A simple procedure for fault detectors design in SISO systems [*]

D. Tena[a], I. Peñarrocha-Alós[a]

[a] *Departament d'Enginyeria de Sistemes Industrials i Disseny. Universitat Jaume I de Castelló, Spain*
(e-mail: {david.tenatena,ipenarro}@uji.es).

Abstract

In this work, we present a novel approach for fault detectors design and implementation in the case of actuator faults. Both the design and the implementation are focused on simplicity. The fault detector is based in an output observer that estimates the fault signal followed by a decision mechanism that detects the presence of a fault from the estimation. The observer consists of two transfer functions fed by the process manipulated variable and the sensor measurement. For the synthesis of the fault detector, we just need an input-output model of the process and two tuning parameters; one used in the observer, and the other in the decision mechanism. We present simple rules for the design considering the trade-off between the detection time, the minimum detectable fault and the false alarm rate. Our implementation method uses standard tools available in industrial control systems and we have applied it to a real two-tank system setup. The main contribution of this work is the simplicity of the design and implementation of the fault detector, making it suitable for process industry and for being managed by not experts in control systems. Another contribution is the *a priori* design based in intuitive engineering performance indices.

Keywords: Fault detection, Transfer functions, Continuous-time systems, Implementation.

1. Introduction

The area of process control has made great advances in the last decades thanks to the emergence of industrial computers, suitable to implement control algorithms and automation sequences. Field control actions related with actuator equipment, which human operators used to carry out, are becoming more and more automated. This has increased the reliability and efficiency of industrial plants and it has reduced the amount of low added-value tasks as well as repetitive operations performed by the workers. Despite this, a critical control task remains mainly as a manual task, which is to detect abnormal events, to diagnose its causes and to make proper adjustments in order to keep the process into a safe operating state. This task has become more difficult over the years, due to the size and complexity of today's plants and the different kind of faults likely to happen. This entails an information overload for the human operator that makes it difficult to find out an abnormal event and realize the root cause. Therefore, fault detection and diagnosis has emerged as an area of knowledge with high research activity.

A complete introduction to fault diagnosis can be found in [11]. There are several diagnostic algorithms available [31], among those we can distinguish between model-based methods and process history based or data driven. Review references about model-based fault diagnosis are [9] and [6]. Within model-based methods, there are quantitative and qualitative methods. Regarding the quantitative model-based techniques, this work focus on observer-based techniques, a recent review of which can be found in [38]. The classical approach is usually based in state observers [32, 10], which imply the use of state space models. It is popular to use extended observers, also called PI-observers, and assume the fault signal as a new state [20, 14, 13]. Then, state observer techniques can estimate the fault signal. Regarding observer design, classical techniques are based on pole placement or eigenstructure assignment [23, 22], which performance in terms of detection time or false alarm rate is checked after the design, not before. Robust design is also covered in the literature [41, 33]. Even if these are state of the art techniques in control systems, the use of state space models and matrix operations required for the observer design and implementation are not so tractable for process engineers or industrial operators. This results in a low use of these techniques in process industry. Additionally, performance cannot be set before the design.

Most of the recent work is about dealing with nonlinear and uncertain systems [19, 36, 35], where advanced techniques as sliding mode observers are used [2, 15]. In [39, 25, 26, 27], the authors focus the observer design in achieving the desired performance, but that implies to solve complex and computationally expensive optimization problems, which make it less attractive for its implementation.

If we compare fault detection with regulatory control

[*] This work has been supported by projects UJI-B2018-39 from Universitat Jaume I de Castelló and MINECO project number TEC2015-69155-R.

field, we realize that one of the main reasons why industrial control is so widely extended is the simplicity and intuitiveness of the PID controller [12, 4]. It allows us to control different processes by only setting two or three parameters. Moreover, these parameters can be tuned with simple rules once the process model is known or after some experiment is performed [28, 3, 7]. The implementation is also straightforward since almost every industrial control system has preset tools configured for that.

In case of fault detection techniques, we have not found in the bibliography solutions as simple and easy to implement so that they can become a new standard in process industry (refining, oil and gas or chemical industry). In this field, it is common the use of observers to estimate unmeasured variables due to the lack of appropriate estimating devices or to replace high-priced sensors in a plant; disturbance and fault detection is one of the main application of this observers [1]. We can find examples of observers based in statistical methods like principal components analysis [17, 42], AI-based techniques [34], and fuzzy-logic [8, 37]. Model-based techniques for fault detection found in literature mainly use state observers [29, 21, 5, 24]. Furthermore, the design procedure in the cited works does not depend on the desired performance, and it must be checked after the design, leading to iterative procedures.

Motivated by this fact, the aim of this work is to pose an approach for the design of fault detectors trying to achieve a design procedure as straightforward as PID design. We look for an approach with few tuning parameters easy to understand and intuitive. Motivated also by the easy implementation of PID controllers in industrial control systems, we look for an approach easy to implement in those systems. Our proposal only requires an input-output model of the process and it allows us to reach targets related with the detection time, the minimum detectable fault and the false alarm rate, considering trade-offs between them as in [39]. An inspiration for this work is the coprime factorization technique explained in [9], which decomposes the model of the process in an invertible part and a non-invertible part. We propose a similar technique for the design of the fault observer, as it is described in the next sections. The proposed fault detector is composed of a fault estimator followed by a fault detection mechanism. We use the input and the output measurement of the process to estimate the fault signal and then use it on a decision mechanism to decide if a fault has occurred. The development of a fault estimator is also important for further applications on active fault tolerant control [16, 40, 18].

Main contributions of this work are:

- Fault detection structure based in input-output models instead of state space models, allowing a major understanding of the method for practitioners.
- Performance-based design in terms of three main features: false alarm rate, minimum detectable fault and detection time.
- Simple tuning rules based in the process model, even if only a simple first order plus time delay model is available.
- Just two continuous tuning parameters and an optional discrete one.
- Easy implementation technique, using tools from industrial control systems.

On the other hand, to achieve this simplicity in the fault detector design and implementation, we sacrifice some properties. There are not enough degrees of freedom to optimize the three features (false alarm rate, minimum detectable fault and detection time) so two of these features will be optimized and the third one will be given in consequence.

The structure of this work is the following. Section 2 exposes the problem formulation and describes the main objectives. Section 3 shows the proposed structure for the fault estimator. Section 4 explains the tuning rules for the fault estimator design and compares our approach with other methods. Section 5 indicates the detection mechanism that follows the fault estimator, and it gives simple rules for the tuning of the fault detector. In Section 6 we adapt the results for real systems with an experimental approach, and we analyze the advantages and disadvantages of this procedure. In Section 7 we implement and test the proposed fault detector in a real system, recreated in a laboratory. Last, in Section 8 we expose the main conclusions.

2. Problem Formulation

Let us consider a linear time-invariant process $G(s)$ with an available input u and a measurable output y, which can be affected by an actuator fault signal f. The output is also altered by a noise signal v, which we consider high frequency and Gaussian. Therefore, the system is defined by

$$y(s) = G(s)\left(u(s) + f(s)\right) + v(s). \tag{1}$$

Let us express the transfer function of the process as

$$G(s) = \frac{K\prod_{i=1}^{p}(1+\beta_i s)\prod_{i=1}^{q}(1-\delta_i s)e^{-Ts}}{s^k\prod_{i=1}^{r}(1+\tau_i s)}, \tag{2}$$

where $k = 1$ if the system has an integrator and $k = 0$ if not. K is the process static gain, β_i is the time constant of the half-left zeros in the complex plane ($\Re(\beta_i) > 0$) and δ_i the one for those in the half-right plane ($\Re(\delta_i) < 0$). T is the dead time and τ_i the time constant of both the stable and unstable poles. We assume that the transfer function $G(s)$ is proper (i.e., $p + q \leq k + r$)

The main objectives of this paper are

- To build a model-based estimator of the fault $f(s)$ using continuous transfer functions and few tuning parameters.
- To provide a design strategy which ensures certain estimation performance requirements.
- To define a decision mechanism to detect if a fault has occurred.
- To provide a simple experimental procedure for its tuning and industrial implementation.

3. Fault Estimator Structure

In this Section we present the Fault Estimator (FE) and we justify the reasons that led us to adopt its particular structure, based in the fulfillment of the some performance requirements defined below. We propose the FE of the figure 1, composed of the transfer functions H_u and H_y, so that the fault estimation is

$$\hat{f}(s) = H_u(s)u(s) + H_y(s)y(s), \qquad (3)$$

where $H_u(s)$ and $H_y(s)$ are to be defined.

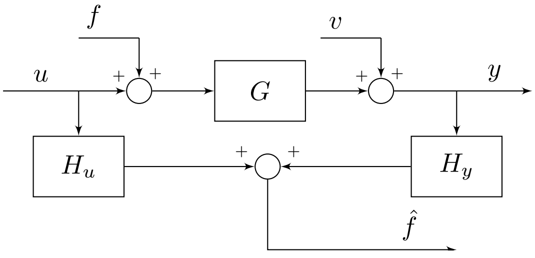

Figure 1: System with Fault Estimator

We define the fault estimation error as $\tilde{f}(s) = f(s) - \hat{f}(s)$ so

$$\begin{aligned}\tilde{f}(s) &= [1 - H_y(s)G(s)]\, f(s) - \\ &\quad - [H_u(s) + H_y(s)G(s)]\, u(s) - \\ &\quad - H_y(s)v(s). \qquad (4)\end{aligned}$$

For optimal fault estimation, the objective is to minimize the fault estimation error $\tilde{f}$. In presence of fault f, this means $\hat{f} = f$. In presence of noise v and no fault, this means $\hat{f} = 0$. This approach is slightly different to the residual approach commonly used, where the objective is to maximize residual sensitivity to fault f and minimize residual sensitivity to noise v.

3.1. Initial analysis

The choice of an structure for $H_u(s)$ and $H_y(s)$ is based in the fulfillment of some desirable requirements for the FE. We define the following indicators to evaluate the performance of the FE with respect to the three exogenous signals of the system: the fault, the input and the noise signals. The fault is unknown in advance but, for design purposes, the fault estimator has to fulfill some requirements over faults of a certain type, such as step or ramp faults:

- Step fault: $f(s) = \frac{1}{s}$.
- Ramp fault: $f(s) = \frac{1}{s^2}$.

The proposed indicators are the following[1]:

A1. The steady-state fault estimation error versus step fault, defined as

$$\lim_{t\to\infty} \tilde{f}(t) = \lim_{s\to 0} (1 - H_y(s)G(s)) \qquad (5)$$

A2. The cumulative fault estimation error versus step fault, defined as

$$\lim_{t\to\infty} \int_0^t \tilde{f}(t)dt = \lim_{s\to 0} (1 - H_y(s)G(s))\frac{1}{s}.$$

A3. The steady-state fault estimation error versus ramp fault, defined as

$$\lim_{t\to\infty} \tilde{f}(t)dt = \lim_{s\to 0} (1 - H_y(s)G(s))\frac{1}{s}.$$

Since the ramp fault is the integral of the step fault, this indicator coincides with the previous one.

A4. The noise effect, which according to (4) and considering the noise as a high frequency signal can be approximated by

$$\lim_{s\to\infty} -H_y(s).$$

A5. The sensitivity from the system input $u(s)$, defined with the $\mathcal{H}_\infty$ norm

$$\left\| H_u(s) + H_y(s)G(s) \right\|_\infty .$$

We look for a simple structure for $H_u(s)$ and $H_y(s)$ which achieves as much as possible a good performance in terms of previous indicators. Specifically, they have to fulfill the next requirements:

B1. $H_u(s)$ and $H_y(s)$ must be stable, proper and causal (realizable).

B2. Considering (4), $[1 - H_y(s)G(s)]$ and $[H_u(s) + H_y(s)G(s)]$ must be stable. This ensures a stable response of the FE from faults and manipulable inputs.

B3. $H_y(s)$ must fulfill $\lim_{s\to 0} H_y(s)G(s) = 1$ to ensure null steady-state fault estimation error, according to **A**1.

[1] We use the final value theorem: $\lim_{t\to\infty} x(t) = \lim_{s\to 0} sX(s)$

B4. $H_y(s)$ must fulfill $\lim_{s\to 0}(1 - H_y(s)G(s))\frac{1}{s} \leq \gamma,\ \gamma \in \mathbb{R}^+$ to ensure finite error versus a ramp fault, according to **A**3.

B5. $H_y(s)$ must be proper to ensure finite noise amplification, according to **A**4.

B6. The sensitivity from the system input must be bounded, i.e., $\|H_u(s) + H_y(s)G(s)\|_\infty \leq \epsilon,\ \epsilon \in \mathbb{R}^+$, according to **A**5.

Considering this, in the next section we justify the selected structure for H_u and H_y.

3.2. Structure choice

Considering the dynamics of the fault error expressed in (4), we can achieve

– perfect fault estimation if

$$H_y(s) = G(s)^{-1}.$$

– perfect input decoupling if

$$H_y(s) = -H_u(s)G(s)^{-1}.$$

– perfect noise attenuation if

$$H_y(\infty) = 0.$$

Obtaining perfect fault estimation has the main drawback of breaking requirement **B**1 in the following cases: $G(s)$ is strictly proper (the most common case), as it would lead to a non-proper $H_y(s)$; $G(s)$ has right half plane zeros, as it would lead to an unstable $H_y(s)$; or $G(s)$ has some delay, as it would lead to non-causal $H_y(s)$. Furthermore, the achievement of perfect noise attenuation would imply $H_y(s)$ being strictly proper, what is incompatible with perfect fault estimation, as $G(s)^{-1}$ will never be strictly proper. We see here that in the design of the transfer function $H_y(s)$ there is a trade-off between achieving fault estimation and noise attenuation.

Let us take now some considerations on the inverse of the process transfer function. We can decompose $G(s)$ as

$$G(s) = G_I(s)G_N(s),$$

where

$$G_I(s) = \frac{K\prod_{i=1}^{p}(1+\beta_i s)}{s^k\prod_{i=1}^{r}(1+\tau_i s)},$$
$$G_N(s) = \prod_{i=1}^{q}(1-\delta_i s)e^{-Ts}.$$

This allows us to easily consider the part whose inverse leads to an stable and causal system ($G_I(s)$) and the part whose inverse leads to an unstable system or non-causal system ($G_N(s)$). If the relative degree of $G_I(s)$ given by

$$d = k + r - p,$$

is greater than zero, its inverse will not be proper. In order to achieve a proper transfer function, we propose to include a pole with multiplicity n equal or greater than d, i.e., $n \geq d$. The time constant of this pole is α. With all of these considerations, we propose

$$\begin{aligned} H_y(s) &= \frac{G_I(s)^{-1}}{(1+\alpha s)^n} = \\ &= \frac{s^k\prod_{i=1}^{r}(1+\tau_i s)}{K\prod_{i=1}^{m}(1+\beta_i s)\,(1+\alpha s)^n}, \end{aligned} \tag{6}$$

This fulfills **B**2, **B**3, **B**4 and **B**5. We use multiplicity $n = d$ to achieve the simplest solution, although solutions with multiplicity $n > d$ can be also used.

We can find an structure for $H_u(s)$ which leads to $\epsilon = 0$ in **B**6, i.e., perfect input decoupling, and also fulfills **B**1, i.e., $H_u(s)$ and $H_y(s)$ realizable, as

$$\begin{aligned} H_u(s) &= -H_y(s)G(s) = \frac{-G_N(s)}{(1+\alpha s)^n} = \\ &= \frac{-\prod_{i=1}^{n}(1-\delta_i s)e^{-Ts}}{(1+\alpha s)^n}. \end{aligned} \tag{7}$$

If we consider $G(s)$ known, then the design of the FE consists in the tuning of a continuous parameter, α, and a discrete one, n (that must fulfill $n \geq d$).

4. Fault Estimator Design

In the previous section we have proposed a structure for the transfer functions H_y and H_u. In this section, we analyze how to tune the parameter α depending on previous performance indicators **A**1 to **A**5.

4.1. Fault Estimator error analysis

We first rewrite the fault estimation error dynamics (4) when using the proposed fault estimator structure given by (6) and (7), leading to

$$\tilde{f}(s) \;=\; \left[1 - \frac{G_N(s)}{(1+\alpha s)^n}\right]f(s) - \frac{G_N(s)}{(1+\alpha s)^n}v(s), \tag{8}$$

where we appreciate the cancellation of the effect of input u over the fault estimation error, and that the poles that define the dynamics of the fault estimation error are $s = -1/\alpha$ with multiplicity n.

From dynamics (8) we can compute indicators **A**1 to **A**5 for $n = d$ as a function of α as follows:

A1. f step: $\lim_{t\to\infty}\tilde{f}(t) = 0$

A2. f step: $\lim_{t\to\infty}\int_0^t \tilde{f}(t)dt = n\alpha + \sum\delta_i + T$.

A3. f ramp: $\lim_{t\to\infty}\tilde{f}(t)dt = n\alpha + \sum\delta_i + T$

A4. v high frequency noise: $\lim_{s\to\infty} -H_y(s) = \frac{-\prod\tau_i}{\alpha^n K\prod\beta_i}$.

A5. sensitivity to u: $\|H_u(s) + H_y(s)G(s)\|_\infty = 0$.

The value of the tuning parameter α has an impact on the measurement noise amplification as well as the tracking error of the FE. In order to refer to a measure of the fault tracking ability of the estimator we will use the time delay versus unitary ramp fault **A**3, leading to the following expression

$$t_d = n\alpha + \sum \delta_i + T. \tag{9}$$

This is also the expression for the cumulative fault estimation error versus step fault.

We consider the noise as a high frequency random signal with certain statistical distribution. For convenience, we approximate this frequency to infinite, which is a good approximation if $H_y(s)$ is strictly proper, i.e it has direct gain, which is always true when we set multiplicity $n = d$. Then, the variance of the estimator due to measurement noises is

$$\sigma_f^2 = \left(\lim_{s\to\infty} H_y(s)\right)^2 \sigma_v^2 = \left(\frac{\prod \tau_i}{K \prod \beta_i \, \alpha^n}\right)^2 \sigma_v^2 \tag{10}$$

where $\sigma_f^2 = var(\tilde{f})$ and $\sigma_v^2 = var(v)$.

If values greater than d are used in the multiplicity of the additional pole, i.e., $n > d$, $H_y(\infty)$ would not be a good approximation for noise amplification since $H_y(\infty) = 0$. In those cases, we should compute the the frequency response for the sampling frequency (ω_s) in a digital implementation to have an approximation of high frequency noise effect, i.e., $|H_y(j\omega_s)|$. That would lead to

$$\begin{aligned} \sigma_f^2 &= |H_y(j\omega_s)|^2 \sigma_v^2 \\ &= \frac{\omega_s^{2k} \prod(1+\omega_s^2\tau_i^2)}{K^2 \prod(1+\omega_s^2\beta_i^2)(1+\omega_s\alpha^2)^n} \sigma_v^2 \end{aligned} \tag{11}$$

From (9), (10) and (11) we realize that a high value of α reduces the noise effect but slows the fault signal tracking while a lower α speeds up the fault signal tracking but increases the noise effect. Therefore, there is a trade-off between requirements **B**4 and **B**5.

4.2. Parameter tuning

The design of the FE consists in choosing an appropriate value for α. We can follow two strategies:

C1. To set the desired tracking error delay, t_d^*. From (9), we set

$$\alpha = \frac{t_d^* - \sum \delta_i - T}{n}. \tag{12}$$

Then, the resulting estimator error variance can be obtained from (10) or (11).

C2. To set the noise amplification by setting the desired fault signal variance, σ_f^{*2}. For $n = d$, we get from (10),

$$\alpha = \sqrt[2n]{\frac{\sigma_v^2 \left(\prod \tau_i\right)^2}{K^2 \left(\prod \beta_i\right)^2 \sigma_f^{*2}}}. \tag{13}$$

In the case of $n > d$, we get from (11),

$$\alpha = \frac{1}{\omega_s}\left(\sqrt[n]{\frac{\sigma_v^2 \omega_s^{2k} \prod\left(1+\omega_s^2\tau_i^2\right)}{K^2 \prod\left(1+\omega_s^2\beta_i^2\right) \sigma_f^{*2}}} - 1\right)^{1/2} . \tag{14}$$

Then, the resulting tracking delay can be obtained from (9).

For a given n, there is just one tuning parameter α so we can only achieve one of the objectives, either the tracking delay or the noise amplification.

Fault Estimator design has been explained in the last two sections, including the structure choice for $H_u(s)$ and $H_y(s)$ and the procedure for tuning the parameters α and n. The proposed FE is able to track fault signals with no error in steady state and it is decoupled from the system input signal u. The tuning of α and n determines the performance in terms of tracking error delay and noise amplification, so we have presented equations to design the FE according to these performance indices.

4.3. Comparison with other techniques

Other fault estimators found in the literature use state space models and the fulfillment of conditions equivalent to those presented here. For instance, the inclusion of an integral term in the observer leads to null steady state error under step disturbances. This is called a PI observer. Let us assume that we have a model for the system as

$$\dot{x} = A\,x + B\,(u + f), \tag{15}$$

$$y = C\,x + v. \tag{16}$$

Let us also consider an integrator for the generation of fault f as

$$\begin{bmatrix}\dot{x}\\ \dot{f}\end{bmatrix} = \begin{bmatrix}A & B\\ 0 & 0\end{bmatrix}\begin{bmatrix}x\\ f\end{bmatrix} + \begin{bmatrix}B\\ 0\end{bmatrix} u + \begin{bmatrix}0\\ I\end{bmatrix}\delta f \tag{17}$$

$$y = \begin{bmatrix}C & 0\end{bmatrix}\begin{bmatrix}x\\ f\end{bmatrix} + v, \tag{18}$$

being δf the fault generator signal. Then, we can construct an observer as

$$\begin{bmatrix}\dot{\hat{x}}\\ \dot{\hat{f}}\end{bmatrix} = \begin{bmatrix}A & B\\ 0 & 0\end{bmatrix}\begin{bmatrix}\hat{x}\\ \hat{f}\end{bmatrix} + \begin{bmatrix}B\\ 0\end{bmatrix} u + L\left(y - \begin{bmatrix}C & 0\end{bmatrix}\begin{bmatrix}\hat{x}\\ \hat{f}\end{bmatrix}\right) \tag{19}$$

$$\hat{f} = \begin{bmatrix}0 & I\end{bmatrix}\begin{bmatrix}\hat{x}\\ \hat{f}\end{bmatrix}, \tag{20}$$

where L is the observer gain to be designed. The ability of the observer to track the fault and to reject the effect of the measurement noises depends on that matrix L. In this approach, the observer transfer functions in figure 1 are computed as

$$H_u(s) = \bar{C}(sI - \bar{A})^{-1}\bar{B}, \quad H_y(s) = \bar{C}(sI - \bar{A})^{-1}L$$

where

$$\bar{A} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} - L \begin{bmatrix} C & 0 \end{bmatrix}, \qquad \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \qquad \bar{C} = \begin{bmatrix} 0 & I \end{bmatrix}.$$

The observer design (i.e., computing matrix L) relies on the resulting extended model. For that, there are several techniques as pole placement, optimal estimation or bounding some system norm. When those designs try to show the trade-offs between tracking speed and noise amplification, they require the solution of an optimization problem, mainly through matrix inequalities. In some cases, those problems are convex and, therefore, standard solver can solve the problem. In other cases, those optimization problems are not convex and we must apply iterative or heuristic procedures. Furthermore, those fault estimator designs try to bound some generic system norms but they do not focus the design on the desired performance for the observer, so the user must evaluate them *a posteriori*.

In that sense, our procedure differs in the use of input-output models as well as in the design techniques, avoiding the computation of complex optimization problems. Furthermore, in our proposal we focus the design on the desired performance indices, instead of checking them after the fault estimator design.

Example 1. *In order to illustrate the differences between our fault estimator and others based on state space models we use in this section a numeric example with the following transfer function for the plant model:*

$$G(s) = \frac{1}{(1+s)^3}. \tag{21}$$

An state space model for this system is given by matrices

$$A = \begin{bmatrix} -3 & -1.5 & -0.5 \\ 2 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \; B = \begin{bmatrix} 0.5 \\ 0 \\ 0 \end{bmatrix}, \; C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}.$$

We compare these different designs:

1. *Our approach with $n = 3$ and $\alpha = 2$.*

2. *Our approach with $n = 4$ and $\alpha = 2$.*

3. *State Observer with poles on $\{-1/2, -1/2, -1/2, -10/2\}$*

4. *State Observer with poles on $\{-1/2, -1/2, -1/2, -1/2\}$*

In order to compare the results of the different proposals we show the frequency response from fault signal to its estimation, and from measurement noise to fault estimation. We see in figures 2 and 3 that the second and fourth approaches are equivalent as they look for the same poles for the fault estimation dynamics. The first and third approach behave practically the same for frequencies under 1 rad/s. With the state observer approach, we can set other poles to achieve different responses, which is where the state space approach increases its flexibility (degrees of freedom) with respect to ours, as we use one pole with a given multiplicity.

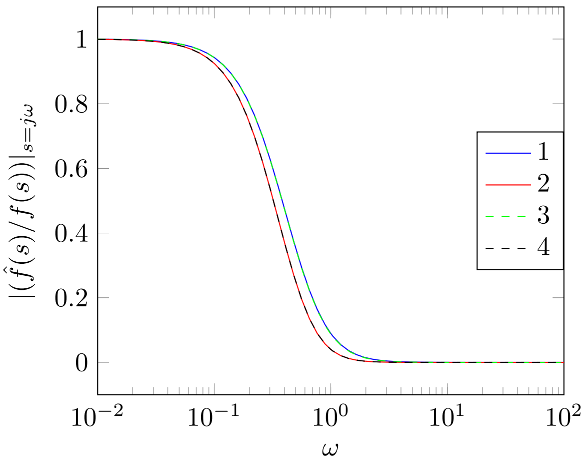

Figure 2: Frequency response of the fault estimation under faults.

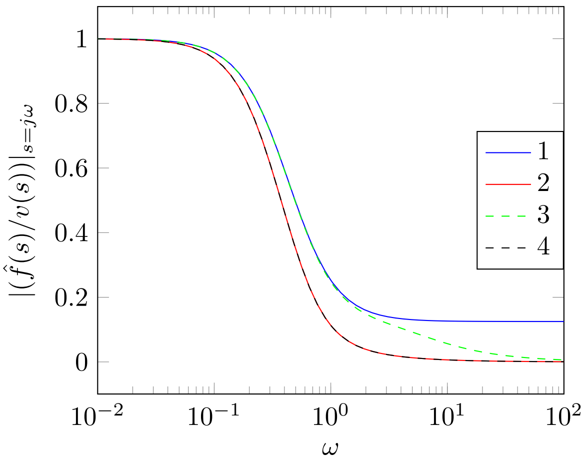

Figure 3: Frequency response of the fault estimation under measurement noises.

5. Fault Detection

Fault Detection (FD) consists in generating a binary signal which is 0 if there is no fault and 1 if there is a fault. This binary signal results from a Fault Detection Mechanism (FDM), which performs the following comparison:

$$\begin{cases} \text{if } |\hat{f}| > J, & \text{Fault,} \\ \text{otherwise,} & \text{No Fault.} \end{cases} \tag{22}$$

Here, $\hat{f}$ is the FE output signal or fault estimation defined in (3) and J is a threshold to be tuned.

5.1. Initial analysis

Let us define the following three indices to evaluate the performance of the FD.

D1. The false alarm rate ϕ, which is the likelihood of detecting a fault when no fault has occurred:

$$\phi = \mathbf{Pr}\{\exists t: \ f(t) = 0, \ |\hat{f}(t)| \geq J\}. \tag{23}$$

A false alarm can occur due to the effect of noise on the estimator.

D2. The minimum detectable fault, f_{min}, which is the smallest amplitude of a step fault f which would trigger $\hat{f}$ above J.

$$f_{min} = \{\min \ k_s: \ |\hat{f}(t \to \infty)| \geq J, \ f(t) = k_s\}. \tag{24}$$

D3. The acknowledgment time t_a, which is the time elapsed between the instant when the fault signal f reaches J if a ramp fault occurred and the instant when the fault estimation $\hat{f}$ reaches J.

$$t_a = \{\min \ t: \ |f(t_0)| \geq J, \ |\hat{f}(t_0 + t)| \geq J\}, \tag{25}$$

assuming t_0 the instant in which the fault reaches J.

In this work we consider Gaussian distributed noise. Then, the fault estimation error is also Gaussian distributed, with variance σ_f (given by (10) or (11)). The false alarm rate is the probability that the fault estimation error is over J due to noise in a fault-free scenario. Therefore, ϕ can be determined as

$$\phi = 2\left(1 - \Phi\left(\frac{J}{\sigma_f}\right)\right), \tag{26}$$

where Φ is the cumulative distribution function of a Gaussian variable.

Under a null error in steady state of the fault estimator, we have

$$f_{min} = J. \tag{27}$$

Both the false alarm rate ϕ and the minimum detectable fault f_{min} depend on the chosen threshold J. Since the fault variance σ_f^2 is independent of the detection mechanism (it has been set through α in the fault estimator), by setting J we can just achieve either a given false alarm rate or a given minimum detectable fault, but not each one independently. A high value for the threshold J reduces the false alarm rate but increases the minimum detectable fault while a lower threshold J reduces the minimum detectable fault but increases the false alarm rate. Then, there is a trade-off between both targets.

In the case of the acknowledgment time, if we consider that the fault estimation reaches the threshold once the fault estimation error is steady state, t_a coincides with the time delay versus ramp fault, t_d, which according with (9) is not related with J. If that is not the case, and the fault estimator has no oscillatory modes, t_a would be lower than t_d. Then, t_d is an appropriate measure for the worst-case acknowledgment time t_a, and it has no relationship with the threshold J.

5.2. Fault Detection Mechanism design

The design of the FDM consists in choosing an appropriate value for the threshold J. We can follow two strategies:

E1. To set the desired minimum detectable fault, f^*_{min}, so we set

$$J = f^*_{min}. \tag{28}$$

Therefore, the false alarm rate ϕ is given by (26).

E2. To set the desired false alarm rate, ϕ^*. Considering (26), the minimum detectable fault f_{min} can be determined as the quantile to get the confidence interval of level $(1 - \phi/2)$, so

$$J = \Phi^{-1}\left(1 - \frac{\phi^*}{2}\right)\sigma_f. \tag{29}$$

where Φ^{-1} is the inverse cumulative distribution function of a Gaussian variable. Therefore, the minimum detectable fault is equal to the threshold J.

As we said before, acknowledgment time t_a is independent from the threshold J, so achieving a certain t_a is inherent to the FE design and cannot be done in the FDM design.

5.3. Fault estimation and detection co-design

As shown in previous sections, by setting α we can achieve either a determined acknowledgment time t_a or a determined variance of the fault estimation σ_f. With the chosen α, we can set the detection threshold J to achieve either a determined minimum detectable fault f_{min} or a determined false alarm rate ϕ. However, after setting α we cannot change the acknowledgment time t_a by setting the threshold J. Therefore, if we want to tune α and J

to achieve targets considering the acknowledgment time, the minimum detectable fault and the false alarm rate, we have to design the FE and the FDM at the same time, in a parallel design or co-design where

$$\begin{aligned} \phi &= f(\alpha, J), \\ f_{min} &= f(J), \\ t_a &= f(\alpha). \end{aligned} \tag{30}$$

Since we have three targets (ϕ, f_{min}, t_a) and only two decision variables (α, J) it is not feasible to achieve all the targets. Hence, the co-design strategies proposed consist in setting α and J to achieve two targets and then the third one is given by the relations defined in previous sections. We present a summary of the strategies in the table 2.

If we want to bound the three performance indices, we must use also the multiplicity index $n \geq d$ as a decision parameter and, given a sampling frequency ω_s, look for values (n, α, J) such that the following set of inequalities hold:

$$\begin{aligned} \phi &\leq f(n, \alpha, J), \\ f_{min} &\leq f(J), \\ t_a &\leq f(n, \alpha). \end{aligned} \tag{31}$$

The cumulative distribution of a Gaussian variable is easily found in literature, but we provide here its value for some reasonable values of ϕ^*:

$\boldsymbol{\phi^*}$	10^{-3}	10^{-6}	10^{-9}	10^{-12}
$\boldsymbol{\Phi^{-1}\left(1-\frac{\phi^*}{2}\right)}$	3.29	4.89	6.11	7.13

Table 1: Inverse cumulative distribution of Gaussian variables

6. Experimental approach

In the previous sections, we have shown how to tune and implement a Fault Detector for all kind of processes no matter the model complexity. However, in process industry it is popular to use data-based techniques to get a FOTD (First Order and Time Delay) model of a process, from simple experiments like input step tests. Processes use to be non-linear, coupled with other processes and the sensors add measurement noise, so it is often difficult to find a model of higher complexity that fits well the experimental data. Usually, there is not a big fitting improvement by using models that are more complicated. Another motivation is that most of the techniques for controller tuning use FOTD models and industry workers are used to manage them. In this section, we adapt the techniques explained before to the case of FOTD models that may include modeling errors, and we present solutions for the inherent drawbacks of this approach.

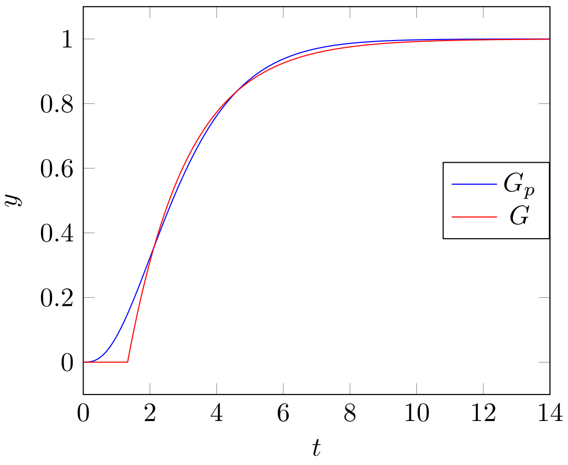

Figure 4: Step response of the system and its FOTD approximation.

Let us assume that we have a FOTD model of the process,

$$G(s) = \frac{K\,e^{-Ts}}{1+\tau s}. \tag{32}$$

In order to apply directly the ideas of the previous sections to this specific FOTD model, we have simplified the formulas in table 2, shown in table 3, giving the rules to tune α and J.

The use of a FOTD model obtained by identification with a step signal around a given operation point tends to neglect the high frequency modes of the process, which depend on non-dominant poles and zeros. It also neglects the potential gain non-linearity due to the use of incremental variables during the identification process. In the following, we will give an insight of the problems derived from applying directly this rules to a FOTD model that simplifies the real dynamics of the system, and we will propose some solutions to these problems.

Example 2. *In order to illustrate the problems caused by the model mismatch and the effect of the proposed solutions, we use in this section a numeric example with a process with the next transfer function for the plant model:*

$$G_p(s) = \frac{1}{(1+s)^3}. \tag{33}$$

We assume that we have a FOTD model for this plant given by

$$G(s) = \frac{1}{(1+1.8s)}e^{-1.33s}. \tag{34}$$

Figure 4 compares the step response of the process $G_p(s)$ and the FOTD approximation $G(s)$.

6.1. Fault estimation

Following the initial proposal in this work, the fault estimation is given by

$$\hat{f}(s) = H_u(s)u(s) + H_y(s)y(s). \tag{35}$$

Table 2: FD design strategies for $n = d$

Targets	$\boldsymbol{\alpha}$	$\boldsymbol{J}$	**Resulting free index**
ϕ^*, f^*_{min}	$\sqrt[2n]{\frac{\sigma_v^2}{K^2 f^*_{min}{}^2}\left(\frac{\prod\tau_i}{\prod\beta_i}\right)^2\Phi^{-1}\left(1-\frac{\phi^*}{2}\right)^2}$	f^*_{min}	$t_a = n\,\alpha + \sum\delta_i + T$
ϕ^*, t^*_a	$\frac{t^*_a-\sum\delta_i-T}{n}$	$\frac{\sigma_v}{K\left(\frac{t^*_a-\sum\delta_i-T}{n}\right)^n}\frac{\prod\tau_i}{\prod\beta_i}\Phi^{-1}\left(1-\frac{\phi^*}{2}\right)$	$f_{min} = J$
f^*_{min}, t^*_a	$\frac{t^*_a-\sum\delta_i-T}{n}$	f^*_{min}	$\phi = 2\left(1-\Phi\left(\frac{K\alpha^n J}{\sigma_v}\frac{\prod\beta_i}{\prod\tau_i}\right)\right)$

Table 3: Simplified FD design strategies in the FOTD case

Targets	$\boldsymbol{\alpha}$	$\boldsymbol{J}$	**Resulting free index**
ϕ^*, f^*_{min}	$\frac{\sigma_v\tau\Phi^{-1}\left(1-\frac{\phi^*}{2}\right)}{Kf^*_{min}}$	f^*_{min}	$t_a = \alpha + T$
ϕ^*, t^*_a	$t^*_a - T$	$\frac{\sigma_v\tau\Phi^{-1}\left(1-\frac{\phi^*}{2}\right)}{K(t^*_a-T)}$	$f_{min} = J$
f^*_{min}, t^*_a	$t^*_a - T$	f^*_{min}	$\phi = 2\left(1-\Phi\left(\frac{K\alpha J}{\sigma_v\tau}\right)\right)$

According to Section 3 if we use $n = d = 1$, the FE transfer functions are

$$H_y(s) = \frac{1+\tau s}{K(1+\alpha s)}, \quad H_u(s) = \frac{-e^{-Ts}}{1+\alpha s}, \tag{36}$$

which can be obtained from (6), (7) and (32).

The fault estimation error, $\tilde{f}(s) = f(s) - \hat{f}(s)$, considering the model plant $G_p(s)$, is

$$\begin{aligned} \tilde{f}(s) &= [1 - H_y(s)G_p(s)]\,f(s) - \\ &- [H_u(s) + H_y(s)G_p(s)]\,u(s) - \\ &- H_y(s)v(s). \end{aligned} \tag{37}$$

With the defined structure for the fault estimator we can compute the indicators **A**1 to **A**5 as a function of the parameter α and model mismatch w.r.t. model plant $G_p(s)$. First, we present the results assuming no modeling error (i.e., $G(s) = G_p(s)$):

A1. step in f: $\lim_{t\to\infty}\tilde{f}(t) = 0$

A2. step in f: $\lim_{t\to\infty}\int_0^t \tilde{f}(t)dt = \alpha + T$.

A3. ramp in f: $\lim_{t\to\infty}\tilde{f}(t)dt = \alpha + T$

A4. high frequency noise in v: $\lim_{s\to\infty} -H_y(s) = \frac{-\tau}{\alpha K}$.

A5. sensitivity to u: $\|H_u(s) + H_y(s)G_p(s)\|_\infty = 0$.

These results only particularize those obtained in 2. If the model does not fit the model dynamics (i.e., $G(s) \neq G_p(s)$), then, the previous indicators result as follows:

A1. When the model gain K is different from the process gain (let us call it K_p), the steady-state fault estimation error versus step fault will have a bias given by

$$\lim_{t\to\infty}\tilde{f}(t) = 1 - \frac{K_p}{K}.$$

Then, if K mismatches K_p, we expect a bias in the steady state of $\hat{f}$ w.r.t f.

A2. If K differs from the process gain K_p, the cumulative fault estimation error versus step fault will be infinite:

$$\begin{aligned} &\lim_{t\to\infty}\int_0^t \tilde{f}(t)dt = \\ &= \lim_{s\to 0}\left(1 - \frac{1+\tau s}{1+\alpha s}\frac{1}{K}G_p(s)\right)\frac{1}{s} = \infty. \end{aligned}$$

A3. The steady-state fault estimation error versus ramp fault coincides with the previous item since the ramp fault is the integral of the step fault, being also infinite under gain mismatch.

A4. The noise effect depends only on the selected $H_y(s)$ but not on the process transfer function, and therefore:

$$\lim_{s\to\infty} -H_y(s) = \frac{-\tau}{\alpha\, K}$$

A5. If we consider some modeling error, $G(s) \neq G_p(s)$. Then, $H_u(s) + H_y(s)G_p(s) \neq 0$, i.e., the sensitivity w.r.t. input u is not null and it is given by

$$\frac{\hat{f}(s)}{u(s)} = \frac{-e^{-Ts}}{1+\alpha s} + \frac{1+\tau s}{K(1+\alpha s)} G_p(s) \neq 0.$$

Considering this, we expect some effect in $\hat{f}$ due to changes in u both in transient response and in steady state. In fault-free scenarios, the estimated fault in steady state due to constant inputs of size $\bar{u}$ will be

$$\lim_{t\to\infty} \hat{f}(t) = \left(-1 + \frac{K_p}{K}\right)\bar{u}$$

To summarize, there are two main drawbacks when we consider a FOTD model for the FE in real systems:

- Biased fault estimation $\hat{f}$ w.r.t. both fault f and process input u.
- Transient effect in the fault estimation $\hat{f}$ due to process input u changes.

Example 3. *We define for the system* (34) *the following fault estimation transfer functions*

$$H_y(s) = \frac{1+1.8s}{1+s}, \qquad H_u(s) = \frac{-e^{-1.33s}}{1+s},$$

where we have set $\alpha = 1$. We assume in this example that the system has a gain $G_p(0) = 0.97$, different from the initial one. Figure 5 illustrates the effects of model simplification and gain mismatch on the example. We see the transient response of the fault estimation $\hat{f}$ due to a unitary step change in u, and also a bias in the steady state. We also appreciate a bias in the fault estimation $\hat{f}$ steady state after a unitary step fault f occurs.

The bias on the fault estimator does not encourage us to directly use fault detection mechanism (22) since it requires an unbiased fault estimation to fulfill its properties about minimum detectable fault and false alarm rate. We propose the following changes to correct these problems:

- In order to overcome the steady state estimation error, we present a fault estimator for incipient faults, that is, a fault estimator that tries to estimate the high frequencies of the faults but does not detect the low frequencies, i.e., the constant value of the faults. The estimator tries to avoid the bias when there is no fault on the system and tries to estimate properly only the initial time when the fault occurs. This does not solve the complete fault estimation, but enables posterior use of fault detection mechanisms.

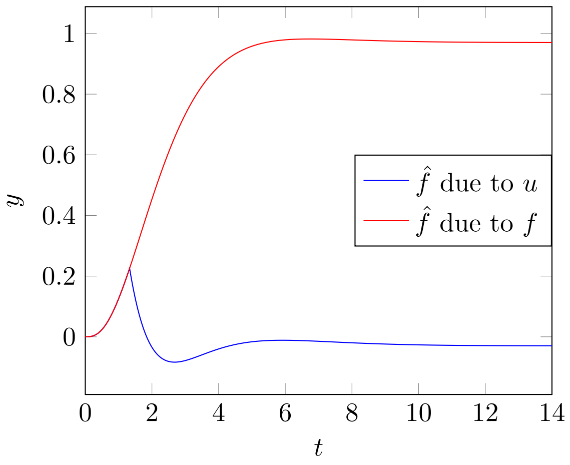

Figure 5: FOTD model drawbacks due to model mismatch under step inputs and step faults.

- In order to overcome the transient error under input changes, we propose the use of the incipient fault estimator together with an adaptive threshold that depends on the control action.

In the following subsections, we address these two issues.

6.2. Incipient Fault estimation

The model mismatch in the low frequency range due to non-linear gain implies a steady state error in the fault estimator in absence of fault, as we showed before. We have this problem even in linear systems because models only behave linear for incremental variables around an operating point.

To avoid this, we propose to filter the initial fault estimate through a washout filter H_w, i.e. a high pass filter that rejects steady state inputs, and we define as $\hat{f}^+$ its output signal. This signal will estimate the initial transients of the fault signal but it will ignore its steady state value, so it will be appropriate for fault detection issues. We propose for the washout filter the next transfer function, including a derivative and a relatively slow pole:

$$H_w(s) = \frac{\tau_w s}{1+\tau_w s}. \tag{38}$$

Then, the incipient fault estimator is given by

$$\hat{f}^+(s) = H_w(s)\hat{f}(s). \tag{39}$$

We define the parameter τ_w in order to keep the dominant frequencies of the fault estimator. The approximate dynamics of the fault estimator is

$$\frac{\hat{f}(s)}{f(s)} = G_p(s)H_y(s) \approx G(s)H_y(s) = \frac{e^{-Ts}}{1+\alpha s}. \tag{40}$$

We choose the washout filter time constant τ_w to fulfill $\tau_w \gg \alpha + T$ in order to guarantee that the main dynamics of the fault estimator are still present in the incipient fault

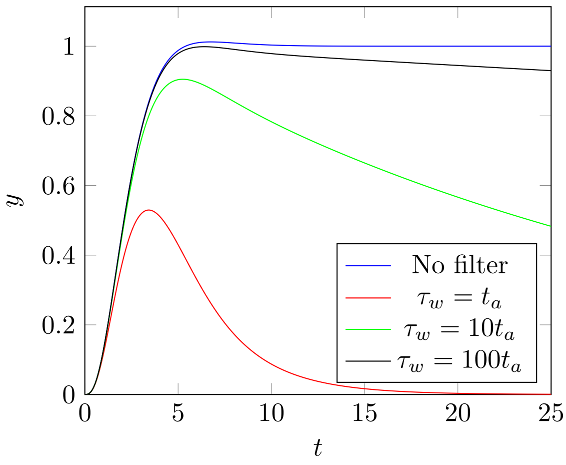

Figure 6: Fault estimation response after the washout filter for different filter parameter τ_w values.

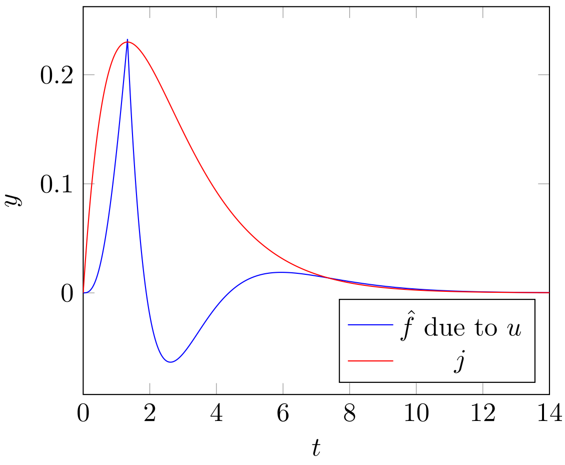

Figure 7: Response of $\hat{f}(t)$ under u step and envelope j.

estimator. However, τ_w is also related to the time needed to reject the bias in fault-free scenarios, and, therefore, it is the initialization time too. Therefore, we also choose τ_w small enough to obtain a reasonable initialization time.

A reasonable initial tuning value is $\tau_w = 10(\alpha + T)$, but the performance should be checked in each particular application.

Note that the high frequency response of $H_w(s)$ is a unitary gain, i.e., $H_w(\infty) = 1$, so the high frequency noise amplification is the same for $\hat{f^+}$ and $\hat{f}$. Adding $H_w(s)$ in the FE structure has no effect in noise amplification.

Example 4. *Figure 6 shows the fault incipient estimation response under unitary step faults for different values of the washout filter parameter τ_w, where $t_a = T + \alpha = 2.33$.*

6.3. Adaptive threshold

The model mismatch in the high frequency range generates a transient in the error response from input changes. To solve this issue, we propose an adaptive threshold for the Fault Detection Mechanism. The new threshold is the sum of the static threshold J and a new signal j that depends on u. This signal j is generated by the transfer function H_j and the input u.

$H_j(s)$ must have zero static gain and its response versus u has to be an envelope for $\hat{f}$. First, the transfer function response has to fit the peak of $\hat{f}(s)$ under a step in u. Next, the response must have a smooth decrease. With these requirements, we propose

$$j(s) = H_j(s)u(s), \tag{41}$$

where

$$H_j(s) = \frac{k_j s}{(1 + \tau_j s)^2}. \tag{42}$$

The effect of u in $\hat{f}$ cannot be predicted before implementing the FD because it depends on the model mismatch. The proposed strategy then consists in performing some steps in u once the initial FE is implemented and then measure the peak (p_j) of the resulting transient in $\hat{f}$ as well as the time (t_j) at which this peak appears. Then, we have to set k_j and τ_j to get a similar peak in the step response of $H_j(s)$. The following expressions give an initial guess for those values:

$$\begin{aligned} \tau_j &= t_j, \\ k_j &= p_j \, \tau_j \, e, \end{aligned} \tag{43}$$

being e the basis of the natural logarithm. These expressions have been tested with several processes and model mismatches and usually give a good starting point. Other envelope transfer functions can be used for this proposal.

Example 5. *In the example we show the drawbacks of the initial approach. We see in figure 7 that the peak under an step in the input is achieved at instant $t_j = 1.33$ (this coincides with the delay of the FOTD model) with a peak value $p_j = 0.23$. Then, we propose*

$$H_j(s) = \frac{0.8315s}{(1 + 1.33s)^2}.$$

In figure 7 we compare the transient response of the fault estimator under an input step and the response of the envelope function. Note that the envelope propose is to increase the threshold during transient changes in the input and it is added to the static threshold J. Even if the envelope response have some values slightly below the transient response, as the ones observed from instants $t = 8$ to $t = 10$, the static threshold J is able to avoid false alarms.

In summary, the new threshold of the FDM is

$$J'(t) = J + |j(t)|. \tag{44}$$

The modified FD proposed for implementation, including the washout filter and the adaptive threshold detailed before, is shown in Figure 8.

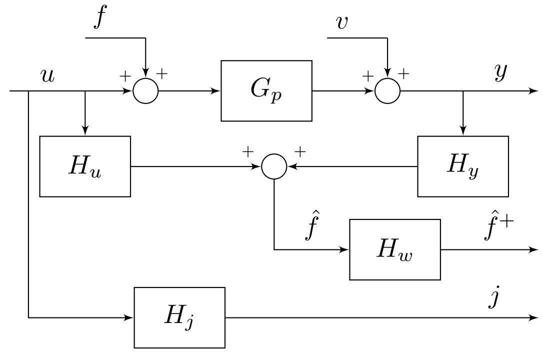

Figure 8: System with Fault Estimator, including H_j and H_w

The resulting Fault Detection Mechanism will be

$$\begin{cases} \text{if } |\hat{f}^+(t)| > J + |j(t)|, & \text{Fault} \\ \text{otherwise,} & \text{No Fault.} \end{cases}$$

Since

$$\frac{\hat{f}^+(s)}{f(s)} = H_w(s)H_y(s)G_p(s) \tag{45}$$

and $H_w(s)$ includes a derivator, the steady state value of $\hat{f}^+(s)$ versus a ramp fault with slope r is

$$\lim_{t\to\infty} r \int_0^t \hat{f}^+(t)dt = \lim_{s\to 0} H_w(s)H_y(s)G_p(s)\frac{r}{s} = r\ \tau_w.$$

In order to ensure the fault detection versus ramp faults, we have to guarantee

$$r_{min}\ \tau_w \geq f_{min}, \tag{46}$$

being r_{min} the minimum fault slope detectable by the FD and f_{min} the value obtained from threshold J and FAR. This is a drawback because, if the slope of the fault signal is lower than that, the FD will not detect the fault. This has to be considered carefully during the tuning of τ_w.

7. Experimental results

This section presents an example of the proposed FD in a real process, depicted in the figure 9. The system consists of two water tanks. The pump *P-1* boosts water from the feed tank to the other, through a pipeline where the valve *V-1* can regulate the flow. The pump has a driver so we can regulate its speed. The higher tank has a level transducer, *L-1*. There is a drain pipeline with a valve, *V-2*, to return water to the feed tank. At the end of this pipeline there is a connection, *C-1*, which reduces the flow. We have configured this in the laboratory for testing the proposed FD.

We consider this process as a SISO system. The input u is the duty cycle of the pulse-width modulated signal applied to the DC regulator which drives the pump, in the range $[0, 100]$ %. The applied voltage to the pump is then

$$\frac{u}{100} V_{cc}, \tag{47}$$

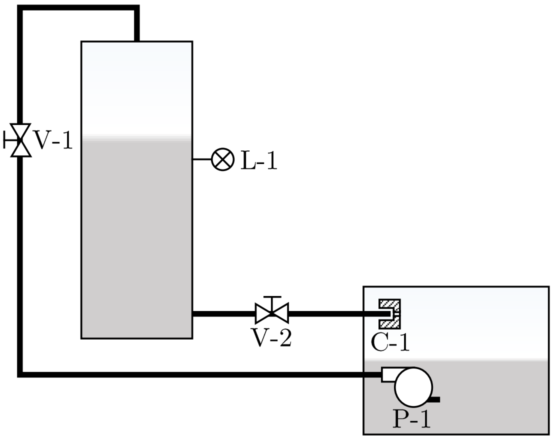

Figure 9: Process for FD testing

being V_{cc} the voltage of the DC regulator. Initially that voltage is set to $V_{cc} = 15$ V. The output y is the tank level *L-1*, in the range $[0, 50]$ *cm*. The valves *V-1* and *V-2* are initially fully open and the connection *C-1* is properly installed.

We are going to consider and test these faults:

- f_1: *V-1* valve small closure, around 20% of its range.
- f_2: *V-1* valve big closure, around 50% of its range.
- f_3: *C-1* connection removal.
- f_4: *V-2* valve small closure, around 20% of its range.
- f_5: *V-2* valve big closure, around 50% of its range.
- f_6: V_{cc} increase in *P-1* driver, from $15V$ to $16V$.

7.1. System identification

First, we need to get a model for the process. With the system in steady state, we have done a step test by suddenly reducing u by 10%. We have identified the output response approximating it to a FOTD model using the SK method described in the Appendix A, as shown in figure 10. The process model, with time measured in seconds, is

$$G(s) = \frac{1.19}{1 + 132.80s} e^{-35s}. \tag{48}$$

7.2. FD tuning

In this example we require a FD with $\phi^* = 10^{-9}$ and $f^*_{min} = 5\%$. From the identification data, we have obtained $\sigma_v^2 = 0.0033$. With the formulas in the table 2, we can easily get the proper tuning for the FD, $\alpha = 7.78$, leading to

$$H_u(s) = \frac{-e^{-35s}}{(1 + 7.78s)}, \tag{49}$$

$$H_y(s) = \frac{1 + 132.80s}{1.19(1 + 7.78s)}, \tag{50}$$

Figure 11 shows the direct application of the Fault Estimator, after some changes on u have also been applied.

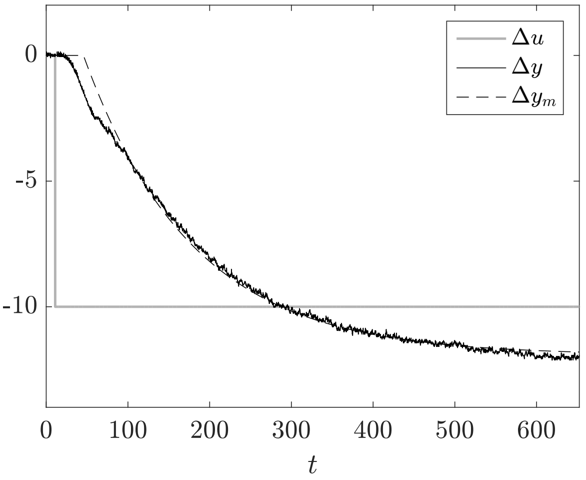

Figure 10: SK method identification. Δy_m is the step response of the FOTD identified model.

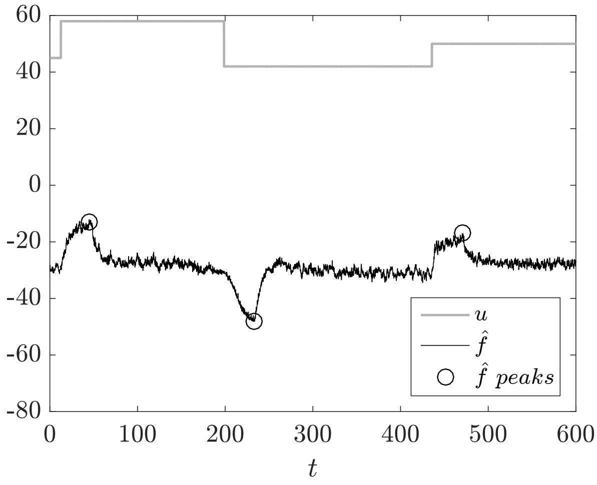

Figure 11: Offset in $\hat{f}$ due to a different operating point. Transient response in $\hat{f}$ due to changes in u.

7.3. Washout filter and adaptive threshold tuning

We can see in figure 11 that $\hat{f}$ has an offset: it is not laying around 0. This is due to the operating point at which the system has been identified. We can appreciate also that the steady state value is not the same over the time, it changes depending on the current value of u. For avoiding this effect caused by process non-linearity, we can include the washout filter H_w explained above. For the tuning of H_w, we propose to set a pole ten times slower than the acknowledgment time t_a, so

$$\tau_w = 10\ t_a = 10\ (\alpha + T) = 428. \tag{51}$$

With this, we expect a negligile effect of H_w in the main frequencies of G but in the long term we are able remove the effect of the non-linearity due to the operating point. Then,

$$H_w(s) \quad = \quad \frac{428s}{1+428s}. \tag{52}$$

On the other hand, due to model mismatches, we expect some sensitivity between the system input and the fault estimation signal. As explained above, we propose to implement an adaptive detection threshold considering the effect of the input. After implementing H_u and H_y we have made some steps in u to measure the effect in $\hat{f}$. The peak p_w of the transient response requires attention since it is necessary for the adaptive threshold tuning. Figure 11 shows the result of this experiment.

Being conservative, we should tune H_j taking into account the highest transient peak and the smallest peak time. This fulfills for the third transient, with peak value $p_w = \Delta\hat{f}/\Delta u = 1.8$ and peak time, since input u changes, $t_p = 35$. From (43),

$$H_j(s) \quad = \quad \frac{171.25s}{(1+35s)^2}, \tag{53}$$

After implementing both H_j and H_w in the configuration shown in Figure 8, we have tested how the FD responds when u changes. In figure 12 we can see how the threshold is able to envelope the transient response of the fault estimation, avoiding false alarms. We can also see that the steady state value of $\hat{f}^+$ is now around 0.

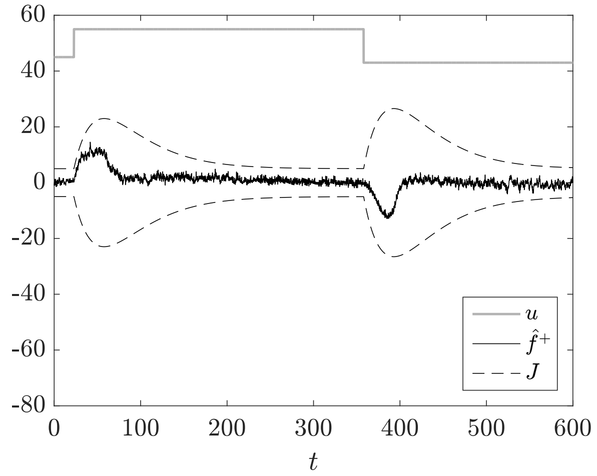

Figure 12: The washout filter and the adaptive threshold are able to avoid the estimation offset and false alarms due to changes in u.

7.4. Faults test

Figure 13 shows the results of the faults test from f_1 to f_6 as described above. All of them have been successfully detected since $\hat{f}^+$ has exceeded the threshold limits. It is shown that the acknowledgment time t_a is smaller when the fault is bigger. The fault signal f in the figure 13 has been deducted *a posteriori* using the steady state of $\hat{f}^+$ and it is just for illustration purpose; it is not possible to know the actual fault signal value. During the test, we have measured the occurrence time of the faults. The washout filter ensures non-biased $\hat{f}^+$ and is slow enough to not miss the fault detection.

The test of the last fault, f_6, is especially illustrative. The fault estimation signal $\hat{f}^+$ is in actuator units, in the range $[0, 100]$ %. This fault has been produced by increasing $1V$ the DC driver voltage from $15V$ to $16V$, which is equivalent to an increase of 6.6% in the duty cycle u keeping the original voltage, and we can see how $\hat{f}^+$ matches this value.

The effect of changes in u are significant considering its range and this limits the performance of the proposed FD, getting a wide adaptive threshold. Nevertheless, we should realize that in most industrial applications control loops are tuned so that actuators respond slowly to avoid disturbances in other processes that are commonly coupled. In that case, the adaptive threshold will be narrower.

7.5. Results comparison with other methods

In this section we want to compare our previous results with those obtained by implementing a PI state observer for the fault estimator, as described in section 4.3. Our motivation is to show that our method has some advantages and that our design procedure is simpler.

First we need an state space representation of the model (48) with no time delays. For that, we use the Padé approximation which substitutes the time delay by a number of poles and zeros. We use a first order approximation (one pole and one zero) because higher order approximations led to oscillatory observer response.

Then, we have applied pole placement to design an observer with all the poles in $-1/\alpha$ with $\alpha = 7.78$. The first disadvantage of this method is that pole placement is done by trial and error and the fault estimation performance is checked after the implementation. We have finally placed the poles in the same values that in our procedure. There is not a procedure to relate the pole placement with fulfilling some false alarm rate or minimum detectable fault, as we do in our method with the formulas in tables 2 and 3.

Therefore, the previous pole placement technique gives us the following transfer functions for the fault estimator, after following the explanation in section 4.3,

$$H_u(s) = \frac{-(1 - 17.5s)}{(1 + 7.78s)^3}$$

$$H_y(s) = \frac{0.84034(1 + 17.5s)(1 + 132.8s)}{(1 + 7.78s)^3}.$$

Due to the fault integrator and the Padé approximation, the order of the observer is now 3 instead of 1. Figure 14 shows the results for the same fault test detailed previously. We can observe that the higher order of the observer implies a more filtered fault estimation signal, and therefore a slower detection time. We have tried to reduce the value of α in the pole placement but the observer behavior presented oscillations. With this procedure (delay approximation plus PI observer design with pole placement) we have a trade-off between the order of the Padé approximation and the placement of the poles. This trade-off not only involves the fault detector behavior (time until detection, minimum detectable fault or false alarm rate) but also the oscillations derived from the use of an approximation for the time delay. Our method tackles directly the time delay and, thus, we do not have an extra source of model inaccuracy.

8. Conclusion

We have presented along this work a novel approach for fault detectors design and implementation in the case of actuator faults. We have proposed a Fault Estimator composed of two transfer functions which include information of the process input-output model and two tuning parameters for achieving objectives based on the fault tracking error and the measurement noise effect. The tuning parameters on the transfer functions are the time constant α and its multiplicity n. Then, we have included a Fault Detection Mechanism that compares the fault estimation with a threshold. The setting of this threshold J as unique tuning parameter for the FDM fulfills objectives based in the false alarm rate, the minimum detectable fault and the acknowledgment time. We show that for considering all the objectives we need to design altogether the FE and the FDM (called Fault Detector co-design). This consists in the tuning of the three free parameters of the FD (α, n and J). Furthermore, we have stated the simpler case in which the multiplicity n is equal to the relative degree d. We have proposed simple tuning strategies through straightforward formulas considering the process model and the fulfillment of the desired objectives for the FD. We have explicitly written the formulas for the FOTD case, which is a common approach for modeling in process industry. In that case we have also analyzed the implications of assuming that model, giving solutions for the drawbacks. Last, we have implemented it in a real system recreated in the laboratory and compared it with methods based on state space observers.

The main advantages of the proposed method are described below. It is simpler because it is based in input-output models instead of state space models. The design is based in the performance of three intuitive engineering indices such as false alarm rate, minimum detectable fault and detection time. Our approach tackles directly the time delay in the model. We provide easy tuning rules based in the process model. The procurement of a process model is a well-covered topic and it is used also for controller design, so it is familiar for workers in industry; furthermore, we explicitly address the case of FOTD models, which is the most widely used model in process industry. The implementation technique is straightforward and uses tools available in industrial control systems.

This work establishes a solid base for future work. We pretend to test this approach in real industrial cases and

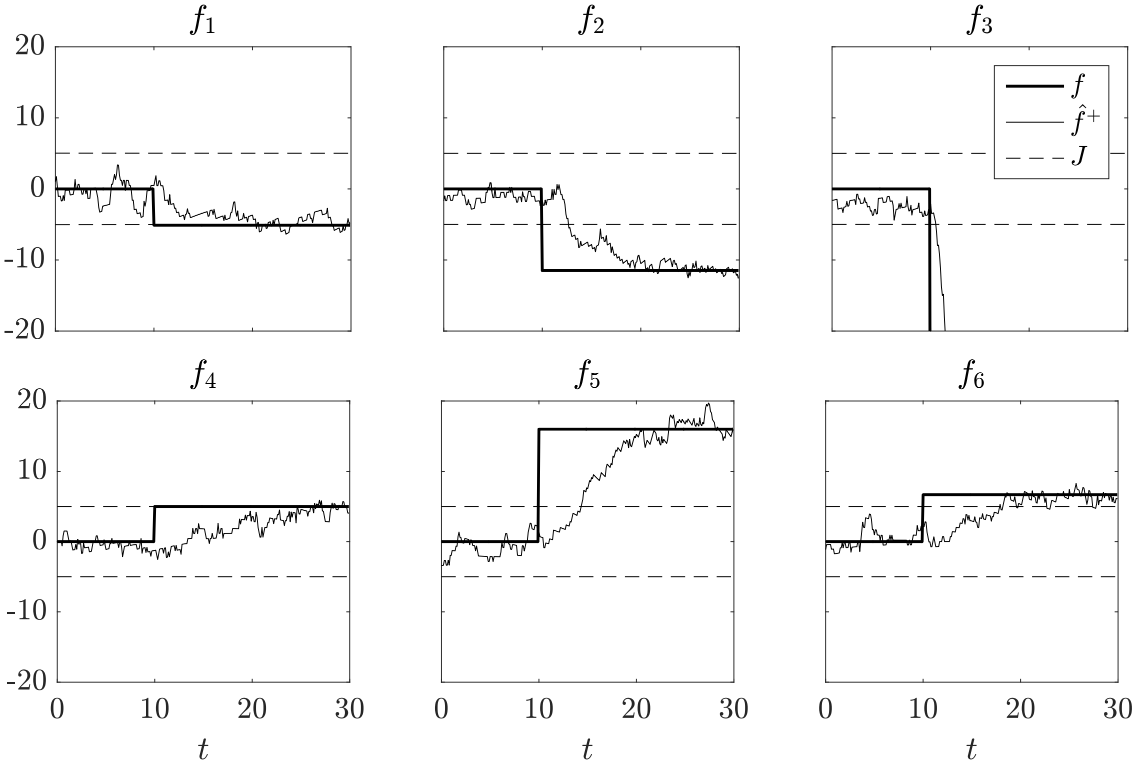

Figure 13: Faults test results with the proposed approach as described in section 7.

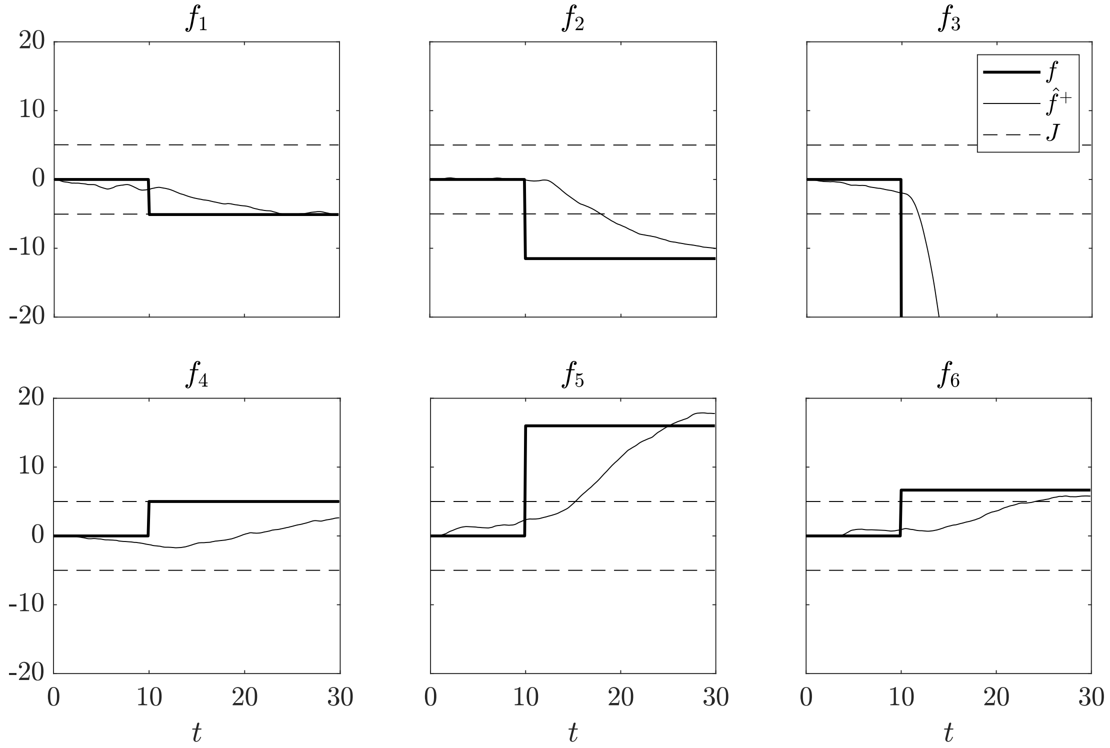

Figure 14: Faults test results with a PI state observer and Padé approximation of the time delay.

to extend the scope to the MIMO case. We will also use this approach in fault tolerant control based in fault estimation.

Appendix A. FOTD identification

For the process identification, due to its simplicity, we propose to follow the SK method [30], which approximates the system to a FOTD model matching the value at 35% and 85% of its step response. This usually gives a good approximation of the process gain and slow dynamics but partially ignoring the high frequency response, which has an impact in the behavior of the fault estimation signal $\hat{f}$ as we said before. For the proposed example (33), our FOTD model will be

$$G(s) = \frac{1}{(1 + 1.8s)} e^{-1.33s}. \tag{A.1}$$

References

[1] Jarinah Mohd Ali, N Ha Hoang, Mohamed Azlan Hussain, and Denis Dochain. Review and classification of recent observers applied in chemical process systems. *Computers & Chemical Engineering*, 76:27–41, 2015.

[2] Hao An, Jianxing Liu, Changhong Wang, and Ligang Wu. Approximate back-stepping fault-tolerant control of the flexible air-breathing hypersonic vehicle. *IEEE/ASME Transactions on Mechatronics*, 21(3):1680–1691, 2015.

[3] Karl Johan Åström and Tore Hägglund. Revisiting the ziegler–nichols step response method for pid control. *Journal of process control*, 14(6):635–650, 2004.

[4] Terrence L Blevins. Pid advances in industrial control. *IFAC Proceedings Volumes*, 45(3):23–28, 2012.

[5] F Caccavale, F Pierri, M Iamarino, and V Tufano. An integrated approach to fault diagnosis for a class of chemical batch processes. *Journal of process control*, 19(5):827–841, 2009.

[6] Jie Chen and Ron J Patton. *Robust model-based fault diagnosis for dynamic systems*, volume 3. Springer Science & Business Media, 2012.

[7] P Cominos and N Munro. Pid controllers: recent tuning methods and design to specification. *IEE Proceedings-Control Theory and Applications*, 149(1):46–53, 2002.

[8] Sourabh Dash, Raghunathan Rengaswamy, and Venkat Venkatasubramanian. Fuzzy-logic based trend classification for fault diagnosis of chemical processes. *Computers & Chemical Engineering*, 27(3):347–362, 2003.

[9] Steven X Ding. *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Springer Science & Business Media, 2008.

[10] Paul M Frank, Steven X Ding, and Teodor Marcu. Model-based fault diagnosis in technical processes. *Transactions of the Institute of Measurement and Control*, 22(1):57–101, 2000.

[11] Rolf Isermann. *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2006.

[12] Manabu Kano and Morimasa Ogawa. The state of the art in chemical process control in japan: Good practice and questionnaire survey. *Journal of Process Control*, 20(9):969–982, 2010.

[13] Atef Khedher, Kamel Benothman, Didier Maquin, and Mohamed Benrejeb. State and sensor faults estimation via a proportional integral observer. In *2009 6th International Multi-Conference on Systems, Signals and Devices*, pages 1–6. IEEE, 2009.

[14] Damien Koenig. Unknown input proportional multiple-integral observer design for linear descriptor systems: application to state and fault estimation. *IEEE Transactions on Automatic control*, 50(2):212–217, 2005.

[15] Salah Laghrouche, Jianxing Liu, Fayez Shakil Ahmed, Mohamed Harmouche, and Maxime Wack. Adaptive second-order sliding mode observer-based fault reconstruction for pem fuel cell air-feed system. *IEEE Transactions on Control Systems Technology*, 23(3):1098–1109, 2014.

[16] Jianglin Lan and Ron J Patton. A new strategy for integration of fault estimation within fault-tolerant control. *Automatica*, 69:48–59, 2016.

[17] Gang Li, S Joe Qin, Yindong Ji, and Donghua Zhou. Reconstruction based fault prognosis for continuous processes. *Control Engineering Practice*, 18(10):1211–1219, 2010.

[18] Xiaohang Li, Hamid Reza Karimi, Yueying Wang, Dunke Lu, and Shenghui Guo. Robust fault estimation and fault-tolerant control for markovian jump systems with general uncertain transition rates. *Journal of the Franklin Institute*, 355(8):3508–3540, 2018.

[19] Yang Liu, Zidong Wang, Xiao He, and Dong-Hua Zhou. Filtering and fault detection for nonlinear systems with polynomial approximation. *Automatica*, 54:348–359, 2015.

[20] Benoıt Marx, Damien Koenig, and Dıdier Georges. Robust fault diagnosis for linear descriptor systems using proportional integral observers. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, volume 1, pages 457–462. IEEE, 2003.

[21] Prashant Mhaskar, Adiwinata Gani, Nael H El-Farra, Charles McFall, Panagiotis D Christofides, and James F Davis. Integrated fault-detection and fault-tolerant control of process systems. *AIChE Journal*, 52(6):2129–2148, 2006.

[22] Jaehong Park and Giorgio Rizzoni. An eigenstructure assignment algorithm for the design of fault detection filters. *IEEE Transactions on Automatic Control*, 39(7):1521–1524, 1994.

[23] RJ Patton and J Chen. Robust fault detection using eigenstructure assignment: A tutorial consideration and some new results. In *Decision and Control, 1991., Proceedings of the 30th IEEE Conference on*, pages 2242–2247. IEEE, 1991.

[24] Francesco Pierri, Gaetano Paviglianiti, Fabrizio Caccavale, and Massimiliano Mattei. Observer-based sensor fault detection and isolation for chemical batch reactors. *Engineering Applications of Artificial Intelligence*, 21(8):1204–1216, 2008.

[25] Ester Sales-Setien, Ignacio Peñarrocha, Daniel Dolz, and Roberto Sanchis. Performance-based design of pi observers for fault diagnosis in lti systems under gaussian noises. In *2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 407–412. IEEE, 2016.

[26] Ester Sales-Setién and Ignacio Peñarrocha-Alós. Multiobjective performance-based designs in fault estimation and isolation for discrete-time systems and its application to wind turbines. *International Journal of Systems Science*, 50(6):1252–1274, 2019.

[27] Ester Sales-Setién and Ignacio Peñarrocha-Alós. Trade-offs on fault estimation via proportional multiple-integral and multiple-resonant observers for discrete-time systems. *IET Control Theory & Applications*, 13(5):659–671, 2019.

[28] Sigurd Skogestad. Simple analytic rules for model reduction and pid controller tuning. *Journal of process control*, 13(4):291–309, 2003.

[29] Oscar AZ Sotomayor and Darci Odloak. Observer-based fault diagnosis in chemical plants. *Chemical Engineering Journal*, 112(1-3):93–108, 2005.

[30] KR Sundaresan and PR Krishnaswamy. Estimation of time delay time constant parameters in time, frequency, and laplace domains. *The Canadian Journal of Chemical Engineering*, 56(2):257–262, 1978.

[31] Venkat Venkatasubramanian, Raghunathan Rengaswamy, Kewen Yin, and Surya N Kavuri. A review of process fault detection and diagnosis: Part i: Quantitative model-based methods. *Computers & chemical engineering*, 27(3):293–311, 2003.

[32] H Wang and S Daley. Actuator fault diagnosis: an adaptive observer-based technique. *IEEE transactions on Automatic Control*, 41(7):1073–1078, 1996.

[33] Jian Liang Wang, Guang-Hong Yang, and Jian Liu. An lmi approach to h-index and mixed h_{-}/h_{∞} fault detection observer

design. *Automatica*, 43(9):1656–1665, 2007.

[34] Kajiro Watanabe, Ichiro Matsuura, Masahiro Abe, Makoto Kubota, and DM Himmelblau. Incipient fault diagnosis of chemical processes via artificial neural networks. *AIChE journal*, 35(11):1803–1812, 1989.

[35] Yunkai Wu, Bin Jiang, and Ningyun Lu. A descriptor system approach for estimation of incipient faults with application to high-speed railway traction devices. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.

[36] T Youssef, Mohammed Chadli, Hamid Reza Karimi, and Rongrong Wang. Actuator and sensor faults estimation based on proportional integral observer for ts fuzzy model. *Journal of the Franklin Institute*, 354(6):2524–2542, 2017.

[37] Dong Yuhua and Yu Datao. Estimation of failure probability of oil and gas transmission pipelines by fuzzy fault tree analysis. *Journal of loss prevention in the process industries*, 18(2):83–88, 2005.

[38] Ke Zhang, Bin Jiang, Peng Shi, and Vincent Cocquempot. *Observer-based fault estimation techniques*. Springer, 2018.

[39] Ping Zhang and Steven X Ding. An integrated trade-off design of observer based fault detection systems. *Automatica*, 44(7):1886–1894, 2008.

[40] Youmin Zhang and Jin Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual reviews in control*, 32(2):229–252, 2008.

[41] Maiying Zhong, Steven X Ding, James Lam, and Haibo Wang. An lmi approach to design robust fault detection filter for uncertain lti systems. *Automatica*, 39(3):543–550, 2003.

[42] David Zumoffen and Marta Basualdo. From large chemical plant data to fault diagnosis integrated to decentralized fault-tolerant control: pulp mill process application. *Industrial & Engineering Chemistry Research*, 47(4):1201–1220, 2008.