Nonlinear FDI based on state derivatives, as provided by inertial measurement units

Julien Marzat, Hélène Piet-Lahanier, Frédéric Damongeot, Eric Walter

Nonlinear FDI based on state derivatives, as provided by inertial measurement units

Julien Marzat [*,**] **Hélène Piet-Lahanier** [*]
Frédéric Damongeot [*] **Eric Walter** [**]

[*] *Office National d'Etudes et de Recherches Aérospatiales (ONERA), France, julien.marzat@onera.fr*
[**] *Laboratoire des Signaux et Systèmes (L2S), CNRS-SUPELEC-Univ Paris-Sud, France, eric.walter@lss.supelec.fr*

Abstract: Various strategies based on differential geometry or system inversion have been proposed to deal with fault detection and isolation (FDI) for nonlinear systems. Many of them require the computation of successive derivatives of inputs and outputs, which might be unrealistic in practical applications where measurements suffer noise and disturbances. In this paper, we take advantage of the fact that, in domains such as aerospace or robotics, sensors allow the measurement of first-order derivatives of state variables. This information, along with the redundancy provided by the control module can be used to generate residuals. Such a procedure is proposed and applied to a generic 2D aeronautical case study.

Keywords: fault detection and isolation, fault diagnosis, input reconstruction, nonlinear systems, system inversion

1. INTRODUCTION

The purpose of Fault Detection and Isolation (FDI) strategies is to detect and identify early unexpected changes in the system (referred to as *faults*) before they lead to a complete breakdown. A traditional way to tackle FDI is to rely on hardware redundancy (sensors or actuators with the same function). However, this approach implies higher costs and lower autonomy because of the additional weight, volume and power required. Another strategy is to rely on analytical redundancy, which exploits the relations between measured or estimated variables in order to detect possible dysfunctions (Isermann (1997)).

FDI methods for linear systems are well known and formalized. The main methods use parameter or state estimation, to check whether the estimates are following an acceptable behavior by comparing them to expected values. Pioneer work by Chow and Willsky (1984) has introduced *parity space methods*, which aim at checking the consistency between inputs and outputs by eliminating the state variables from the model equations, exploiting temporal redundancy on a short horizon. This kind of approach has later been formalized in geometrical terms by Massoumnia et al. (1989). The idea remains the exploitation of the null-space of the observability matrix to generate residuals, with additional concern for robustness.

Whenever extreme performance is sought, most real systems behave nonlinearly and are appropriately described by nonlinear models, e.g., in aerospace and robotics. The immediate idea would be to make the existing linear methods applicable by linearization or polynomial approximation (Witczak (2007)), but this may add more uncertainty to the already inaccurate model, leading to higher false-alarm or non-detection rates. Nonlinear FDI methods are thus required to deal with these systems. Recently, dedicated methods for fault diagnosis for nonlinear systems have emerged. They aim at extending parity space methods to nonlinear control-affine systems via differential geometry or differential algebra tools. A description and a comparison of methods based on these two frameworks can be found in (Bokor and Szabó (2009)) or in (Shumsky and Zhirabok (2006)).

Using a *differential-geometric* approach, De Persis and Isidori (2001) have been the first to suggest a nonlinear parity technique for nonlinear control-affine systems, using the observability tools defined in (Isidori (1995)). More recently, another extension for the same class of nonlinear systems has been proposed in (Leuschen et al. (2005)) with similar assumptions, but closer to the original idea of parity equations. *Differential-algebraic* methods have also been designed (see Martinez-Guerra and Diop (2004)). They use a transformation of the nonlinear system into a set of differential polynomials, which are only functions of the inputs, outputs and their successive derivatives. It may then be possible to extract the fault variable from these expressions to detect and estimate faults with the help of elimination theory. Another interesting idea is *inversion-based* fault diagnosis, where the left-inverse of the nonlinear system (Hirschorn (1979)) is computed to obtain a dynamical model whose outputs are the fault signals while the inputs are the original inputs and outputs and their successive derivatives (see Edelmayer et al. (2004)).

Many of these techniques require the successive computation of time derivatives of the inputs and outputs of nonlinear systems from the computed control inputs and the noisy and disturbed output measurements. Numerical differentiation of these quantities may lead to serious com-

putational errors. This problem is indeed known to be ill posed: small perturbations of the signal may lead to large errors in the computed derivatives (Ramm and Smirnova (2001)).

Following an inversion-based strategy, we propose an FDI procedure for nonlinear control-affine systems exploiting measured state derivatives to avoid numerical differentiation. The procedure takes advantage of the fact that autonomous vehicles (in aerospace or robotics) are most often equipped with an IMU (Inertial Measurement Unit) that measures accelerations, which are the derivative of the velocity state variables. Along with this information, the particular structure of the system allows the estimation of the actual control inputs, as achieved by the actuators. Residuals that are indicative of actuator faults are then obtained by comparing these estimates with the desired control inputs as computed by the control module. Section 2 presents the class of nonlinear systems considered and the principles of the procedure. An algorithm to perform the FDI residual generation is detailed in Section 3. This strategy is then applied in Section 4 to a representative aeronautical case study to detect, isolate and even identify actuator faults. Conclusions and perspectives are discussed in Section 5.

2. METHOD

2.1 Class of systems considered

We consider the nonlinear state-space model

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x}) \cdot \mathbf{u} + \mathbf{w}_\mathrm{d} + \mathbf{w}_\mathrm{f} \\ \mathbf{y} = \mathbf{C} \cdot \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{bmatrix} + \mathbf{v} \end{cases} \tag{1}$$

where $\mathbf{x}$ is the state vector, $\mathbf{u}$ is the control input vector, $\mathbf{y}$ is the output vector, $\mathbf{f}, \mathbf{G}$ and $\mathbf{C}$ are a smooth vector field and matrices with appropriate dimensions, $\mathbf{w}_\mathrm{d}$, $\mathbf{w}_\mathrm{f}$ and $\mathbf{v}$ are the perturbation, fault and measurement noise vectors. The nominal model to be used as a basis for fault diagnosis is

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x}) \cdot \mathbf{u} \\ \mathbf{y} = \mathbf{C} \cdot \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{bmatrix} \end{cases} \tag{2}$$

The special form of the observation equation is a key point of the method. It indicates that the output vector depends not only on state variables, as usual, but also on derivatives of state variables. More restrictive assumptions (but nevertheless realistic) needed by our algorithm will be described in Section 3.1.

This type of observation equation is, for example, obtained in aeronautics where most of the vehicles are equipped with a inertial navigation system (INS), comprising a inertial measurement unit (IMU) coupled with a computer. The IMU measures non-gravitational acceleration and angular velocity, and the INS integrates them to evaluate position, velocity and orientation. Position, velocity, orientation and angular velocity are usually forming the state vector $\mathbf{x}$, while acceleration is part of $\dot{\mathbf{x}}$.

2.2 Principle of the procedure

Most FDI methods compare estimated outputs with their measured values to generate residuals that are indicative of the presence of faults. We propose to generate similar signals, but studying the inputs of the system. This requires distinguishing several types of input vectors $\mathbf{u}$, as illustrated by Figure 1.

- $\mathbf{u}_\mathrm{c}$ is the computed control input vector, provided by a user-defined control module, either in open or closed loop, and sent to the actuators.
- $\mathbf{u}_\mathrm{a}$ is the actual (unknown) control input vector as achieved by the actuators.
- $\widehat{\mathbf{u}}_\mathrm{a}$ is an estimate of $\mathbf{u}_\mathrm{a}$, which may be obtained with the algorithm proposed in Section 3.

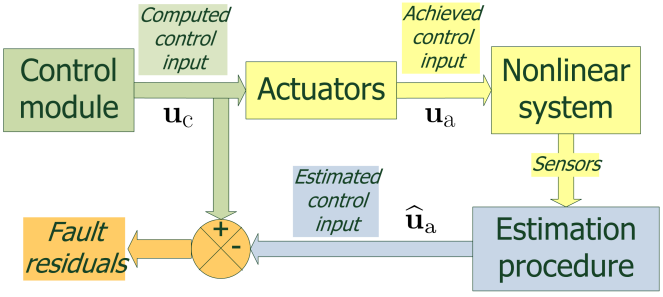

Fig. 1. Residual generation

The unusual form of the observation equation may imply that there are some state equations where *all* the variables are measured, except for the control input $\mathbf{u}$. Using these relations along with the control-affine structure of the model, it is trivial to compute an estimate $\widehat{\mathbf{u}}_\mathrm{a}$ of the achieved control input $\mathbf{u}_\mathrm{a}$. Fault residuals are then obtained by comparing $\widehat{\mathbf{u}}_\mathrm{a}$ to the computed control input $\mathbf{u}_\mathrm{c}$.

3. ALGORITHM

3.1 Model extraction

A preliminary step of the procedure is to extract from the nonlinear state-space model (2) all state equations containing control inputs, and involving only measured state variables and derivatives of state variables.

Algorithm 1 shows the extraction procedure. The number of state variables is denoted by N. The i^th state equation in (2) is

$$\dot{x}_i = f_i(\mathbf{x}) + \mathbf{g}_i^\mathrm{T}(\mathbf{x}) \cdot \mathbf{u}$$

where $\mathbf{g}_i^\mathrm{T}$ is the i^th row of matrix $\mathbf{G}$.

If n is the number of equations and m the number of control inputs, the resulting set of equations can be written as

$$\begin{bmatrix} \widetilde{f}_1(\mathbf{y}) \\ \vdots \\ \widetilde{f}_n(\mathbf{y}) \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{g}}_1^\mathrm{T}(\mathbf{y}) \\ \vdots \\ \widetilde{\mathbf{g}}_n^\mathrm{T}(\mathbf{y}) \end{bmatrix} \cdot \mathbf{u} = \begin{bmatrix} \widetilde{g}_{11}(\mathbf{y}) & \cdots & \widetilde{g}_{m1}(\mathbf{y}) \\ \vdots & \ddots & \vdots \\ \widetilde{g}_{1n}(\mathbf{y}) & \cdots & \widetilde{g}_{mn}(\mathbf{y}) \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} \tag{3}$$

or more concisely $\widetilde{\mathbf{f}}(\mathbf{y}) = \widetilde{\mathbf{G}}(\mathbf{y}) \cdot \mathbf{u}$. Dependency in $\mathbf{y}$ will be omitted in the following sections.

Algorithm 1.

$j \leftarrow 0$
for $i = 1$ to N **do**
if $\dot{x}_i$ and all the state variables that are arguments of f_i and $\mathbf{g}_i$ are measured, and $\mathbf{g}_i$ is not identically zero **then**

$$j \leftarrow j + 1$$
$$\widetilde{f}_j(\mathbf{y}) = \dot{x}_i - f_i(\mathbf{x})$$
$$\widetilde{\mathbf{g}}_j^{\mathrm{T}}(\mathbf{y}) = \mathbf{g}_i^{\mathrm{T}}(\mathbf{x})$$

end if
end for

3.2 Direct residual generation (DRG)

Consider each equation in (3) separately. Provided that the entry of $\widetilde{\mathbf{G}}$ which multiplies this equation is nonzero, each control input involved can be expressed as a function of the measurement vector $\mathbf{y}$ and the other control inputs present in the equation. It implies that there are as many ways to express each control input as there are equations where it appears. Let $\widehat{u}_{\mathrm{a}ij}$ be the estimate of the i^{th} control input from the j^{th} equation of (3). If $\widetilde{g}_{ij}$ is not identically zero, we have

$$\widehat{u}_{\mathrm{a}ij}(\mathbf{u}) = \frac{1}{\widetilde{g}_{ij}} \left[\widetilde{f}_j - \sum_{\substack{k=1 \\ k \neq i}}^{m} \widetilde{g}_{kj} \cdot u_k \right]$$

These estimates are computed by substituting the computed control input vector $\mathbf{u}_{\mathrm{c}}$ for the unknown control input vector $\mathbf{u}$. Residuals r_{ij} can now be generated by comparing each computed control input $u_{\mathrm{c}i}$ (as given by the control module) to the corresponding estimated inputs $\widehat{u}_{\mathrm{a}ij}$. The residual r_{ij} is defined as

$$r_{ij} = u_{\mathrm{c}i} - \widehat{u}_{\mathrm{a}ij}(\mathbf{u}_{\mathrm{c}})$$
$$= \frac{\widetilde{g}_{ij}}{\widetilde{g}_{ij}} u_{\mathrm{c}i} - \frac{\widetilde{f}_j}{\widetilde{g}_{ij}} + \frac{1}{\widetilde{g}_{ij}} \sum_{\substack{k=1 \\ k \neq i}}^{m} \widetilde{g}_{kj} \cdot u_{\mathrm{c}k}$$

so

$$r_{ij} = -\frac{\widetilde{f}_j}{\widetilde{g}_{ij}} + \frac{1}{\widetilde{g}_{ij}} \sum_{k=1}^{m} \widetilde{g}_{kj} \cdot u_{\mathrm{c}k}$$

Algorithm 2 describes the resulting DRG procedure.

Algorithm 2.

for $i = 1$ to m **do**
for $j = 1$ to n **do**
if $\tilde{g}_{ij}$ is not identically 0 **then**

$$r_{ij} = \frac{1}{\tilde{g}_{ij}} \left[-\tilde{f}_j + \tilde{\mathbf{g}}_j^{\mathrm{T}} \cdot \mathbf{u}_{\mathrm{c}} \right]$$

end if
end for
end for

This first step produces as many residuals as there are nonzero terms in $\widetilde{\mathbf{G}}$.

3.3 Additional residual generation (ARG)

Some equations in (3) may contain only one control input. It is then possible to express this control input *only* as a function of the measurements. Suppose the j^{th} equation in (3) contains only the k^{th} control input, with $k \in [1, ..., m]$. This equation has the form

$$\widetilde{f}_j = \widetilde{g}_{kj} \cdot u_k$$

An estimate of u_k from equation j is then $\widehat{u}_{kj} = \widetilde{f}_j / \widetilde{g}_{kj}$. The idea of this ARG step is to substitute $\widehat{u}_{kj}$ for the k^{th} term of the computed input vector $\mathbf{u}_{\mathrm{c}}$ (as given by the control module). Then, the formula given in Algorithm 2 may be used to generate additional residuals.

These residuals are sensitive to a reduced number of actuator faults, because less computed inputs are involved in their evaluation. This will be illustrated in the case-study described in Section 4.

If there is no equation in (3) where only one input is present, the ARG step is not performed.

Algorithm 3 describes the preparatory step of ARG.

Algorithm 3.

$\mathbf{\Gamma} = \mathrm{zeros}(n, m)$
$\mathbf{H} = \mathrm{zeros}(n, m)$
$\mathcal{D} = \{\}$
for $j = 1$ to n **do**
for $i = 1$ to m **do**
if $\widetilde{g}_{ij}$ is not identically 0 **then**
$\Gamma_{ij} = 1$
end if
end for
if $\sum_{i=1}^{m} \Gamma_{ij} = 1$ **then**

$$h_{ij} = \widehat{u}_{ij} = \frac{\widetilde{f}_j}{\widetilde{g}_{ij}}$$
$$\mathcal{D} = \{\mathcal{D}, [i, j]\}$$

end if
end for

The matrices $\mathbf{\Gamma}$ and $\mathbf{H}$ will be used to compute the additional residuals. The set $\mathcal{D}$ comprises all the indexes $[i, j]$ of the nonzero terms in $\mathbf{H}$.

To allow all possible substitutions, all combinations of the feasible indexes have to be identified from $\mathcal{D}$. If there are two different ways to express directly u_i from equations j_1 and j_2, namely $\widehat{u}_{ij_1}$ and $\widehat{u}_{ij_2}$, then we choose that only one at a time can replace u_{ci} in u_{c}. Other strategies could of course be considered. Algorithm 4 shows the recursive procedure employed (named *Group*($\mathcal{S}$)) to extract all the allowed combinations. The size (number of elements) of a set $\mathcal{S}$ is denoted $n_{\mathcal{S}}$.

The result of Algorithm 4 is a set $\mathcal{P}$ containing all the combinations of indexes $[i, j]$ where u_i can be expressed from equation j, excluding combinations where the i indexes take the same value (as explained above). The MAPLE procedure *choose* can be used to implement Algorithm 4, but the excluded combinations should be removed afterwards.

Algorithm 4. Group($\mathcal{D}$)

$\mathcal{P} = \{\}$
$\mathcal{Q} = \{\}$
for $k = 0$ to $n_{\mathcal{D}}$ **do**
  if $k = 0$ **then**
    $\mathcal{P} = \{\emptyset\}$
  else
    $\mathcal{P} = \text{Group}(\mathcal{P})$
    for all subsets $\mathcal{E} \in \mathcal{P}$ **do**
      $\mathcal{Q} = \mathcal{E} \cup \mathcal{D}[k]$
    end for
    for $i = 1$ to $n_{\mathcal{Q}}$ **do**
      if $n_{\mathcal{Q}[i]} > 1$ **then**
        for $j = 1$ to $n_{\mathcal{Q}[i]}$ **do**
          for $l = 1$ to $n_{\mathcal{Q}[i]}$ **do**
            if $\mathcal{Q}[i][j][1] = \mathcal{Q}[i][l][1]$ **then**
              Remove $\mathcal{Q}[i]$ from $\mathcal{Q}$
            end if
          end for
        end for
      end if
    end for
    $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{Q}$
  end if
end for

Algorithm 5 builds all the possible vectors $\widetilde{\mathbf{u}}_c$ from $\mathbf{u}_\text{c}$ by substituting u_{ij} for u_{ci}, where $[i, j]$ is in the subset $\mathcal{P}[k]$ of $\mathcal{P}$. $\mathbf{1}_m$ is a vector of size m whose components are all equal to 1, and $\mathbf{I}_m$ is the identity matrix of dimension m. Additional residuals $\widetilde{r}_{ij}^k$ are then obtained with the formula given in Algorithm 2, but with the new $\widetilde{\mathbf{u}}_\text{c}^k$ instead of the original $\mathbf{u}_\text{c}$.

Algorithm 5.

for $k = 1$ to $n_{\mathcal{P}}$ **do**
  $\mathbf{\Delta}^k = \text{zeros}(n, m)$
  for $j = 1$ to n **do**
    for $i = 1$ to m **do**
      for $l = 1$ to $n_{\mathcal{P}[k]}$ **do**
        if $i = \mathcal{P}[k][l][1]$ and $j = \mathcal{P}[k][l][2]$ **then**

$$\mathbf{\Delta}_{ij}^k = 1$$

        end if
      end for
    end for
  end for

$$\widetilde{\mathbf{u}}_\text{c}^k = \left(\mathbf{\Delta}^k\right)^\text{T} \cdot \mathbf{H} \cdot \mathbf{1}_m + \left(\mathbf{I}_m - \left(\mathbf{\Delta}^k\right)^\text{T} \cdot \mathbf{\Gamma}\right) \cdot \mathbf{u}_\text{c}$$

  for $j = 1$ to n **do**
    for $i = 1$ to m **do**
      if $\widetilde{g}_{ij} \neq 0$ **then**

$$\widetilde{r}_{ij}^k = \frac{1}{\widetilde{g}_{ij}} \left[-\widetilde{f}_j + \widetilde{\mathbf{g}}_j^\text{T} \cdot \widetilde{\mathbf{u}}_\text{c}^k \right]$$

      end if
    end for
  end for
end for

3.4 Remarks

- This procedure is completely analytical. It only needs to be run once with a symbolic mathematics software such as MAPLE. Then, the obtained residuals could easily be hard-coded for embedded processing at a low computation cost. The residuals are indeed explicit functions of the measurements so that no integration of differential equations is required.
- The algorithm may be extended to systems where all the variables involved in the equations are not directly measured but could be obtained with an observer. The abstracted equations will be the state equations containing control inputs, whose state variables are measured or observed and derivatives of state variables are measured.
- Other inversion algorithms could be considered.

4. AERONAUTICAL APPLICATION

The following case-study corresponds to a 2D longitudinal model of a surface-to-air missile (Marzat et al. (2009a)). A more complete 3D study is currently under way.

4.1 State-space model

The dynamics of the missile is described by the following state equations.

$$\begin{cases} \dot{x} = \cos(\theta) v_\text{bx} + \sin(\theta) v_\text{bz} \\ \dot{z} = \cos(\theta) v_\text{bz} - \sin(\theta) v_\text{bx} \\ \dot{v}_\text{bx} = -q v_\text{bz} - \sin(\theta) g - \dfrac{Q s_\text{ref}}{M} \left[c_\text{x0} + c_\text{xa} \alpha + c_{\text{x}\delta_\text{m}} \delta_\text{m} \right] \\ \qquad + \dfrac{1}{M} \left[f_\text{min} + (f_\text{max} - f_\text{min}) \eta \right] \\ \dot{v}_\text{bz} = q v_\text{bx} + \cos(\theta) g - \dfrac{Q s_\text{ref}}{M} \left[c_\text{z0} + c_\text{za} \alpha + c_{\text{z}\delta_\text{m}} \delta_\text{m} \right] \\ \dot{q} = \dfrac{Q s_\text{ref}}{b} \left[c_\text{m0} + c_\text{ma} \alpha + c_{\text{m}\delta_\text{m}} \delta_\text{m} + \dfrac{l_{ref}}{\sqrt{v_\text{bx}^2 + v_\text{bz}^2}} c_\text{mq} q \right] \\ \dot{\theta} = q \end{cases} \tag{4}$$

where

- b is the inertia term,
- $[x, z]$ is the position in the inertial frame,
- $[v_\text{bx}, v_\text{bz}]$ is the speed in body coordinates,
- θ is the orientation in the (x, z) plane,
- q is the angular velocity,
- δ_m is the rudder deflection angle,
- η is the propulsion rate,
- $Q = \frac{1}{2}\rho(v_\text{bx}^2 + v_\text{bz}^2)$ is the dynamic pressure,
- $\alpha = \arctan(\frac{v_\text{bz}}{v_\text{bx}})$ is the angle of attack,
- M is the aircraft mass,
- f_min and f_max are constants of the propulsion model,
- s_ref and l_ref are characteristic dimensions,
- $c_{(.)}$ are the aerodynamic coefficients, known piecewise continuous nonlinear functions of the Mach value and angle of attack.

The state vector is $\mathbf{x} = [x, z, v_\text{bx}, v_\text{bz}, q, \theta]^\text{T}$, the input vector is $\mathbf{u} = [\delta_\text{m}, \eta]^\text{T}$. An IMU provides measurements of the non-gravitational acceleration $(a_\text{bx}, a_\text{bz})$ and the angular rate q. The navigation system integrates these signals to estimate position, speed and orientation. The expressions of the non-gravitational accelerations are $a_\text{bx} = \dot{v}_\text{bx} + q v_\text{bz} + \sin(\theta) g$ and $a_\text{bz} = \dot{v}_\text{bz} - q v_\text{bx} - \cos(\theta) g$. The output

vector is thus $\mathbf{y} = [x, z, \dot{x}, \dot{z}, v_{\rm bx}, v_{\rm bz}, \theta, q, a_{\rm bx}, a_{\rm bz}]^{\rm T}$. The computed control input vector $\mathbf{u}_{\rm c} = [\delta_{\rm mc}, \eta_{\rm c}]^{\rm T}$ is obtained via a guidance and control module, the description of which is not needed here. Further details can be found in Marzat et al. (2009b).

4.2 Model extraction

In (4), two state equations fulfill the requirements of Algorithm 1. The abstracted equations are

$$\begin{cases} \dot{v}_{\rm bx} = -qv_{\rm bz} - \sin(\theta)g - \dfrac{Qs_{\rm ref}}{M}\left[c_{x0} + c_{xa}\alpha + c_{x\delta_{\rm m}}\delta_{\rm m}\right] \\ \qquad\qquad + \dfrac{1}{M}\left[f_{\min} + (f_{\max} - f_{\min})\eta\right] \\ \dot{v}_{\rm bz} = qv_{\rm bx} + \cos(\theta)g - \dfrac{Qs_{\rm ref}}{M}\left[c_{z0} + c_{za}\alpha + c_{z\delta_{\rm m}}\delta_{\rm m}\right] \end{cases}$$

They can be expressed under the form (3) as

$$\begin{bmatrix} \widetilde{f}_1 \\ \widetilde{f}_2 \end{bmatrix} = \begin{bmatrix} \widetilde{g}_{11} & \widetilde{g}_{21} \\ \widetilde{g}_{12} & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta_{\rm m} \\ \eta \end{bmatrix} \tag{5}$$

with

$$\begin{cases} \widetilde{f}_1 = a_{bx} + \dfrac{Qs_{\rm ref}}{M}\left[c_{x0} + c_{xa}\alpha\right] - \dfrac{f_{\min}}{M} \\ \widetilde{f}_2 = a_{bz} + \dfrac{Qs_{\rm ref}}{M}\left[c_{z0} + c_{za}\alpha\right] \\ \widetilde{g}_{11} = -\dfrac{Qs_{\rm ref}}{M}c_{x\delta_{\rm m}} \\ \widetilde{g}_{12} = \dfrac{f_{\max} - f_{\min}}{M} \\ \widetilde{g}_{21} = -\dfrac{Qs_{\rm ref}}{M}c_{z\delta_{\rm m}} \end{cases}$$

4.3 Direct Residual Generation

Applying Algorithm 2 to (5) gives the residuals

$$\begin{cases} r_{11} = \dfrac{-\widetilde{f}_1 + \widetilde{g}_{11}\delta_{\rm mc} + \widetilde{g}_{21}\eta_{\rm c}}{\widetilde{g}_{11}} \\ r_{21} = \dfrac{-\widetilde{f}_1 + \widetilde{g}_{11}\delta_{\rm mc} + \widetilde{g}_{21}\eta_{\rm c}}{\widetilde{g}_{21}} \\ r_{12} = \dfrac{-\widetilde{f}_2 + \widetilde{g}_{12}\delta_{\rm mc}}{\widetilde{g}_{12}} \end{cases}$$

4.4 Second step and final residuals

The abstracted equations (5) comprise one equation where only the rudder control input $\delta_{\rm m}$ appears. Thus, the second step of the procedure can be performed. Algorithm 3 gives the following results:

$$\mathbf{\Gamma} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \mathbf{H} = \begin{bmatrix} 0 & 0 \\ \dfrac{\widetilde{f}_2}{\widetilde{g}_{12}} & 0 \end{bmatrix}, \mathcal{D} = \{\{[1,2]\}\}$$

Because there is only one element in the unique subset of $\mathcal{D}$, there is only one possible combination and Algorithm 4 returns $\mathcal{P} = \mathcal{D}$. Therefore, in Algorithm 5 we have $n_{\mathcal{P}} = 1$ and thus

$$\mathbf{\Delta}^k = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

$$\widetilde{\mathbf{u}}_{\rm c}^1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ \dfrac{\widetilde{f}_2}{\widetilde{g}_{12}} & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \left(\mathbf{I}_2 - \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \right) \begin{bmatrix} \delta_{\rm mc} \\ \eta_{\rm c} \end{bmatrix}$$

$$\widetilde{\mathbf{u}}_{\rm c}^1 = \begin{bmatrix} \dfrac{\widetilde{f}_2}{\widetilde{g}_{12}} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \eta_{\rm c} \end{bmatrix} = \begin{bmatrix} \dfrac{\widetilde{f}_2}{\widetilde{g}_{12}} \\ \eta_{\rm c} \end{bmatrix}$$

Substituting $\widetilde{\mathbf{u}}_{\rm c}^1$ for $\widetilde{\mathbf{u}}$ in the residual generation formula gives the two additional residuals

$$\begin{aligned} \widetilde{r}_{11}^1 &= \frac{-\widetilde{f}_1 + \frac{\widetilde{g}_{11}\widetilde{f}_2}{\widetilde{g}_{12}} + \widetilde{g}_{21}\eta_{\rm c}}{\widetilde{g}_{11}} \\ \widetilde{r}_{21}^1 &= \frac{-\widetilde{f}_1 + \frac{\widetilde{g}_{11}\widetilde{f}_2}{\widetilde{g}_{12}} + \widetilde{g}_{21}\eta_{\rm c}}{\widetilde{g}_{21}} \end{aligned} \tag{6}$$

Taking (5) into account, we can rewrite (6) as

$$\begin{aligned} \widetilde{r}_{21}^1 &= \frac{-\left(\widetilde{g}_{11}\delta_m + \widetilde{g}_{21}\eta\right) + \frac{\widetilde{g}_{11}\widetilde{g}_{12}\delta_m}{\widetilde{g}_{12}} + \widetilde{g}_{21}\eta_{\rm c}}{\widetilde{g}_{21}} \\ \widetilde{r}_{21}^1 &= \frac{\widetilde{g}_{11}(-\delta_m + \delta_m) + \widetilde{g}_{21}(-\eta + \eta_{\rm c})}{\widetilde{g}_{21}} \\ \widetilde{r}_{21}^1 &= -\eta + \eta_{\rm c} \end{aligned}$$

Therefore, if $\eta = \eta_{\rm c}$ (non-faulty situation), the residual is zero and if $\eta = \eta_{\rm c} + \eta_{\rm fault}$, the residual identifies the magnitude of the fault affecting propulsion. This result also shows that the residual is structurally insensitive to faults affecting the rudder input, even with measurements uncertainty. By the same token, identification of the rudder fault is feasible via r_{12}. The signature table of the residuals with respect to the faults is given in Table 1.

Table 1. Fault-signature table

	$\delta_{\rm m}$	η
r_{11}	1	1
r_{21}	1	1
r_{12}	1	0
$\widetilde{r}_{11}^1$	0	1
$\widetilde{r}_{21}^1$	0	1

To get an idea of the robustness of the method to model uncertainty, consider a modeling error $\widetilde{g}_{11} = \widetilde{g}_{11} + \varepsilon$, where ε is a small parameter. A calculation similar to that above yields

$$\widetilde{r}_{21}^1 = -\varepsilon\delta_{\rm m} - \eta + \eta_{\rm c}$$

The residual is now sensitive to the other control input, but its effect is limited as ε is small and $\delta_{\rm m}$ bounded. If a bound on the value of ε is available, this information can be taken into account in the analysis of the residual.

4.5 Results

IMU measurements are affected by uncertainty, modeled as biases, scale factors and noise. Considering, e.g., a one-axis sensor $\tilde{q}$ measuring the angular rate q, the measurement is expressed as $\tilde{q} = k_q q + b_q + w_q$ where k_q is the scale factor, b_q the bias and w_q follows a zero-mean Gaussian distribution with standard deviation σ_q. These three parameters (for each sensor) typically belong to intervals characteristic of the IMU and during simulation they were given values taken at either bound of these intervals. In the simulation, the IMU is also assumed to suffer a delay of two time steps.

To illustrate the relevance of the approach, we consider the half-loss of propulsion η at time 10s followed by the locking of the rudder δ_{m} in place at time 15s. Figure 2 shows that the residuals allow the detection, isolation and identification of the two successive faults affecting the actuators. We have $\widetilde{r}^1_{21} \approx 0.5$, which means that half of the propulsion has been lost, and $r_{12} \approx 0.35$ rad, which means that the rudder is locked in place around this value. Residuals r_{11} and r_{21} react to the two faults successively, while $\widetilde{r}^1_{11}$ reacts to the propulsion fault only (but does not allow its identification due to the functions involved). The propulsion fault is an abrupt one and may be detected almost immediately, whereas the rudder locking is an incipient fault and is therefore detected with a delay.

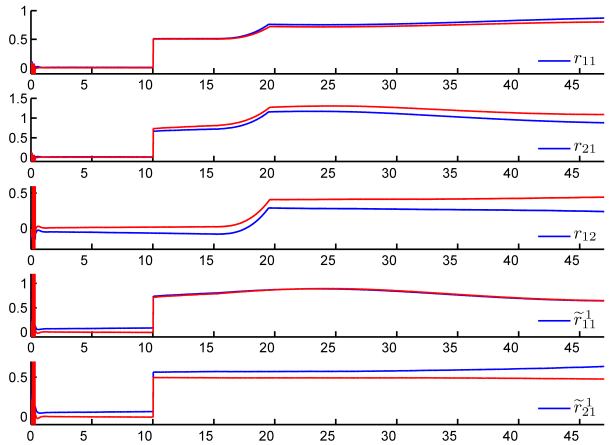

Fig. 2. Residuals for two successive faults for extreme values of IMU uncertainty parameters (blue: min, red: max)

Since functions $\widetilde{g}_{ij}$ are involved in the denominator of the residuals, their value should be checked. If one of the denominators is too close to zero at a considered time instant, the residual should not be taken into account for decision at this moment.

5. CONCLUSIONS AND PERSPECTIVES

A new strategy based on system-inversion for nonlinear fault diagnosis has been presented. The key point is to use *measured* derivatives of state variables instead of evaluating them numerically. The control-affine structure of the class of nonlinear systems considered allows the estimation of the control inputs as achieved by the actuators. Residuals are then obtained by comparing these estimates to the control input as computed by the control module.

This FDI strategy has been successfully applied to a 2D longitudinal aeronautical case study to detect, isolate and identify simultaneous faults, with IMU measurements affected by strong uncertainty. Robustness regarding model uncertainty and disturbances should also be considered in the future, along with sensor faults. The study will also be extended to the 3D case.

This paper was centered on fault-residual generation. A strategy for the analysis of these residuals remains to be chosen to decide when the residuals have reached a significant level. Adaptive thresholds and statistical tests should be considered. False-alarm and non-detection rates, detection delays and computational complexity will then need to be analyzed, and compared with those obtained by other FDI methods on the same aeronautical benchmark. The inner parameters of all these FDI strategies should be tuned to optimal performance by a systematic methodology to maintain as much objectivity as possible in the comparison.

REFERENCES

Bokor, J. and Szabó, Z. (2009). Fault detection and isolation in nonlinear systems. *Annual Reviews in Control*, 33(2), 113–123.

Chow, E. and Willsky, A. (1984). Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control*, 29(7), 603–614.

De Persis, C. and Isidori, A. (2001). A geometric approach to nonlinear fault detection and isolation. *IEEE Transactions on Automatic Control*, 46(6), 853–865.

Edelmayer, A., Bokor, J., Szabo, Z., and Szigeti, F. (2004). Input reconstruction by means of system inversion: A geometric approach to fault detection and isolation in nonlinear systems. *International Journal of Applied Mathematics and Computer Science*, 14(2), 189–200.

Hirschorn, R. (1979). Invertibility of multivariable nonlinear control systems. *IEEE Transactions on Automatic Control*, 24(6), 855–865.

Isermann, R. (1997). Supervision, fault-detection and fault-diagnosis methods: An introduction. *Control Engineering Practice*, 5(5), 639–652.

Isidori, A. (1995). *Nonlinear Control Systems: An Introduction*. Springer-Verlag, Berlin-Heidelberg.

Leuschen, M., Walker, I., and Cavallaro, J. (2005). Fault residual generation via nonlinear analytical redundancy. *IEEE Transactions on Control Systems Technology*, 13(3), 452–458.

Martinez-Guerra, R. and Diop, S. (2004). Diagnosis of nonlinear systems using an unknown-input observer: An algebraic and differential approach. *IEE Proceedings-Control Theory and Applications*, 151(1), 130–135.

Marzat, J., Piet-Lahanier, H., Damongeot, F., and Walter, E. (2009a). Autonomous fault diagnosis: State of the art and aeronautical benchmark. In *Proceedings of the 3rd European Conference for Aero-Space Sciences, Versailles*.

Marzat, J., Piet-Lahanier, H., Damongeot, F., and Walter, E. (2009b). A new model-free method performing closed-loop fault diagnosis for an aeronautical system. In *Proceedings of the 7th Workshop on Advanced Control and Diagnosis ACD'2009, Zielona Gora, Poland*.

Massoumnia, M., Verghese, G., and Willsky, A. (1989). Failure detection and identification. *IEEE Transactions on Automatic Control*, 34(3), 316–321.

Ramm, A. and Smirnova, A. (2001). On stable numerical differentiation. *Mathematics of Computation*, 70(235), 1131–1153.

Shumsky, A. and Zhirabok, A. (2006). Nonlinear diagnostic filter design: Algebraic and geometric points of view. *International Journal of Applied Mathematics and Computer Science*, 16(1), 115–127.

Witczak, M. (2007). *Modelling and Estimation Strategies for Fault Diagnosis of Non-Linear Systems: From Analytical to Soft Computing Approaches*. Springer-Verlag, Berlin-Heidelberg.