 & -1/T_2 & A_3/(A_2T_3) & 0 \\ 0 & 0 & -1/T_3 & 0 \\ 0 & 0 & 0 & -1/T_4 \end{bmatrix}$$

$$B = \begin{bmatrix} \gamma_1 k_1/A_1 & 0 \\ 0 & \gamma_2 k_2/A_2 \\ (1-\gamma_1)k_1/A_3 & 0 \\ 0 & (1-\gamma_2)k_2/A_4 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Considering: $T_i = A_i\sqrt{\frac{2h_i^0}{g}} /\ a_i, (i = 1, \dots, 4)$

Model parameters: $A_{i=1:4} = 706.85\ cm^2$; $a_{i=1:2} = 5.39\ cm^2$

$a_{i=3:4} = 1.89\ cm^2$; $g = 981\ cm/s^2$;

Operating point: $\gamma_{i=1:2} = 0.7$; $k_{i=1:2} = 1$; $h_{i=1:2}^0 = 25\ cm$;

$h_{i=3:4}^0 = 18.30\ cm$; $u_{i=1:2}^0 = 1193.73\ cm^3/s$;

In the scheme of Fig. 6, each observer is managed by the input vector u and the output of each sensor. In this way for n sensors, there should be n observers. These observers generate a vector of estimated states. The residuals are generated by comparing them with the outputs and can be used to detect and isolate faults.

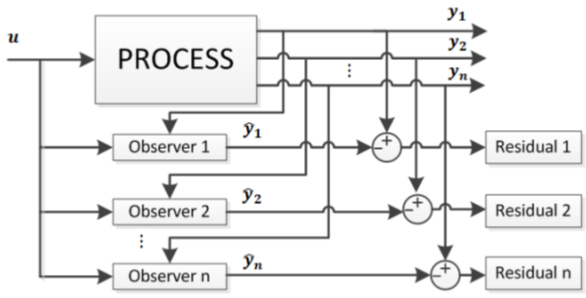

Fig. 6. DOS Residual Generation

Thus, a fault in the n^{th} sensor will cause the estimated state of the n^{th} observer to differ from the estimated state of the n-1 observers, which allows the detection and isolation of the fault directly.

Equations (5) and (6) correspond to the discrete modeling of the system, which can be represented by:

$$X(k+1) = \Phi X(k) + \Gamma u(k) \quad (5)$$
$$y(k) = CX(k) \quad (6)$$

Where Φ y Γ contain the coefficients of the matrices of the discrete model. This model can be used to generate a discrete observer of the form:

$$\hat{X}(k+1) = \Phi\hat{X}(k) + \Gamma u(k) + K_e[y(k) - C\hat{X}(k)] \quad (7)$$

Where K_e is the observer's gain matrix that can be calculated by solving Riccati's equation.

$$K_e = \Delta_{ed}(\phi)\begin{pmatrix} C \\ C\phi \\ \vdots \\ C\phi^{n-2} \\ C\phi^{n-1} \end{pmatrix}^{-1}\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \quad (8)$$

Where:

$$\Delta_{ed}(\phi) = \phi^n + a_{n-1}\phi^n + \dots + a_1\phi + a_0 I = 0 \quad (9)$$

To apply the dedicated observer scheme, it is necessary to check the observability of the system using (10).

$$Ob = \begin{pmatrix} C \\ C\phi \end{pmatrix} \quad (10)$$

After analyzing the observability of the system according to (10) for the multivariable system, it is obtained that the system is completely observable from h_1, h_2, h_3 and h_4.

Applying (8) and (9) the gains of the observers are:

$$K_{e1} = \begin{pmatrix} 31.61 \\ 0 \\ 0 \\ 0.015 \end{pmatrix}, \quad K_{e2} = \begin{pmatrix} 0 \\ 31.62 \\ 0.015 \\ 0 \end{pmatrix}, \quad K_{e3} = \begin{pmatrix} 0 \\ 0.001 \\ 31.62 \\ 0 \end{pmatrix}, \quad K_{e4} = \begin{pmatrix} 104.99 \\ 0 \\ 0 \\ 361.19 \end{pmatrix}$$

The FDI scheme was designed so that the system is able to detect anomalies in h_1, h_2, h_3 and h_4 sensors at a stationary point of the four tanks systems. Four full-order observers were developed from the measurements of h_1, h_2, h_3 and h_4, which estimate $\hat{h}_1, \hat{h}_2, \hat{h}_3, \hat{h}_4$. According to the FDI scheme proposed in Fig. 6, we define:

$$r_1 = |h_1 - \hat{h}_1| \quad r_2 = |h_2 - \hat{h}_2|$$
$$r_3 = |h_3 - \hat{h}_3| \quad r_4 = |h_4 - \hat{h}_4| \quad (11)$$

Where r_1, r_2, r_3, r_4 are the residuals used to evaluate the fault existence of a fault. In some approaches, it is sufficient that a residual exceeds an established threshold to affirm the occurrence of a fault and take action on the event. Next, to implement this solution in the VLE is necessary: based on (7) develop observers algorithms in Python (or Matlab), also, choose a threshold value.

b) Practical experience: Dedicated Observer Scheme

- Program the observer equations with gains calculated offline. Collect the inputs and outputs signal from the virtual 4TP.

- Compare the estimated outputs with the signals from the virtual 4TP sensors to obtain the residuals bank.
- Test the FDI system, setting up a fault in any of the sensors.
- Verify detectability and isolability of faults in graphs of the residuals shown in the virtual environment.

Online performance of designed FDI system is show in Fig. 7, setting up fault scenario:
- Fault in sensor h_1 occurred between 20 and 25 seconds.
- Fault in sensor h_2 occurred between 40 and 45 seconds.

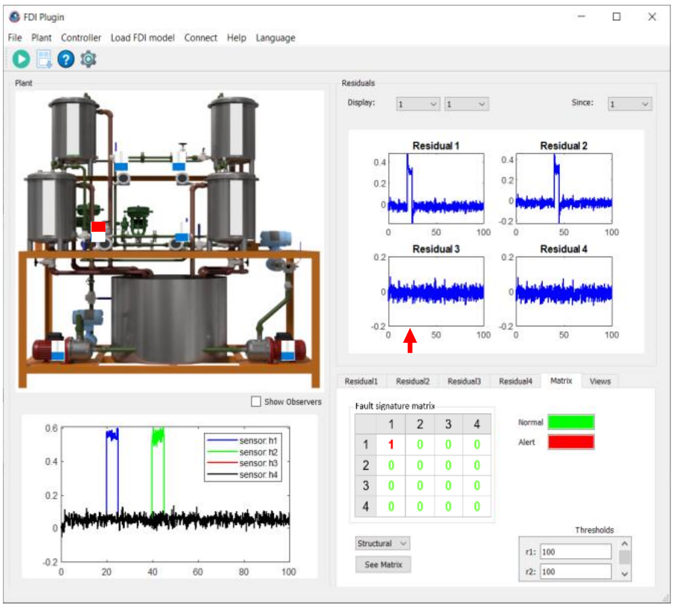

Fig. 7. Desktop app for faults in sensors h_1 and h_2

5.2 Structural analysis approach

As a second case, Structural Analysis approach (Blanke, 2016), (Pérez, 2017) is used to design a FDI system of actuators, sensors faults and process leaks. For this goal, structural model of the process is needed.

a) Theoretical design

Let the 4TP process description consist of a set of 18 equations involving a set of variables partitioned into a set Z of 6 measured variables and a set X of 14 internal unmeasured variables. Likewise, the processes is impacted by the presence of a set F of 8 faults, 2 actuators faults $f_{bi}(i=1,2)$ associated with pumps $P_i(i=1,2)$, 4 sensor faults $f_{mi}(i=1,2,3,4)$ associated with level sensor $LT_i(i=1,2,3,4)$ and 2 faults related to the process $f_{pi}(i=3,4)$ associated with leaks in tanks $T_i(i=3,4)$, (Sánchez, 2019).

The structural model of the system $\Sigma(z,x,f)$ is obtained abstracting the functional equations (1) to (4) of the 4TP model. It retains a representation of which variables are involved in the equations, as shown in equations of Table 1.

Based on the structural model, the incidence matrix for the 4TP process is computed, this matrix is shown in Fig. 8.

Table 1. Equations of the 4TP used to structural model.

$e_1: q_1 = k_1 u_1 + f_{b1}$	$e_2: q_2 = k_2 u_2 + f_{b2}$
$e_3: \dot{h}_1 = -\frac{q_{o1}}{A_1} + \frac{q_{o4}}{A_1} + \frac{\gamma_1}{A_1} q_1$	$d_4: \dot{h}_1 = \frac{dh_1}{dt}$
$e_5: q_{o1} = a_1\sqrt{2gh_1}$	$e_6: y_1 = h_1 + f_{m1}$
$e_7: \dot{h}_2 = -\frac{q_{o2}}{A_2} + \frac{q_{o3}}{A_2} + \frac{\gamma_2}{A_2} q_2$	$d_8: \dot{h}_2 = \frac{dh_2}{dt}$
$e_9: q_{o2} = a_2\sqrt{2gh_2}$	$e_{10}: y_2 = h_2 + f_{m2}$
$e_{11}: \dot{h}_3 = -\frac{q_{o3}}{A_3} + \frac{(1-\gamma_1)q_1}{A_3} + f_{p3}$	$d_{12}: \dot{h}_3 = \frac{dh_3}{dt}$
$e_{13}: q_{o3} = a_3\sqrt{2gh_3}$	$e_{14}: y_3 = h_3 + f_{m3}$
$e_{15}: \dot{h}_4 = -\frac{q_{o4}}{A_4} + \frac{(1-\gamma_2)q_2}{A_4} + f_{p4}$	$d_{16}: \dot{h}_4 = \frac{dh_4}{dt}$
$e_{17}: q_{o4} = a_4\sqrt{2gh_4}$	$e_{18}: y_4 = h_4 + f_{m4}$

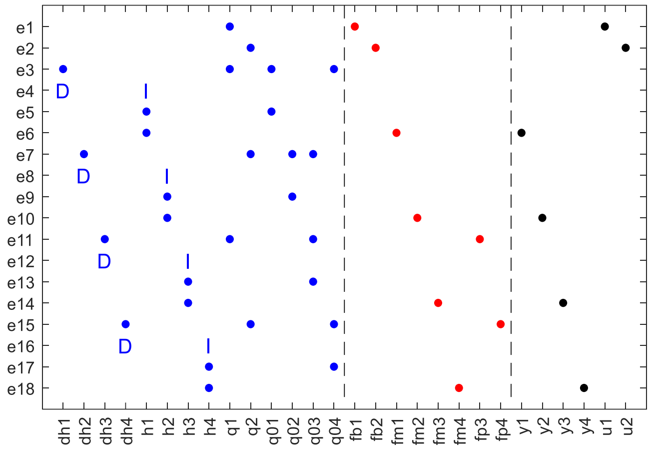

Fig 8. Incidence Matrix

Then, the Dulmage Mendelshon (DM) decomposition is applied to the incidence matrix as a tool to compute redundant sets using structural analysis. Making use of this permutation, the process model is divided into three parts: the structurally overdetermined (SO) part Σ^+; the structurally just determined part Σ^0 and the structurally underdetermined part Σ^-. Based in the SO part, a Fault-Driven Minimal Structurally Overdetermined (FMSO) sets (Pérez, 2017) are computed in order to detect all the fault of interest for this process. A FMSO set φ is defined as a minimal structurally overdetermined set of $\Sigma(z,x,f)$ whose fault support is not empty.

Using the algorithm proposed in (Perez, 2015) to calculate the FMSO sets for the 4TP process, 30 FMSO sets are found that can detect all process faults, (Sánchez, 2019). After, within the set of 30 FMSO sets found, four FMSO sets are selected to isolate the 8 faults of interest, see (12) to (15). For example, FMSO sets φ_1 and φ_3 are sensitives to fault f_{b1} (pump 1 fault).

$$\varphi_1 = \{e_1, e_{11}, e_{12}, e_{13}, e_{14}\} \quad (12)$$
$$\varphi_2 = \{e_2, e_{15}, e_{16}, e_{17}, e_{18}\} \quad (13)$$
$$\varphi_3 = \{e_1, e_3, e_4, e_5, e_6, e_{17}, e_{18}\} \quad (14)$$
$$\varphi_4 = \{e_2, e_7, e_8, e_9, e_{10}, e_{13}, e_{14}\} \quad (15)$$

Analytical redundancy relations (ARR) are equations that only involve measured variables. Let $\Sigma(z,x,f)$ be a system. Then, a relation $arr(z,\dot{z},\ddot{z},\dots) = 0$ is an ARR for $\Sigma(z,x,f)$ if for each z consistent with $\Sigma(z,x,f)$ the relation is fulfilled (Pérez, 2017). Finally, an analytical redundancy relation is obtained for each FMSO set.

b) Practical experience: Structural Analysis

- Solve the set of residual equations found with the structural analysis method.
- Obtain the known plant signals $u_1, u_2, y_1, y_2, y_3, y_4$
- Program all the residual generators based only in known variables of the model.
- Configure the fault signature matrix to meet the objectives of fault detection and isolation.
- Test FDI system setting up a fault scenario.

Online performance of designed FDI system is show in Fig. 9, setting up fault scenario:
- Fault in pump 1 (f_{b1}) occurred between 200 and 245 seconds.

In this case, residuals 1 and 3 indicate fault in f_{b1} according to fault signature matrix. Fig. 9 shows the sensitivity of the 4 residuals (arr_1,arr_2, arr_3 and arr_4) to the fault f_{b1}, which guarantees its detection and isolation.

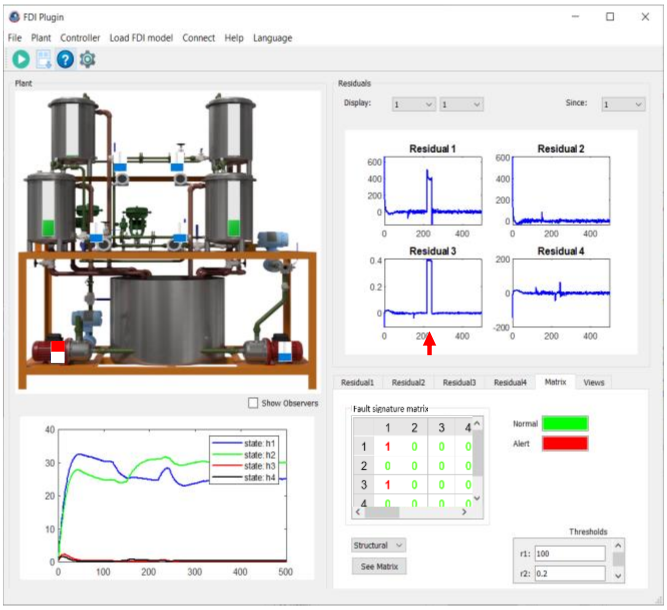

Fig. 9. Desktop app for fault in pump 1. FDI plugin for 4 defined residuals.

At last, a report with all the FDI system design (charts and algorithms) will be uploaded into the Web server application.

6. CONCLUSIONS

An online VLE was proposed, with it, students can easily set up faults in sensors, actuators or in the process itself, program model-based FDI algorithms and test FDI system performance. The proposed Desktop application integrates the model process dynamics with the FDI design through a user-friendly interface, which can handle Python and Matlab scripts codes. It also accepts symbolic representation of the FDI method. Practical experience for the 4TP under different fault scenarios was described. Finally, the use of this proposal with the two model-based FDI methods shown in this paper, can serve as a guide for organizing practical experiences.

AKNOWLEDGMENTS

This work was funded by *Proyecto de Mejoramiento y Ampliación de los Servicios del Sistema Nacional de Ciencia Tecnología e Innovación Tecnológica* 8682-PE, Banco Mundial, CONCYTEC and FONDECYT through grant E041-01[N48-2018-FONDECYT-BM-IADT-MU].

REFERENCES

Alvarado, I., Limon, D., García, W., Alamo, T., and Camacho, E. (2006). An Educational Plant Based on the Quadruple-Tank Process. *7th IFAC Symposium on Advances in Control Education*. vol. 8, no.6, pp. 82-87. Elsevier.

Blanke, M., Kinnaert, M., Lunze, J., and Staroswiecki, M. (2016). Diagnosis and Fault-Tolerant Control. Springer.

Cass, S. (2019). The Top Programming Languages 2019. Journal IEEE Spectrum.

Clark, R. N. (1979). The dedicated observer approach to instrument failure detection. 18th Decision and Control including the Symposium on Adaptive Processes, Vol 2.

Chen, J. and Patton, R.J. (1999). Robust Model-Based Fault Diagnosis for Dynamic Systems. Kluwer Academic Publishers.

Ding, S. X. (2013). Model-Based Fault Diagnosis Techniques - Design Schemes, Algorithms and Tools. London, U.K: Springer-Verlag.

Dormido, R.,Varga, H., Duro, N., Sánchez, J., Dormido-Canto, S., Farias, G., Esquembre, F., and Dormido, S. (2008). Development of Web-Based Control: The Three-Tank System. IEEE Trans. on Educ., vol.51, pp. 35-44.

Gertler, J. (1998). Fault Detection and Diagnosis in Engineering Systems. Marcel Dekker, New York.

Gomes, L. (2009). Current trends in remote laboratories, IEEE Trans. on Industrial Electronics 56, pp. 4744-4756.

Heradio, R., De la Torre, R., and Dormido, S. (2016). Virtual and remote labs in control education: A survey. UNED.

Isermann, R. (2006). Fault-Diagnosis Systems. Berlin, Germany: Springer-Verlag.

Johansson, K. (2000). The Quadruple-Tank Process; A Multivariable Laboratory Process with an Adjustable Zero. IEEE T on Con. Sys. Tec., vol.8, no.3., pp. 456-465.

Pérez, C.G., Travé-Massuyès, L., Chanthery, E., Sotomayor J. (2015) Decentralized Diagnosis in a Spacecraft Attitude Determination and Control System. Journal of Physics: Conference Series 659 (1), 012054

Pérez, C., Chanthery, E., Travé-Massuyès, L., Sotomayor, J. (2017) Fault-Driven Structural Diagnosis Approach in a Distributed Context, IFAC-PapersOnLine, Volume 50, Issue 1, Pages 14254-14259, ISSN 2405-8963.

Sánchez, V., Pérez, G, Sotomayor, J. (2019). Reconfigurable Model Predictive Control applied to the Quadruple Tank Process. 15th European Workshop on Advanced Control and Diagnosis ACD. Bologna, Italy.

Sotomayor, J., Pérez, G., Soto, M. (2019). A Virtual Laboratory Environment for Control Design of a Multivariable Process. Journal IFAC-PapersOnLine, 52 (9), pp. 15-20.