Structural Analysis Approach for the Generation of Structured Residuals for Aircraft FDI

M. L. FRAVOLINI
V. BRUNORI
Perugia University
G. CAMPA, Member, IEEE
M. R. NAPOLITANO
West Virginia University
M. LA CAVA
Perugia University

A systematic methodology is described for calculating structured residuals with high fault diagnostic capabilities for detecting sensor and actuators failures. The effort addresses implementation issues for real-time applications such as residual computation complexity and sensitivity to measurement noise. These specific requirements have been rigorously introduced through a cost function measuring the quality of the residual signal. A structural analysis approach of the nonlinear model of the system in conjunction with the unknown variables elimination method is used to derive subsets of residual equations. An algorithm is proposed for selecting the residual equations with maximum "failure isolability" and minimum cost, according to the selected performance criteria. The methodology has been applied to the design of a real-time residual generator for a nonlinear model of a remotely controlled semi-scale YF-22 research aircraft.

Manuscript received May 18, 2007; revised February 4, 2008; released for publication June 12, 2008.

IEEE Log No. T-AES/45/4/935105.

Refereeing of this contribution was handled by T. Robertazzi.

Authors' addresses: M. L. Fravolini, V. Brunori, M. La Cava, Dept. of Electronic and Information Engineering, Perugia University, 06100 Perugia, Italy, E-mail: (fravolini@diei.unipg.it); G. Campa, The Mathworks, 400 Continental Blvd., El Segundo, CA 90245; M. R. Napolitano, Dept. of Mechanical and Aerospace Engineering, West Virginia University, Morgantown, WV 26506-6016.

I. INTRODUCTION

Research on fault detection and isolation (FDI) has traditionally been an important topic in avionics [1–3]. Flight control systems for high performance aircraft (both commercial and military) are required to cope with detection and identification of failures of different components, such as sensors, actuators, and flight computers [4–5]. The importance of a correct and prompt FDI cannot be underestimated since undetected and/or misclassified faults may lead to unstable and/or unrecoverable flight conditions. Typically, FDI methods are based on the comparison between the behavior of the plant and the response of mathematical model. Two main conceptual approaches are currently pursued. The first approach is based on dynamic observers [6, 7] while the second approach is based on the use of nonlinear analytical redundancy relations (ARRs) [8–10]. The effort described in this paper is related to the ARR methodology. Particularly, in ARR-based FDI, the model equations are manipulated to derive diagnostic signals (called ARRs, residuals, or consistency relations) which are based only on measurable variables. Ideally, ARRs are zero in fault-free conditions and are different from zero after the occurrence of a failure. Each ARR is characterized by its binary fault signature indicating the set of faults to which the ARR is sensitive. The need for isolating different faults requires the simultaneous evaluation of multiple ARRs (called "structured residuals" subset) whose signatures define the binary signature matrix. Those ARRs should be designed such that different faults have different binary codes in the signature matrix thus allowing the unambiguous isolation of the faults.

Methodologies for the design of structured ARRs have been extensively explored for linear systems and are also known as parity space approach methods [11, 12]. However, the design of structured residuals for nonlinear systems is a more complex task. To date, a general approach to deal with this problem has not been introduced. Specific ad hoc methods have been proposed for restricted classes of nonlinear systems where the design of ARRs having desired fault sensitivities is formulated as a general problem of variables elimination from the set of relations constituting the mathematical model of the system. For differential algebraic systems a constructive method based on Ritt's algorithm was proposed in [13]. Similarly nonlinear structured ARRs were calculated through exploitation of the properties of the Groebner basis in [14].

Structural analysis (SA) [15–17] is a powerful methodology that has been proposed for deriving information about the part of the system from which ARRs could be generated. However, SA only deals with the structural information contained in the model,

i.e., in which variables appear in each equation, and it does not account for other information enclosed in the analytical model. Furthermore, while SA can provide indication on the variable elimination sequences that allow the generation of ARRs having desired fault sensitivity, it does not address implementation issues of ARRs for real-time applications. In fact, issues such as ARR computational complexity, equations bijectivity respects to the involved variables, requirements for eliminating from the ARRs undesired noise-corrupted variables and high-order derivatives are seldom considered. Some efforts dealing with the computational aspects of ARRs are described in [18–20]. The effort here described is oriented toward the design of structured ARRs with desired fault sensitivity in a systematic way while also accounting for computational issues. Specifically, the main contribution is the formulation of an automated methodology that, based on a large set of ARRs generated from the SA of the model, selects a "suboptimal" subset of structured ARRs while minimizing a performance index which is defined with the main objective of allowing real-time applications of the FDI scheme.

The proposed methodology has been tested using the nonlinear mathematical model of a semi-scale YF-22 research aircraft built at West Virginia University. Different optimality criteria have been proposed, and the corresponding optimal structured ARRs have been evaluated. Time domain simulation studies have been performed for assessing the detection and isolation capabilities of the residual generators.

II. FAULTS AND NONLINEAR AIRCRAFT MODELING

A. Fault Modelling

In this work, failure of actuators and sensors are considered; the faults are modeled as independent additive signals f_i as shown in Fig. 1. The signals x_i represent "physical" signals that are not directly available but are known through their measurement provided by the sensors y_i. The vector of internal variables is defined as X while the vector of sensors measurements is defined as Y. Similarly, system input signals m_i represent the "physical" signals delivered by the actuators while the actuator command signals (actuator inputs) are named u_i. The vector M (having components m_i) represents the "physical" system inputs while U (having components u_i) represents the vector of command inputs. Occasionally, due to the occurrence of faults, a signal x_i may be different from y_i, and m_i may be different from u_i. In this schematization, signals belonging to X and M are considered as "internal" unmeasurable unknown

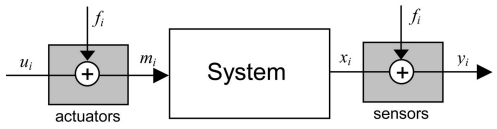


Fig. 1. Additive faults acting on sensors and actuators.

signals while signals belonging to U and Y are considered as "external" measurable known signals. Fault signals f_i are considered as unknown signals.

B. Aircraft Modeling

Consistency relations were derived from the nonlinear mathematical model of the semi-scale YF-22 research aircraft, shown in Fig. 2. The model equations were derived following the approach outlined in [21, 22], while the modeling parameters were computed through a nonlinear parameter identification (PID) analysis performed on several collections of flight data, as described in [23] and used for this same purpose. The aircraft dynamics is described by the following differential equations (denoted also as relationships r_j):

$$
\begin{aligned}
r_1 &: \dot{V} - \frac{F_X\cos(\alpha)\cos(\beta) + F_Y\sin(\beta) + F_Z\sin(\alpha)\cos(\beta)}{m} \\
&\quad = 0 \\
r_2 &: \dot{\alpha} - \frac{-F_X\sin(\alpha) + F_Z\cos(\alpha)}{mV\cos(\beta)} - q \\
&\quad + (p\cos(\alpha) + r\sin(\alpha))\tan(\beta) = 0 \\
r_3 &: \dot{\beta} - \frac{-F_X\cos(\alpha)\sin(\beta) + F_Y\cos(\beta) - F_Z\sin(\alpha)\sin(\beta)}{mV} \\
&\quad - p\sin(\alpha) + r\cos(\alpha) = 0 \\
r_4 &: \dot{p} - P_lL - P_nN - P_{pq}pq - P_{qr}qr = 0 \\
r_5 &: \dot{q} - Q_mM - Q_{pp}p^2 - Q_{pr}pr - Q_{rr}r^2 = 0 \\
r_6 &: \dot{r} - R_lL - R_nN - R_{pq}pq - R_{qr}qr = 0 \\
r_7 &: \dot{\psi} - \frac{q\sin(\phi) + r\cos(\phi)}{\cos(\theta)} = 0 \\
r_8 &: \dot{\theta} - q\cos(\phi) + r\sin(\phi) = 0 \\
r_9 &: \dot{\phi} - p - \dot{\psi}\sin(\theta) = 0 \\
r_{10} &: \dot{h} - u\sin(\theta) + [(v)\sin(\phi) + (w)\cos(\phi)]\cos(\theta) = 0.
\end{aligned}
\tag{1}
$$

In (1) V is the airspeed speed; α is the angle of attack; β is angle of sideslip; p, q, r are the angular rates; ψ, θ, φ are the attitude angles; and h is the altitude. Since, the proposed FDI scheme was tested on all the state variables except x and y (see Section VII), the respective dynamics equations were not included in the available constraints. This simplification does not affect the generality of the proposed approach.

In (1) the total forces and moments acting on the aircraft expressed in the body-axes are

$$\begin{aligned} F_X &= X_a + X_T + X_{gr} & L &= L_a + L_T \\ F_Y &= Y_a + Y_T + Y_{gr} \quad \text{and} & M &= M_a + M_T \\ F_Z &= Z_a + Z_T + Z_{gr} & N &= N_a + N_T \end{aligned} \tag{2}$$

where the T subscript indicates the contribution from the engine thrust force. The aerodynamic forces and moments in (2) generated by the control surface deflections are given by

$$\begin{aligned} X_a &= C_{Xa} q_{\text{dyn}} S & L_a &= C_{la} q_{\text{dyn}} S b \\ Y_a &= C_{Ya} q_{\text{dyn}} S \quad \text{and} & M_a &= C_{ma} q_{\text{dyn}} S \bar{c} \\ Z_a &= C_{Za} q_{\text{dyn}} S & N_a &= C_{na} q_{\text{dyn}} S b \end{aligned} \tag{3}$$

where S is the aircraft wing surface, b is wingspan and $\bar{c}$ is the mean aerodynamic chord. The variable $q_{\text{dyn}} = 0.5\rho V^2$ is the aerodynamic dynamic pressure; this variable depends on temperature air density ρ, velocity V, and altitude and represents the basic measurement required to compute the aerodynamic force and moments. In (3) aerodynamic coefficients are used to represent the aerodynamic forces and moments acting on the aircraft [21], these are expressed by affine functions of the aircraft states and inputs through the following relations:

$$\begin{aligned} C_{Xa} &= C_{X0} + C_{X\alpha}\alpha + C_{Xde}\delta_e \\ C_{Ya} &= C_{Y0} + C_{Y\beta}\beta + C_{Yp}\frac{pb}{2V} + C_{Yr}\frac{rb}{2V} \\ &\quad + C_{Yda}\delta_a + C_{Ydr}\delta_r \\ C_{Za} &= C_{Z0} + C_{Z\alpha}\alpha + C_{Zq}\frac{q\bar{c}}{2V} + C_{Zde}\delta_E \\ C_{la} &= C_{l0} + C_{l\beta}\beta + C_{lp}\frac{pb}{2V} + C_{lr}\frac{rb}{2V} \\ &\quad + C_{lda}\delta_a + C_{ldr}\delta_r \\ C_{ma} &= C_{m0} + C_{m\alpha}\alpha + C_{mq}\frac{q\bar{c}}{2V} + C_{mde}\delta_e \\ C_{na} &= C_{n0} + C_{n\beta}\beta + C_{np}\frac{pb}{2V} + C_{nr}\frac{rb}{2V} \\ &\quad + C_{nda}\delta_a + C_{ndr}\delta_r. \end{aligned} \tag{4}$$

In (4) δ_e, δ_s, and δ_r are thrust, elevators, aileron and rudders input deflections, respectively.

The body axes components of the gravity force in (2) are

$$\begin{aligned} X_{gr} &= -mg_0 \sin(\theta) \\ Y_{gr} &= mg_0 \cos(\theta)\sin(\phi) \\ Z_{gr} &= mg_0 \cos(\theta)\cos(\phi) \end{aligned} \tag{5}$$

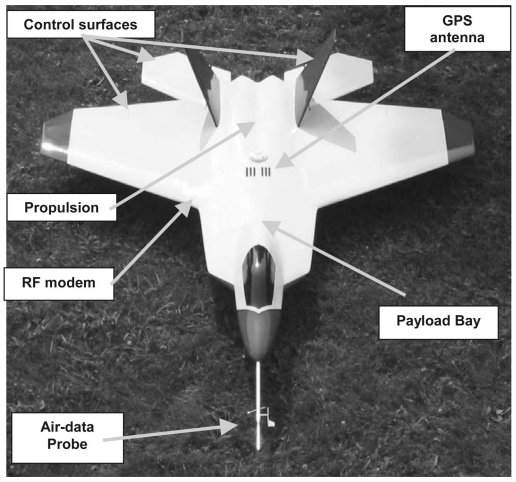


Fig. 2. YF-22 semi scale model.

TABLE I
YF-22 Modeling Parameters

$b = 1.96$	$S = 1.36$	$m = 20.63$	$C_{X0} = 0.0068$	$C_{Y0} = 0.020$
$C_{Z0} = 0.0038$	$C_{l0} = -0.0015$	$C_{m0} = 0.0063$	$C_{n0} = 0.00013$	$C_{X\alpha} = 0.43$
$C_{Z\alpha} = 2.45$	$C_{m\alpha} = -0.23$	$C_{Y\beta} = 0.30$	$C_{l\beta} = -0.045$	$C_{n\beta} = 0.054$
$C_{Yp} = 0.83$	$C_{lp} = -0.22$	$C_{np} = -0.11$	$C_{Zq} = 0.035$	$C_{mq} = -2.69$
$C_{Yr} = -1.077$	$C_{lr} = 0.09$	$C_{nr} = -0.26$	$C_{X\delta e} = -0.24$	$C_{Zde} = -0.32$
$C_{m\delta e} = -0.26$	$C_{Yda} = 0.21$	$C_{l\delta a} = -0.05$	$C_{nda} = -0.02$	$C_{Ydr} = -0.44$
$C_{ldr} = 0.01$	$C_{ndr} = -0.06$	$P_l = 0.62$	$P_n = -0.02$	$P_{pq} = -0.02$
$Pqr = 0.19$	$Q_m = 0.13$	$Q_{pp} = 0.03$	$Q_{pr} = 0.74$	$Q_{rr} = -0.03$
$R_l = -0.02$	$R_n = 0.13$	$R_{pq} = -0.82$	$R_{qr} = 0.02$	$\bar{c} = 0.76$

$g_0 = 9.80\ \text{ms}^{-2}$ $p_s = 97190\ \text{Nm}^{-2}$

where m is the aircraft mass and g_0 is the gravitational acceleration at sea level. The components of the total aircraft velocity in the body-axes are

$$\begin{aligned} u &= V\cos(\alpha)\cos(\beta) \\ v &= V\sin(\beta) \\ w &= V\sin(\alpha)\cos(\beta). \end{aligned} \tag{6}$$

Parameters P_l, P_n, P_{qp}, P_{qr}, Q_l, Q_{pp}, Q_{pr}, Q_{rr}, R_l, R_n, R_{pq}, R_{qr}, in (1) are coefficients that are derived from of the aircraft inertia matrix [22].

The complete set of values for the modeling parameters identified for the YF-22 model are reported in Table I. As discussed in [23], the performance of the identified nonlinear model was deemed satisfactory, since for a wide range of aircraft inputs, there was a substantial agreement between the flight data and the data obtained from the simulation of the nonlinear model.

C. Sensors and Actuators

The proposed FDI approach was tested on a subset of sensors and actuators that are installed on the actual semi-scale YF-22. A total of 15 faults were considered including 4 on the actuators and 11 on the

sensors:

1) *Actuators*: The considered control surfaces are the ailerons, stabilators, and rudders; these are indicated respectively with δ_a, δ_e, δ_r; the fourth actuator is the engine thrust force X_T. Considering the notation introduced in figure 1, the physical quantities supplied by the actuators are grouped in the vector M:

$$M = [X_T \quad \delta_e \quad \delta_a \quad \delta_r] \tag{7}$$

while the vector of the commands generated by the UAV control systems is

$$U = [X_t \quad \delta_e \quad \delta_a \quad \delta_r]_{\text{command}}. \tag{8}$$

2) *Sensors*: The vector X of the 11 selected internal variables is defined as

$$X = [V \quad \alpha \quad \beta \quad p \quad q \quad r \quad \psi \quad \theta \quad \phi \quad h \quad q_{\text{dyn}}]. \tag{9}$$

In (9) variables $x_1 - x_{10}$ characterize the dynamics of the aircraft described by (1) while $x_{11}(q_{\text{dyn}})$ is needed to calculate the aerodynamic forces and moments.

The output sensor signals are grouped together into the output measurement vector:

$$Y = [V \quad \alpha \quad q \quad \theta \quad h \quad \beta \quad p \quad r \quad \phi \quad \psi \quad q_{\text{dyn}}]_{\text{measured}}. \tag{10}$$

In (9) and (10) the variables have been reordered so that y_1–y_5 are associated with the longitudinal dynamics while y_6–y_{10} are associated with the lateral dynamics.

D. Measurement Equations

The measurement equations for the 4 actuators are

$$r_{10+i} : u_i - (m_i - f_i) = 0, \qquad i = 1 \ldots 4 \tag{11}$$

where $f_1 \ldots f_4$ are additive functions modeling actuator faults. Similarly, the measurement equations for the 11 sensors are given by

$$\begin{aligned}
r_{15} &: y_5 - (x_1 + f_5) = 0 \\
r_{16} &: y_6 - (x_2 + f_6) = 0 \\
r_{17} &: y_7 - (x_5 + f_7) = 0 \\
r_{18} &: y_8 - (x_8 + f_8) = 0 \\
r_{19} &: y_9 - (x_{10} + f_8) = 0 \\
r_{20} &: y_{10} - (x_3 + f_{10}) = 0 \\
r_{21} &: y_{11} - (x_4 + f_{11}) = 0 \\
r_{22} &: y_{12} - (x_6 + f_{12}) = 0 \\
r_{23} &: y_{13} - (x_9 + f_{13}) = 0 \\
r_{24} &: y_{14} - (x_7 + f_{14}) = 0 \\
r_{25} &: y_{15} - (x_{11} + f_{15}) = 0
\end{aligned} \tag{12}$$

where $f_5 \ldots f_{15}$ are the fault functions acting on the 11 sensors.

From an SA point of view the YF-22 structural model consists of 25 relationships $r_1, \ldots r_{25}$, that depend on $N_{uy} = 15$ known (measurable) inputs and output variables ($U \cup Y$), on $N_{xm} = 15$ unknown (not measurable) variables ($M \cup X$), and on $N_f = 15$ unknown fault variables F.

III. STRUCTURAL ANALYSIS FOR FDI

A structural model of a system consists of a set of relations $R = \{r_1, r_2, \ldots, r_m\}$ which depend on a set of variables $Z = \{z_1, z_2, \ldots, z_n\}$. Analytically, the relationships are expressed as follows:

$$r_j(z_1, z_2, \ldots z_n) = 0 \tag{13}$$

where r_j represents a dynamic or static, linear or nonlinear equation.

In this work the relations set R is constituted of the set of the 25 relations $R = \{r_1, r_2, \ldots, r_{25}\}$ while the set Z is the union of all known and unknown variables $Z \equiv U \cup M \cup X \cup Y \cup F$.

Usually, the set of relations R is not directly usable for the real-time computation of residual signals, because these may depend on unmeasurable variables (x_i, m_i). However, by appropriately combining the relations in R it is possible to derive special subsets of relations where all the unknown variables x_i, and m_i are eliminated so that these relations depend only on measurable input-output variables (u_i, y_i) and on unknown fault variables f_i. These input-output relations will be referred to as AARs. In order to generate AARs with desired faulty sensitivity it is important to investigate the possibility of solving for (eliminating) unknown variables from the set of constraints R. The possibility of eliminating variables in a relation r_j is based on the concept of "calculability."

A. Calculability

A variable z_i is said to be "calculable" from a relation r_j if its value can be derived from the relationship r_j under the condition that the values of the other variables z_k $(k \neq i)$ are known. In this study the "calculability" property was not associated with the theoretical existence of a local inverse function g_j (implicit function theorem [17]) such that locally $z_i = g_j(z_1, z_{i-1}, z_{i+1}, \ldots, z_n)$. Instead, it was restricted to the possibility of actually solving analytically the relationship r_j as a function of z_i. This definition of calculability was motivated by the practical need for automatically solving (through a symbolic computation software) the constraint r_j in function of a dependent variable z_i. The structural model associated with the relation set R and their calculability property were represented using the

following constrained structural matrix S_C:

$$S_C(i,j) = \begin{cases} 2 & \text{iff } r_j(\cdot) \text{ depends on } z_i \text{ and } z_i \text{ is not calculable from } r_j(\cdot) \\ 1 & \text{iff } r_j(\cdot) \text{ depends on } z_i \text{ and } z_i \text{ is calculable from } r_j(\cdot) \\ 0 & \text{iff } r_j(\cdot) \text{ does not depend on } z_i \end{cases}. \tag{14}$$

B. Note on Variables Calculability

Generally, the relationships r_j are differential involving a variable z_i and its time derivatives. While the knowledge of the variable z_i at each time instant allows the computations of its time derivatives—at least in their approximated form—the opposite is generally not true since the integration process requires the initial condition value that is usually unknown [17]. For this reason, a relation r_j that depends on z_i and its time derivatives is considered unsolvable for that variable and a "2" is placed in $S_C(i,j)$ indicating that z_j must be calculated from another relationship. During the variable elimination process, once z_i becomes available, its time derivatives are calculated through analytic differentiation. Following this approach, time derivatives are not considered as additional variables but simply as variables that are calculable only after the explicit calculation of z_i.

C. Fault Signatures of an ARR

The set of faults to which an ARR is sensitive is called the "ARR fault signature." The grouping of the ARR fault signatures generates the signature matrix S_F, which is defined as follows:

$$S_F(i,j) = \begin{cases} 1 & \text{iff } \mathrm{ARR}_j(\cdot) \text{ depends on } f_i \text{ or its derivatives} \\ 0 & \text{otherwise} \end{cases}. \tag{15}$$

A fault f_i is said to be detectable if there is at least one non-null entry in the ith row of S_F. Two faults f_1, f_2 are structurally isolable if the i_1th and the i_2th rows of S_F are different. The calculation of the S_F matrix is therefore critical for the purpose of analyzing the detection and isolation property of a subset of ARRs. In practice, for real-time fault isolation purposes, it is desirable to evaluate a minimum number of ARRs whose S_F matrix guarantees the isolation of a maximum number of faults. Such subset are referred to as minimal isolating ARR subset (MI-ARRS). Generally, the MI-ARRS is not unique.

IV. APPROACH TO ARRS GENERATION

In general SA does not allow the generation of residual equations; however, the analysis of the structural table can provide guidelines toward evaluating the set of constraints from which ARRs can be generated [16]. A single specific methodology for the generation of ARRs does not exist. If fact, any mathematical manipulation of the constraint equations leading to the generation of consistency relations that depend only on known (U and Y) variables can be considered a valid approach for the generation of ARRs. A simple method for the generation of ARRs is based on the variables elimination technique. In this method a constraint is used to calculate an unknown variable z_i as a function of the others; this expression is then substituted in all the remaining equations that depend on z_i. Iterating this scheme it is possible to eliminate sequentially all the unknown variables. For FDI purposes the case when there are more relations r_j than unknown variables z_i is of practical interest. In this case some relations are used to eliminate the unknown variables and the remaining relations are used as possible ARR for FDI. The sequence of variables elimination and the selection of the constraints from which a variable can be calculated are in general not unique; in fact, different elimination sequences would lead to different ARRs. The association between variables and constraints is known as "matching" [15]. Specifically, a matching is an assignment, which associates some system variables with the system constraints from which they can be calculated. A matching is said complete—with respect to a certain subset of unknowns—if each unknown variable can be associated with a relation belonging to a subset of the available constraint equations. Since there are different alternatives for matching the set of unknown variables to the set of relations, it is important to define algorithms able to calculate all the possible complete matching sets that can be originated from the SA of the S_C matrix. It should be noticed that the calculation of the entire set of ARRs that could be theoretically generated from the S_C matrix would involve the calculation of all the complete matching sets between unknowns and relations. Among the unknown variables, a different role is played by fault variables F, and the X and M variables. In fact, for FDI purposes, a matching is useful only if it generates ARRs that are insensitive to all the X and M variables but are sensitive to some fault variables of F. Therefore, while all the X and M variables must always be eliminated in the matching process, in order to generate ARRs that are sensitive only to a defined subset F_o of the faulty variables ($F = F_e \cup F_o$ and $F_e \cap F_o = \emptyset$), it is necessary to consider the remaining subset of fault variables F_e as the "true unknown" fault variables to be eliminated.

A way to generate ARRs having all the possible fault signatures is to exhaustively repeat the unknowns-relations matching process for each one of the possible combination of faults in the subset F_e. For each matching process, the set X_e containing all the variables that have to be eliminated ($X_e = X \cup M \cup F_e$) may contain a different number of variables N_e, depending on the actual number of fault variables contained in the particular combination of faults in F_e.

A. The Algorithm for the Generation of ARRs

For each specific fault signature (i.e., for each subset of variables X_e), the calculation of the associated ARRs that can be derived from the set of available relations R is approached as an unknown variable elimination problem based on the constrained S_C matrix. An algorithm that solves the variables elimination problem by eliminating one unknown variable at the time was implemented.

The algorithm performs the matching by generating a tree structure where each node contains the information of the matched/unmatched relations and unknowns. That is, different nodes represent different matchings. A branch that connects a source node to a destination node represents the selection of an additional calculable variable/relation matching between an unknown unmatched variable and an unmatched relation of the source node.

A branch at level k may generate one (or more) branches at level $k + 1$. A new branch and a new destination node is added if the following two conditions are both met.

1) By considering all the unknowns and relations matched in a source node, there is an unmatched computable unknown z_i that can be matched to an unmatched relation r_j.

2) The unmatched unknown z_i found in condition 1 is the only unmatched unknown left in the relation r_j.

Level after level the unknown variables are eliminated. The nodes at level N_e (if the algorithm is able to reach that level) have surely eliminated all the N_e unknown variables and therefore contain complete matches. For each complete match, the sequence of branches crossed to go from level 1 to level N_e indicates the unknown variable elimination sequence, while the set of ARRs is simply the set of the remaining relations that do not belong to the complete matching (associated to X_e).

While this algorithm is able to find all the matches than can be generated by the sequential elimination of unknown variables one at the time (satisfying the above condition 1 and 2, it does not allow the calculation of complete matching that requires the simultaneous elimination of 2 or more unknown variables. In practice, as it will be seen later, models used within practical FDI applications often feature a number of output equations, each one featuring only one variable to be eliminated. In these cases the algorithm is able to yield a relevant number of ARRs. The pseudocode of the algorithm described in this section is reported in Table II.

TABLE II
The Pseudocode of ARRs Algorithm

For each one of the possible configurations of the eliminating variables X_e
  Define the subset of unknown variables
  Define the subset of known variables

  Number of Levels = N_e = Number of eliminating unknowns

  Level 1
  Find the subset J_1 of the columns of S_M having a single "1," no "2," and all "0" not considering the rows associated to known variables
  For each column $j_1 \in J_1$:
    Match the relation j_1 to the unknown variable i_1 such that $S_M(i_1, j_1) = 1$
    Create a new branch k_1 at level-1: Branch-1(k_1)
    Create the matching list: Match-1(k_1) = (i_1,j_1); increment k_1
  End
  Level 2
  For each Branch-1(k_1)
    Find the subset J_2 of the columns of S_M having a single "1," no "2," and all "0" not considering the rows associated to known variables and the rows and columns already matched in the list Match-1(k_1)
    For each column $j_2 \in J_2$:
      Match the relation j_2 to the unknown i_2 such that $S_M(i_2, j_2) = 1$
      Create a new branch at level-2: Branch-2(k_2)
      Match-2(k_2) = Match-1(k_1)
      Add to the list Match-2(k_2) the match (i_2,j_2); increment k_2
  End
……
……
  Level N_e
  …
  End
  For each Branch-N_e
  ARR-RELATIONS = find in S_M the relations not matched in Match-N_e
  end
End

B. Simple Example

A simple example is reported to illustrate the algorithm-matching process. Considering the following system:

$$\begin{aligned} r_1 &: \dot{x}_1 = -x_1 + 2x_2 + u_1 \\ r_2 &: \dot{x}_2 = +x_1 - 3x_2 + u_2 \\ r_3 &: x_1 = x_2 + u_3 \\ r_4 &: y_1 = x_1 + f_1 \\ r_5 &: y_2 = x_2 + f_2 \end{aligned} \tag{16}$$

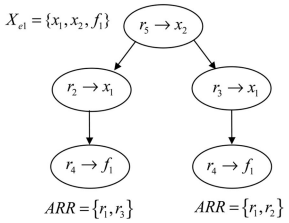


Fig. 3. Example of matching process.

TABLE III
The S_C Matrix for the Illustrative Example

Var	r_1	r_2	r_3	r_4	r_5
x_1	2	1	1	1	0
x_2	1	2	1	0	1
f_1	0	0	0	1	0
f_2	0	0	0	0	1

the set of available constraints is $R = \{r_1, r_2, r_3, r_4, r_5\}$, the set of internal variables is $X = \{x_1, x_2\}$, the set of measured variables is $Y = \{y_1, y_2\}$, while the set of control commands is $U = \{u_1, u_2, u_3\}$. In this example no faults were assumed on the actuators therefore $M \equiv U$ and for this reason M is omitted; the set of (sensor) faults is $F = \{f_1, f_2\}$. The constrained structural matrix S_C for the system (16) is reported in Table III. For this system there are two possible choices for the subsets of the variables to be eliminated: $X_{e1} = \{x_1, x_2, f_1\}$ and $X_{e2} = \{x_1, x_2, f_2\}$. If, for instance, we consider the computation of the possible complete matchings between the relations set R and the set of unknown elimination variables X_{e1}, then the algorithm will generate ARRs that are sensitive to f_2. If matches are calculated with respect to X_{e2} then ARRs that are sensitive to f_1 will be obtained. Fig. 3 shows the process of matching relations and variables with respect to X_{e1}. Considering, for instance, the left path, first, variable x_2 is computed from r_5; then x_1 is computed from r_2 (the computation of x_1 allows also the computation $\dot{x}_1$); finally f_1 is computed from r_4. The unmatched relations are r_1, and r_3, which constitute the ARRs associated to the X_{e1} matching.

Following the elimination sequence for the left path in Fig. 3, the following expression for the ARRs = $\{r_1, x_3\}$ can be computed:

$$\begin{aligned} \mathrm{ARR}_1 &= \ddot{y}_2 - \ddot{f}_2 + 4\dot{y}_2 - 4\dot{f}_2 - \dot{u}_2 + y_2 - f_2 - u_2 - u_1 \\ \mathrm{ARR}_3 &= \ddot{y}_2 - \ddot{f}_2 + 3\dot{y}_2 - 3\dot{f}_2 - \dot{u}_2 - y_2 + f_2 - u_3. \end{aligned} \tag{17}$$

Similarly, ARRs = $\{r_1, r_2\}$ for the right path can be computed. The matching for X_{e2} can be easily computed following the same procedure.

V. CRITERIA FOR ARRS RANKING

After the computation of the ARRs, originated from all the possible combinations of the elimination variables, the next step is the selection of the MI-ARR subsets. Since many of such equivalent subsets can be obtained from the calculated ARRs, the selection of the "best" MI-ARRS was performed using an optimality criterion. A priority selection was first introduced through ranking the ARRs using a cost function J. An ARR with a lower cost function J should have a higher selection priority than an ARR with a higher cost function. Different cost functions were analyzed; some of them required the explicit analytic calculation of the ARRs.

A. Derivative Cost

It is well known that the numerical calculation of the time derivative of a signal is usually unreliable due to the presence of the high frequency measurement noise. For this reason, the calculation of high-order derivatives should be avoided or minimized in the ARRs. Therefore, a quadratic cost function penalizing high-order derivatives was defined for the generic jth ARR:

$$J_1(j) = \sum_{i=1}^{Nuy} \mathrm{Der}_j(i)^2 \tag{18}$$

where $\mathrm{Der}_j(i)$ is equal to the maximal order of derivation of a variable y_i or u_i plus one. That is, if a variable i is not present in the ARR then $\mathrm{Der}_j(i) = 0$; if the variable is present but not its time derivatives then $\mathrm{Der}_j(i) = 1$. Note that the cost function (18) penalizes also the presence of the nonderived signals; this penalization was appositely inserted to favor the ranking of ARRs having a small number of variables.

B. Computational Cost

In real time applications the analytical complexity of an ARR is an important issue. In fact, dealing with complex ARRs may require substantial computational resources. Therefore, an index J_2 was introduced for the specific purpose of quantifying the ARR computational cost using the relationship,

$$\begin{aligned} J_2(j) = {} & \alpha \cdot \mathrm{multiplications}_j + \beta \cdot \mathrm{additions}_j \\ & + \chi \cdot \mathrm{divisions}_j + \delta \cdot \mathrm{functions}_j \end{aligned} \tag{19}$$

where multiplications, additions, divisions, and functions represent the number of operations required for the calculation of the ARR. The coefficients α, β, χ, δ, are weight coefficients; for simplicity the values $\alpha = \beta = \chi = \delta = 1$ were used in the following analysis.

C. Maximum Fault Independence Cost

Each ARR is characterized by a specific binary fault signature. A signature with many "1" is sensitive to many faults. In practical situations, if the gain

coefficient of a fault in the ARR is too small, then the increase of the residual signal following the occurrence of that fault could be less than the corresponding detection threshold and therefore unable to trigger the binary transition from zero to one, generating as a result an erroneous binary fault code. This problem is clearly not present within the ARRs that do not depend on that fault, since both before and after the fault the residual signal generated by such ARRs is simply not affected by that fault variable. In other words while following a fault occurrence a 1 can be equivocated with a "0," the opposite—that is a 0 equivocated with a 1 due to noise when no fault is present—is generally less likely. Therefore, whenever the weak coupling of some faults in the ARRs is a concern, it might be preferable to select ARRs having signatures with a minimum number of 1, so that the ARRs will have a "maximum fault independence." A cost function that penalizes the presence of 1 was defined by simply counting the number of 1 in the jth row of the sensitivity matrix S_F:

$$J_3(j) = \sum_{i=1}^{N_F} S_F(i,j). \tag{20}$$

D. Maximum Fault Dependence Cost

When fast detection of faults is a fundamental issue for the safety of the system, then it could be desirable to select ARRs that are sensible to a large number of fault ("maximum fault dependence") so that a fault occurrence can be detected by more than one ARRs. A cost function that favors the selection of ARRs with a large number of 1 is given by

$$J_4(j) = \sum_{i=1}^{N_F} NOT(S_F(i,j)). \tag{21}$$

This cost function essentially counts the number of faults that a given ARR is insensitive to.

VI. CRITERIA FOR THE SECTION OF A SUBOPTIMAL MI-ARRS

The purpose of the following algorithm is to extract from the set of the ranked ARRs a MI-ARRS having a minimum cumulative cost J. In the proposed strategy the MI-ARRS is generated sequentially by selecting one ARR at a time. The algorithm works as follows. The best-ranked ARR is selected and its fault binary signature is calculated. Next, the subsequent ARRs in the ranked list are sequentially evaluated (following an increasing cost order). A new ARR is selected only if its binary signature allows increasing the number of groups of faults that can be isolated compared with the current set of faults that can be isolated from the current set of selected ARRs. An ARR is discarded if its signature does not increase the number of isolation groups. The algorithm stops when the number of isolable faults equals the maximum number of faults N_f or when all the possible ARRs have been evaluated. Since all the AARs are potentially evaluated, the algorithm selects a subset of ARRs that allows isolating a maximum subset of faults; therefore, an MI-ARRS is achieved. Since the ranked AARs are analyzed sequentially the algorithm favors the selection of ARRs having a suboptimal minimal cost. Clearly, the number of isolable faults in the MI-ARRS does not depend on the ranking criteria J; however, different cost functions J may lead to the selection of MI-ARRS having a variable number of ARRs.

A. Implementation Details

The SA part of the code was implemented in Matlab. The inputs of this code are the S_C matrix along with the state, input, output, and fault variables. The code automatically performs the SA and finds the set of complete matches and the related ARRs. The results of this analysis are saved to a file and later retrieved by a Maple code containing the symbolic dynamic model equations of the YF-22. The code computes the symbolic expression of the ARRs and evaluates the cost according to the selected performance index J. Finally, the results from the Maple code are saved in a different file and retrieved by the Matlab algorithm performing the selection of the MI-ARRS. It should be clarified that the proposed design procedure has to be repeated only once in the analysis phase; therefore, the required computational effort is not a limiting factor. For the YF-22 model the procedure for the ARR generation required 1537 s, the Maple code for the analytical calculation of the ARR and cost required 6465 s, while the procedure for the selection of the MI-ARRS required 1.55 s on a desktop computer featuring a 2.6 GHz Pentium-4 processor with, 1024 Megabytes of RAM.

VII. APPLICATION OF THE FDI SCHEME TO THE MATHEMATICAL MODEL OF THE WVU YF-22 AIRCRAFT

A. Structural Analysis Results

The starting point for the application of the FDI scheme to the YF-22 is the computation of the S_C matrix that relates the 25 relations (introduced in Section II) to the unknown and known variables. The S_C matrix for the YF-22 is shown in Table IV. Due to space constraints only the dependence of the relationships on the 25 (X,U,Y) variables is reported.

TABLE IV
The Constrained Structural Matrix for the YF-22 Aircraft

	X										U				Y										
	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	r_{11}	r_{12}	r_{13}	r_{14}	r_{15}	r_{16}	r_{17}	r_{18}	r_{19}	r_{20}	r_{21}	r_{22}	r_{23}	r_{24}	r_{25}
x_1	2	2	2	2	2	2	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
x_2	2	2	2	0	1	0	0	0	0	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
x_3	2	2	2	1	0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
x_4	1	1	1	2	2	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
x_5	0	1	0	1	2	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
x_6	1	1	1	1	2	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
x_7	0	0	0	0	0	0	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
x_8	2	2	2	0	0	0	2	2	2	2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
x_9	2	2	2	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
x_{10}	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
x_{11}	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
f_1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_2	1	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
f_3	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
f_4	1	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
f_5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
f_6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
f_7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
f_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
f_9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
f_{10}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
f_{11}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
f_{12}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
f_{13}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
f_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
f_{15}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

TABLE V
Performance Comparison for the Difference Cost Functions

Optimization Criteria	Number of ARRs	NOT Isolable Faults	Cost of the selected ARRs in MI-ARR	J_1 Tot	J_2 Tot	J_3 Tot	J_4 Tot
J_1: Derivative	9	8–13	[7 8 8 9 10 10 12 12 14]	**90**	264	48	87
J_2: Computational	10	8–13	[6 11 13 13 17 23 30 30 68 56]	102	**267**	60	90
J_3: Max Independence	9	8–13	[4 4 4 4 4 6 6 7 8]	143	1452	**47**	88
J_4: Max dependence	12	8–13	[2 3 3 3 3 3 3 3 3 3 3 3]	234	2808	145	**35**

Based on the S_C matrix, the algorithm for the automatic generation of the ARRs was executed. The algorithm examined all the possible ways of eliminating variables, giving as a result 21515 ARRs with 766 different binary fault signatures. Later (for comparison purposes) these ARRs were ranked according to the 4 previously introduced cost functions; finally, the algorithm for the calculation of the MI-ARRS was run for each cost function separately.

The overall results are reported in Table V where the column "Number of ARRs" indicates the number of ARRs that constitutes the MI-ARRS while the column "Not isolable faults" indicates the faults (fault index number) that are not isolable (i.e., the faults that have the same fault signature in the MI-ARRS). It should be emphasized that it was not possible to find MI-ARRS allowing the isolation of all the 15 faults, in fact fault 8 and fault 13—associated respectively with θ and φ—were indistinguishable, in the sense that all the ARRs that are sensitive to θ are also sensitive to φ and vice versa. This is a direct consequence of the aircraft model equations and of the procedure of AARs generation. In fact, through a careful examination of the model equations, it can be observed that while both θ and φ are present within each equation of the set $r_1,r_2,r_3,r_7,r_8,r_9,r_{10}$, they cannot be evaluated from any equation of the mentioned set, (see the "2" in rows 8 and 9 of Table IV) entailing that the algorithm can only calculate θ and φ form the measurement equations pair r_{18}, and r_{23}. As a consequence (within the AAR generation process) $r_1,r_2,r_3,r_7,r_8,r_9,r_{10}$ will always be used either for the calculation of other calculable variables or as redundancy equations. In both cases all the generated ARRs will have the same fault sensitivity with respect to $\dot{\theta}$ and φ leading to indistinguishable binary fault signatures.

TABLE VI
Details of the Analytic Structure of the MI-ARR Associated to J_1

	ARR Ranking J_1	J_1	No. of Derivated Signals	No. of Additions	No. of Multiplications	No. of Functions	No. of Divisions
ARR 1	1	7	1	2	2	2	0
ARR 2	557	8	1	3	3	4	1
ARR 3	1113	8	1	6	7	3	1
ARR 4	1122	9	1	3	10	10	0
ARR 5	3343	10	2	22	7	1	0
ARR 6	5662	10	2	4	5	3	1
ARR 7	7041	12	2	8	22	1	0
ARR 8	7052	12	2	48	17	3	0
ARR 9	7672	14	2	40	10	13	2

The column "Cost of the selected ARRs" indicates the cost of each ARR in the MI-ARRS evaluated with the different J_i. For instance, considering the J_1 ranking criteria, the MI-ARRS contains 9 ARRs having individual costs J_1 = [7 8 8 9 10 10 12 12 14]. Considering the rows associated to J_3 and J_4, the "cost of the selected ARRs" indicates the number of faults the ARRs are sensitive or insensitive to, respectively. The remaining columns of Table V indicate, for each optimization criteria, the cumulative cost of the MI-ARRS evaluated using all the cost functions. It is interesting to compare the dimensions of the MI-ARRS generated for each J_i. While the MI-ARRs associated with J_2 and J_4 include 10 and 12 ARRs, respectively, those associated to J_1 and J_3 contain only 9 ARRs. For all the optimization criteria the number of not isolable fault is the same.

Since the cost functions J_3 and J_4 do not take into account the computational complexity, the corresponding ARRs have a very high computational cost. On the other hand, J_3 produces an MI-ARRs with a minimum number of 1 while J_4 has a minimum number of "0" in the fault isolation matrix. A significant reduction in the analytical complexity was achieved by the MI-ARRs associated with the indexes J_1 and J_2. Therefore, these MI-ARRSs are more suitable for online real-time implementation. Overall, the derivative cost J_1 provides the best performance; in fact, the minimization of the derivative contribution is achieved while maintaining a reasonable computational complexity and a significant fault independence in the MI-ARRs; additionally, it requires only 9 ARRs. Within the following time domain analysis, the discussion will focus on the results associated with J_1. Table VI shows in detail the analytical structure of each AAR of the MI-ARR associated with J_1. Finally, Tables VII, VIII, IX, and X show the binary signature matrix of the MI-ARRs for the different cost functions.

B. Time Domain Analysis

The computed MI-ARRSs can be used as residual generators for FDI schemes dealing with sensors and actuators failures. The time domain analysis was performed through a Simulink-based simulation study. Flight conditions were simulated to replicate the actual flight conditions of the WVU YF-22 flight testing [23]. A 200 s time interval was selected to perform this study. The recorded values of the 4 aircraft inputs in (8) were used as inputs to the Simulink model of the WVU YF-22 to generate realistic time histories of the states of the aircraft. A Simulink block was added to implement the real-time calculation of the MI-ARRs equations. In this block the derivatives of the 15 measured signals were approximated evaluating the incremental ratio of the signals with a sampling time of 0.01 s. Additive faults f_i were sequentially injected, 4 as actuator faults and 11 as sensor faults. A linear ramp-like shape in addition to a small magnitude sinusoidal perturbation was selected as incipient sensor fault modeling function; the failure modeling is described by

$$F_i(t-t_f) = \begin{cases} 0 & t < t_f \\ A_i \cdot (t-t_f)/T_R & t_f \le t < t_f + t_R \\ A_i \cdot (1 + A_{oi}) \sin(\omega_{oi} t) & t \ge t_f + T_R \end{cases}, \quad i = 1 \ldots 15 \qquad (22)$$

where t_f is the failure occurrence time, T_R is the time duration of the ramp, A_i and A_{oi} are, respectively, the final offset values and sinusoidal perturbation amplitude for the ith signal and ω_{oi} is the sinusoidal frequency. Due to space constraints, in the following analysis only the results provided by the MI-ARRSs generated with the derivative cost J_1 are analyzed.

C. Ideal Case

In the ideal case no measurement noise was superimposed either to the input or output signals. Therefore, in the simulation the residual signals are identical to zero until the occurrence of the fault.

TABLE VII
MI-ARRS Fault Signature Matrix (S_F^T) Generated by J_1

	f1 X_T	f2 de	f3 da	f4 dr	f5 V	f6 α	f7 q	f8 θ	f9 h	f10 β	f11 p	f12 r	f13 φ	f14 ψ	f15 q_{dyn}
ARR 1	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0
ARR 2	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0
ARR 3	0	0	0	0	0	0	1	1	0	0	1	1	1	0	0
ARR 4	0	0	0	0	1	1	0	1	1	1	0	0	1	0	0
ARR 5	0	1	0	0	1	1	1	0	0	0	**0**	**0**	0	0	**0**
ARR 6	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0
ARR 7	0	0	1	1	1	0	**0**	0	0	1	1	1	0	0	**0**
ARR 8	0	0	1	0	1	0	**0**	0	0	1	1	1	0	0	1
ARR 9	1	1	0	0	1	1	**0**	**0**	0	1	1	1	**0**	0	1

TABLE VIII
MI-ARRS Fault Signature Matrix (S_F^T) Generated by J_2

	f1 X_T	f2 de	f3 da	f4 dr	f5 V	f6 α	f7 q	f8 θ	f9 h	f10 β	f11 p	f12 r	f13 φ	f14 ψ	f15 q_{dyn}
ARR 1	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0
ARR 2	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0
ARR 3	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0
ARR 4	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0
ARR 5	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0
ARR 6	0	0	0	0	1	1	0	1	1	1	0	0	1	0	0
ARR 7	0	1	0	0	1	1	1	0	0	0	1	1	0	0	1
ARR 8	0	0	1	1	1	0	1	0	0	1	1	1	0	0	1
ARR 9	0	0	1	0	1	0	1	0	0	1	1	1	0	0	1
ARR 10	1	1	0	0	1	1	0	1	0	1	1	1	1	0	1

TABLE IX
MI-ARRS Fault Signature Matrix (S_F^T) Generated by J_3

	f1 X_T	f2 de	f3 da	f4 dr	f5 V	f6 α	f7 q	f8 θ	f9 h	f10 β	f11 p	f12 r	f13 φ	f14 ψ	f15 q_{dyn}
ARR 1	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0
ARR 2	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0
ARR 3	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0
ARR 4	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0
ARR 5	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0
ARR 6	0	0	1	1	1	0	0	0	0	1	1	0	0	0	1
ARR 7	0	1	0	0	1	1	1	0	0	0	0	1	0	0	1
ARR 8	0	0	1	0	1	0	1	0	0	1	1	1	0	0	1
ARR 9	1	1	0	0	1	0	1	1	0	1	0	0	1	0	1

TABLE X
MI-ARRS Fault Signature Matrix (S_F^T) Generated by J_4

	f1 X_T	f2 de	f3 da	f4 dr	f5 V	f6 α	f7 q	f8 θ	f9 h	f10 β	f11 p	f12 r	f13 φ	f14 ψ	f15 q_{dyn}
ARR 1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1
ARR 2	1	1	1	1	1	1	1	1	0	1	1	0	1	0	1
ARR 3	1	1	1	1	1	1	0	1	0	1	1	1	1	0	1
ARR 4	1	1	1	1	1	1	1	1	0	1	0	1	1	0	1
ARR 5	1	1	1	1	1	1	1	1	0	1	0	0	1	1	1
ARR 6	1	1	1	1	1	1	1	1	0	0	1	1	1	0	1
ARR 7	1	1	1	1	1	0	1	1	0	1	1	1	1	0	1
ARR 8	1	1	1	1	1	1	1	1	0	1	1	1	1	0	0
ARR 9	1	1	1	0	1	1	1	1	0	1	1	1	1	0	1
ARR 10	1	1	0	1	1	1	1	1	0	1	1	1	1	0	1
ARR 11	1	0	1	1	1	1	1	1	0	1	1	1	1	0	1
ARR 12	0	1	1	1	1	1	1	1	0	1	1	1	1	0	1

After the failure occurrence the residuals sensitive to that fault diverge from zero. In this ideal situation the selection of the residual detection threshold is not an important issue since the insensitive residuals remain exactly at zero. Therefore any small threshold could be selected to detect the faults occurrence. In this case neither the "false alarm" problem nor the "residual filtering" problem are present; thus, perfect detection and isolation for each of the 13 isolable faults was achieved, in accordance with the results in Table VII. As expected, in the case of absence of any source of noise it is ideally possible to isolate faults of arbitrary small amplitudes.

D. Noisy Case and Decision Test

The presence of measurement noise creates more challenges. In fact, in this case, the residual signals are not zero even in the fault-free conditions. Therefore, small amplitude faults could be erroneously interpreted as noise fluctuations. In realistic conditions the fault detection and isolation is not a simple task, and an appropriate evaluation test based on the temporal evolution of the residuals should be used to make a decision regarding the occurrence of a fault. A well-known filter for the detection of moderate persistent shift in the mean value of the residual is the cumulative sum (CUSUM) [24]. This filter is used to detect both positive and negative changes $\pm\nu_i$ in the mean value μ_{io} ($\mu_{if} = \mu_{io} \pm \nu_i$) of the residual $r_i(k)$ caused by the occurrence of a fault. The filter is defined by the following equations [25]:

$$S_i^+(k) = \sup\left(0, S_i^+(k-1) - \frac{\nu_i(\nu_i - 2r_i(k))}{2\sigma_i^2}\right) \quad (23)$$

$$S_i^-(k) = \sup\left(0, S_i^-(k-1) + \frac{\nu_i(-\nu_i - 2r_i(k))}{2\sigma_i^2}\right) \quad (24)$$

where $S_i^+(k)$ and $S_i^-(k)$ are the CUSUM states and σ_i^2 is an estimation of the variance of $r_i(k)$. The output of the filter is defined as $S_i(k) = \max[S_i^-(k), S_i^+(k)]$, when $S_i(k)$ exceeds predefined thresholds an alarm level is activated. The CUSUM-based algorithm has shown desirable failure detection capabilities along with lower false alarm rates when compared with simple threshold logic applied to unfiltered signals [25]. A Simulink block was introduced in the simulation to implement the CUSUM filtering of the residual signals.

E. Simulation Results

The fault amplitudes were selected depending on the amplitude of the signals in normal flight

TABLE XI
Faults and Noise Characteristics

	X_T N	de rad	da rad	dr rad	1) V m/s	α rad	q rad/s	θ rad	h m	β rad	p rad/s	r rad/s	φ rad	ψ rad	q_{dyn} N/m^2
Fault Amplitude	10.9	0.001	0.003	0.001	8	0.015	0.012	0.012	48	0.003	0.001	0.008	0.009	0.4	161
Noise STD	—	—	—	—	0.066	0.0087	0.0058	0.0021	0.23	0.0087	0.0058	0.0058	0.0021	0.048	0.13

conditions. The fault amplitudes A_i were selected to be 20% of the mean of the absolute value of the signals evaluated along the entire test flight, and represent small amplitude faults that may occur in real situations. The amplitude of the faults are shown in Table XI.

An identical shape was assumed for the 15 faults characterized by the following values: $T_R = 1$ s, $A_{oi} = 0.1$, $\omega_{oi} = 10$ rad/s, and $t_f = 100$ s.

To perform a realistic study, it is also important to take into account the noise affecting the input-output signals involved in the computation of ARRs. For example the measured signals Y are typically corrupted by sensor measurement noise. The selection of a realistic noise level within the simulation is crucial because it has a direct effect on the amplitude of the residual signal $S_i(k)$ which in turn strongly affects the possibility of detecting faults. Considering this aspect, in [23], it was experimentally determined that the (short term) noise for the actual sensors of the WVU YF-22 UAV can be accurately modeled by an additive zero mean Gaussian noise having a standard deviation σ reported in Table XI [23] for each of the 11 sensors. Note that since the 4 actuator commands U represent the commands computed by the flight controller, it was assumed that these signals are not affected by noise. The parameters of the CUSUM filters were estimated based on the statistics of the residuals evaluated in fault-free conditions. Specifically, the σ_i for (23–24) were estimated as the standard deviations of the residual signals in fault-free conditions. The values of the ν_i coefficients strongly affect each fault on each residual signal $S_i(k)$. The ν_i were expressed as functions of σ_i ($\nu_i = \text{gain}_i \cdot \sigma_i$) and the gain_i values were determined with the objective of maximizing the post failure signal rise of the sensitive faults versus the insensitive ones.

To analyze the faults isolation ability and determine the gain_i parameters of the 9 CUSUM filters, 15 simulations were performed where single faults were sequentially injected in the sensors; the corresponding time responses $S_i(k)$ of the CUSUM signals for the 15 simulations were recorded. The ν_i values were computed according the following offline analysis. For each ARRi, the maxim value of the $S_i(k)$ signal was computed for each one of the 15 faults (evaluated at post failure condition $t > t_f$); these maximum values were divided in two vectors: $\text{MAX}_{\text{sen\_}i}$ containing values associated to the sensitive faults and $\text{MAX}_{\text{insen\_}i}$ containing values associated to the insensitive ones; then the ratio: $\text{Ratio}_i = \min(\text{MAX}_{\text{sen\_}i})/\min(\text{MAX}_{\text{insen\_}i})$ was computed.

Clearly, the higher the Ratio_i, the easier the separation will be (through thresholds) between the sensitive and insensitive signals. Since the $S_i(k)$ signals are affected by the ν_i values, the optimal ν_i values were derived by varying the gain_i coefficients and calculating the corresponding Ratio_i (for the gain_i, the values in the range [0.01–5] were explored with step 0.01). For each ARR_i, the values of gain_i giving the highest Ratio_i were selected. The result of this analysis gave the following values for the ν_i for the 9 filters: $[0.51\sigma_1, 0.045\sigma_2, 1.16\sigma_3, 0.17\sigma_4, 0.59\sigma_5, 0.04\sigma_6, 0.041\sigma_7, 0.33\sigma_8, 0.15\sigma_9]$. The detection thresholds for the 9 CUSUM signals $S_i(k)$ were fixed to 1.5 times the maximum value of the corresponding CUSUM signals evaluated in fault-free conditions. It was verified that these thresholds guarantee the correct detection of all the 15 faults while at the same time providing a good security margin for possible noise peeks.

The theoretical sensitivity of each ARR in the MI-ARRSs to the 15 faults can be easily determined analyzing the ith line in Table VII. In any case, it is important to verify whether this signature can actually be recovered from the filtered CUSUM responses in presence of noisy measurements. Considering the first ARR, this analysis was performed by showing on the same plot (upper part of Fig. 4) the 15 time histories of the CUSUM signals generated from the first ARR in correspondence with the 15 faults. Residuals that were theoretically sensible to a fault (e.g., presence of 1 in the first row of Table VII) were plotted using a black line, while insensitive residuals were plotted with gray lines (presence of 0 in the first row of Table VII). Following this representation, after the fault occurrence, all the black lines should overlay the detection threshold while the gray lines should remain below. As shown in Fig. 4, a perfect agreement with the fault sensitivity predicted by first row of Table VII was achieved; in fact the CUSUM filtered signal associated with the first ARR resulted sensitive only to faults F7, F8, F12, and F13. Similarly, Figs. 5, 6, 7,

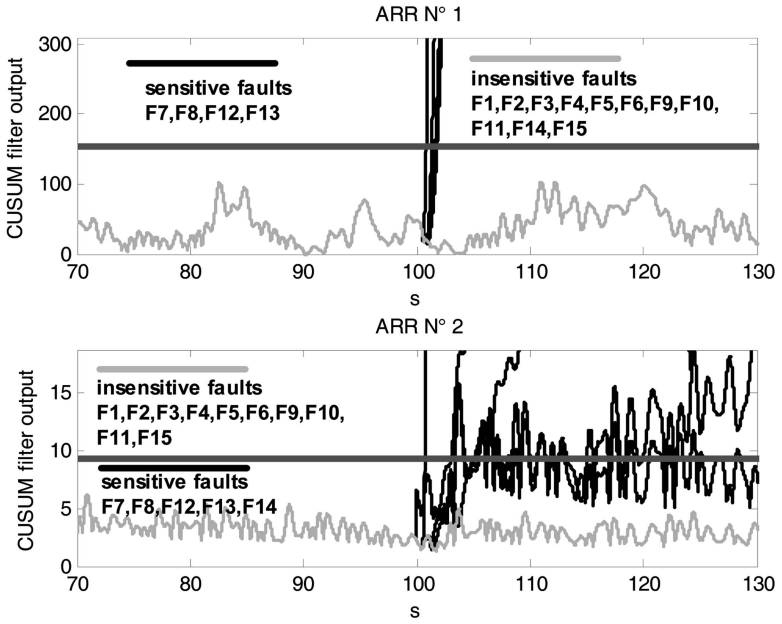


Fig. 4. Filtered output of ARR 1 and ARR 2.

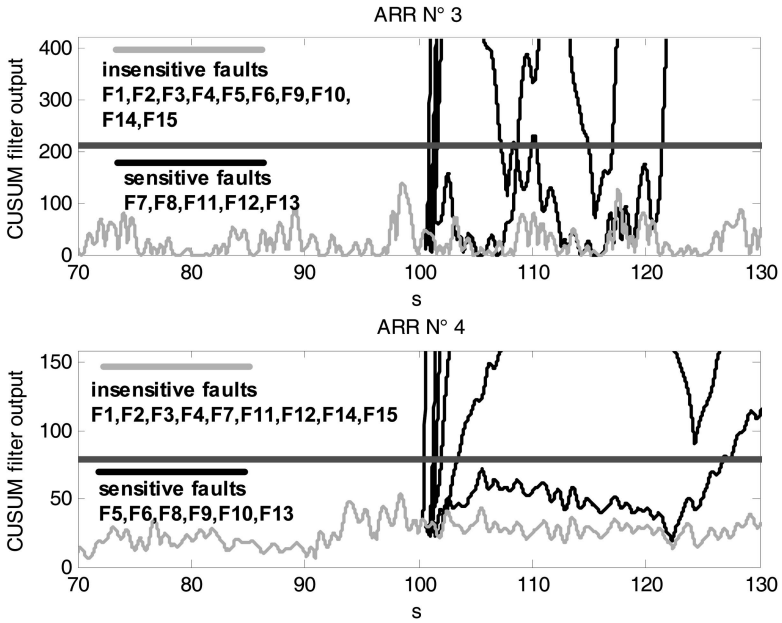


Fig. 5. Filtered output of ARR 3 and ARR 4.

8 show the response to the 15 faults for the remaining 8 CUSUM filtered ARRs. It can be observed from all the figures that the sensitive residuals overstep the threshold while the insensitive residuals do not. This confirms the feasibility of reconstructing the binary isolations signatures of Table VII from the filtered ARRs.

VIII. CONCLUSIONS

This paper presents the results of an effort towards defining an automated procedure for the design of a subset of structured ARRs suitable for the real-time applications of FDI schemes. Practical implementation issues have been considered in the design methodology through the introduction of a ranking scheme for the residuals selection priority. Particularly, through the selection of different performance indices it was possible to select structured residuals subsets with high detection capabilities along with desirable robustness properties. The methodology was validated using a detailed mathematical model of a semi-scale remotely controlled YF-22 research aircraft. Despite the complexity of the model the algorithm provided a large number of possible ARRs within a reasonable

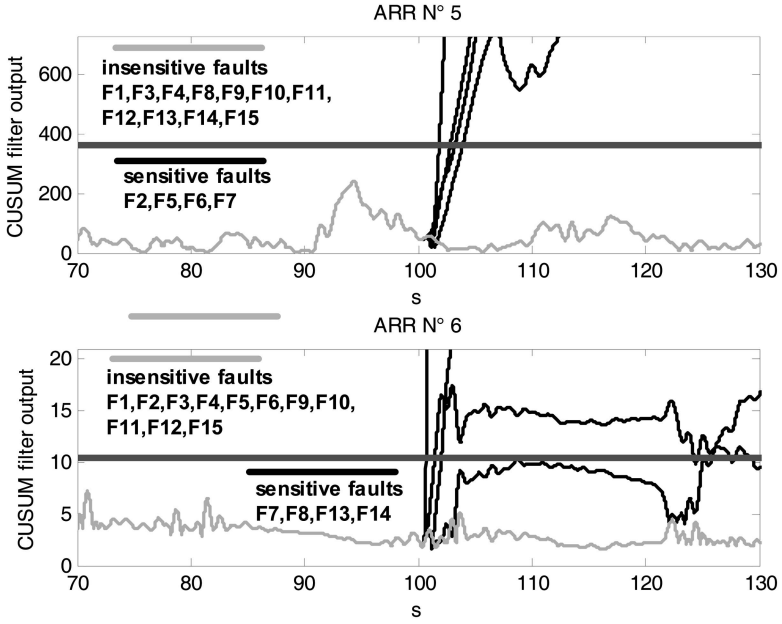


Fig. 6. Filtered output of ARR 5 and ARR 6.

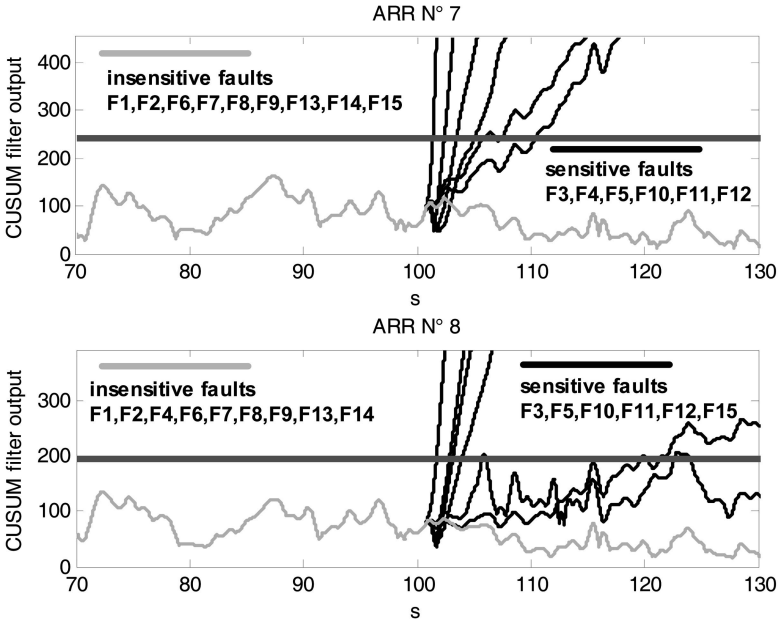


Fig. 7. Filtered output of ARR 7 and ARR 8.

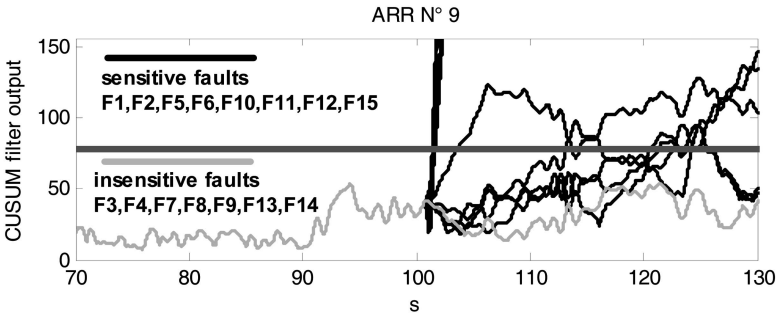


Fig. 8. Filtered output of ARR 9.

time. The analysis shows that minimizing the total number of derivatives to be calculated in the residuals provided the best overall performance in the sense that the minimization of the derivative contribution also allowed for maintaining an acceptable computational complexity along with significant fault independence. This desirable behavior was also confirmed by time domain simulations where the MI-ARRs generated from the derivative cost in conjunction with CUSUM filtering provided desirable robustness for fault isolations in presence of measurement noise.

REFERENCES

[1] Patton, R. J.
Fault detection and diagnosis using analytical redundancy.
Computing and Control Engineering Journal, (May 1991), 127–156.

[2] Chen, J., and Patton, R. J.
Robust Model-Based Fault Diagnosis for Dynamic Systems.
London: Kluwer, 1999.

[3] Zhou, D. H., and Frank, P. M.
Fault diagnostics and fault tolerant control.
IEEE Transactions on Aerospace and Electronic Systems, **34**, 2 (1998), 420–427.

[4] Perhinschi, M.G., Napoletano, M. R., Campa, G., Seanor, B., Burken, J., and Larson, R.
Design of safety monitor schemes for a fault tolerant flight control system.
IEEE Transactions on Aerospace and Electronic Systems, **42**, 2 (2006), 562–571.

[5] Zhang, Y., and Li, X. R.
Detection and diagnosis of sensor and actuator failures using IMM estimator.
IEEE Transactions on Aerospace and Electronic Systems, **34**, 4 (1998), 1293–1313.

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

[7] Gauthier, J. P., and Kupka, I.
Observability and observers for nonlinear systems.
SIAM Journal of Control Optimization, **32**, 4 (1994), 975–994.

[8] Staroswiecki, M., and Comtet-Varga, G.
Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems.
Automatica, **37** (2001), 687–699.

[9] Frisk, E., and Aslund, J.
Lowering orders of derivatives in nonlinear residual generation using Realization theory.
Automatica, **41** (2005), 1799–1807.

[10] Guernez, C., Casar, J. Ph., and Staroswiecki, M.
Extension of parity space to nonlinear polynomial dynamic systems.
In *Proceedings of IFAC Safeprocess*, Hull, UK, (1997), 861–866.

[11] Gertler, J.
Fault detection and isolation using parity relations.
Control Engineering Practice, **5**, 5 (1997), 653–661.

[12] Zhang, Y., and Li, X. R.
Detection and diagnosis of sensor and actuator failures using IMM estimator.
IEEE Transactions on Aerospace and Electronic Systems, **34**, 4 (1998), 1293–1313.

[13] Zhang, Q., Basseville, M., and Benveniste, A.
Fault detection and isolation in nonlinear dynamic systems: A combined input-output and local approch.
Automatica, **34**, 11 (1998), 1359–1373.

[14] Frisk, E.
Residual generator design for nonlinear, polynomial systems a Groebner basis approach.
In *Proceedings of IFAC Safeprocess*, Budapest, Hungary, 2000, 979–984.

[15] Staroswiecki, M., Cassar, J., and Declerck, P.
A structural framework for the design of FDI in large scale industrial plants.
In R. Patton, P. Frank, and R. Clark (Eds.), *Issues of Fault Diagnosis for Dynamic Systems*.
Berlin: Springer, 2000.

[16] Frisk, E., Coquempot, V., Krysander, M., Staroswiecki, M., and Dustegör, D.
Structural analysis of fault isolability in the DAMADICS benchmark.
Control Engineering Practice, **14**, 6 (2006), 597–608.

[17] Zamanabadi, R. I., Blankeb, M., and Katebic, S.
Cheap diagnosis using structural modeling and fuzzy-logic-based detection.
Control Engineering Practice, **11**, 4 (2003), 415–422.

[18] Krysander, M. Åslund, J., and Nyberg, M.
An efficient algorithm for finding over-constrained sub-systems for construction of diagnostic tests.
In *16th International Workshop on Principles of Diagnosis* (DX-05), Pacific Grove, CA, 2005. Available: http://www.vehicular.isy.liu.se/Publications/stru.html.

[19] Dustegor, D., Cocquempot, V., and Staroswiecki, M.
Structural analysis for residual generation: Towards implementation.
In *Proceedings of IEEE International Conference on Control Applications*, Taipei, Taiwan, vol. 2, 2004, 1217–1222.

[20] Rattfalt, L.
Comparative study of two structural methods for fault isolability analysis.
Master's Thesis, Dept. of Electrical Engineering, Linköping University, 2004. Available: http://www.fs.isy.liu.se/Publications/MSc.

[21] Stevens, B., and Lewis, F.
Aircraft Control and Simulation.
New York: Wiley, 1992.

[22] Rauw, M. O.
FDC 1.2—A simulink toolbox for flight dynamics and control analysis.
Zeist, The Netherlands, 1997. Available http://www.dutchroll.com/.

[23] Campa, G., Gu, Y., Seanor, B., Napolitano, M. R., Pollini, L., and Fravolini, M. L.
Design and flight-testing of nonlinear formation control laws.
Control Engineering Practice, **15**, 9 (2007), 1077–1092.

[24] Basseville, M., and Nikiforov, I.
Detection of Abrupt Changes—Theory and Practice.
Englewood Cliffs, NJ: Prentice-Hall, 1993.

[25] Campa, G., Fravolini, M. L., Seanor, B., Napolitano, M. R., Del Gobbo, D., Yu, G., and Gururajan, S.
On-line learning neural networks for sensor validation for the flight control system of a B777 research scale model.
International Journal of Robust and Nonlinear Control, **12**, 11 (2002), 987–1007.

Mario L. Fravolini was born in Perugia, Italy. He received his Ph.D. degree in electronic engineering from the University of Perugia in 2000.

Currently he is research assistant with the Department of Electronics and Information Engineering University of Perugia. In 1999 he was with the Control Group at the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta. He has been a visiting research assistant professor at the Department of Mechanical and Aerospace Engineering West Virginia University, for several years. He teaches courses in the area of feedback control systems at Perugia and Terni University. His research interests include: fault diagnosis, intelligent and adaptive control, predictive control, optical feedback, and active control of structures.

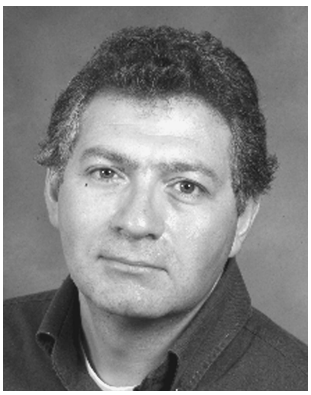
Valerio Brunori was born in 1981. He graduated in 2006 from the University of Perugia in information and telecommunication engineering. He is a Ph.D. student in industrial engineering at the University of Perugia.

His main research interests are motor control, optical feedback control of UAVs, fault diagnosis of nonlinear systems, MagLev control for automotive applications.

Giampiero Campa (M'99) was born in Taranto, Italy. He received both his M.S. degree in electrical and electronic engineering (1996), and his Ph.D in robotics and automation (2000) from the University of Pisa, Italy.

He has been a visiting student at the Industrial Control Centre, Strathclyde University, UK, and at the Department of Aerospace Engineering, Georgia Institute of Technology. In 2000 he joined the Department of Aerospace Engineering, West Virginia University, where he is currently a research assistant professor.

Dr. Campa is a Member of the IEEE Control System Society and of the Mathematical Association of America and his current research interests include systems modeling, identification and simulation, nonlinear and hybrid control systems, and real-time embedded computing.



Marcello R. Napolitano was born in Pomigliano D'Arco, Italy. He received his M.S. degree in aeronautical engineering from the University of Naples, Italy, 1985, and his Ph.D. degree in aeronautical engineering from Oklahoma University, 1989.

In 1990 he joined the Department of Aerospace Engineering, West Virginia University, where he is currently a full professor. His current research interests include flight control systems, unmanned aerial vehicles, fault tolerance, and neural networks.

Michele La Cava received his degree in electronic engineering from the University of Naples in 1965.

From 1968 to 1976, he was a research engineer at Fondazione U. Bordini in Rome, Italy. From 1976 to 1989, he was with the Department of Electronics, Informatics and Systems at the University of Calabria in Cosenza, Italy, as assistant professor and then as full professor. Since 1989 he has been full professor of Feedback Control Systems at the University of Perugia, Italy. His current research interests are in the field of nonlinear motor control and soft computing applications to robotics and process control.