New Method of Faults Diagnostic based on Neuro-Dynamic Sliding Mode for Flat Nonlinear Systems

O.Dhaou[1], L.Sidhom[2], A.Abdelkrim[3]
Research Laboratory L.A.R.A in Automatic control, National Engineering School of Tunis
University of Tunis El Manar BP 37, Le Belvédère, 1002[1, 2, 3]
National Engineering School of Carthage ENICarthage
University of Carthage 45, Street of Entrepreneurs, Charguia II, 2035, Tunis, Tunisia[1, 3]

***Abstract*—This paper addresses the problem of simultaneous actuator, process and sensor Fault Detection and Isolation (FDI) for nonlinear system having flatness properties with the presence of disturbances and which are operating in closed-loop. In particular, the nonlinear system is corrupted with additive actuator, process or sensor faults with simultaneous occurrence. In this case, the residual signals might be sensitive to all of these faults that can appear in the system. The proposed FDI method is based on both input and parameter estimators that are designed in parallel. With the flatness property of such system, the design of these two estimators requires information on the measured outputs and their successive derivatives. To estimate these last one, a new scheme of the 2nd-order dynamic sliding mode differentiator is proposed. Residuals are next defined as the difference between the estimated and expected behavior. In order to isolate the faults, dynamic neural networks technique is employed. Besides, comparative study between this new differentiator and the well-known 2nd-order Levant's differentiator is provided to show the pros and cons of the proposed FDI method. This latter is validated by the simulation results and is carried out on a three tank system.**

***Keywords*—Flat system; fault detection and isolation; inputs/parameters estimator; higher order sliding mode differentiator; dynamic neural network**

I. Introduction

Recently, FDI problem has received a great deal of attention especially for systems for which the faults occurrence can lead to irreparable damages. Moreover, the FDI problem is addressed in many applications and is important in safety critical systems such as energy domain [1], chemical domains [2] and industry machines [3]. In the literature, FDI methods can be classified in two approaches: process history based methods and model-based ones. A thorough review of such FDI methods can be viewed via the following papers: [1] [2] [3]. The first approach does not need any knowledge of the process mathematical model. This can be a main interest over the second category of FDI approach. In fact conventional methods of such FDI approach for sensors or actuators can basically be defined by thehardware redundancy. However, its main disadvantage is that it requires knowledge of a large amount of data. Moreover, this method cannot deal with the effects caused by measurement noises. Actually, it cannot proceed in the real time case.

For this approach, there are different methods. In [4] the authors present an FDI method to supervise the aircraft gas turbine engine actuator and sensors. This method uses only the system input/output data. Then, it does not require any prior knowledge of the linear model of the system. It has given satisfactory results, but the problem is with regarding the robustness and accuracy of the fault estimation especially in the practical case.

Other methods have been used FDI for nonlinear systems based on Neural Networks (NN) and there are many research results about this subject. However, the limit of the NN method concerns the updating rate of learning weights especially in facing of sudden faults. To overcome this problem, many solutions are proposed as in [3]. In this latter, the Weightless Neural Networks (WNN) based on random access memory devices method is investigated to carry out the detection and identification per fault groups. Even though satisfactory results obtained by this method without residual generation, the proposed of the training and testing method is very complicated and its implementation is also complex. In general way, the methods of this first approach have many disadvantages, such as the complexity of the structure and the need for many learning and tuning parameters that require in-depth analysis and studies to obtain the optimal selection of the structure. In addition, these methods require a large amount of data for proper training of the adjustable parameters available.

The model-based approaches are defined by the analytical redundancy FDI methods. Much kind of methods have been developed for the case of nonlinear systems [5], [6]. In [5], the problem of sensor faults detection for Unmanned Aerial Vehicle (UAV) system is studied. This method is based on NN observer to detect the presence of faults and the weighting parameters which are updated using kalman filter. This method gives efficient results only for sensors faults in the UAV systems. On the other hand, actuator faults may cause significant impact problem in the UAV control system. That's why, other researchers have proposed a fault detection and isolation method in sensor and actuator faults occurring UAV system [6].

However, there is a major disadvantage to such method that is likely going to lead to false results because it uses simple model for observer's design which is far from reality. Thereby, there are methods based on sliding mode technique which have been strongly used especially in the presence of uncertainties and matched perturbations on the system model. Then, sliding mode observers or differentiators based techniques have been widely considered in FDI methods. For example, in [7], the proposed method is based on multi-models to solve the fault detection and isolation problem for a certain class of nonlinear system. The derivatives of the system's output are calculated by the well-known HOSM Levant's differentiator [8] and the system states are estimated using the HOSM multiple observers. The equivalent output injection of the HOSM multiple observers is used to detect the presence of faults in the system. This method is able to be detected and isolated simple and multiple faults but it has a great drawback consisting in the difficulty implementation especially for the rather complicated systems.

In addition, the result of this method depends on the quality of the derivative signals. In practice, the differentiation operation must provide a best compromise between accuracy and the noise rejection. As well known, the sliding mode differentiator has a major benefit which is known by the simplicity in real-time implementation. But, it presents a main drawback concerning the adjustment of its gains. Indeed, these tuning gains require information of the Lipschitz constant of the derivative signals. So, in the practice case, it is hard to obtain beforehand exact information about the value of this constant. To solve this problem, different new schemes of sliding mode differentiators have been proposed in the literature, let us mention some examples [9], [10], [11].

In this paper, we have investigated a novel model-based fault Diagnosis strategy for a particular class of nonlinear systems called flat ones [12].The proposed method offers a fast performance in on-line closed loop operation and ensures robustness with respect to additive measurement and plant noises. For our research work, the considered flatness property of the model system leads to estimate the successive derivatives using some differentiation scheme. The outputs of this scheme are used to design both inputs and parameters estimators. For this aim, a High Order Sliding Mode Differentiator with Dynamic Gains (SMDDG) defined in [13] is employed to detect faults. Furthermore, we consider a simultaneous occurrence of actuator, process or sensor faults. Thus, the obtained analytic residual signals might be sensitive to all of these faults which make fault isolation very difficult. That's why, the isolation procedure is formulated here as a pattern recognition problem using a Dynamic Neural Networks (DNN) on the residual signals. Comparing with FDI methods in the existing literature, the main contributions of this paper are:

1) A new scheme of adaptive higher order sliding mode differentiator is used to define a new FDI method [13]. Where this algorithm offers a full segregation between the actual faults and measurement noises, which keeps the false alarms rate at a very low level compared to other proposed adaptive differentiators.

2) The fault isolation is realized using an Adaptive Neural Network. In this step, the approach can easily distinguish between the three types of faults that can occur in the system, even though they appeared simultaneously.

3) A comparative study with the well-known Levant's differentiator is proposed to show the performance of the proposed approach.

This paper is organized as follows. The problem is formulated in Section 2. In Section 3 the proposed FDI method is explained. The system application is presented in the next section. The simulations cases are given in Section 5 to discuss the results and to show the effectiveness of the proposed method with different fault scenarios.

II. Problem Formulation

Consider the following form of the nonlinear system:

$$\begin{cases} \dot{x} = f\left(t, x, \theta, \mathrm{F}_{x,\theta}\right) + g(x, F_u)u \\ y = h\left(t, x, \mathrm{F}_y\right) + \eta \end{cases} \tag{1}$$

Where f and h are nonlinear functions; $x = \left(x_1, \ldots, x_n\right)$ $u = \left(u_1, \ldots, u_m\right)$ and $y = \left(y_1, \ldots, y_p\right)$ are the state, the input and the output vectors system respectively. $\theta = \left(\theta_1, \ldots, \theta_r\right)$ is a finite set of unknown parameters which can be a combination of the system's physical parameters. η is a sensor noise considered here by noises on the outputs system. $F_{x,\theta}, F_u, F_y$ are finite set of process, actuators and sensor faults variables respectively. The non-linear systems present unknown parameters of additional faults that can appear at the sensors, actuators or process. Based on Fliess' work [12], for this class of system defined in (1), its input and state vectors can be characterized by a set of variables called flat outputs and their finite number of its successive derivatives. According to [12], the system (1) presents a flatness property if it can be described by the following equations:

$$\begin{cases} u_i = \varphi_{1j}\left(L_f^0 y_1, L_f^1 y_1, \ldots, L_f^k y_1, \ldots, L_f^0 y_m, L_f^1 y_m, \ldots, L_f^k y_q\right); \\ x_j = \varphi_{2j}\left(L_f^0 y_1, L_f^1 y_1, \ldots, L_f^k y_1, \ldots, L_f^0 y_m, L_f^1 y_m, \ldots, L_f^k y_q\right); \end{cases} \tag{2}$$

Where φ_{1j} and φ_{2j} are two nonlinear functions.

$j = \left\{1, \ldots, n\right\}$ and $k = \left\{1, \ldots, q\right\}$, k is the differentiation order where q is the flatness order, m is the input number and $L_f h(.)$ is the Lie derivative defined by $\frac{\partial h}{\partial x} f\left(\cdot\right)$.

Then, considering the flat nonlinear system (2), the aims of this paper are: firstly, detect the presence of faults taking into account of sensor noises. Then, isolate possible acts of faults on the system (actuator, sensor or process). In order to obtain these goals and basing on the flatness property of the system, a generic residual signal is obtained as follows:

$$r_l = \Gamma_l\left(t,[u]_e,[\theta]_e,y,[\dot{y}]_e \ldots,\left[y^{(k)}\right]_e\right) \tag{3}$$

Where $y=\left(y_1,\ldots,y_q\right)$ is the flat output vector, $u=\left(u_1,\ldots,u_m\right)$ is the input vector and $\theta=\left(\theta_1,\ldots,\theta_r\right)$ is the parameter vector.

$[u]_e$ and $[\theta]_e$ are the estimated input and parameter vectors.$\left[y^{(k)}\right]_e$ is the estimated derivatives outputs vector.

Note that the obtained residual expression (3) contains information about any type faults which can affect the system. This residual dealt with two issues. The first one is the choice of the differentiation algorithm where it plays a crucial role having good quality of the estimated signal and consequently the capability of accurate fault detection. The second issue concerns the fault isolation. Indeed, for the residual given by (3), it is difficult to dissociate the different types of fault, besides; this problem becomes more important with the presence of other disturbances. To solve these issues, the proposed FDI method is described in the following section.

III. Proposed FDI Method

To explain the proposed FDI method, Fig. 1 is presented. Fig. 1 is defined by three blocks: control loop, fault detection and isolation blocks. The FDI procedure is based on residual generation signal r_i which is defined by input $[u_i]_e$, parameter $[\theta_r]_e$ estimated vectors respectively and also the flat output vector with their successive derivatives $\left[y^{(k)}\right]_e$. Fig. 1 is illustrated for the case where twoflat outputs are present in the system which explains the use of second order differentiator.

But the proposed approach may well be used in the general case where the system presents q flat outputs. As it is explained above, the choice of the differentiation algorithm is very delicate to give a good quality of the derivatives estimations especially with noisy signal. In the literature, there are different differentiators that have been proposed such as [9] [10] [11]. For instance, there are a HOSM differentiators proposed by Levant in [8]. In this paper, a new scheme of HOSM Levant's differentiator is employed. It is defined by the higher order Sliding Mode Differentiator with Dynamic Gains (SMDDG) proposed in [13].As already indicated, the studied system presents only two flat outputs which limit the order of the differentiation to two. In the following, we start by exposing the sliding mode algorithms.

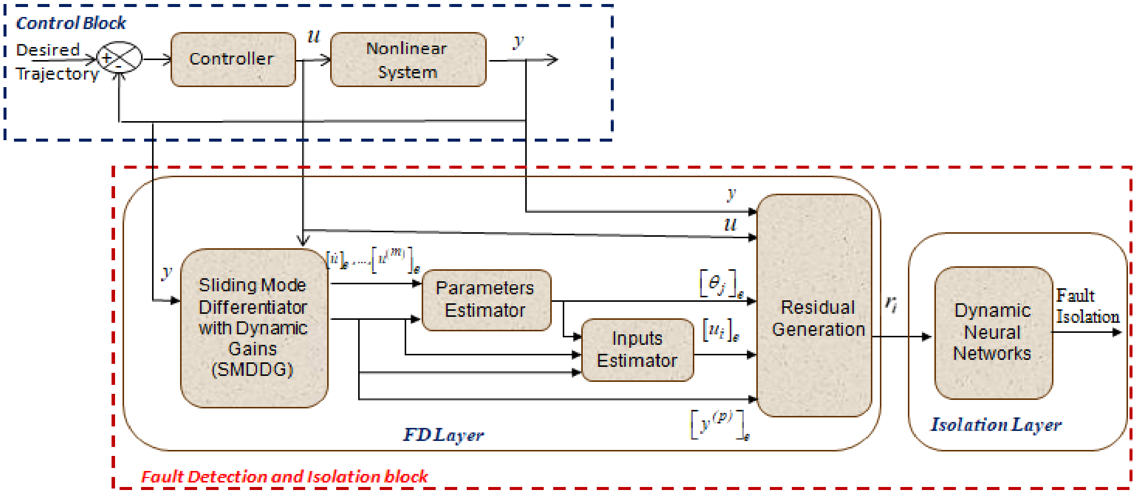


Fig. 1. Block Scheme of Proposed FDI Method.

A. AbbreviationsDifferentiation Algorithms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract.

Considering the input function to differentiate as follows

$$y(t)=y_0(t)+\xi(t) \tag{4}$$

Where $y_0(t)$ is an unknown base signal where the $(\partial+1)^{th}$ derivative is bounded by a positive Lipschitz constant C and $\xi(t)$ is a noise which has only a known upper limit such that $|\xi(t)|<\varepsilon$. Where ε very much less than 1 and $|y(t)-y_0(t)| \le \varepsilon$.

1) HOSM *differentiator:* The dynamic equations of the second order differentiator proposed by Levant are as follows:

$$\begin{cases} \dot{z}_0=\delta_0 \\ \delta_0=-\lambda_0\left|z_0-y\right|^{\frac{2}{3}} sign\left(z_0-y\right)+z_1 \\ \dot{z}_1=\delta_1 \\ \delta_1=-\lambda_1\left|z_1-\delta_0\right|^{\frac{1}{2}} sign\left(z_1-\delta_0\right)+z_1 \\ \dot{z}_2=-\lambda_2 sign\left(z_1-\delta_0\right) \end{cases} \tag{5}$$

Note that $s_0 = z_0 - y(t),\ \ s_1 = z_1 - \delta_0$ are the two sliding surfaces of the differentiator which define the derivatives estimation errors. Theoretically and after a convergence time, we have two outputs: $\delta_0 = \dot{y}(t),\ \ s_1 = \ddot{y}(t)$. In [11] the authors give the inequalities that are established in finite time, for a positive constant μ_i depending exclusively on the differentiator parameters:

$$\left\{ \left| z_i - y_0^{(i)}(t) \right| \le \mu_i C^{\frac{i}{\partial+1}} \varepsilon^{\left(\frac{\partial-i+1}{\partial+1}\right)}, \quad i = 0,\ldots,\partial \right. \tag{6}$$

From (6) it can be deduced that the precision is degraded with the increase of the derivative order.

2) SMDDG differentiator: The advantage of this new scheme is that its gains are adjusted in real time regardless of the input signal. Considering the input signal defined by (4) and under the same assumption on the noise signal, the second order differentiator with dynamic gains is described by (7):

$$\begin{cases} \dot{z}_0 = \delta_0 \\ \delta_0 = -\hat{\lambda}_0 \left| s_0 \right|^{\frac{2}{3}} sign\left(s_0\right) - K_0 s_0 + z_1 \\ \dot{z}_1 = \delta_1 \\ \delta_1 = -\hat{\lambda}_1 \left| s_1 \right|^{\frac{1}{2}} sign\left(s_1\right) - \hat{\lambda}_2 \int_0^t sign\left(s_1\right) dt - K_1 s_1 \end{cases} \tag{7}$$

Where K_0, K_1 are positive convergences gains and $\hat{\lambda}_0, \hat{\lambda}_1, \hat{\lambda}_2$ are positive dynamic gains of the algorithm computed in real time and depending on the sliding surfaces S_0, S_1. The dynamic equations are defined by (8).

$$\begin{cases} \dot{\hat{\lambda}}_0 = \left[\left| s_0 \right|^{\frac{2}{3}} sign\left(s_0\right) \right] s_0,\ \ \hat{\lambda}_0(0) \ge 0 \ \ and\ \ \dot{\hat{\lambda}}_0 > 0 \ \ \forall\ t > 0 \\ \dot{\hat{\lambda}}_1 = \left[\left| s_1 \right|^{\frac{1}{2}} sign\left(s_1\right) \right] s_1,\ \ \hat{\lambda}_1(0) \ge 0 \ \ and\ \ \dot{\hat{\lambda}}_1 > 0 \ \ \forall\ t > 0 \\ \dot{\hat{\lambda}}_2 = s_1 \int_0^t sign\left(s_1\right) dt \end{cases} \tag{8}$$

With Equations (8), the trajectories of the system (7) converge towards the point $s_0 = s_1 = 0$ under the assumption that existed unknown positive constants λ_0^* and λ_1^* defined by (9):

Assumption:

$$\begin{cases} \dot{y} = -\lambda_0^* \left| s_0 \right|^{\frac{2}{3}} sign\left(s_0\right) + z_1 \\ \ddot{y} = -\lambda_1^* \left| s_1 \right|^{\frac{1}{2}} sign\left(s_1\right) - \lambda_2^* \int_0^t sign\left(s_1\right) dt \end{cases} \tag{9}$$

In the absence of noise, the convergence time of this differentiation algorithm becomes so fast if the tuning of the convergence gains values is high. However, in the case of noisy signals, there is a trade-off between the convergence time and the reduction of the noise amplification. Indeed, the linear terms added on the dynamics equation of the algorithm represent a solution to have more smoothing outputs. Then, in this case, it is necessary not to choose a high value for $K_i,\ i \in \{0,1\}$. For more details see [13].

B. Parameters and Inputs Estimation

Started by the proposed parameter estimator, for this fact, we assume that the system parameters $\theta^T = [\theta_1, \theta_2, \ldots, \theta_r]$ satisfy some identifiability conditions in order to be obtained only from measured data. For this aim the observability test is obtained by rank test condition based on the calculation of the space spanned by gradients of the Lie derivatives of the output functions of the system.

$$rank\left(J_0\right) = rank\left(\left[\frac{\partial}{\partial x_{j1}} L_f^k y_{i1}\right]\right) = n; \tag{10}$$

With $\quad j_1 = \{1,2,\ldots,n\};\ \ k = \{1,2,\ldots,n-1\}\ \ ;i1 = \{1,2,\ldots,p\}$

Where $L_f h(.)$ is the Lie derivative defined by $\frac{\partial h}{\partial x} f(.)$ and J_0 is the Jacobian matrix.

If $rank\left(J_0\right) = n$, so the system is observable. Therefore, by considering parameters as state variables P and have derivatives are equal to zero $\dot{P} = 0$. So, for the described system the parameter identifiability can be treated as a particular case of the observability problem. The rank test of the following matrix is then used to determine the parameters to be identified.

$$rank\left(J_1\right) = rank\left(\left[\frac{\partial}{\partial P_{j2}} L_f^k y_{i1}\right]\right) = n;$$

$$j_2 = \{1,2,\ldots,q\};\ \ k = \{1,2,\ldots,n-1\}\ \ ;i1 = \{1,2,\ldots,p\}. \tag{11}$$

Suppose that the parameters to be identified $\theta^T = [\theta_1, \theta_2, \ldots, \theta_r]$ satisfy the identifiability condition, the estimation of the parameters $[\theta_{\bar{\lambda}}]_e$ could be written by (12)

$$[\theta_{\bar{\lambda}}]_e = \phi_{\bar{\lambda}}\left(t, [y]_e, [\dot{y}]_e, \ldots, \left[y^{(p)}\right]_e, u, [\dot{u}]_e, \ldots, \left[u^{(m)}\right]_e\right)$$
$$with\ \ \bar{\lambda} = \{1,2,\ldots,r\} \tag{12}$$

Where r is the parameters number.

For the input estimator, the estimation of the system's inputs is based on the flatness expression presented in equation (2) and the estimated parameters defined by (12). When including the equations (12) and (2), a relation between the parameter estimations and the system's outputs and their derivatives is defined as follows:

$$[u_i]_e = \gamma\left(t,[y],[\dot{y}]_e,...,\left[y^{(k)}\right]_e,[\theta_1]_e,...,[\theta_r]_e\right)$$
$$with \quad i=\{1,2,...,m\} \qquad (13)$$

Where γ is a nonlinear function, $[y],[\dot{y}]_e,...,\left[y^{(k)}\right]_e$ are the output and their derivatives estimation given using the SMDDG differentiator and $[\theta_1]_e,...,[\theta_r]_e$ are the parameter estimations.

After the estimation of the system's inputs, a comparison between the estimated inputs u_e and their nominal values u_n is performed. The residual signal is then defined as:

$$r_i = u_n - u_e \qquad (14)$$

$$with \quad i=\{1,2,...,m\} \qquad (15)$$

Where Γ_i is a nonlinear function.

The residual signal r_i converge to zero in fault free case meaning that the input estimated $[u]_e$ is equal to the nominal system input $[u]_n$. Besides, r_i makes a deviation in the presence of any anomaly in the system. Note that the residual vector has the same size of the input vector $i=\{1,2,...,m\}$ since it is a result of the difference between the nominal and the estimated inputs vector.

$$r_i = \begin{cases} 0 & [u]_e = [u]_n \\ \neq 0 & [u]_e \neq [u]_n \end{cases} \qquad (16)$$

C. Fault isolation with Neural Network

After the fault has been detected by the proposed FD method, it is necessary to obtain information about it in order to isolate faults. Indeed, it is important to determine which component is faulty in the system and also identify the successive and simultaneous fault cases. This problem is solved by using a Dynamic Neural Network algorithm (DNN). This algorithm is implemented by feed-forward technique explained in Fig. 2. DNN is an adaptive NN with multiple hidden layers of neurons used to model complex non-linear systems. The training consists in giving the proper output according to the corresponding input patterns. For our case, this algorithm is based on the learning method for the residual signal $r=(r_1,r_2,...,r_m)$ obtained by the comparison between the nominal inputs $(u_1,u_2,...,u_m)$ and the estimated ones $(u_{1e},u_{2e},...,u_{me})$.

The DNN's output d_i is trained to present "0" in the fault free case of the system and "1" in case of fault.

Considering the on-line use, the network training is performed for all possible fault scenarios with using a sliding window T_{sw} where $T_{sw}=10T_e$ for the residual signal with a step of $1ms$. So, the signal is transformed into a matrix P as follows:

$$P = \begin{bmatrix} r_i(0) & r_i(T_e) & \ldots & r_i(9T_e) \\ r_i(T_e) & r_i(2T_e) & \ldots & r_i(10T_e) \\ \vdots & \vdots & \ddots & \vdots \\ r_i(\Delta t-9T_e) & r_i(\Delta t-8T_e) & \cdots & r_i(\Delta t) \end{bmatrix},$$

In order to enable the neural network to learn the imposed input-output pattern, the network weights are adjusted by a specific algorithm called back propagation algorithm which is used in [10]. The main purpose is to bring the network function as close as possible to a given function. So, the learning problem consists to find better weight combinations.

However, the function is not given explicitly but implicitly though some examples. Consider a feed-forward network with m input and output. This network can have no fixed number of hidden units and can exhibit any desired feed-forward connection pattern. A training set $\{(r_1,d_1),\ldots,(r_l,dp_{NN})\}$ is given and consists of p_{NN} ordered pairs of m and p_{NN} dimensional input and output patterns vectors. The primitive functions of the networks are considered continuous and differentiable. The initial weights are considered as random numbers. In general, when the input r_i is presented to the network, it produces an output t_i different to from the desired d_i.The aim is to make t_i and d_i identical by using a learning algorithm. So, it is necessary to minimize the error function of the network, this error is defined by (17).

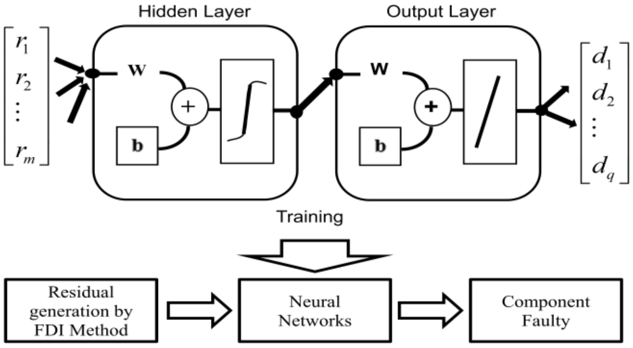


Fig. 2. Neural Network Pattern and Fault Isolation.

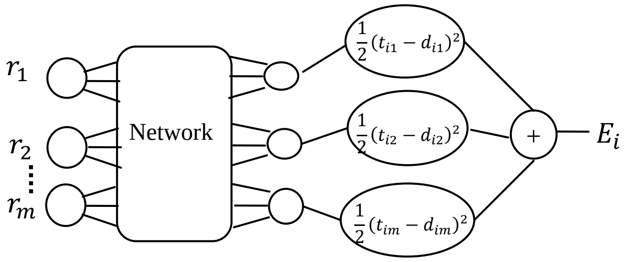

r_1
r_2
r_m
Network
$\frac{1}{2}(t_{i1}-d_{i1})^2$
$\frac{1}{2}(t_{i2}-d_{i2})^2$
$\frac{1}{2}(t_{im}-d_{im})^2$
+
E_i

Fig. 3. Network for the Computation of the Error Function.

$$E = \frac{1}{2}\sum_{i=1}^{p_{NN}} \left\| t_i - d_i \right\|^2 \tag{17}$$

After minimizing this error function, new unknown input patterns will be presented to the network and meanwhile it's interpolate. In order to obtain a local minimum of the error function, a back-propagation algorithm is used. Based on this algorithm, the network is initialized with random weight. After that we correct the initial weight by calculation of the gradient of the error function. Fig. 3 shows the computation technique of the error function by using the back-propagation technique. The minimization process consists at the first on extending the network to computes automatically the error function.

All the j output units of the networks evaluate the function $\frac{1}{2}\left(t_{ij} - d_{ij}\right)^2$. The addition of the m outputs nodes gives E_i as a result. The same network extension has to be built for each pattern d_i. All quadratic errors are added in one sum unit $E_1 + \ldots + E_p$ to define the error function E. Then, the error for a given training set is calculated. In order to make the quadratic error E as low as possible, the weights are the only parameter that can be modified in the network. We now have w network capable of calculating the total error for given training set. E is calculated by the extended network exclusively through composition of the node functions, it is a continuous and differentiable function of the weights $w_1, w_2, \ldots, w_l$ in the network. We can thus minimize E by using an iterative process of gradient descent, for which we need to calculate the gradient.

$\nabla E = \left(\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_l}\right)$. Each weight is updated using the increment $\Delta w_i = -\gamma \frac{\partial E}{\partial w_i} \quad for \quad i = 1, \ldots, l$. Where γ represents a learning constant, i,e, a proportionality parameters which defines the step length of each iteration in the negative gradient direction. With this extension of the original network the whole learning problem now reduces to the question of calculating the gradient of a network function with respect to its weights. Once we have a method to compute this gradient, we can adjust the network weights iteratively. In this way we expect to find a minimum of the error function, where $\nabla E = 0$.

IV. Application to the Three-Tank-System

In order to validate the proposed FDI method, the chosen study system is the three-tank-system shown in Fig. 4. This system consists of three cylindrical tanks with identical section S. The three tanks are interconnected by two cylindrical pipes of section S_p and the outflow coefficients of tank 1 and tank 2 are q_{31} and q_{32} respectively. The liquid flows from the tank 3, with a valve of section S_p which supplies the two pumps P_1 and P_2 collected in the tank R_0.

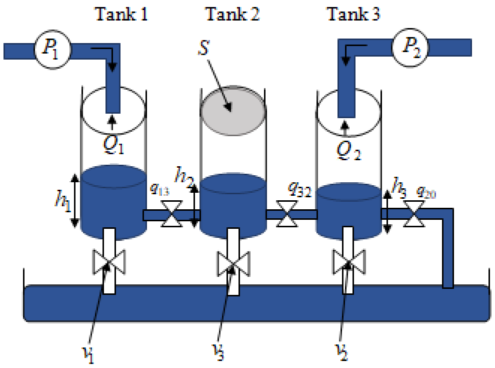


Fig. 4. Three-Tank System.

The two pumps are collected by DC motors, supply the tanks 1 and 2 with flow rates Q_1 and Q_2. The water levels in the three tanks are noted by h_1, h_2 and h_3 respectively and verify the following inequality: $h_1(t) > h_3(t) > h_2(t)$. The three tanks are connected with the tank R_0 by three valves v_1, v_2 and v_3 where their sections are flexible to simulate various failures such as more or less significant leaks on each tank.

The nominal behavior of this system is defined by:

- The valves v_1, v_2 and v_3 must be closed in order to have no leakage rates;
- The fluid is supposed to be perfect. Its characteristics are considered unchanged. So the parameters such as the coefficients of viscosity supposed to be known, certain and have a constant average value;
- The aging and the environment effects on the system's parameters are not considered.

Using the mass balance this system can be presented as follows

$$\begin{cases} S\frac{dh_1}{dt} = Q_1 - q_{13} \\ S\frac{dh_2}{dt} = Q_2 + q_{32} - q_{20} \\ S\frac{dh_3}{dt} = q_{13} - q_{32} \end{cases} \tag{18}$$

$q_{ij};\ i, j = \{1, 2, 3\}\ \ \forall i \neq j$ represent the flow of water flowing from the tank i to j. The expression of the flow is given by: $q_{ij} = \theta_i.S_p.sign\left(h_i - h_j\right).\sqrt{2g\left|h_i - h_j\right|}$ where q_{20} represent the outflow that can be expressed by $q_{20} = \theta_2.S_p.\sqrt{2g\left|h_2\right|}$.For the next, the following notations are considered to facilitate the model exploitation:

$x=(x_1,x_2,x_3)^T=(h_1,h_2,h_3)^T$ and $u=(u_1,u_2)^T=(Q_1,Q_2)^T$. Then the full nonlinear system model is given by

$$\begin{cases}\dot{x}_1=-\dfrac{q_{13}}{S}\sqrt{x_1-x_3}+\dfrac{1}{S}u_1,\\ \dot{x}_2=\dfrac{q_{32}}{S}\sqrt{x_3-x_2}-\dfrac{q_{20}}{S}\sqrt{x_2}+\dfrac{1}{S}u_2,\\ \dot{x}_3=-\dfrac{q_{13}}{S}\sqrt{x_1-x_3}+\dfrac{q_{32}}{S}\sqrt{x_3-x_2},\end{cases}$$

$$\begin{cases}y_1=x_2,\\ y_2=x_3,\end{cases} \qquad (19)$$

This system has two flat outputs defined by y_1 and y_2, see equations (20) and (21).

$$\begin{cases}x_1=y_1\\ x_2=y_2-\left(\dfrac{q_{13}\sqrt{y_1-y_2}-S\dot{y}_2}{q_{32}}\right)^2\\ x_3=y_2\end{cases}$$

$$\begin{cases}u_1=S\dot{y}_1+q_{13}\sqrt{y_1-y_2},\\ u_2=-\dfrac{\dfrac{2S}{q_{32}}\left(q_{13}\sqrt{y_1-y_2}-S\dot{y}_2\right)}{q_{32}}\\ \qquad\times\left(\dfrac{q_{13}(\dot{y}_1-\dot{y}_2)}{2\sqrt{y_1-y_2}}-S\ddot{y}_2\right)\\ \qquad-q_{13}\sqrt{y_1-y_2}\\ \qquad+q_{20}\sqrt{y_2-\left(\dfrac{q_{13}\sqrt{y_1-y_2}-S\dot{y}_2}{q_{32}}\right)^2}\end{cases} \qquad (20)$$

According to (20), the second input u_1 is expressed in terms of these two flat outputs and their successive derivatives but its expression is still complex compared to the first input u_1.

For the parameter estimation of the system, it is necessary to test the identifiability of the latter. So, to obtain information about system's parameters it is necessary to calculate the Jacobean matrix of Lie-derivatives (21):

$$J_I=\begin{bmatrix}\dfrac{\partial}{\partial P_1}L_f y_1 & \dfrac{\partial}{\partial P_2}L_f y_1 & \dfrac{\partial}{\partial P_3}L_f y_1\\ \dfrac{\partial}{\partial P_1}L_f y_2 & \dfrac{\partial}{\partial P_2}L_f y_2 & \dfrac{\partial}{\partial P_3}L_f y_2\\ \dfrac{\partial}{\partial P_1}L_f y_3 & \dfrac{\partial}{\partial P_2}L_f y_3 & \dfrac{\partial}{\partial P_3}L_f y_3\end{bmatrix} \qquad (21)$$

Where $\dfrac{\partial}{\partial P_2}L_f y_1=\dfrac{\partial}{\partial P_3}L_f y_1=\dfrac{\partial}{\partial P_1}L_f y_2=\dfrac{\partial}{\partial P_2}L_f y_3=0$

and $det(J_1)=-\dfrac{\partial}{\partial P_2}L_f y_2.\dfrac{\partial}{\partial P_1}L_f y_1.\dfrac{\partial}{\partial P_3}L_f y_3\neq 0;$ for $x_1>x_3>x_2$.

So, the rank of the matrix (21) is 3 and also $n=3$. Therefore, the system's parameters are identifiable. Then, the observability of the system is justified. Based on equation (19) it can be concluded that the parameter estimation as follows

$$\begin{cases}\left[\theta_1\right]_e=\dfrac{-S\left[\dot{y}_1\right]_e+u_1}{S_p sign\left(y_1-y_3\right)\sqrt{2g\left|y_1-y_3\right|}}\\ \left[\theta_2\right]_e=\dfrac{-S\left(\left[\dot{y}_1\right]_e+\left[\dot{y}_2\right]_e+\left[\dot{y}_3\right]_e\right)-\left(u_1+u_2\right)}{S_p sign\left(y_2\right)\sqrt{2g\left|y_2\right|}}\\ \left[\theta_3\right]_e=\dfrac{-S\left(\left[\dot{y}_1\right]_e+\left[\dot{y}_3\right]_e\right)+u_1}{S_p sign\left(y_3-y_2\right)\sqrt{2g\left|y_3-y_2\right|}}\end{cases} \qquad (22)$$

For the input estimator, it is easy to obtain the estimation of the system's inputs using (20) and (22):

$$\begin{cases}\left[u_1\right]_e=S\left[\dot{y}_1\right]_e+q_{13}\sqrt{y_1-y_2},\\ \left[u_2\right]_e=-\dfrac{\dfrac{2S}{q_{32}}\left(q_{13}\sqrt{y_1-y_2}-S\left[\dot{y}_2\right]_e\right)}{q_{32}}\\ \qquad\times\left(\dfrac{q_{13}\left(\left[\dot{y}_1\right]_e-\left[\dot{y}_2\right]_e\right)}{2\sqrt{y_1-y_2}}-S\left[\ddot{y}_2\right]_e\right)\\ \qquad-q_{13}\sqrt{y_1-y_2}\\ \qquad+q_{20}\sqrt{y_2-\left(\dfrac{q_{13}\sqrt{y_1-y_2}-S\left[\dot{y}_2\right]_e}{q_{32}}\right)^2}\end{cases} \qquad (23)$$

The residual expression is given by (24)

$$\begin{cases}r_1=\left[u_1\right]_e-u_1\\ r_2=\left[u_2\right]_e-u_2\end{cases} \qquad (24)$$

In the following paragraph simulation results of the proposed method are presented.

V. Simulation Results

For the estimation tests, the Matlab SIMULINK is used with sampling period $T_e=10^{-3}s$ and in the presence of a Gaussian white noise with noise power equal to 0.01. The known nominal parameters of the system are: the gravity constant $g=9.81m.s^{-2}$, the tank section $S=0.0154m$

and the pipes section $S_p = 5.10^{-5} m$. The nominal values of the viscosity coefficient of the system are: $\theta_1 = 0.5,\ \theta_2 = 0.675,\ \theta_3 = 0.5$. For the SMDDG, $K_0 = 150$ and $K_1 = 100$ are considered and the dynamic gains initialized to zero. The simulations are achieved under operating conditions that allow the full rank condition for parameter identifiability. To satisfy this condition, the operating conditions are chosen as with operating condition $x_1 > x_3 > x_2$.

To prove the effectiveness of the proposed method, various fault scenarios are defined and summarized in Table I. The first one consists on simple actuator fault first one consists on simple actuator fault f_{act1} which appeared in the first input u_1 of the system. The second scenario is a simple process fault f_{proc1} that occurs on the first parameter of the system θ_1 . For scenario 3 , three faults are generated. Two faults appear successively at the first, then the second actuator, f_{act1} and f_{act2} respectively and another one appears on the output sensor. The last scenario considers the case of simultaneous process faults that arise on the system, f_{proc1} and f_{proc2} respectively.

For our studied system, the process faults are generally consisting on leakage faults. However the actuator faults can be defined by a reduction in control effectiveness.

An evaluation with known method in literature will be performed.

A. Analysis and Validation of the Fault Detection Schemes

Before presenting the simulation results, it is important to specify the types of faults that are applied. Table II summarizes the forms of the various faults. Here, the actuator fault is a positive intermittent signal, the process faults are an intermittent negative signal and the fault sensors is a step signal.

The effectiveness of the proposed FDI method Fig. 1 presented in the comparison results between the using of the SMDDG Differentiator (7) and the case when we used the Second Order Sliding Mode Differentiator (5).

1) Simple Actuator fault detection: Scenario 1: In this case, an intermittent fault f_{act1} is introduced in the first actuator at $20ms$. The simulation results are shown in Fig. 5. Based on equation (23), Fig. 5(a) shows the first and the second input estimation $[u_1]_e$ and $[u_2]_e$ respectively. This figure represents the curves given by using the SMDDG and HOSM compared by the nominal input values with any faults.

Remark that the red signal obtained by the SMDDG presents a positive peak which appeared at $t = 20ms$ and a negative one appeared at $t = 25ms$.

This explains that despite of the noise presence, it is well possible to detect an intermittent fault occurring on the first input u_1 .

First output estimation is provided by the SMDDG. However, with the HOSM differentiator fault detection is much more difficult since the failure is almost lost in the noise.

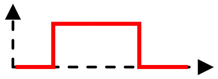
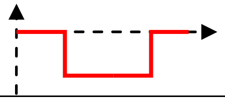
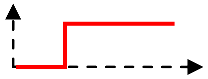
Consequently, only a very small variation appeared in both input and output of the estimated signal at the interval time $t \in [20 \quad 25] ms$. In Fig. 5(c) it is clear to see that, the first residual given by the SMDDG can introduce the appearance of actuator fault on u_1 . Since this residual is considered as DNN inputs so learning is better than the residual signal obtained using the HOSM.

2) Simple process *fault: scenario 2:* The second scenario represents a case of an intermittent fault f_{proc2} introduced in the second parameter of the system at $30ms.$ Fig. 6(a) and Fig. 6(b) represents the first and the second inputs and their estimations by using the SMDDG and the HOSM differentiator. These figures show that a fault occurred at $t = 30ms.$

TABLE I. SIMULATION SCENARIOS

Fault Types and its magnitude	Actuator 1 $f_{act1} = 0.6$	Actuator 2 $f_{act2} = 0.4$	Process 1 $f_{proc1} = -0.5$	Process 2 $f_{proc2} = -0.3$	Sensor 1 $f_{sen1} = 0.01$
Fault appearance interval	$[20, 25] ms$	$[25, 30] ms$	$[20, 28] ms$	$[30, 36] ms$	$t = 30ms$
Scenario 1	✓				
Scenario 2				✓	
Scenario 3	✓	✓			✓
Scenario 4	✓		✓	✓	

TABLE II. FAULT MODEL

Fault Types	**Model**
Actuator Fault	
Process Fault	
Sensor Fault	

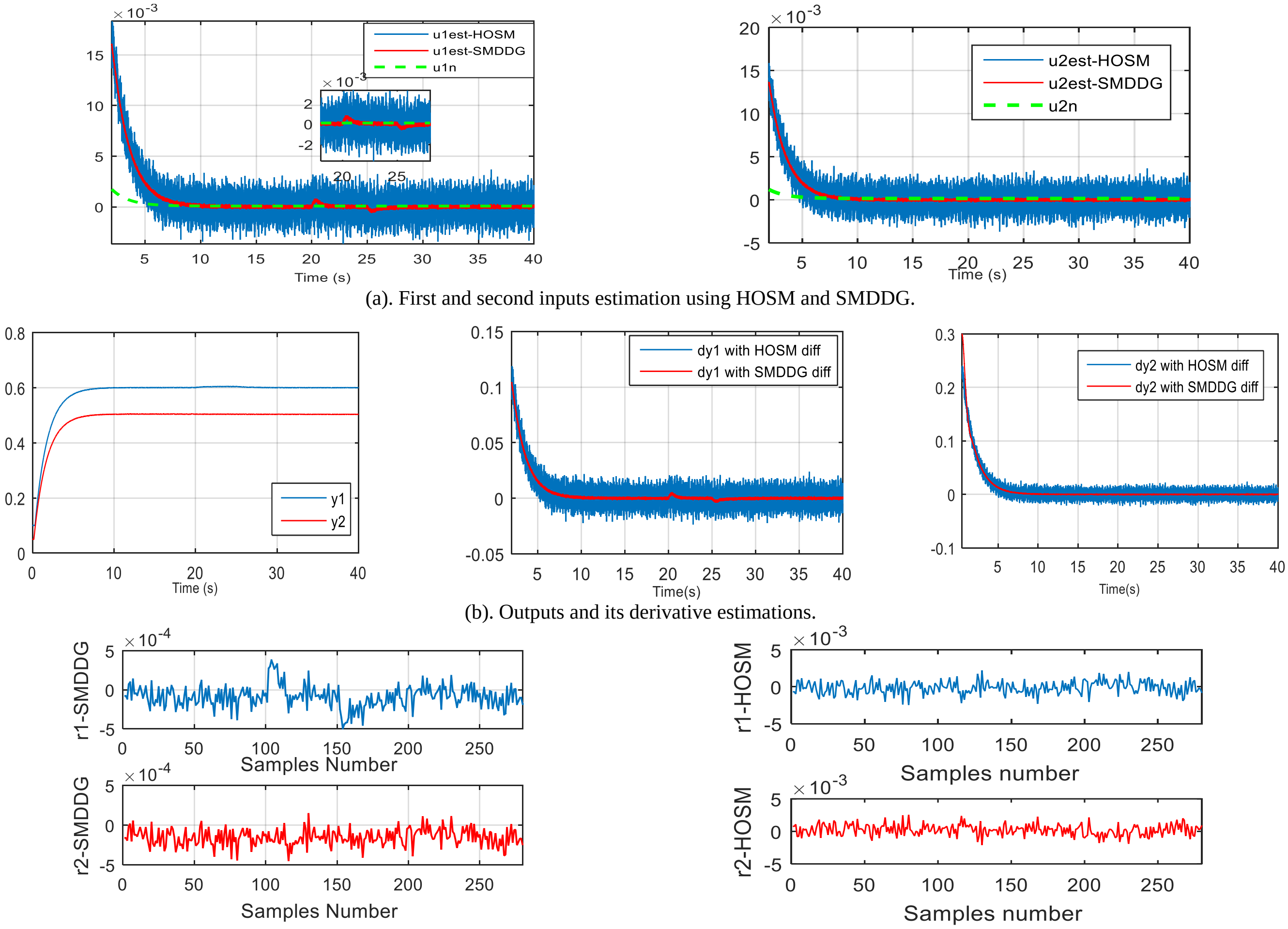
In the same time, a perturbation is showed in the second parameter $[\theta_2]_e$ estimated by using the equation (22).

With the SMDDG differentiator, the occurrence of fault is clearly shown compared it to the case of using the HOSM differentiator algorithm. The occurrence of the fault only in the second estimated input explained by its expression in equation (23), where the first input estimation is independent of the second parameter of the system. This difference shows the performance of our differentiator compared to the HOSM differentiator which presents more noise amplification. In Fig. 6(e) of the residual signal r_2 the fault starts with a negative peak and ends with a positive one.

This explains the presence of process fault, because the process defect is described by a leak so the residual, which is the difference between $[u_2]_e$ and u_2, gives a decay to the residual curve.

3) Successive actuator faults detection case scenario 3: In the practical case of non-linear systems, not only simple faults can appear, but many faults can occur in the same time or/and in a successive way. In this case, successive faults are tested. A fault f_{act2} is introduced in actuator 2 at $25ms$. In the same way, successive faults, f_{act1} in actuator 1 at $20ms$ and f_{sen1} in sensor 1 at $30ms$ have been simulated. Fig. 7 shows the simulation results of this case.

Thanks to the flatness properties, the system presents a strong relation between all parameters; inputs, outputs, there derivatives and even all the parameters of system itself shown in equation (22) and (23). Basing on Fig. 7 and with using the SMDDG, we remark the presence of fault in the output derivatives, Fig. 7(a) and also in the input estimation, Fig. 7(c).



(a). First and second inputs estimation using HOSM and SMDDG.

(b). Outputs and its derivative estimations.

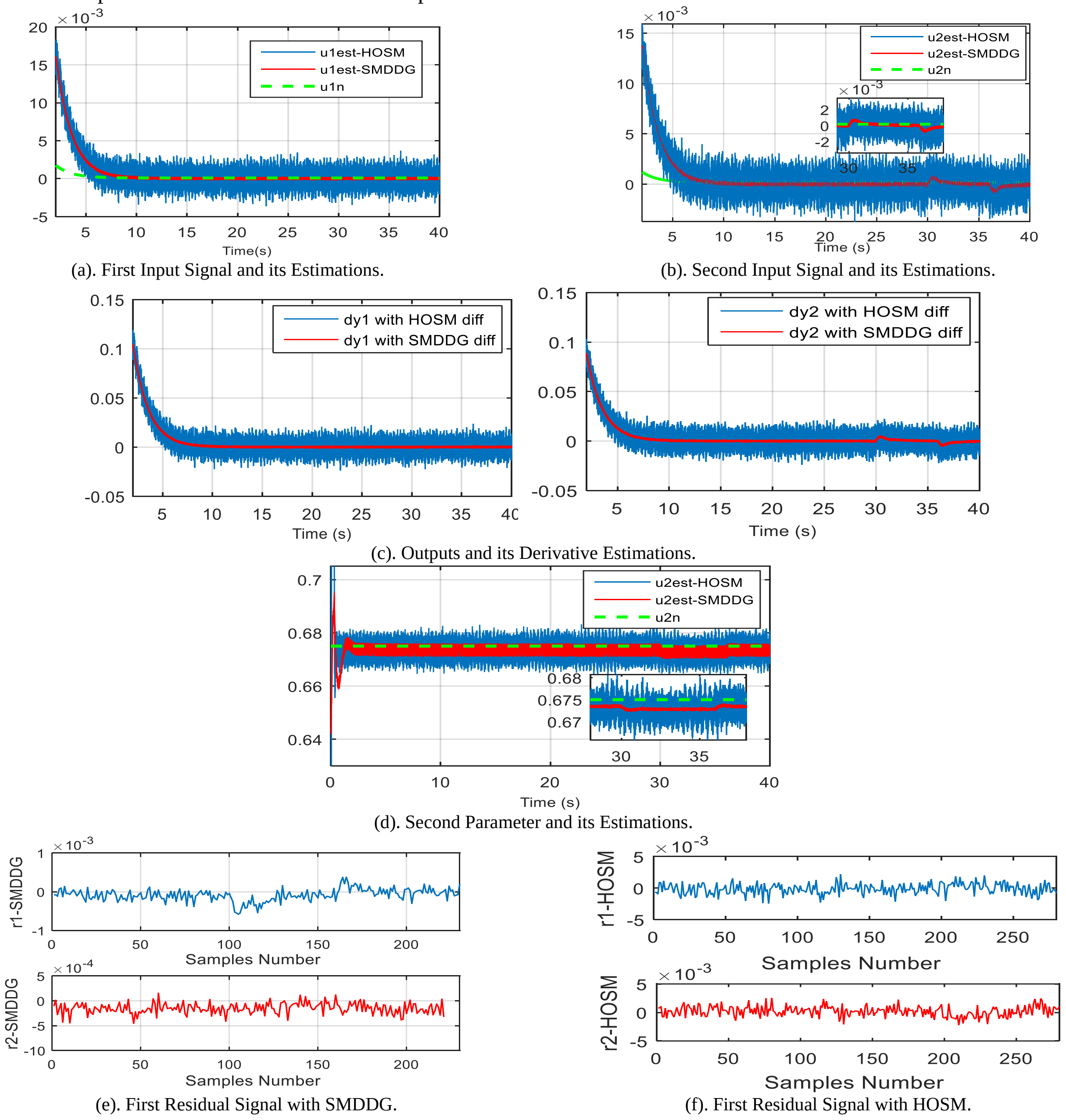
(c). First residual signal with SMDDG

(d). First residual signal with HOSM.

Fig. 5. First Scenario: Simple Actuator Fault.

But for the results that are given by using the HOSM differentiator, a small disturbance is shown, see Fig. 7(d) due to the presence of chattering on the estimated signals. For the residual signals given in Fig. 7(e), the actuator fault is shown like in the previous example and the sensor fault is presented by a high positive magnitude peak.

4) Simultaneous fault detection case scenario 4: Now, the scenario considering simultaneous faults is presented. The faults present in this case are: an actuator and a two process faults respectively $f_{act1} = 0.6$, $f_{proc1} = -0.5$, and $f_{proc2} = -0.3$. The f_{act1} and f_{proc1} appeared in the same time on the first input and also a fault in the second parameter systems f_{proc2} appeared in next time. The first and the second inputs estimations are shown in Fig. 8(a). The curve of the estimated parameter θ_2 shows the presence of two faults. The first one caused by f_{proc1} and the second one by $f_{proc2} = -0.3$. This proves the difficulty of the isolation step. In Fig. 8(d) which corresponds to the residual signals; it is clear that in the case where the SMDDG differentiator is used, faults are visible more clearly than the using of HOSM differentiator. Finally, the proposed simulation results show the effectiveness of the proposed technique in detecting faults compared to the well-known Levant's differentiator.



(a). First Input Signal and its Estimations.

(b). Second Input Signal and its Estimations.

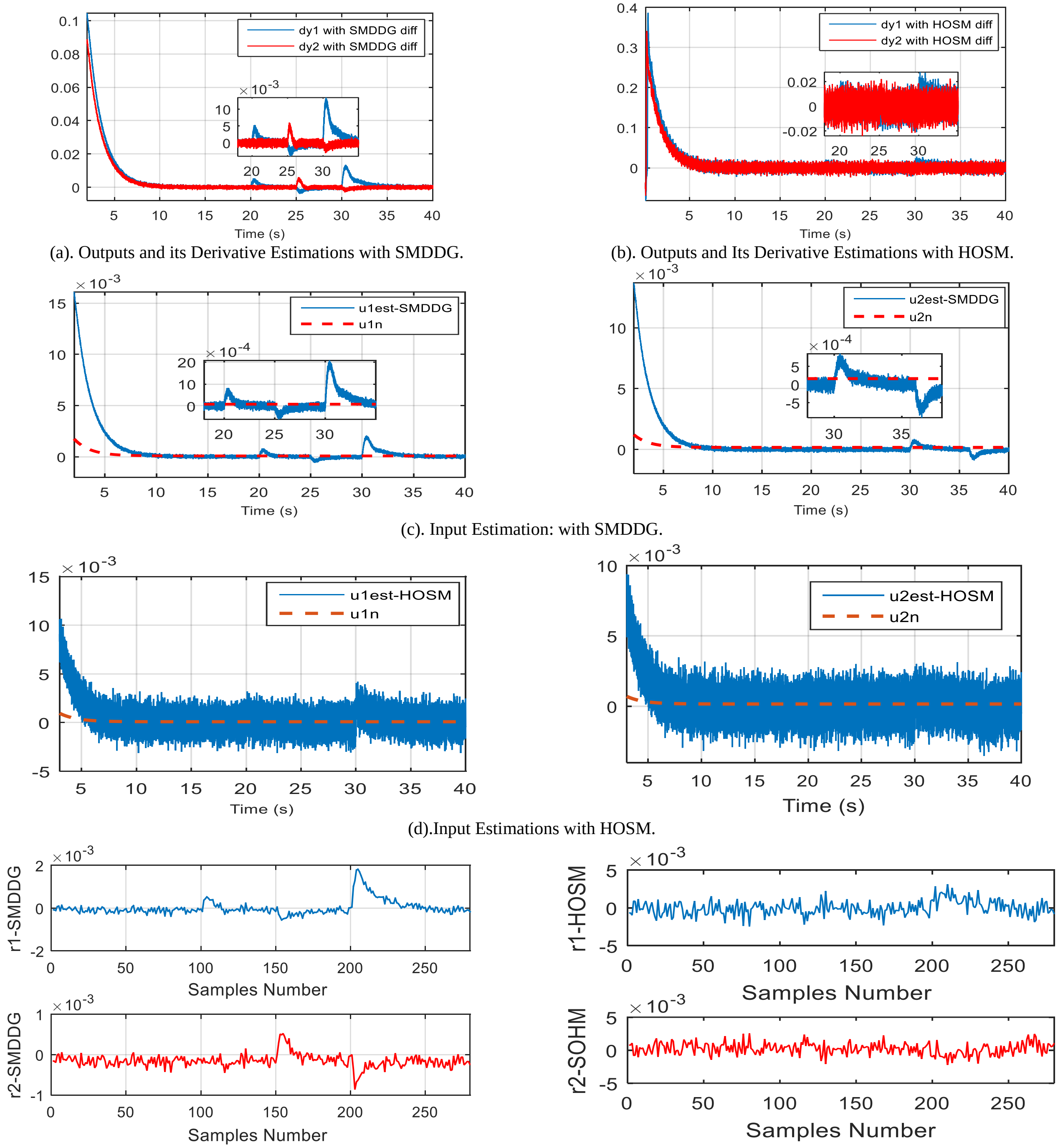
(c). Outputs and its Derivative Estimations.

(d). Second Parameter and its Estimations.

(e). First Residual Signal with SMDDG.

(f). First Residual Signal with HOSM.

Fig. 6. Second Scenario Simple Process Fault.



(a). Outputs and its Derivative Estimations with SMDDG.

(b). Outputs and Its Derivative Estimations with HOSM.

(c). Input Estimation: with SMDDG.

(d).Input Estimations with HOSM.

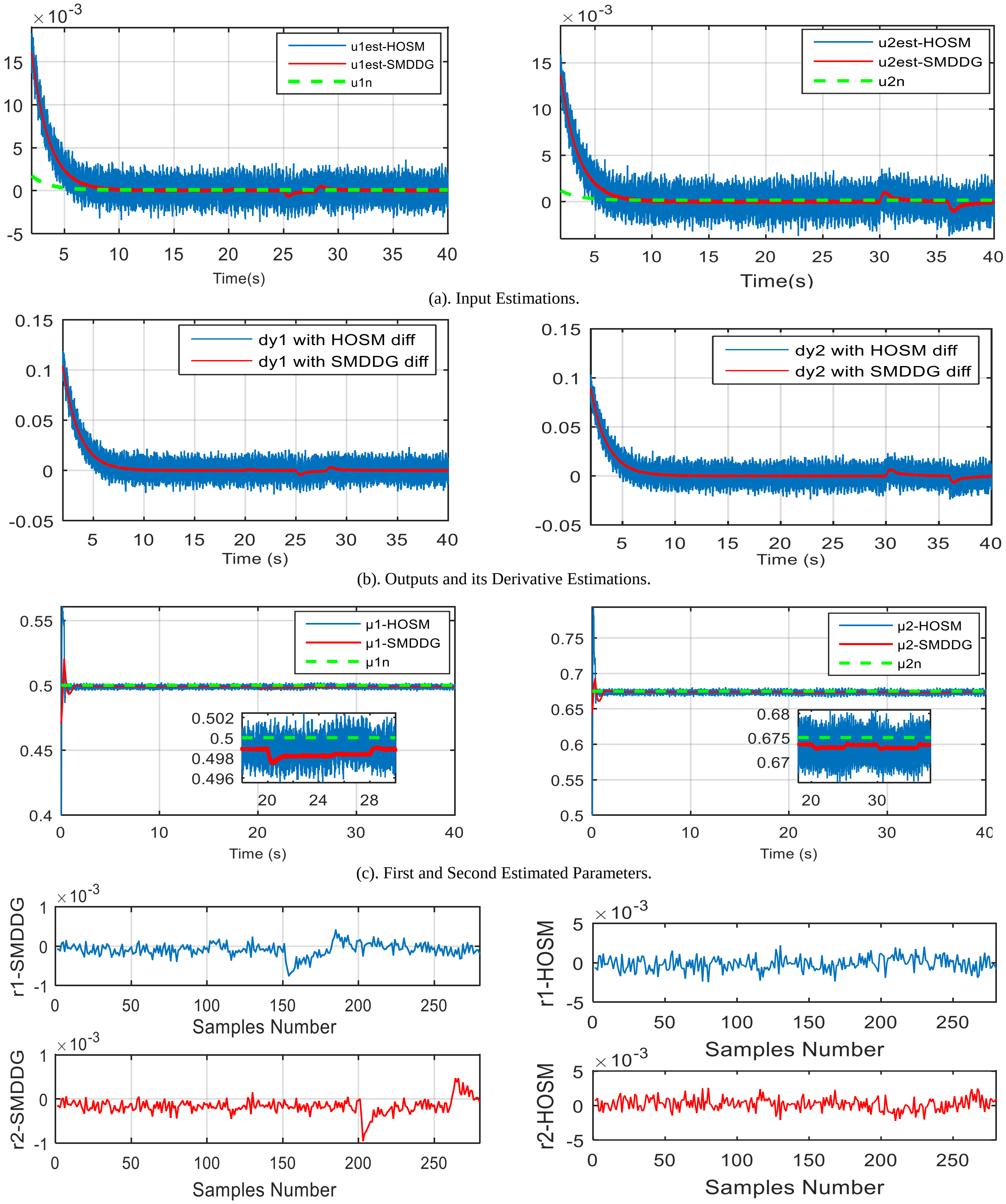
(e). Residual Signals.

Fig. 7. Third Scenario: Case of Successive Faults.

B. Fault Isolation

After the detection fault, the fault type is identified by the use of the DNN. The results of the four scenarios are summarized in Tables III and IV.

The first table shows the results in the case of using the SMDDG differentiator. With this last one, the fault kinds are clearly identified in all of the simulation scenarios. For example, in the case of successive actuator faults, we remark that the table presents the value 1 in the box of the first and the second actuators and also in the box of the first. Table II represents the results of the isolation fault by the DNN in the case of using the HOSM differentiator. In this table, some faulty alarm is presented. For example, in the case of scenario4, where the first actuator fault and the first process fault occurs simultaneously and a fault occurs in the second process, the value 1 is present in the first and the second actuator, in the first process and in the first sensor. Also, in the scenario3, it shows the presence of fault in the second process and the first sensor, which represents a faulty alarm.



(a). Input Estimations.

(b). Outputs and its Derivative Estimations.

(c). First and Second Estimated Parameters.

(d). Residual Signals.

Fig. 8. Ford Scenario Case of Simultaneous Faults.

TABLE III. NEURAL NETWORK RESULTS: SMDDG CASE

Residual		f_{act1}	f_{act2}	f_{proc1}	f_{proc2}	f_{sen1}	f_{sen2}
S1	r_1	1	0	0	0	0	0
	r_2	0	0	0	0	0	0
S2	r_1	0	0	0	0	0	0
	r_2	0	0	0	1	0	0
S3	r_1	1	0	0	0	1	0
	r_2	0	1	0	0	0	0
S4	r_1	1	0	1	0	0	0
	r_2	0	0	0	0	1	0

TABLE IV. NEURAL NETWORK RESULTS: HOSM CASE

Residual		f_{act1}	f_{act2}	f_{proc1}	f_{proc2}	f_{sen1}	f_{sen2}
S1	r_1	1	0	0	0	1	0
	r_2	0	1	0	0	0	0
S2	r_1	1	0	1	0	0	0
	r_2	0	0	1	1	0	0
S3	r_1	1	0	0	1	1	0
	r_2	0	1	0	1	0	1
S4	r_1	1	0	1	0	0	0
	r_2	0	1	0	0	1	0

VI. CONCLUSION

In this paper, a new FDI method is proposed to solve the fault detection and isolation problem in particular class of nonlinear system having the flatness property. This property gives a generic residual signal which is obtained by the difference between the input estimation and the nominal one. The residual signals are ready to be sensitive to all types of fault that can appear in the system which makes the identifiability of the fault very difficult. The effectiveness of the proposed fault detection method is given by the using of the SMDDG differentiator which gives satisfactory results in comparison with results obtained by the Levant's HOSM differentiator. After the fault detection, the isolation phase is done by the use of DNN pattern with back propagation. Many scenarios are tested to improve the effectiveness of the proposed method which contains an actuator, process or sensor fault. These faults are presented in different situation; simple, successive and simultaneous cases.

Several future works can be considered as a perspective. The first is to apply this approach in the practical case and the second is to extend this approach to be applied on other types of nonlinear systems.

REFERENCES

[1] Seogpil, Cho, Zhen et al, "Model based fault detection, fault isolation and fault tolerent control of a blad pitch system in floating wind turbine", Renewable Energy Jornal, 2018.

[2] H. Shahnazari, P. Mhaskar"Actuator and sensor fault detection and isolation for nonlinear systems subject to uncertainty", International Journal Robust Nonlinear Control, 2017.

[3] José Carlos M. Oliveira, Karen V. Pontes , Isabel Sartori , Marcelo Embiruçu,"Fault Detection and Diagnosis in dynamic systems using Weightless Neural Networks", Expert Systems With Applications, 2017.

[4] E. Naderi, K. Khorasani, "Data-driven fault detection, isolation and estimation of aircraft gas turbine engine actuator and sensors", Mechanical Systems and Signal Processing, 2018.

[5] P. Aboutalebi, A.Abbaspour, P. Forouzannezhad, A. Sargolzaei "A Novel Sensor Fault Detection in an Unmanned Quadrotor Based on Adaptive Neural Observer "Intell Robot Syst Journal ,2017.

[6] A. Abbaspour, P. Aboutalebi, Kang K.Yen, A. Sargolzaei,"Neural adaptive observer-based sensor and actuator fault detection in nonlinear systems: Application in UAV", ISA Transactions 2017.

[7] H. Rios, J. Davila, L.Fridman and C.Edwards, "Fault Detection and Isolation for Nonlinear System via HOSM Multiple-Observer", International Journal of Robust and Nonlinear Control, 2012.

[8] A. Levant, "Robust exact differentiation via sliding mode technique", Automatica, vol.34, 1998.

[9] Tiago Roux Oliveira, Victor Hugo Pereira Rodrigues, Antonio Estrada, Leonid Fridman," Output-feedback variable gain super-twisting algorithmfor arbitrary relative degree systems", International Journal of Control, 2043-2059, 2018.

[10] Hussein Obeid, Leonid Fridman, Salah Laghrouche, Mohamed Harmouche and Mohammad Ali Golkani, "Adaptation of Levant's Differentiator Based on Barrier Function",International Journal of Control, 2017.

[11] Carlos Vazquez, StanislavAranovskiy, Leonid B. Freidovich, Leonid Fridman "Time-Varying Gain Differentiator: A Mobile Hydraulic System Case Study", IEEE Transactions on Control Systems Technology, 1063-6536, 2016.

[12] Fliess, M., Levine, J., Martin, P. and Rouchon, P. Sur les systmes non linaires diffrentiellement plats, Comptes rendus de l'Acadmie des sciences, Srie1:Mathmatiques 315(5): 619–624, 1992.

[13] L. Sidhom, M. Smaoui, X. Brun and E. Bideaux, « Robust Estimator Design for Control of Electropneumatic System », IETE Journal of Research, Vol. 7, No.5, pp. 689-701, September 2018.