JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 31, No. 1, January–February 2008

Efficient Nonlinear Actuator Fault Detection and Isolation System for Unmanned Aerial Vehicles

Guillaume Ducard* and Hans P. Geering†
ETH Zurich, 8092 Zurich, Switzerland

DOI: 10.2514/1.31693

In this paper, three main limitations of the classical implementation of the multiple-model adaptive-estimation method to isolate faults based on predefined fault hypotheses are highlighted. The first concerns the number of filters that must be designed to span the range of possible fault scenarios, which must be limited due to computational load. The second limitation appears when an actuator is locked at an arbitrary nonzero position that biases the residuals of the Kalman filters, leading to inaccurate fault detection and state estimation. Third, most of the implementations of a multiple-model adaptive-estimation method only work efficiently around predefined operating conditions. This paper presents a nonlinear actuator-fault detection and isolation system, which properly works over the entire operating envelope of an aircraft. Locked-in-place and floating actuator faults can be handled. The robustness of the fault detection and isolation system is enhanced by the usage of auxiliary excitation signals. The fault detection and isolation system is also capable of handling two simultaneous actuator failures with no increase of the computational load. The complete system was demonstrated in simulation with a nonlinear model of a model aircraft in moderate to severe wind conditions.

Nomenclature

$\boldsymbol{B}$	=	control input matrix
b	=	aircraft wing span, m
C_L	=	dimensionless roll moment coefficient
C_M	=	dimensionless pitch moment coefficient
C_N	=	dimensionless yaw moment coefficient
$\bar{c}$	=	mean aerodynamic chord, m
F_T	=	aircraft thrust force along the body x axis, N
$\boldsymbol{I}$	=	aircraft inertia matrix in the body-fixed frame, kg.m^2
L	=	roll moment, N · m
M	=	pitch moment, N · m
N	=	yaw moment, N · m
$\boldsymbol{P}$	=	state estimation error covariance matrix
p	=	body-axis roll rate, rad/s
p_k, Pbaf$_k$	=	probability signal indicating whether actuator k has failed
q	=	body-axis pitch rate, rad/s
$\bar{q}$	=	freestream dynamic pressure, N/m^2
$\boldsymbol{R}$	=	sensor noise covariance
r	=	body-axis yaw rate, rad/s
$\boldsymbol{r}_k$	=	residuals generated by the kth (extended) Kalman filter
ref	=	controller reference signal
S	=	wing area, m^2
T_s	=	sampling period, s
$\boldsymbol{u}$	=	vector of the actuator control command
V_T	=	air speed of the aircraft, m/s
$\boldsymbol{v}$	=	virtual control command, $\boldsymbol{v} = [C_L \;\; C_M \;\; C_N]_u^T$
$\hat{\boldsymbol{x}}$	=	estimate of the aircraft state vector
$\boldsymbol{y}$	=	vector of measurements
α	=	angle of attack, rad
β	=	sideslip angle, rad
δ_{a1}	=	left aileron deflection, rad
δ_{a2}	=	right aileron deflection, rad
δ_{e1}	=	left elevator deflection, rad
δ_{e2}	=	right elevator deflection, rad
δ_r	=	rudder deflection, rad
$\hat{\delta}_k$	=	estimated deflection of actuator k, rad
$\bar{\delta}_k$	=	unknown deflection of a faulty actuator k, rad
ρ	=	air density, kg/m^3

Subscripts

$f_1, f_2, \ldots, f_i$	=	fault hypothesis $1, 2, \ldots, i$
nf	=	no fault

Superscript

b	=	aircraft body-fixed frame

Received 19 April 2007; revision received 1 August 2007; accepted for publication 1 August 2007.

*Ph.D. Student, Department of Mechanical and Process Engineering. Student Member AIAA.

†Professor, Department of Mechanical and Process Engineering. Associate Fellow AIAA.

I. Introduction

NEW generations of unmanned aerial vehicles (UAVs) will be designed to achieve their mission not only with increased efficiency, but also with more safety and security. Future UAVs will be operated with algorithms capable of monitoring the aircraft health and of taking action if needed. Fault-tolerant control systems for small and low-cost UAVs should not significantly increase the number of actuators or sensors to achieve the safer operation. The paper describes a computationally efficient online nonlinear fault detection and isolation system (FDI) that monitors the actuators' health without requiring sensors to measure the deflection of the control surfaces.

II. Multiple-Model Adaptive Estimation Schemes

A. Fault Detection and Isolation Using Multiple-Model Adaptive Estimation Schemes

One approach to detect and isolate actuator or sensor faults is the multiple-model adaptive-estimation (MMAE) method [1], as depicted in Fig. 1. It is based on a bank of parallel Kalman filters (KFs), each of which is matching a particular fault status of the system. A hypothesis-testing algorithm uses the residuals from each KF to assign a conditional probability to each fault hypothesis. Several papers have demonstrated how the MMAE method can be used in the context of fault detection and isolation for aircraft [2–4]

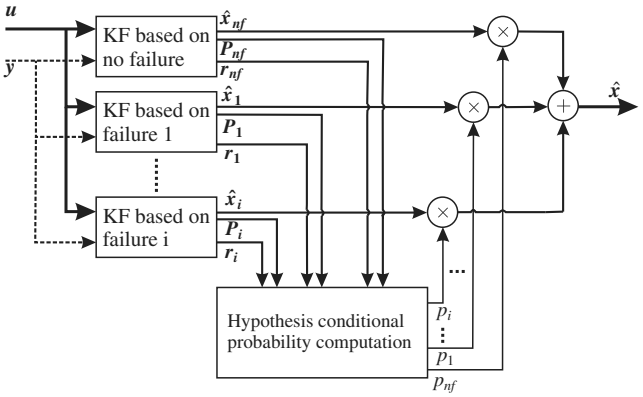

Fig. 1 Classical MMAE scheme.

and underwater vehicles [5]. The main advantage of the method lies in its responsiveness to parameter variations, leading to faster fault isolation than that attained by other methods without multiple-model structure. It also enables the reconstruction of a correct state estimate even when an actuator or sensor fault occurs, because the estimated state vector is the sum of each KF estimate weighted by its corresponding probability.

B. Limitations of the MMAE Method

The MMAE method can be applied in practice as long as the expected faults can be hypothesized by a reasonable number of Kalman filters. However, the number of addressable faults is limited due to the computational load required for each filter.

The method also does not provide satisfactory results when an actual fault does not closely match a predefined fault hypothesis. This may occur when an actuator is locked in place at an arbitrary position that affects the dynamics of the system. Because lock-in-place faults cannot be predicted, they can have detrimental effects on the filter performance. Because of the biased residuals, the Kalman filter provides inaccurate estimates of the state variables, which causes severe problems with the probability calculation. Therefore, neither the fault detection nor the fault isolation works properly, and the state estimation is useless for control purposes.

Moreover, in most of the fault-tolerant applications that use the MMAE method, the KFs are designed based on a linear model of the unmanned vehicle operated at some nominal conditions. Very few papers use the MMAE method in the nonlinear case, when the system operates over the entire range of possible operating conditions. The authors of [6] used a multiple-model approach for the sensor fault detection of nonlinear systems. However, the assumption is made that the nonlinear system can be approximated by a finite number of interpolated linear time-invariant models, which constitute the banks of Kalman filters, for which the residuals are used to determine the effective operating regime and isolate the faulty sensor. In this paper, a unique bank of filters is designed, which can operate over the whole span of the aircraft's flying conditions.

C. New Extensions to the MMAE Method

To make the MMAE method applicable for any flight conditions and capable of isolating lock-in-place or floating actuator faults, the MMAE algorithm is combined with extended Kalman filters (EKF) used for the nonlinear estimation of some (unknown) fault parameter: the deflection of a faulty control surface (or actuator). The resulting method is called "extended multiple-model adaptive estimation" (EMMAE) [7,8]. In this paper, it is explained why the online estimation of the deflection of a faulty actuator enables the EMMAE method to cope with lock-in-place or floating actuator-fault scenarios and drastically reduces the number of filters needed.

Moreover, this method takes advantage of the estimated faulty-actuator deflection to reconfigure the settings of a control allocator to efficiently compensate for the fault. Furthermore, some techniques were added to enhance the robustness and the performance of the EMMAE-FDI system when there is very low excitation of the aircraft during steady flight and to improve the speed and accuracy of the fault isolation.

Finally, the results of simulations are presented and demonstrate the complete system on a nonlinear model of an aircraft experiencing consecutive actuator faults with severe wind disturbance.

III. New FDI Scheme Based on the EMMAE Method

A. Modeling Actuator Faults

A lock-in-place or floating actuator fault in the system can be seen as if the desired control input δ_j was disconnected and replaced by a faulty control signal $\bar{\delta}_j$ that takes control over the plant, as shown in Fig. 2. In a concise manner, the true input of the plant can be written [9] as $u_i(t) = \delta_i(t) + \sigma_{A_i}[\bar{\delta}_i(t) - \delta_i(t)]$. In case of actuator failure(s), the vector of the (unknown) inputs is $\bar{\boldsymbol{\delta}}(t) = [\bar{\delta}_1(t) \;\; \bar{\delta}_2(t) \cdots \bar{\delta}_m(t)]^T$, $\boldsymbol{\sigma} = \mathrm{diag}([\sigma_{A1} \;\; \sigma_{A2} \cdots \sigma_{Am}])$, and

$$\sigma_{Aj} = \begin{Bmatrix} 1 & \text{if the } j\text{th actuator fails} \\ 0 & \text{otherwise} \end{Bmatrix}$$

In the method presented next, the unknown parameters $\bar{\delta}_j$ are constantly estimated by their respective EKF. The conditional fault-hypothesis probabilities p_j assign the value for σ_{Aj}.

B. EMMAE Method

The MMAE method is to be made applicable for any arbitrary lock-in-place faults or uncontrolled varying faults and at all flying conditions. Therefore, the original MMAE algorithm is modified by replacing the linear Kalman filters with extended Kalman filters used as nonlinear estimators of the state vector and a fault parameter: namely, the deflection of a faulty control surface (or actuator). The implementation of the EMMAE is depicted in Fig. 3. Contrary to the FDI designs with the classical MMAE method in which several KF are designed for several faulty deflections for one actuator, in the EMMAE method, only one EKF is responsible for completely monitoring the actuator health. Therefore, the EMMAE method

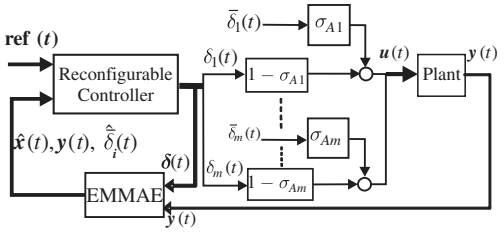

Fig. 2 Modeling of actuator faults.

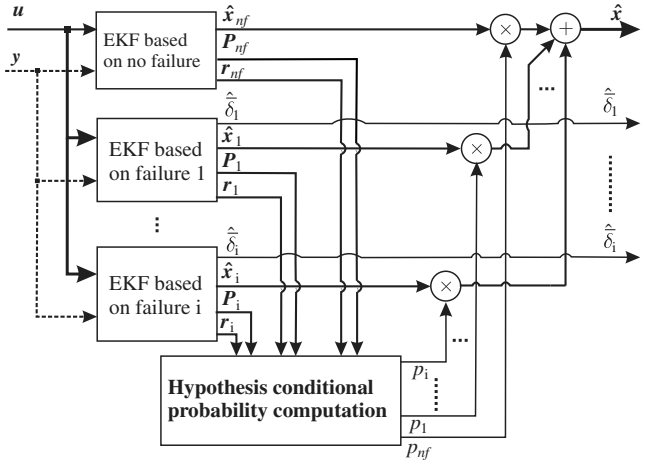

Fig. 3 EMMAE scheme: each EKF monitors its assigned actuator.

drastically reduces the number of filters required for actuator health monitoring.

The addition of the actuator deflection estimate in the system state vector enables the EMMAE method to work for all the possible positions in which an actuator can be locked or floating. To better illustrate the difference between the EMMAE and MMAE methods, we recall how the models are defined in the regular MMAE method [2–5] for an actuator failure. The MMAE considers a bank of linear models of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \qquad \boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} \tag{1}$$

where each model matches a fault scenario.

For example, the model that describes a failure of the jth actuator will have its $\boldsymbol{B}$ matrix modified such that the jth column of the $\boldsymbol{B}$ matrix is replaced by the very same column times a factor λ_j that varies from zero (complete loss of the actuator) to one (fully functioning actuator) [see Eq. (2)]. Any intermediate value of λ_j indicates a reduction in the effectiveness of the jth actuator to modify the dynamics of the aircraft.

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}_j\boldsymbol{u}$$

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \begin{bmatrix} b_{11} & \cdots & b_{1j}\lambda_j & \cdots & b_{1N} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ b_{l1} & \cdots & b_{lj}\lambda_j & \cdots & b_{lN} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ b_{p1} & \cdots & b_{pj}\lambda_j & \cdots & b_{pN} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_j \\ \vdots \\ u_N \end{bmatrix} \tag{2}$$

If we are to design a filter to detect the failure of the ith sensor, the ith row of the $\boldsymbol{C}$ matrix will be replaced by the very same row times a factor λ_i that varies from zero (total sensor failure such as sensor disconnection) to one (no sensor failure), as shown in Eq. (3).

$$\boldsymbol{y} = \boldsymbol{C}_i\boldsymbol{x}$$

$$\boldsymbol{y} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1k} & \cdots & c_{1p} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ c_{i1}\lambda_i & c_{i2}\lambda_i & \cdots & c_{ik}\lambda_i & \cdots & c_{ip}\lambda_i \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mk} & \cdots & c_{mp} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \\ \vdots \\ x_p \end{bmatrix} \tag{3}$$

However, this kind of approach for the modeling of actuator and sensor failure is very restrictive. Indeed, in case of a total loss of the jth actuator, the factor λ_j equals zero. This means that whatever control input the controller generates for the jth actuator, it has no influence on the dynamics of the aircraft, and the faulty-actuator deflection is considered to be zero. Note that if the jth actuator is actually locked at a nonzero deflection angle, the control signal to the jth actuator has no influence on the dynamics of the aircraft; however, the faulty-actuator deflection does have an influence on the dynamics of the aircraft. This condition results in an unknown bias term that will prevent the jth Kalman filter in the MMAE method from working properly; therefore, the residuals will be biased, and the state estimation and the computation of the probabilities will be incorrect as well.

In the EMMAE method, we not only modify the control input matrix, but also the dynamics matrix. Indeed, to define a model that describes a failure of the jth actuator, the jth column of the control input matrix is zeroed and the state vector is augmented with the jth actuator deflection $\bar{\delta}_j$. The dynamics matrix is also augmented with the original jth column of the control input matrix. This way, the control inputs from the controller to the jth actuator are totally ignored, but the faulty deflection $\bar{\delta}_j$ that is constantly estimated ($\hat{\bar{\delta}}_j$) in the state vector contributes to modifying the dynamics of the aircraft model of the jth filter, yielding residuals that are the smallest for the filter matching the occurring fault. Let us illustrate how the filters in the EMMAE are constructed in practice.

IV. Practical Implementation of the EKFs in the EMMAE FDI

A. Aircraft Configuration

The five control surfaces of the aircraft under consideration are one left aileron, one right aileron, one left elevator, one right elevator, and one rudder, as shown in Fig. 4. All actuators are fully independent, which means that ailerons (or elevators) can individually move up, down, or together in the same direction. This configuration permits some pitch torque to be produced with ailerons or some roll torque to be produced with elevators.

The state vector of the FDI filters is chosen to take only the most relevant states to reduce the computation load when running the EKFs. The state vector is $\boldsymbol{x} = [p \;\; q \;\; r \;\; \alpha \;\; \beta]^T$. The control vector for the aircraft is $\boldsymbol{u} = [\delta_{a_1} \;\; \delta_{a_2} \;\; \delta_{e_1} \;\; \delta_{e_2} \;\; \delta_r \;\; F_T]^T$.

B. Aircraft Nonlinear Dynamics

Among the nonlinear equations that describe the dynamics of the aircraft, those involving the turn rates are of high interest. They show the explicit relationship between turn rates and the torques applied to the aircraft (i.e., L, M, and N, expressed in the body axes x_b, y_b, and z_b of the aircraft):

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \boldsymbol{I}^{-1}\left(\begin{bmatrix} L \\ M \\ N \end{bmatrix}^b - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \boldsymbol{I} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) \tag{4}$$

In the context of this paper, the aircraft is a small unmanned aerial vehicle for which the aerodynamic moments were modeled as follows [10]:

$$\begin{aligned} L &= \bar{q}SbC_L(\delta_{a1}, \delta_{a2}, \delta_{e1}, \delta_{e2}, p, r, \beta) \\ M &= \bar{q}S\bar{c}C_M(\delta_{a1}, \delta_{a2}, \delta_{e1}, \delta_{e2}, \alpha, q) \\ N &= \bar{q}SbC_N(\delta_{a1}, \delta_{a2}, \delta_{e1}, \delta_{e2}, \delta_r, \beta, r) \end{aligned} \tag{5}$$

where $\bar{q} = \rho V_T^2/2$ is the dynamic pressure.

The aerodynamic derivatives are expressed as a linear combination of the state elements and control inputs as

$$\begin{aligned} C_L &= C_{L_{a1}}\delta_{a1} + C_{L_{a2}}\delta_{a2} + C_{L_{e1}}\delta_{e1} + C_{L_{e2}}\delta_{e2} + C_{L\tilde{p}}\tilde{p} + C_{L\tilde{r}}\tilde{r} + C_{L\beta}\beta \\ C_M &= C_{M_{a1}}\delta_{a1} + C_{M_{a2}}\delta_{a2} + C_{M_{e1}}\delta_{e1} + C_{M_{e2}}\delta_{e2} + C_{M\tilde{q}}\tilde{q} + C_{M\alpha}\alpha \\ C_N &= C_{N_{\delta_r}}\delta_r + C_{N\tilde{r}}\tilde{r} + C_{N\beta}\beta \end{aligned} \tag{6}$$

with

$$\tilde{p} = \frac{bp}{2V_T}, \qquad \tilde{q} = \frac{\bar{c}q}{2V_T}, \qquad \tilde{r} = \frac{br}{2V_T} \tag{7}$$

The last two nonlinear differential equations concern the angle of attack α and the sideslip angle β as follows:

$$\begin{aligned} \dot{\alpha} &\approx q + \frac{g}{V_T}\left\{1 + \frac{\bar{q}S}{mg}([C_{X_1} + C_{Z_2}]\alpha + C_{Z_1})\right\} \\ \dot{\beta} &\approx -r + \frac{\bar{q}SC_{Y1}}{mV_T}\beta \end{aligned} \tag{8}$$

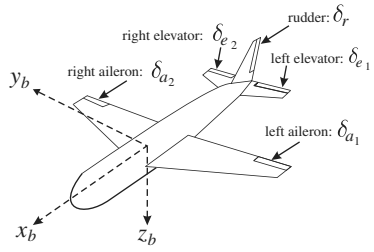

Fig. 4 Aircraft configuration.

with the drag derivative C_{X_1}, the side-force derivative C_{Y_1} and the lift derivatives C_{Z_1} and C_{Z_2} are constants (also see Table A1 in the Appendix). The inertia matrix is

$$\boldsymbol{I} = \begin{bmatrix} I_{xx} & 0 & I_{xz} \\ 0 & I_{yy} & 0 \\ I_{zx} & 0 & I_{zz} \end{bmatrix}$$

and the measurement vector is $\boldsymbol{y} = [p \;\; q \;\; r \;\; \alpha \;\; \beta]^T$.

C. Extended Kalman Filter Equations

The extended Kalman filters are designed based on a set of continuous nonlinear differential equations that describe the plant under consideration [11] as follows:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{w} \tag{9}$$

where $\boldsymbol{x}$ is the system state vector, $\boldsymbol{u}$ is the control vector, $\boldsymbol{f}(\boldsymbol{x})$ is the set of nonlinear functions of this state vector, and $\boldsymbol{w}$ is random zero-mean process noise. Equation (9) is first linearized around the current operating point and then discretized using the Euler integration method. Note that the Euler integration method is also used for the simulations presented in the following sections. The discrete equations are expressed in state-space form as

$$\boldsymbol{x}_{k+1} = \phi_k \boldsymbol{x}_k + \boldsymbol{G}_k \boldsymbol{u}_k + \boldsymbol{w}_k \tag{10}$$

where the state vector $\boldsymbol{x}_k$ is evaluated at the discrete time instant $t_k = kT_s$, with T_s being the sampling period of the system. The control input vector at time step k is $\boldsymbol{u}_k$, and the discrete random zero-mean process noise $\boldsymbol{w}_k$ is used to describe uncertainties in our model.

Finally, the discrete form of the measurement equation, either a linear or nonlinear function of the states, is $\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{v}_k$, where $\boldsymbol{v}_k$ is a discrete zero-mean random process described by the measurement noise covariance matrix $\boldsymbol{R}_{v,k} = E\{\boldsymbol{v}_k \boldsymbol{v}_k^T\}$ and consists of the variances of each of the measurement noise sources. The discrete transition matrix ϕ_k is approximated by $\phi_k \approx \boldsymbol{I} + \boldsymbol{F}(k)T_s$, where the continuous system dynamics matrix $\boldsymbol{F}(k)$ is obtained by linearizing the continuous nonlinear equations and is evaluated at the latest available state estimate $\hat{\boldsymbol{x}}_{k|k}$ according to

$$\boldsymbol{F}(k) = \left. \frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}} \right|_{\boldsymbol{x} = \hat{\boldsymbol{x}}_{k|k},\, \boldsymbol{u} = \boldsymbol{u}_k}$$

Similarly, the continuous measurement matrix $\boldsymbol{H}(k)$ is computed by linearizing the (possibly nonlinear) set of continuous measurement equations $\boldsymbol{h}(\boldsymbol{x})$ and is successively evaluated at the latest available state estimate $\hat{\boldsymbol{x}}_{k|k-1}$ according to

$$\boldsymbol{H}(\boldsymbol{k}) = \left. \frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}} \right|_{\hat{\boldsymbol{x}}_{k|k-1}}$$

The equations used in the extended Kalman filters are described next [11,12]:

1) The Kalman gains are computed as

$$\boldsymbol{L}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T [\boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T + \boldsymbol{R}_{v,k}]^{-1}$$

and are a function of the last propagated state error covariance matrix $\boldsymbol{P}_{k|k-1}$ and of the measurement noise covariance matrix $\boldsymbol{R}_{v,k}$.

2) The measurement update of the state estimate is the sum of the last extrapolated state estimate $\hat{\boldsymbol{x}}_{k|k-1}$ plus the Kalman gain times a residual made of the difference between the measurement vector $\boldsymbol{y}_k$ and the estimated measurement vector $\boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k-1})$, where $\boldsymbol{h}(.)$ is the set of continuous nonlinear measurement equations, yielding

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{L}_k [\boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k-1})]$$

3) The update of the state error covariance matrix $\boldsymbol{P}_{k|k}$ can be recursively computed as a function of the last predicted state error covariance matrix $\boldsymbol{P}_{k|k-1}$ and the last computed Kalman gain $\boldsymbol{L}_k$ as $\boldsymbol{P}_{k|k} = [\boldsymbol{I} - \boldsymbol{L}_k \boldsymbol{H}_k] \boldsymbol{P}_{k|k-1}$.

4) The propagation forward of the state error covariance matrix is $\boldsymbol{P}_{k+1|k} = \phi_k \boldsymbol{P}_{k|k} \phi_k^T + \boldsymbol{R}_w$.

5) The propagation forward of the state estimate does not have to be done with the transition matrix ϕ_k, but instead it is done directly by integrating the actual nonlinear differential equations forward at each sampling interval.

If the Euler integration technique is used, the extrapolated state estimated is computed with $\hat{\boldsymbol{x}}_{k+1|k} = \hat{\boldsymbol{x}}_{k|k} + \dot{\hat{\boldsymbol{x}}}_{k|k} T_s$, where the derivative is obtained from $\dot{\hat{\boldsymbol{x}}}_{k|k} = \boldsymbol{f}(\hat{\boldsymbol{x}}_{k|k})$.

D. Designing the EKF for the No-Fault Scenario

The continuous system dynamics matrix for the no-fault filter $\boldsymbol{F}_{\boldsymbol{nf}}(k)$ evaluated at time step k can be explicitly derived from the nonlinear model as

$$\boldsymbol{F}_{\boldsymbol{nf}}(k) = \begin{bmatrix} \frac{I_{zz}Sb^2C_{L\bar{p}}}{2D_1V_T}\bar{q} - \frac{N_1}{D_1}q & \frac{-N_1}{D_1}p + \frac{N_2}{D_1}r & \frac{(I_{zz}C_{L\bar{r}} - I_{xz}C_{N\bar{r}})Sb^2}{2D_1V_T}\bar{q} + \frac{N_2}{D_1}q & 0 & \frac{Sb[I_{zz}C_{L\beta} - I_{xz}C_{N\beta}]}{D_1}\bar{q} \\ \frac{I_{xx}-I_{zz}}{I_{yy}}r - \frac{2I_{zx}}{I_{yy}}p & \frac{S\bar{c}^2C_{M\bar{q}}}{2V_TI_{yy}}\bar{q} & -\frac{I_{xx}-I_{zz}}{I_{yy}}p - \frac{2I_{xz}}{I_{yy}}r & \frac{S\bar{c}C_{M\alpha}}{I_{yy}}\bar{q} & 0 \\ \frac{-I_{xz}Sb^2C_{L\bar{p}}}{2D_1V_T}\bar{q} + \frac{N_3}{D_1}q & \frac{N_3}{D_1}p + \frac{N_1}{D_1}r & \frac{(-I_{xz}C_{L\bar{r}} + I_{xx}C_{N\bar{r}})Sb^2}{2D_1V_T}\bar{q} + \frac{N_1}{D_1}q & 0 & \frac{Sb[I_{xx}C_{N\beta} - I_{xz}C_{L\beta}]}{D_1}\bar{q} \\ 0 & 1 & 0 & \frac{\rho V_T S C_{Z2}}{2m} & 0 \\ 0 & 0 & -1 & 0 & \frac{\rho V_T S C_{Y1}}{2m} \end{bmatrix}_{\hat{\boldsymbol{x}}_{\boldsymbol{nf}(k|k)}} \tag{11}$$

with $D_1 = I_{xx}I_{zz} - I_{xz}^2$, $N_1 = I_{xz}(I_{xx} - I_{yy} + I_{zz})$, $N_2 = I_{yy}I_{zz} - I_{xz}^2 - I_{zz}^2$, and $N_3 = I_{xz}^2 - I_{xx}I_{yy} + I_{xx}^2$.

The discrete transition matrix for the no-fault filter is calculated with $\phi_{\boldsymbol{k,nf}} = \boldsymbol{I} + \boldsymbol{F}_{\boldsymbol{nf}}(k)T_s$. We also compute the control input matrix of the no-fault filter $\boldsymbol{G}_{\boldsymbol{NF}}(k)$ as follows:

$$\boldsymbol{G}_{\boldsymbol{nf}}(k) = \begin{bmatrix} \frac{SbI_{zz}C_{La_1}}{D_1}\bar{q} & \frac{SbI_{zz}C_{La_2}}{D_1}\bar{q} & \frac{SbI_{zz}C_{Le_1}}{D_1}\bar{q} & \frac{SbI_{zz}C_{Le_2}}{D_1}\bar{q} & \frac{-SbI_{xz}C_{N_{\delta r}}}{D_1}\bar{q} \\ \frac{S\bar{c}C_{Ma_1}}{I_{yy}}\bar{q} & \frac{S\bar{c}C_{Ma_2}}{I_{yy}}\bar{q} & \frac{S\bar{c}C_{Me_1}}{I_{yy}}\bar{q} & \frac{S\bar{c}C_{Me_2}}{I_{yy}}\bar{q} & 0 \\ \frac{-SbI_{xz}C_{La_1}}{D_1}\bar{q} & \frac{-SbI_{xz}C_{La_2}}{D_1}\bar{q} & \frac{-SbI_{xz}C_{Le_1}}{D_1}\bar{q} & \frac{-SbI_{xz}C_{Le_2}}{D_1}\bar{q} & \frac{SbI_{xx}C_{N_{\delta r}}}{D_1}\bar{q} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{\bar{q}(k)} \tag{12}$$

The discrete control input matrix for the no-fault filter is obtained with $\boldsymbol{G}_{\boldsymbol{k,nf}} = \boldsymbol{G}_{\boldsymbol{nf}}(k)T_s$.

E. Augmenting the State Vector with the Faulty-Actuator Deflection $\bar{\delta}_i$

The state vector of the ith filter is augmented to monitor the occurrence of the ith actuator fault. The deflection of the failed actuator is included in the state vector in a way to be estimated by the EKF. Therefore, the state vector for each filter i is

$$\boldsymbol{z}_i = \begin{bmatrix} \boldsymbol{x} \\ \bar{\delta}_i \end{bmatrix} \tag{13}$$

The augmented state vector leads to the following state-space equations for each filter i

$$\begin{bmatrix} \boldsymbol{x}(k+1) \\ \bar{\delta}_i(k+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}(k) & \boldsymbol{G}^{(i)}(k) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(k) \\ \bar{\delta}_i(k) \end{bmatrix} + \begin{bmatrix} {}^{(0,i)}\boldsymbol{G} \\ 0 \end{bmatrix} \boldsymbol{\delta}(k)$$
$$\boldsymbol{y}(k) = [\boldsymbol{H}(k) \;\; 0] \begin{bmatrix} \boldsymbol{x}(k) \\ \bar{\delta}_i(k) \end{bmatrix} \tag{20}$$

F. Designing the EKF for the Case of a Failure on Aileron 1

To provide an example of how to derive the matrices for the EKF corresponding to the scenario of a lock-in-place or floating actuator failure, we will consider the filter that monitors the health of aileron 1:

$$\boldsymbol{F}_{\delta_{a1}}(k) = \begin{bmatrix} \frac{I_{zz}Sb^2C_{L\tilde{p}}}{2D_1V_T}\bar{q} - \frac{N_1}{D_1}q & \frac{-N_1}{D_1}p + \frac{N_2}{D_1}r & \frac{(I_{zz}C_{L\tilde{r}} - I_{xz}C_{N\tilde{r}})Sb^2}{2D_1V_T}\bar{q} + \frac{N_2}{D_1}q & 0 & \frac{Sb[I_{zz}C_{L\beta} - I_{xz}C_{N\beta}]}{D_1}\bar{q} & \frac{SbI_{zz}C_{La_1}}{D_1}\bar{q} \\ \frac{I_{xx}-I_{zz}}{I_{yy}}r - \frac{2I_{zx}}{I_{yy}}p & \frac{S\bar{c}^2C_{M\tilde{q}}}{2V_TI_{yy}}\bar{q} & -\frac{I_{xx}-I_{zz}}{I_{yy}}p - \frac{2I_{xz}}{I_{yy}}r & \frac{S\bar{c}C_{M\alpha}}{I_{yy}}\bar{q} & 0 & \frac{S\bar{c}C_{Ma_1}}{I_{yy}}\bar{q} \\ \frac{-I_{xz}Sb^2C_{L\tilde{p}}}{2D_1V_T}\bar{q} + \frac{N_3}{D_1}q & \frac{N_3}{D_1}p + \frac{N_1}{D_1}r & \frac{(-I_{xz}C_{L\tilde{r}} + I_{xx}C_{N\tilde{r}})Sb^2}{2D_1V_T}\bar{q} + \frac{N_1}{D_1}q & 0 & \frac{Sb[I_{xx}C_{N\beta} - I_{xz}C_{L\beta}]}{D_1}\bar{q} & \frac{-SbI_{xz}C_{La_1}}{D_1}\bar{q} \\ 0 & 1 & 0 & \frac{\rho V_TSC_{Z2}}{2m} & 0 & 0 \\ 0 & 0 & -1 & 0 & \frac{\rho V_TSC_{Y1}}{2m} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{\hat{z}_1(k|k)} \tag{21}$$

$$\boldsymbol{z}_i(k+1) = \boldsymbol{f}_{z_i}[\boldsymbol{z}_i(k), \boldsymbol{\delta}(k)] + \boldsymbol{w}_k, \qquad \boldsymbol{y}_i(k) = \boldsymbol{h}[\boldsymbol{z}_i(k)] + \boldsymbol{v}_k \tag{14}$$

where

$$\boldsymbol{f}_{z_i}[\boldsymbol{z}_i(k), \delta(k)] = \begin{Bmatrix} \boldsymbol{f}[\boldsymbol{z}_i(k), \boldsymbol{\delta}(k)] \\ \bar{\delta}_i(k) \end{Bmatrix} \tag{15}$$

The linearization of the dynamics matrix yields

$$\boldsymbol{F}_{z_i}(k) = \frac{\partial}{\partial \boldsymbol{z}_i}\boldsymbol{f}_{z_i}[\boldsymbol{z}_i(k), \boldsymbol{\delta}(k)]\bigg|_{z_i = \hat{z}_i(k|k)} = \begin{bmatrix} \boldsymbol{F}(k) & \boldsymbol{G}^{(i)}(k) \\ 0 & 1 \end{bmatrix} \tag{16}$$

with $\boldsymbol{G}^{(i)}$ representing the ith column of the matrix $\boldsymbol{G}$. The input matrix becomes

$$\boldsymbol{G}_{z_i}(k) = \frac{\partial}{\partial \bar{\delta}_i}\boldsymbol{f}_{z_i}[\boldsymbol{z}_i(k), \boldsymbol{\delta}(k)]\bigg|_{z_i(t) = \hat{z}_i(k|k)} = \begin{bmatrix} {}^{(0,i)}\boldsymbol{G}(k) \\ 0 \end{bmatrix} \tag{17}$$

with ${}^{(0,i)}\boldsymbol{G}$ representing the matrix $\boldsymbol{G}$ with the ith column set to zero. The linearization of the measurement matrix is

$$\boldsymbol{H}_{z_i}(k) = \frac{\partial}{\partial \boldsymbol{z}_i}\boldsymbol{h}[\boldsymbol{z}_i(k), \boldsymbol{\delta}(k)]\bigg|_{z_i(t) = \hat{z}_i(k|k-1)} = [\boldsymbol{C}_x(k) \;\; \boldsymbol{C}_{\bar{\delta}_i}(k)] \tag{18}$$

where

$$\boldsymbol{C}_x(k) = \frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{h}[\boldsymbol{z}_i(k)]\bigg|_{z_i(t) = \hat{z}_i(k|k-1)} = \boldsymbol{H}(k)$$
$$\boldsymbol{C}_{\bar{\delta}_i}(k) = \frac{\partial}{\partial \bar{\delta}_i}\boldsymbol{h}[\boldsymbol{z}_i(k)]\bigg|_{z_i = \hat{z}_i(k|k-1)} = 0 \tag{19}$$

Using Eqs. (13–19), the linearized system evaluated at each sampling time can be written as

$$\boldsymbol{G}_{\delta_{a1}}(k) = \begin{bmatrix} 0 & \frac{SbI_{zz}C_{La_2}}{D_1}\bar{q} & \frac{SbI_{zz}C_{Le_1}}{D_1}\bar{q} & \frac{SbI_{zz}C_{Le_2}}{D_1}\bar{q} & \frac{-SbI_{xz}C_{N_{\delta r}}}{D_1}\bar{q} \\ 0 & \frac{S\bar{c}C_{Ma_2}}{I_{yy}}\bar{q} & \frac{S\bar{c}C_{Me_1}}{I_{yy}}\bar{q} & \frac{S\bar{c}C_{Me_2}}{I_{yy}}\bar{q} & 0 \\ 0 & \frac{-SbI_{xz}C_{La_2}}{D_1}\bar{q} & \frac{-SbI_{xz}C_{Le_1}}{D_1}\bar{q} & \frac{-SbI_{xz}C_{Le_2}}{D_1}\bar{q} & \frac{SbI_{xx}C_{N_{\delta r}}}{D_1}\bar{q} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{\bar{q}(k)} \tag{22}$$

G. Computational Complexity

Let us denote by N the number of actuators that are monitored by an FDI system. In the classical implementation of an MMAE FDI, for each actuator k, filters are to be designed for k different possible positions of the failed actuator. Therefore, a number of $kN + 1$ Kalman filters are required ($+1$ refers to the no-fault filter). If the appearance of a second fault is to be checked as well, then a new bank of Kalman filters has to be reloaded, based on the knowledge of the first fault that occurred. In total, $Nk + 1 + (N-1)kN = N^2k + 1$ Kalman filters must be designed.

One major advantage of the EMMAE-FDI method presented in this paper over classical MMAE schemes is that it requires only one filter to completely monitor one actuator. Any possible actuator-fault scenario is taken into account by only one filter. Therefore, for the monitoring of a single actuator fault, a number of only $N + 1$ filters are required with the EMMAE-FDI method. For the monitoring of a second actuator fault with the EMMAE-FDI method, no other bank of filters has to be loaded. Indeed, if actuator i fails, it suffices to feed all the other filters with the estimate of the faulty control surface deflection $\hat{\bar{\delta}}_i$ instead of the input δ_i. Thus, again only $N + 1$ filters are needed with the EMMAE-FDI system to detect and isolate a second fault after a first actuator fault has been introduced.

As a simple example, our UAV is equipped with five actuators. The classical MMAE scheme designed for three possible faulty deflections per actuator requires 16 filters, whereas the EMMAE

method needs only six filters for lock-in-place and floating actuator-fault scenarios.

Finally, the choice of the state vector $\boldsymbol{x} = [p \;\; q \;\; r \;\; \alpha \;\; \beta \;\; \hat{\delta}]^T$ minimizes the number of relevant state elements for the satisfactory operation of the filters, thus limiting the number of computations required for the EKFs. Indeed, in previous works [7,8], up to nine states in the state vector had to be used.

V. Actuator-Fault Isolation

A hypothesis-testing algorithm uses the residuals and the state error covariance matrix from each EKF to assign a conditional probability to each fault scenario. The estimated state vector of the system is the sum of the state vector of each EKF weighted by its corresponding probability:

$$\hat{\boldsymbol{x}}[k] = \sum_i \hat{\boldsymbol{x}}_i[k].p_i[k]$$

where $\hat{\boldsymbol{x}}_i[k]$ is the state estimate computed by the EKF that assumes the fault scenario θ_i. The index i covers all the fault scenarios implemented, including the no-fault case. By $p_i[k]$, we denote the probability that the ith fault scenario is occurring. Now the main difficulty is in computing the online probability $p_i[k]$. To determine which fault-scenario the actual plant is the closest to, we must consider the measurement data from the sensors. The last available measurement vector is $\boldsymbol{y}[k]$. We also define the sequence of the last measurement vectors as $\boldsymbol{Y}_k = \{\boldsymbol{y}[k], \boldsymbol{y}[k-1], \ldots, \boldsymbol{y}[0]\}$. The fault probability $p_i[k]$ can be expressed as the *a posteriori conditional probability* $p_i[k] = p(\theta = \theta_i | \boldsymbol{Y}_k)$; that is, the probability that the actual plant can be categorized in the scenario θ_i, given the sequence of the last measurements $\boldsymbol{Y}_k$:

$$p_i[k] = p[(\theta = \theta_i)|\boldsymbol{Y}_k] = \frac{p[(\boldsymbol{y} = \boldsymbol{y}_k)|(\theta = \theta_i, \boldsymbol{Y}_{k-1})]p_i[k-1]}{\sum_{j=0}^{N} p[(\boldsymbol{y} = \boldsymbol{y}_k)|(\theta = \theta_j, \boldsymbol{Y}_{k-1})]p_j[k-1]} \tag{23}$$

We now derive an explicit formula for the term $p[(\boldsymbol{y} = \boldsymbol{y}_k)|(\theta = \theta_i, \boldsymbol{Y}_{k-1})]$, which corresponds to the probability of obtaining the measurement data $\boldsymbol{y}[k]$ at time $t_k = kT_s$, assuming that scenario θ_i exists and given the sequence of the last measurements $\boldsymbol{Y}_{k-1}$. (The last measurement data available, $\boldsymbol{y}[k]$, are sometimes written as $\boldsymbol{y}_k$.) The probability density is chosen to be a Gaussian function [12] with its characteristic bell-shaped curve according to the following formula:

$$p[(\boldsymbol{y} = \boldsymbol{y}[k])|(\theta = \theta_i, \boldsymbol{Y}_{k-1})] = \lambda_i[k] e^{-\frac{\boldsymbol{r}_i[k]^T \boldsymbol{P}_i^{-1} \boldsymbol{r}_i[k]}{2}} \tag{24}$$

with

$$\lambda_i[k] = \frac{1}{(2\pi)^{m/2} |\boldsymbol{P}_i[k]|^{1/2}}$$

where $|\ldots|$ denotes the determinant of the matrix, m represents the measurement dimension, and $\boldsymbol{P}_i[k]$ is the residual covariance matrix calculated at time step k in the ith extended Kalman filter. The term $\boldsymbol{r}_i[k]$ corresponds to the residuals of the ith extended Kalman filter, when the measurement update step occurs according to the equation $\boldsymbol{r}_i[k] = \boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_i[k|k-1])$. The probability $p[(\boldsymbol{y} = \boldsymbol{y}_k)|(\theta = \theta_i, \boldsymbol{Y}_{k-1})]$ is given by $f[\boldsymbol{r}_i = \boldsymbol{r}_i[k]|(\theta = \theta_i, \boldsymbol{Y}_{k-1})]$, with the probability density defined as a function of the residual $\boldsymbol{r}_i$ with

$$f[\boldsymbol{r}_i|(\theta = \theta_i, \boldsymbol{Y}_{k-1})] = \frac{1}{(2\pi)^{m/2} |\boldsymbol{P}_i[k]|^{1/2}} e^{-\frac{\boldsymbol{r}_i^T \boldsymbol{P}_i^{-1}[k] \boldsymbol{r}_i}{2}}$$

Therefore, the filter that corresponds to the fault scenario produces a value $\hat{\boldsymbol{y}}_i[k] = \boldsymbol{h}[\hat{\boldsymbol{x}}_i(k|k-1)]$ that is very close (apart from noise) to the actual value of the measurement data $\boldsymbol{y}_k$. This means that the conditional probability $p[(\boldsymbol{y} = \boldsymbol{y}_k)|(\theta = \theta_i, \boldsymbol{Y}_{k-1})]$ is the highest for the filter matching the fault scenario. By examining the probabilities, we can determine which is the health status of the system. It is either the no-fault case or a case in which an actuator is locked-in-place or floating.

VI. Simulation Results of the Performance of the EMMAE FDI with no Supervision System

A. Simulation Conditions

To obtain realistic simulations, the sensor measurements are corrupted by zero-mean white Gaussian noise corresponding to typical specifications of low-cost sensors. For the turn rate sensors, the standard deviation is $\sigma_{p,q,r} = 5$ deg/s $= 0.0873$ rad/s, which corresponds to a noise covariance of $\boldsymbol{R}_{p,q,r} = 0.0076 \times \boldsymbol{I}_3$[rad^2/s^2]. For the airflow angle sensors, the noise standard deviation is $\sigma_{\alpha,\beta} = 2$ deg $= 0.0349$ rad, with a noise covariance of $\boldsymbol{R}_{\alpha,\beta} = 0.0012 \times \boldsymbol{I}_2$ [rad^2]. The airspeed sensor noise has a standard deviation of $\sigma_{V_T} = 1$ m/s, with a noise covariance of $\boldsymbol{R}_{V_T} = 1$ m^2/s^2. Poor sensor quality adversely affects the FDI reliability. When there is little excitation and when the faulty-actuator deflection is close to the trim conditions, it becomes even more difficult to detect an actuator failure. Indeed, due to large-sensor noise, the control signals (see Fig. 5) become noisy as well, which reduces the difference between the actual faulty-actuator deflection and its corresponding control signal. The larger this difference, the easier it is to detect the fault. The EKF process noise covariance matrix and the sensor noise covariance matrix are selected as follows: $\boldsymbol{R}_w = 0.002 \times \boldsymbol{I}_5$ and $\boldsymbol{R}_v = \text{diag}[0.1 \times \boldsymbol{I}_3 \;\; 0.02 \times \boldsymbol{I}_2]$.

An actuator fault is declared valid if the corresponding fault probability exceeds 90% for a certain amount of time. A fault is declared removed when the corresponding fault probability is below 5% for a certain amount of time.

Finally, to prevent the possibility that the recursive computation of the fault probability [see Eq. (23)] stays at zero forever as soon as the probability reaches zero, the lower bound of each probability is set to 0.001.

The hypothesis testing uses a Gaussian density function, which assumes that the residuals from the EKFs are Gaussian-distributed. When this is not the case, there is a little inconsistency with the application of the theory. However, assuming that these residuals are Gaussian-distributed is still reasonable, especially when the aircraft dynamics are slow. A Gaussian density function was chosen for simplicity, because it is entirely defined by the two parameters computed by the EKFs: namely, the residual vector and the state error covariance matrix. If another probability density function would be known, then Eq. (24) could be changed accordingly.

The scenario to test the fault detection method is chosen to put the FDI in the most difficult conditions, which are those of minimum excitation of the system. This is achieved when the aircraft is flying straight and level (no maneuvers and no wind) at a constant speed of 30 m/s. The actuator faults are simulated by blocking the control surfaces close to the trim deflections corresponding to straight level flight conditions, because those faults are harder to detect and to estimate. For example, the ailerons and the rudder will fail close to the neutral deflection, and the elevators will fail close to -2 deg. The fault detection with the EMMAE method is tested on a 6-DOF nonlinear aircraft model of a UAV currently in use at the laboratory. Simulations were performed in Matlab/Simulink on closed-loop control architecture, with a nonlinear autopilot that regulates the speed, altitude, and attitude of the aircraft. The EMMAE-FDI system is therefore composed of six EKFs, one for monitoring the no-fault case, two for monitoring each aileron (one on the left and one on the right wing), two EKFs for monitoring each of the two independent elevators, and one EKF for the rudder.

A sequence of consecutive faults is generated. From 10 s $< t <$ 40 s, aileron 1 fails and is locked at -1- deg deflection. For 70 s $< t <$ 100 s, aileron 2 fails and is "floating" between the two positions -1 deg and 1 deg in a square-wave fashion. For 130 s $< t <$ 160 s, the rudder fails and is locked at -1 deg. For 190 s $< t <$ 220 s, elevator 1 gets locked at -1 deg. Finally, for 250 s $< t <$ 280 s, elevator 2 is floating between two uncontrolled positions -0.5 deg and -3.5 deg in a square-wave fashion.

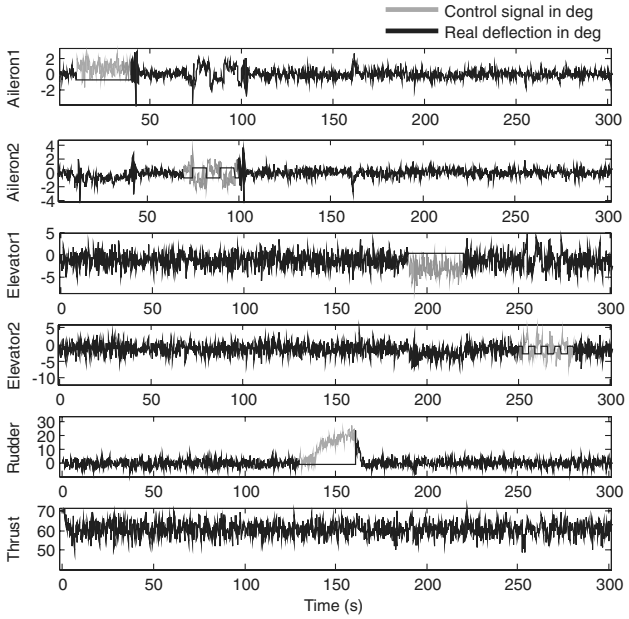

Fig. 5 Control signals and actual actuator deflections.

After this sequence of faults, the aircraft continues to fly straight and level, and no more faults are introduced.

B. Comments on Simulation Results

Figure 6 shows the results obtained by the FDI system after the sequence of faults. The top plot, labeled "No-fault," has a probability of 1 when the EMMAE-FDI system does not detect any fault in the aircraft. An actuator fault is declared valid if the corresponding fault probability exceeds 90% for a certain amount of time. A fault is declared removed when the corresponding fault probability is below 5% for a certain amount of time.

When aileron 1 fails at $t = 10$ s, the no-fault filter needs about 6 s for its probability to go down to almost zero, which means that a failure occurred somewhere. After the aileron 1 fault is introduced, both probabilities for ailerons 1 and 2 to fail (Pbaf1 and Pbaf2) start rising at $t = 11.5$ s. At $t = 17$ s, the FDI begins to distinguish between the two ailerons that have failed and Pbaf2 returns to zero, whereas Pbaf1 rises up to 90% at $t = 34$ s. Therefore, it took 24 s for the FDI to indicate that aileron 1 is experiencing a failure. At $t = 40$s, the fault of aileron 1 is removed and the actuator behaves normally again. Figure 6 shows that the probability Pbaf1, indicating whether aileron 1 fails, decreases slowly and reaches zero again after 10 s, whereas the no-fault probability rises accordingly. It thus takes 10 s for the FDI to indicate that the fault has been removed.

As Fig. 5 shows, at $t = 70$ s, aileron 2 fails and has an uncontrolled square-wave motion between -1 and 1 deg. Figure 6 shows that the FDI takes 2 s to detect that there is a failure somewhere in the system (see the drop of the no-fault probability at $t = 72$ s). However, we observe an ambiguity between the two ailerons for a few seconds before the probability Pbaf2 finally reaches 90% at $t = 80$ s. At $t = 100$ s, the fault of aileron 2 is removed and this actuator behaves normally again. However, the FDI is not capable of quickly detecting that the fault has been removed; it takes about 30 s to do so.

For the rudder, the fault is introduced at $t = 130$ s and is isolated by the FDI system at $t = 139$ s when Pbaf5 exceeds 90%. The fault removal is detected in less than 5 s. It takes less time to isolate a rudder fault and detect its removal, because there is only one rudder, unlike the other actuators that are redundant (two ailerons and two elevators). Therefore, a malfunctioning rudder cannot be compensated by a redundant rudder, thus resulting in no actuator-fault ambiguity.

The introduction of the elevator 1 fault at $t = 190$ s is isolated by its corresponding filter (probability signal Pbaf3 exceeding 90%) at $t = 198$ s. After the fault removal, the FDI system takes 10 s to indicate the fault removed. We observe the same kind of behavior for elevator 2 as for elevator 1. Finally, after the last fault has been removed, 22 s are needed by the FDI system to slowly build up probability in the no-fault filter to indicate that there is no more fault detected in the system.

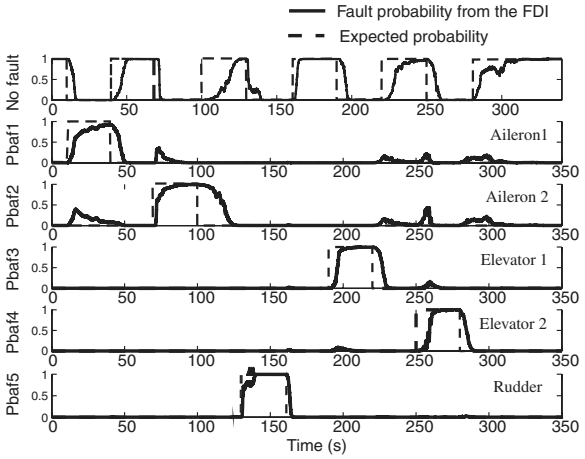

Fig. 6 Probabilities from each filter of the EMMAE FDI after a sequence of faults.

C. Remarks on the First Attempt of Using the EMMAE-FDI System

The results plotted in Fig. 6 indicate that the current implementation of the method is able to detect the fact that a failure occurred, even in a very low-excitation case, but it could not quickly and reliably tell which actuator experienced the fault. In cases of redundant actuators having the same influence on the aircraft aerodynamics, the EMMAE method has difficulties quickly resolving ambiguities between redundant actuators when they cannot be properly excited. It appears that failures of actuators near trim deflection are more difficult to detect and isolate.

Moreover, whenever a fault is removed, the EMMAE method alone requires a long time to detect that fact. It is critical, however, that the FDI can quickly detect the removal of a failure or quickly recognize a false alarm has been triggered due to possible external perturbations such as strong wind gusts.

Finally, it is critical that the probabilities quickly reach the expected values that correctly describe the fault scenario. Indeed, the estimated state vector of the system, which is the sum of the state vectors of each EKF weighted by its corresponding probability, must be sufficiently correct and accurate if this state estimate is fed back to the controller.

VII. Improvements to the EMMAE-FDI System

This section describes the techniques that were added to enhance the performance of the FDI system when there is very low excitation of the system, particularly during steady level flight. To improve the speed and the accuracy of the fault isolation, a supervision module is designed for which the tasks are detailed next.

A. Design of an Active Fault Supervision Module (Supervisor)

The supervision module shown in Fig. 7 is designed to monitor probability signals from the FDI. If an actuator-failure probability exceeds a certain threshold for a certain period of time, then the supervisor is designed to superimpose an artificial control signal on the corresponding actuator.

If an actuator fails, the additional signal will have no effect on the aircraft dynamics, but will help the FDI more quickly confirm the failure of this actuator. On the other hand, if the actuator has not actually failed, the aircraft will respond according to the additional signal, and the FDI will then remove the fault assigned to this actuator. If the corresponding fault probability falls below a certain

threshold for a defined period of time, the supervisor removes the superimposed signal.

Actuator excitation methods have already been developed for system identification purposes. Null-space injection methods were developed that work directly within optimization-based control allocation approaches that generate actuator excitation while still providing the desired pseudocommands [13,14]. In this work, the excitation signals are adaptively controlled within the limits from 1 to 4 deg, with the function

$$\delta_{\mathrm{ex}_i}(t) = \{1 + 3[1 - p_i(t)]\}\cos(2\pi f_i t)$$

Note that an actuator is only being excited when its corresponding fault probability p_i exceeds 5%. Although better performance might be expected if the excitation signals to each actuator were independent and decorrelated, we used the same excitation signal for all actuators. The frequency of the signal is to be chosen within the range of the aircraft bandwidth (in our case, $f_i = 1$ Hz). The excitation signal has an adaptive amplitude dependent on the probability p_i of actuator i to have failed. In this way, when the probability p_i is low, the excitation amplitude is large and vice versa. Simulation results show that this adaptive amplitude for the excitation signal efficiently improves the accuracy and speed for fault isolation compared with a fixed-amplitude excitation signal. This adaptive amplitude ensures that the actuator is excited as little as possible, but still enough to isolate the fault or to remove a false alarm.

This method is therefore a systematic way of testing each triggered failure alarm, to confirm it or to remove it, hence making the FDI more robust. Whereas other proposed schemes [5,15] suggest to have the aircraft perform a health-check maneuver or diagnostic maneuver as soon as a failure is detected, in the method of this paper, the actuator is directly excited by the supervision module rather than by the aircraft autopilot, yielding much faster and more accurate fault isolation.

B. Simulation Results of the Performance of the EMMAE FDI (with Supervision System)

1. *Detection Performance*

Figure 8 shows that the ailerons and rudder faults are quickly (2 s) and accurately detected and isolated. The elevator faults take longer to be isolated (about 5 s). However, the removal of all the faults is detected in less than 2 s. Furthermore, there is no more ambiguity or false detection among the actuator faults. Comparing the results shown in Fig. 8 with those of Fig. 6, we see that the performance and the robustness of the EMMAE-FDI system have greatly improved due to the supervision module.

2. *State Estimation Performance*

Figure 9 shows a comparison between noiseless (and noisy) measurements and the state estimate from the EMMAE method in the implementation of Fig. 7. Clearly, despite a large amount of noise in the sensor measurements, the state estimate accurately tracks the true (noiseless) measurements, and the filtered data enable the controller to generate control signals with less noise, which facilitates fault detection and isolation as well as state estimation from correct probability computations.

Furthermore, as shown in Fig. 10, the faulty control surface deflection is well estimated in the case of a frozen actuator, but it is slowly estimated in the case of uncontrolled square-wave motion of the control surface. An engineering fix to have the EKF track the faulty-actuator deflection is to increase the process noise on the desired control input to the detriment of estimate accuracy. In practice, however, it is more likely that an actuator simply gets locked at a certain deflection or floats around the local and slowly varying angle of attack, thereby reducing the challenge of estimating the faulty-actuator deflection.

As mentioned, the simulations that are presented here are for straight and level flight (i.e., the worst condition for any FDI system). In practice, a small UAV is maneuvering almost all the time, which further improves the results and the performance of the FDI system.

VIII. Realistic Flight Scenario

In this section, the FDI system is tested in a realistic flight scenario, in which the UAV takes off, tracks a predefined trajectory, and follows an altitude and a speed reference profile. Dryden wind turbulences are included in the simulation to test the robustness of the FDI system in the presence of external disturbances. Figure 11 shows the wind speeds in the north, east, and down directions. They correspond to severe wind conditions for the type of aircraft simulated, which is a small aerobatic model aircraft.

A. Testing the FDI System in Wind Conditions and Without Actuator Faults

The left plot of Fig. 12 shows that the aircraft takes off at point 1 and flies successively in the direction of points $2, 3, \ldots, 6$ and then again flies the whole sequence of points. The overshoot during the

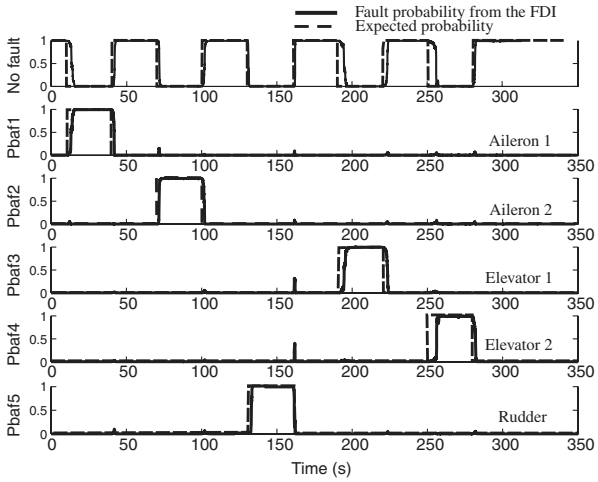

Fig. 8 Probabilities from each filter of the EMMAE FDI after a sequence of faults (with the supervisor).

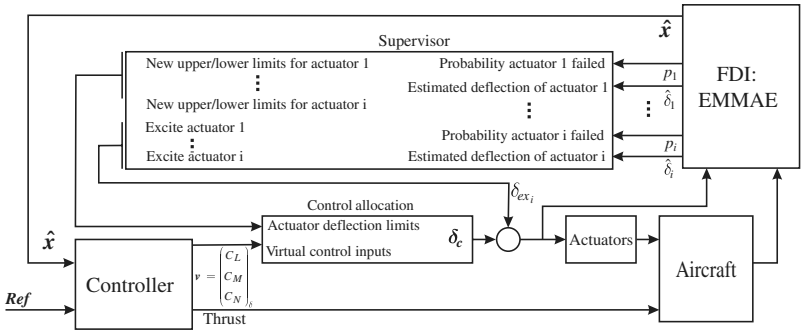

Fig. 7 EMMAE FDI with active fault supervision module system for a reconfigurable flight control system.

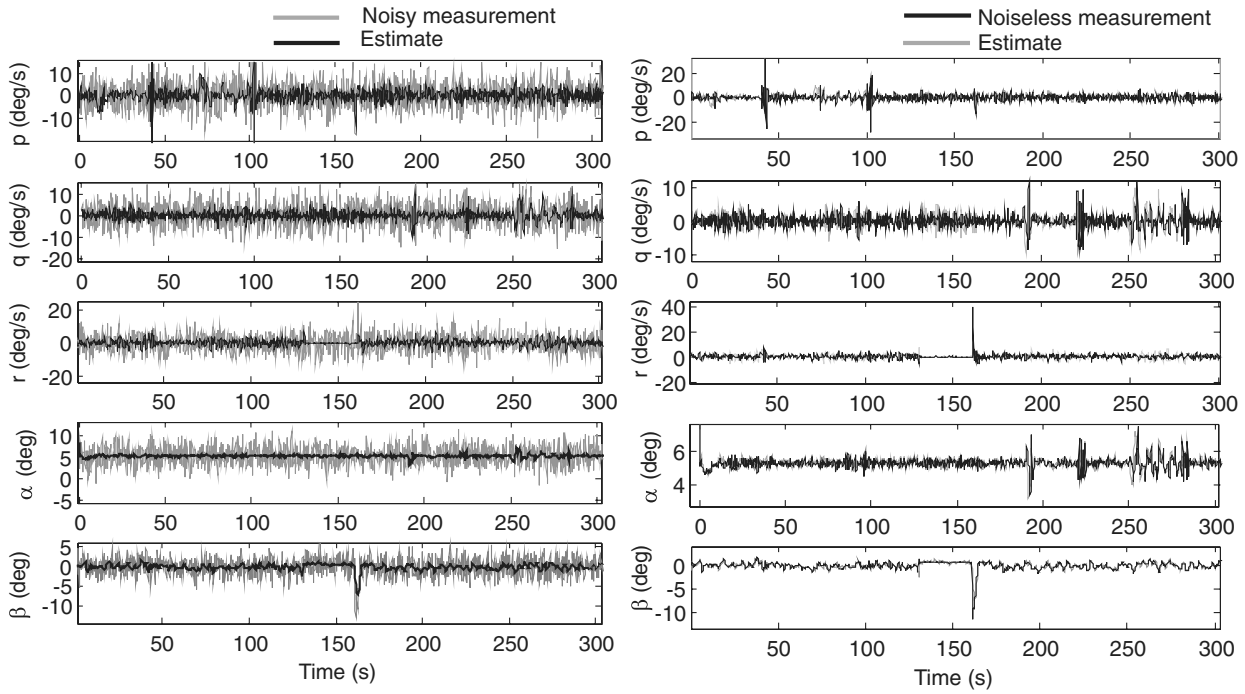

Fig. 9 Comparison between noisy (on the left) or noiseless (on the right) measurements and the probability-weighted state estimate from the EMMAE method.

turns after points 3 and 6 are due to the guidance system and the excessive speed of the aircraft when approaching these points. Lower speeds result in less overshoot. The speed profile shown in the top right plot of Fig. 12 is chosen to cover a significant range of the aircraft speed, to test the performance of the fault detection system at different operating conditions. Obviously, the measured airspeed V is corrupted by noise, like all the other measurement signals defined in Sec. VI that are used by the Kalman filters in the FDI system. Also, the altitude reference signal is chosen such that the aircraft has the vertical motion depicted in the bottom right plot of Fig. 12.

Figure 13 shows the fault probabilities computed by the EMMAE FDI under the condition of strong wind gusts. The usage of the active actuator-fault supervision system helps in removing some false alarms. For example, for 110 s $< t <$ 115 s or 195 s $< t <$ 200 s, the FDI system with no supervision module indicates that the ailerons are failing, whereas with the supervision module, these two false fault alarms do not appear. In such wind conditions, the active supervision system would be more effective if the amplitude of the excitation signals were bigger.

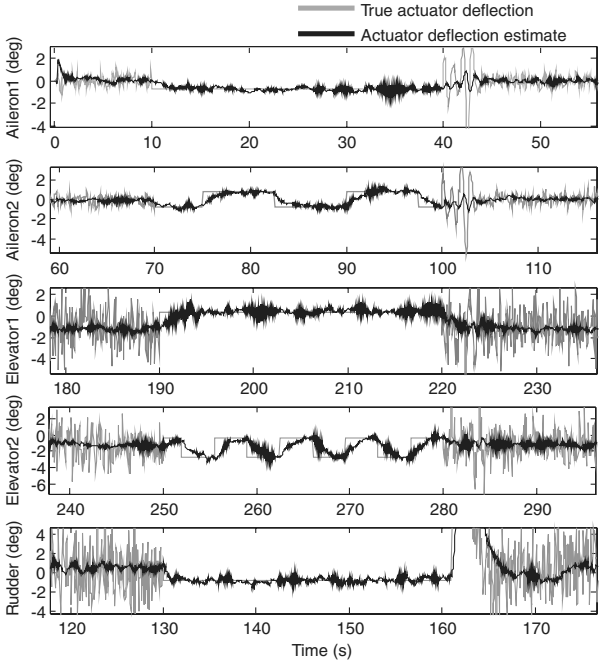

Fig. 10 True actuator positions and associated estimates (a position estimate is only valid during the occurrence of a fault).

B. With Strong Winds, Actuator Faults, and the Supervision Module

In this section, the previous flight scenario is simulated again with the sequence of faults presented in Sec. VI.A. The fault-probability signals for each actuator are generated by the EMMAE-FDI system and are shown in Fig. 14. The FDI system detects and isolates the actuator faults in less than 5 s for aileron 1 (Pbaf1), 3 s for aileron 2 (Pbaf2), 2 s for elevator 1 (Pbaf3), 3 s for elevator 2 (Pbaf4), and 1 s for the rudder (Pbaf5). In all cases, the removal of the fault is accurately detected in less than a second.

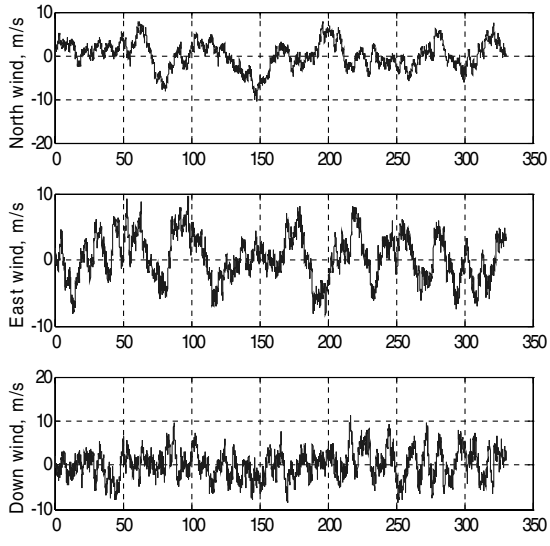

Fig. 11 Wind speed in meters per second.

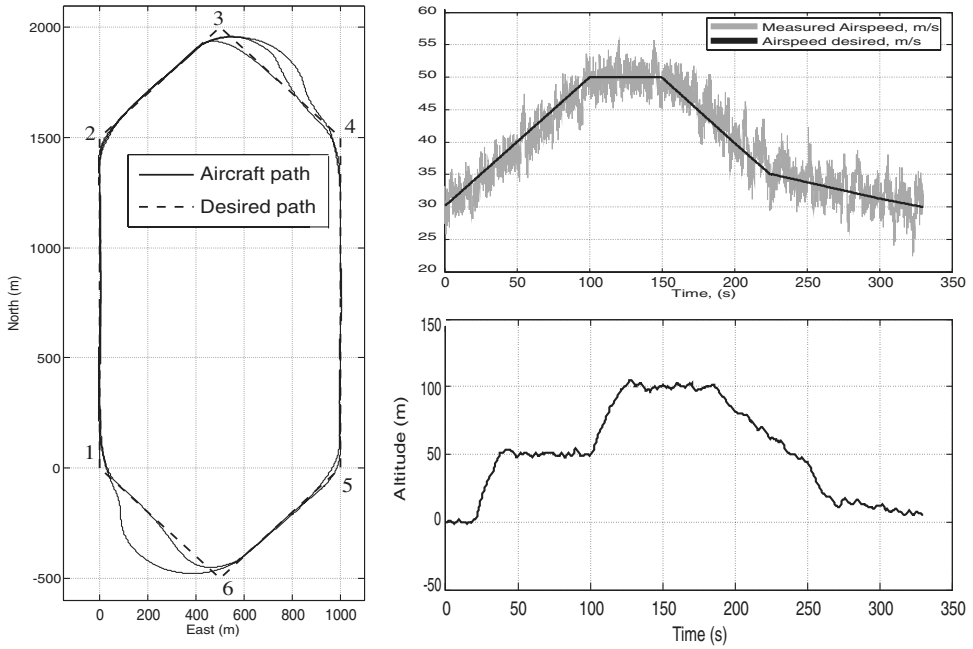

Fig. 12 Flight simulation.

IX. Additional Filtering Stage for the EMMAE-FDI System

The results shown in Fig. 14 are still not completely satisfactory, because the fault-probability signals are noisy. To improve the quality of the final fault diagnosis of the aircraft, an additional filtering stage is added to the EMMAE-FDI system. As shown in Fig. 15, each fault-probability signal $p_1, p_2, \ldots, p_i$ passes through a low-pass filter, for which the cutoff frequency is 0.2 Hz, and a logic block that sets its output to one if its input exceeds 0.6 and sets its output to zero when its input goes below 0.4. After passing through this additional filtering stage, the signals of Fig. 14 are transformed into the filtered actuator-fault probabilities $\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_i$; they are shown in Fig. 16. Although it takes longer to finally isolate the fault in the system, on average 2.5 s, the results shown in Fig. 16 are in accordance with the expected fault probabilities. We recall that this simulation is made in severe wind conditions, flying the flight path shown in Fig. 12 at different speeds and altitudes. Figure 16 shows that noise levels in the probability signals $\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_i$ are reduced and they can be used to select alternative flight modes in a reconfigurable flight controller, for example. Note that the excitation signals are still triggered by the unfiltered probability signals that come directly out of the EMMAE module.

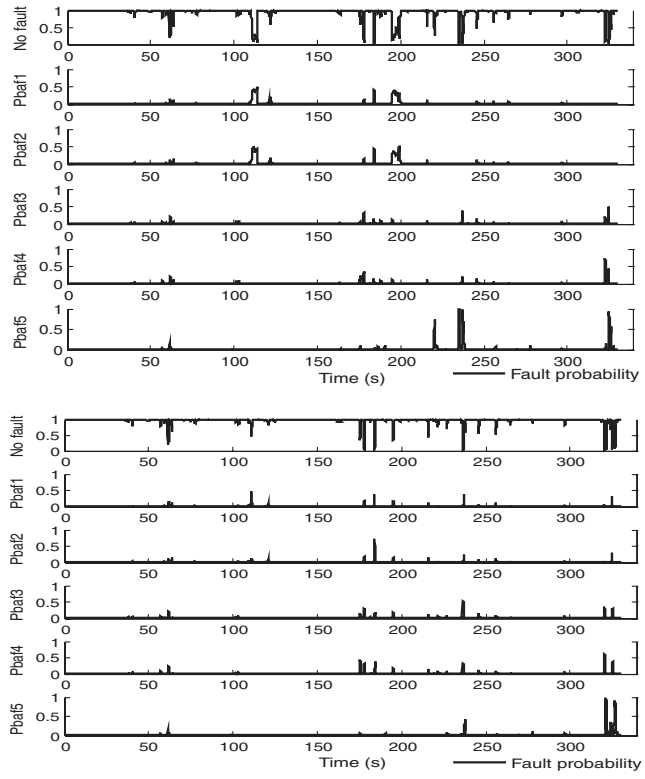

Fig. 13 Fault probabilities in the case of strong winds, no actuator fault, no active fault supervision system (top plot), and with active fault supervision system (bottom plot).

X. Detection and Isolation of Simultaneous Failures

This section addresses the case of a second fault occurring simultaneously in the system. The EMMAE FDI is slightly modified to detect and isolate the occurrence of a second fault and to continue the monitoring of the fault that first occurred. The system here described works properly for the detection and isolation of the second fault as long as the first fault does not change (i.e., the faulty control surface remains locked). In such a case, the filter i that

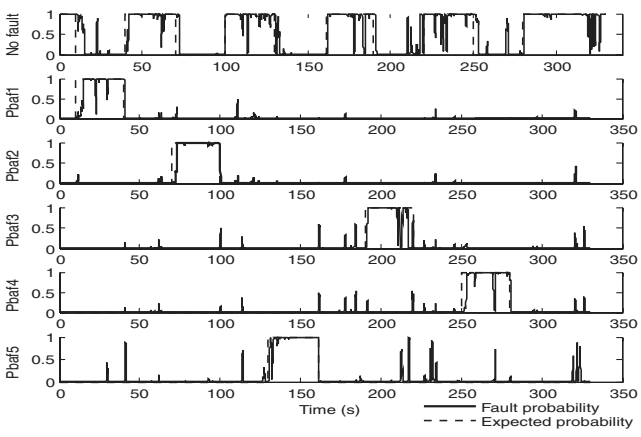

Fig. 14 Fault probabilities in the case of severe wind, with a sequence of actuator faults, with the active fault supervision system.

isolated the first actuator fault provides an estimation of the deflection of the faulty control surface. This estimation $\hat{\bar{\delta}}_i$ is used as a new input instead of the input δ_i for all the other filters in the bank of EKFs in the EMMAE FDI. Note that the value of $\hat{\bar{\delta}}_i$ used as input for the other filters must be the value before the second fault occurred. Indeed, when the second fault k occurs, then the ith filter has an input δ_k that is false, leading to the wrong estimation of $\hat{\bar{\delta}}_i$. Figure 17 shows the modifications that are to be made at the input of the EKFs in the case of the first failure occurring on actuator 1.

Figure 18 shows the probability signals $\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_i$ generated at the output of the FDI system shown in Fig. 15. Aileron 1 fails between $30\text{ s} < t < 110\text{ s}$ and elevator 1 fails between $50\text{ s} < t < 90\text{ s}$. Pbaf11 indicates the probability of aileron 1 failing only, Pbaf13 indicates the probably of both actuators (aileron 1 and elevator 1) failing at the same time. The system can properly detect and isolate the fault situations. The flight scenario that yields the results of Figs. 18 and 19 corresponds to straight level flight with no wind. The maximum time delay needed to isolate the first fault is 3.8 and 2 s for the second failure.

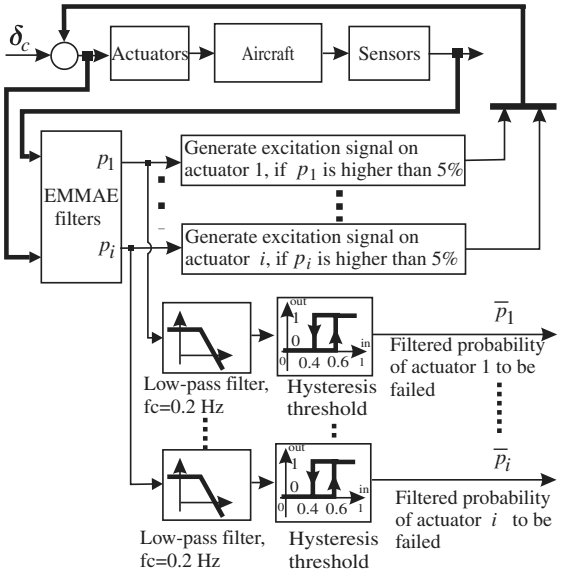

Fig. 15 Complete architecture of the active actuator-fault detection, isolation and supervision system.

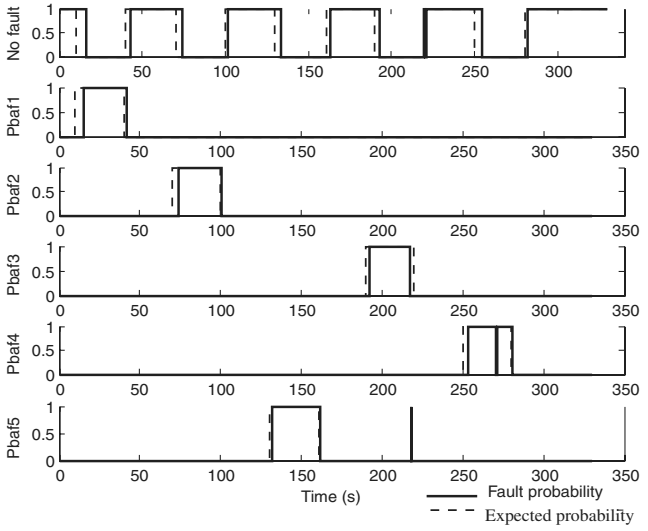

Fig. 16 Filtered actuator-fault probabilities $\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_i$ after the second stage of filtering in the active FDI supervision module.

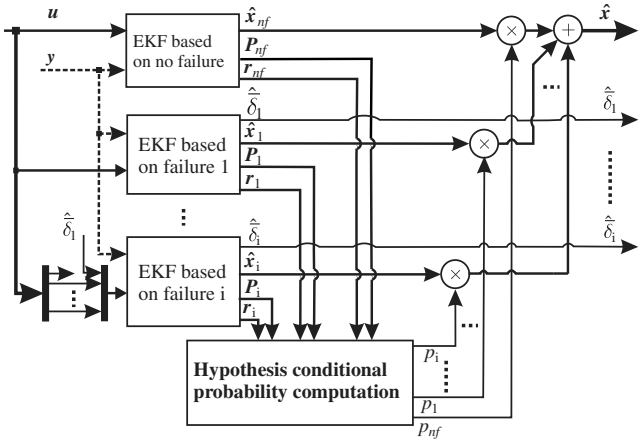

Fig. 17 Modified EMMAE FDI for the isolation of a second simultaneous fault.

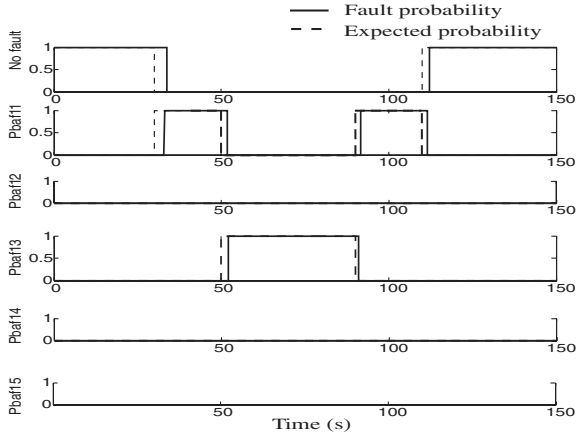

Fig. 18 Fault isolation in the case of two simultaneous failures.

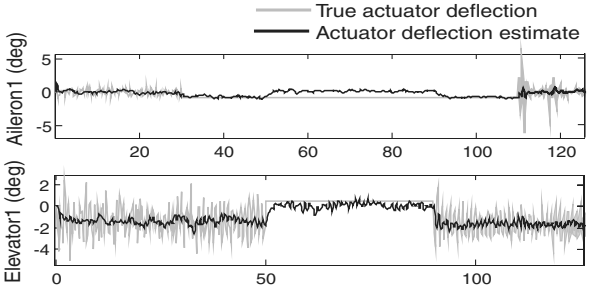

Fig. 19 Estimation of the deflection of the faulty control surfaces.

Figure 19 shows the estimation of the deflection of the faulty control surfaces. As expected, the estimation of the position of the failing aileron is valid as long as the second fault is not introduced (i.e., between the time intervals $30\text{ s} < t < 50\text{ s}$ and $90\text{ s} < t < 110\text{ s}$). The deflection of the failing elevator is properly estimated during the occurrence of the corresponding fault between $50\text{ s} < t < 90\text{ s}$.

XI. Possible Usage of the EMMAE-FDI Module for a Reconfigurable Flight Control System

The design of a reconfiguring control system in the event of a major failure or damage is a challenging task [16]. Many papers show how to build an adaptive controller using a multiple-model structure. An overview of these architectures and their robustness properties can be found in [17]. This section shows a new feature of the EMMAE FDI, which consists of using the estimation of the faulty

Table A1 Aircraft parameters

Name	Description	Value	Name	Description	Value
$C_{L_{a1}} = -C_{L_{a2}}$	Dimensionless aileron roll-torque effectiveness	3.395×10^{-2}	C_{Y1}	Dimensionless side-force derivative	-3.79×10^{-2}
$C_{L_{e1}} = -C_{L_{e2}}$	Dimensionless elevator roll-torque effectiveness	4.85×10^{-3}	C_{Z1}	Dimensionless lift derivative	1.29×10^{-2}
$C_{M_{a1}} = C_{M_{a2}}$	Dimensionless aileron pitch-torque effectiveness	2.725×10^{-1}	C_{Z2}	Dimensionless lift derivative	-3.25
$C_{M_{e1}} = C_{M_{e2}}$	Dimensionless elevator pitch-torque effectiveness	3.89×10^{-2}	$C_{M\alpha}$	Dimensionless pitch derivative for the angle of attack	-9.03×10^{-2}
$C_{L\tilde{p}}$	Dimensionless roll derivative for scaled pitch rate $\tilde{p} = bp/(2V_T)$	-1.92×10^{-1}	$C_{M\tilde{q}}$	Dimensionless pitch derivative for scaled pitch rate $\tilde{q} = \bar{c}q/(2V_T)$	-9.83
$C_{L\tilde{r}}$	Dimensionless roll derivative for scaled yaw rate $\tilde{r} = br/(2V_T)$	3.61×10^{-2}	$C_{N\tilde{r}}$	Dimensionless yaw derivative for scaled yaw rate $\tilde{r} = br/(2V_T)$	-2.14×10^{-1}
$C_{L\beta}$	Dimensionless roll derivative for the sideslip angle	-1.30×10^{-2}	$C_{N\beta}$	Dimensionless yaw derivative for the sideslip angle	8.67×10^{-2}
C_{X1}	Dimensionless drag derivative	-2.12×10^{-2}	C_{X3}	Dimensionless drag derivative	-1.55
C_{X2}	Dimensionless drag derivative	-2.66×10^{-2}	C_{X4}	Dimensionless drag derivative	-4.01×10^{-1}

control surface to reconfigure part of the control system depicted in Fig. 7, using control allocation techniques [18–20].

The controller generates a virtual control command $\boldsymbol{v} = [C_L \;\; C_M \;\; C_N]^T$ in terms of desired roll, pitch, and yaw torque. This virtual command v is passed to the control allocator, which is provided with each actuator's position limits and effectiveness to produce any torque component of the v signal. A constrained optimization problem is then solved online to optimally generate control signals for actuators.

The control allocator is responsible for distributing the desired moments among a set of actuators. Therefore, if the control allocator is provided with information that a faulty control surface is deflected at a certain angle, then the control allocator can generate an appropriate command to the unfailed actuators in an optimal manner (i.e., by taking advantage of the torque already generated by the faulty actuator). A method to indicate to the control allocator which actuator has failed and at what position is to have the supervision module merge the upper and lower deflection limits $\delta_{i,\max}$ and $\delta_{i,\min}$ of the failed actuator i and make these limits equal to the estimated deflection $\hat{\delta}_i$ of actuator i computed by the corresponding EKF. This method allows the faulty actuator to be frozen or floating at any position. Moreover, because $\hat{\delta}_i$ is actually the estimate for the deflection angle of a control surface, this method is still valid for a mechanical-link failure between an actuator (servo) and its corresponding control surface (flap). A complete description of this reconfigurable control allocator can be found in [20].

XII. Conclusions

The extended multiple-model adaptive-estimation fault detection and isolation (EMMAE-FDI) algorithm, combined with an active supervision module, offers fast and accurate fault detection and isolation. Moreover, the addition of the estimation of the faulty control surface deflection in the state vector makes the method applicable for actuator failures such as frozen or floating at an arbitrary position. Only one filter is needed to monitor the health of a single actuator. The filters used in the EMMAE FDI are extended Kalman filters that provide nonlinear state estimation at any flight operating condition. An active fault detection and isolation technique is developed, which generates appropriate artificial excitation of the aircraft when needed. An additional filtering stage for the fault-probability signals was designed to enhance the robustness of the diagnosis, even in the event of severe wind turbulence. The whole system was demonstrated using nonlinear simulations of a realistic flight scenario. The FDI system was shown to be capable of handling two simultaneous actuator failures. Finally, when a fault is clearly isolated, the faulty-actuator deflection estimate can be advantageously used to modify online the settings of a control allocator, making the whole system suitable for flight control reconfiguration without any change in the initial controller or any additional actuator position sensor.

Appendix A: Parameters of the Aircraft Model

The aerodynamic forces are defined in the wind frame w as follows:

$$X^w = \bar{q}S(C_{X1} + C_{X2}\alpha + C_{X3}\alpha^2 + C_{X4}\beta^2)$$

$$Y^w = \bar{q}SC_{Y1}\beta, \qquad Z^w = \bar{q}S(C_{Z1} + C_{Z2}\alpha)$$

aircraft mass $m = 28$ kg, aerodynamic mean chord $\bar{c} = 0.58$ m, wingspan $b = 3.1$ m, wing surface $S = 1.80$ m^2, and inertia matrix

$$\boldsymbol{I} = \begin{bmatrix} 2.56 & 0 & 0.5 \\ 0 & 10.9 & 0 \\ 0.5 & 0 & 11.3 \end{bmatrix} [\text{kg m}^2]$$

Acknowledgments

The authors thank the associate editor and the reviewers of this paper for their helpful comments.

References

[1] Magill, D. T., "Optimal Adaptive Estimation of Sampled Stochastic Processes," *IEEE Transactions on Automatic Control*, Vol. 10, No. 4, 1965, pp. 434–439. doi:10.1109/TAC.1965.1098191

[2] Maybeck, P. S., and Stevens, R. D., "Reconfigurable Flight Control Via Multiple Model Adaptive Control Methods," *IEEE Tranactions on Aerospace and Electronic Systems*, Vol. 27, No. 3, 1991, pp. 470–479. doi:10.1109/7.81428

[3] Eide, P., and Maybeck, P. S., "An MMAE Failure Detection System for the F-16," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 32, No. 3, 1996, pp. 1125–1136. doi:10.1109/7.532271

[4] Maybeck, P. S., "Multiple Model Adaptive Algorithms for Detecting and Compensating Sensor and Actuator Failures in Aircraft Flight Control Systems," *International Journal of Robust and Nonlinear Control*, Vol. 9, No. 14, 1999, pp. 1051–1070. doi:10.1002/(SICI)1099-1239(19991215)9:14<1051::AID-RNC452>3.0.CO;2-0

[5] Lingli, N., "Fault-Tolerant Control of Unmanned Underwater Vehicles," Ph.D. Dissertation, Virginia Polytechnic Inst. and State Univ., Blacksburg, VA, 2001.

[6] Theilliol, D., Sauter, D., and Ponsart, J. C., "A Multiple Model Based Approach for Fault Tolerant Control in Non Linear Systems," *Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS 2003)*, Pergamon, New York, for International Federation of Automatic Control, Oxford, June 2003, pp. 151-156..

[7] Rupp, D., Ducard, G., Shafai, E., and Geering, H. P., "Extended Multiple Model Adaptive Estimation for the Detection of Sensor and Actuator Faults," *Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference 2005*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, Dec. 2005, pp. 3079–3084.

[8] Ducard, G., and Geering, H. P., "A Reconfigurable Flight Control System based on the EMMAE Method," *Proceedings of the 2006 American Control Conference*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, June 2006, pp. 5499–5504.

[9] Tao, G., Chen, S., Tang, X., and Joshi, S. M., *Adaptive Control of Systems with Actuator Failures*, Springer-Verlag, London, 2004, pp. 1–14.

[10] Stevens, B., and Lewis, F., *Aircraft Control and Simulation*, 2nd ed., Wiley, New York, 2003, pp. 59–138.

[11] Zarchan, P., and Musoff, H., *Fundamentals of Kalman Filtering: A Practical Approach*, 2nd ed., Vol. 208, Progress in Astronautics and Aeronautics, AIAA, Reston, VA, 2005, pp. 257–259.

[12] Maybeck, P. S., *Stochastic Models, Estimation, and Control*, Vol. 1, Academic Press, New York, 1979; reprint, Navtech, Arlington, VA, 1994, pp. 1–16.

[13] Buffington, J., Chandler, P., and Pachter, M., "Integration of On-Line System Identification and Optimization-Based Control Allocation," AIAA Paper 1998-4487, 1998.

[14] Doman, D. B., and Ngo, A. D,, "Dynamic Inversion-Based Adaptive/Reconfigurable Control of the X-33 on Ascent," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 2, 2002, pp. 275–284.

[15] Azam, M., Pattipati, K., Allanach, J., Poll, S., and PattersonHine, A., "In-Flight Fault Detection and Isolation in Aircraft Flight Control Systems," IEEE Aerospace Conference, Inst. of Electrical and Electronics Engineers Paper 1429, 2005.

[16] Maciejowski, J. M., "Reconfiguring Control Systems by Optimization," European Control Conference (ECC97), http://www-control.eng.cam.ac.uk/jmm/papers/ecc97.pdf [retrieved 21 Sept. 2007].

[17] Fekri, S., Athans, M., and Pascoal, A., "Issues, Progress and New Results in Robust Adaptive Control," *International Journal of Adaptive Control and Signal Processing*, Vol. 20, No. 10, 2006, pp. 519–579.
doi:10.1002/acs.912

[18] Bodson, M., "Evaluation of Optimization Methods for Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 703–711.

[19] Härkegård, O., "Backstepping and Control Allocation with Applications to Flight Control," Ph.D. Thesis, Linköping Univ., Linköping, Sweden, 2003.

[20] Ducard, G., Geering, H. P., and Dumitrescu, E., "Efficient Control Allocation for Fault Tolerant Embedded Systems on Small Autonomous Aircrafts," *Proceedings of the 1st IEEE Symposium on Industrial Embedded Systems*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, Oct. 2006, pp. 1–10.