The Aeronautical Journal (2022), **126**, pp. 1568–1592
doi:10.1017/aer.2021.66

REGULAR PAPER

An integrated fault-tolerant control strategy for control surface failure in a fighter aircraft

İ. Gümüşboğa[1,*] and A. İftar[2]

[1]Department of Avionics, Eskişehir Technical University, Eskişehir, Turkey and [2]Department of Electrical and Electronics Engineering, Eskişehir Technical University, Eskişehir, Turkey
*Corresponding author. Email: ilkaygumusboga@eskisehir.edu.tr

Received: 8 December 2020; **Revised:** 28 June 2021; **Accepted:** 1 July 2021

Keywords: Fault-tolerant control; Flight control; Failure detection; Flight simulation; Extended Kalman filter; Asymmetric elevator failures

Abstract

Elevator failure may have fatal consequences for fighter aircraft that are unstable due to their high manoeuvrability requirements. Many studies have been conducted in the literature using active and passive fault-tolerant control structures. However, these studies mostly include sophisticated controllers with high computational load that cannot work in real systems. Considering the multi-functionality and broad operational prospects of fighter aircraft, computational load is very important in terms of applicability. In this study, an integrated fault-tolerant control strategy with low computational load is proposed without sacrificing the ability to cope with failures. This control strategy switches between predetermined controllers in the case of failure. One of these controllers is designed to operate in a non-failure condition. This controller is a basic controller that requires very little computational effort. The other controller operates when an asymmetric elevator failure occurs. This controller is a robust fault-tolerant controller that can fly the aircraft safely in case of elevator failure. The switching is decided by a failure detection system. The proposed integrated fault-tolerant control system is verified by non-linear F-16 flight simulations. These simulations show that the proposed method can cope with failures but requires less computational load because it uses a conventional controller in the case of no failure.

Nomenclature

a_{trim}	trim value of a vector (or variable) a
$\dot{a}$	derivative of a vector (or variable) a with respect to time
$\boldsymbol{A}^T$	transpose of a matrix $\mathbf{A}$
h	altitude (m)
L, M, N	rolling, pitching and yawing moments, respectively (N.m)
N_d, E_d	north and east geographic positions of the aircraft, respectively (m)
p, q, r	roll, pitch and yaw rates, respectively (rad/s)
q_r	pitch-rate reference signal (rad/s)
$\hat{\boldsymbol{r}}$	residual vector (rad)
$\hat{\boldsymbol{r}}_f$	filtered residual vector (rad)
$\hat{\boldsymbol{r}}_{\text{decision}}$	decision signal that contains failure decision information
t_f	time of failure (s)
T_d	discretisation period (s)
V	true velocity (m/s)
X, Y, Z	total X-, Y- and Z-axis forces, respectively (N)
α, β	angle-of-attack and sideslip angle, respectively (rad)
δ_e, δ_a, δ_r	elevator, aileron and rudder deflections, respectively (rad)
δ_{ec}, δ_{ac}, δ_{rc}	elevator, aileron and rudder deflection commands, respectively (rad)
δ_{er}, δ_{el}	right and left elevator surface deflections, respectively (rad)

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

δ_{er_f}	failed right elevator deflection command (rad)
δ_T	throttle setting [0–100]
τ	time constant (s)
ϕ, θ, ψ	Euler angles for roll, pitch and yaw, respectively (rad)
$\boldsymbol{0}_{m\times n}$	$m \times n$-dimensional zero matrix
$\boldsymbol{I}_n$	$n \times n$-dimensional identity matrix

1.0 Introduction

Control surface failure in aircraft can cause performance degradation, and even system instability leading to serious accidents. In the aviation industry, developments that will increase the level of safety and minimise the risk in case of critical failure have come to the fore in recent years. In this regard, developments in civil and military aviation generally follow two different approaches. In the civil aviation industry, *hardware redundancy* is the preferred approach [1, 2, 3]. In this approach, if the primary control surface fails, a redundant control surface automatically takes the control authority hierarchically. However, redundant control surfaces and actuators increase the weight of the aircraft, thus this approach is not preferred in military aviation, which requires high manoeuvrability capabilities. In military aircraft, methods that provide flight safety by using other existing control surfaces are generally used. This approach is known as *aerodynamic redundancy* [4, 5]. In this approach, if any one of the control surfaces fails, the flight control system uses other existing control surfaces to achieve stability requirements and even complete the mission. Therefore, Fault-Tolerant Control (FTC) approaches that use aerodynamic redundancy have achieved prominence in the literature recently [6, 7, 8].

The many FTC methods that have been developed can be categorised into two types: *passive* and *active* FTC systems [9, 10, 11, 12]. In passive FTC, controllers are designed to be robust against a class of presumed failures. A single robust controller works for all conditions in this control approach [13, 14, 15, 16]. On the other hand, in active FTC systems, a structure is used that changes controller parameters or switches between pre-designed multiple controllers, depending on the occurrence of particular failures [17, 18, 19, 20]. In addition, some hybrid methods have also been proposed in the literature [21, 22, 23]. For both civil and military aircraft, controlled dynamics are examined under two headings: *longitudinal* and *lateral/directional* [24, 25, 26, 27]. Control of longitudinal dynamics is more important for military combat planes, since these planes have unstable longitudinal dynamics [28, 29, 30]. For longitudinal dynamics, control of the pitch rate is the most important aim, since it is important in both target tracking and in the critical flight phases of approach and landing [31, 32, 33]. The classical approach to pitch-rate tracking is to use a simple proportional–integral (PI) controller, which works satisfactorily in case of no failures [31, 34, 35, 36]. However, it was shown by Gümüşboğa et al. [37] that this approach cannot cope with asymmetric elevator failures. Therefore, a robust FTC approach has been proposed by Gümüşboğa et al. [37] to cope with such failures. Such robust FTC systems can deal with failures very well but are more sophisticated than a classical controller mathematically. In this paper, after briefly presenting these two control structures, we compare them in terms of their computational requirements, revealing that the computational load is considerably higher for the robust FTC than that for the classical controller.

The sophisticated FTC increases the computational load greatly in just one control loop. The use of such controllers in all the control loops operating on a fighter plane would increase the computational load enormously. This motivates a new control approach that is both capable of handling failures and has a reasonable computational load that enables its use in real life. In the present paper, to solve this problem, an integrated FTC approach is proposed. This control system is represented in Fig. 1 (where q and q_r are the pitch rate and its reference value, respectively). This control system approach is based on the fundamental fact that, since control surface failure is a rare event, it does not make sense to use the robust FTC unless a failure occurs. In this structure, the aircraft is controlled by classical control systems when there is no malfunction (nominal operation). Thus, the aircraft can provide a high control loop ratio with computationally effective controllers to fulfil all mission expectations (e.g. in high-manoeuvre

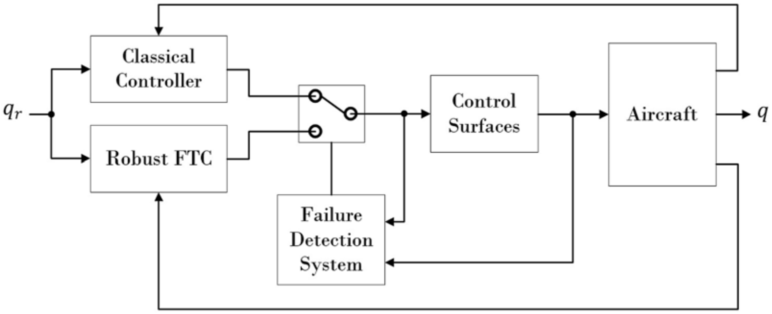

Figure 1. *Block diagram of the integrated FTC system.*

dogfights). When a failure occurs, the fault detection system detects it and activates the robust FTC via a switching mechanism. In this case, high mission expectations are abandoned and flight safety becomes the priority. Thus, using a lower control loop rate, the fighter aircraft can return home safely. An EKF-based failure detection system is proposed to detect the failure of the right elevator. The use of such a failure detection system instead of directly evaluating the measurements increases the robustness of the fault decision process. Therefore, if the failure occurs once, it is always considered faulty in the ongoing process. Thus, after the failure decision, the fault-tolerant controller operates continuously. This is a realistic assumption because, once a control surface on the aircraft is broken, the probability of self-recovery is very low.

This integrated FTC structure proposed herein combines the advantages of active and passive FTC systems. The proposed approach could be applied to all the control systems on an aircraft. In essence, this structure offers a reasonable logic between complex but robust and effective but fragile control systems. To demonstrate the performance of this control approach in a limited space, the robust controller in Fig. 1 is chosen as a robust pitch-rate control augmentation system presented previously [37]. Since there is no universal benchmark in this regard, the results of this proposed approach have been shown to be almost identical to the use of robust FTC alone. Thus, a plausible control approach is proposed that is equally resistant to failure and very effective in terms of computational load.

As a case study, we consider the F-16 aircraft, which is used in many air forces and has been considered widely in the literature [38, 39, 40]. In Section 2, we first briefly present a non-linear model for the F-16 dynamics as previously obtained by Gümüşboğa et al. [41] and modified by Gümüşboğa et al. [42] to accommodate the dynamics in the case of asymmetric elevator failure. This non-linear model is used both for simulations and to derive a linear design model, obtained previously by Gümüşboğa et al. [37] and also briefly presented in Section 2. Then, Section 3 briefly presents and compares the classical controller and the robust FTC designed by Gümüşboğa et al. [37]. The failure detection system to be used in the proposed integrated control strategy is presented in Section 4. The proposed integrated control structure is presented in Section 5, and its performance is verified by non-linear flight simulations. Finally, some concluding remarks are given in Section 6.

2.0 Aircraft and Control Surface Dynamics

In the first subsection of this section, we first briefly present the non-linear model for the dynamics of the F-16 obtained by Gümüşboğa et al. [41, 42]. The non-linear control surface actuation model and the control surface failures considered in this work are presented in the second subsection. Finally, the linear model used in the design of controllers is presented in the third subsection.

2.1 Non-linear aircraft model

The non-linear F-16 dynamics model obtained by Gümüşboğa et al. [41] consists of six sub-models: control surface actuation model, aerodynamic model, propulsion model, gravity model, atmosphere

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

model and the Equations of Motion (EoM). The control surface actuation model is detailed in the next subsection. The aerodynamic model produces the aerodynamic forces (X, Y, Z) and moments (L, M, N) acting on the aircraft. The original model, which is partly data based and partly equation based, was presented by Gümüşboğa et al. [41] and modified by Gümüşboğa et al. [42] to accommodate the dynamics in the case of asymmetric elevator failure. In the present study, we use this modified model (see also Gümüşboğa et al. [37]) for all the simulations. However, the linear design model (Subsection 2.3) is based on the original model presented by Gümüşboğa et al. [41]. The propulsion model is constructed using experimental data and produces the engine thrust value (T) based on the throttle setting (δ_T). The gravity model produces the gravitational acceleration value based on the latitude of the geodetic location on the Earth (which is assumed to be 30° throughout this work) and the altitude of the aircraft (h). The atmosphere model is data based and calculates the speed of sound and the dynamic pressure (to be used in the aerodynamic and propulsion models) based on the altitude and the velocity of the aircraft. Finally, the EoM consist of the kinematic and dynamic equations related to the six-degrees-of-freedom motion of the aircraft. This model consists of 12 non-linear, coupled, first-order differential equations [31, 43] that produce the 12 variables of motion: the true velocity (V), the angle-of-attack (AoA) (α), the sideslip angle (β), the roll (p), pitch (q) and yaw (r) rates, the roll (ϕ), pitch (θ) and yaw (ψ) angles, the north (N_d) and east (E_d) geographic positions and the altitude (h) of the aircraft, based on the aerodynamic forces (X, Y, Z) and moments (L, M, N) calculated by the aerodynamic model and the thrust value (T) calculated by the propulsion model. More detailed information about the non-linear dynamic model of the F-16 aircraft can be found in the works by Gümüşboğa et al. [41, 42], Huo [43], Stevens et al. [31] and Nguyen et al. [44].

2.2 Non-linear control surface actuation models

The aircraft's control surface actuators are modelled as linear first-order lag systems appended by some rate and deflection limits [31, 43, 44, 45]. The control surface deflection commands (δ_{e_c}, δ_{a_c}, δ_{r_c}), specified by the flight control system, enter this model as inputs. The actual angular deflection of the flight control surfaces (δ_e, δ_a, δ_r) are the outputs of this model. Here, the subscripts e, a and r respectively refer to the elevator, ailerons and rudder. The transfer function for the linear lag dynamics is

$$G_{\text{actuator}}(s) = \frac{1}{\tau s + 1} \tag{1}$$

where the time constant is taken to be $\tau = 49.5 \times 10^{-3}$s, for all three control surfaces [44]. However, the deflection and rate limits are different for each control surface. The control surface deflections are limited as follows: $|\delta_e| \leq 0.4363$, $|\delta_a| \leq 0.3752$ and $|\delta_r| \leq 0.5236$rad. Additionally, the control surface deflection rates are limited as follows: $\left|\dot{\delta}_e\right| \leq 1.0472$, $\left|\dot{\delta}_a\right| \leq 1.3963$ and $\left|\dot{\delta}_r\right| \leq 2.0944$rad/s [44, 43].

The above dynamics, however, are valid for a healthy control surface. When control surface failure occurs, these models must be modified. The failures we consider in the present study are freezing (lock-in-place) and floating [46]. In a freezing failure, the control surface becomes locked in its last position before the failure occurs. In a floating failure, the control link that drives the control surface breaks. This corresponds to the zero hinge moment state, and the control surface deflects in accordance with the air flow passing over it, which is determined by the AoA of the tail [36], which can be assumed to be the same as that of the aircraft [42].

Furthermore, we assume that such a failure occurs only in the right elevator (as explained in the work by Gümüşboğa et al. [37], the right and left elevator surfaces can move independently on the F-16 aircraft, giving the possibility of such an asymmetric failure; furthermore, as explained in the work by Gümüşboğa et al. [37], an asymmetric failure produces a more challenging scenario since, besides a loss in the pitch moment, serious roll moment disturbance is also produced). To consider such an asymmetric failure, the right and left elevator angles must first be differentiated (in the healthy aircraft, these two angles are assumed to deflect equally). Thus, hereafter, we let δ_{er} and δ_{el} denote the right and left elevator deflection, respectively. When a failure of the right elevator occurs, the left elevator angle,

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

δ_{el}, continues to move according to the healthy model. For the right elevator, however, instead of the actual elevator command, $\delta_{e_c}(t)$, a modified signal, $\delta_{er_f}(t)$ at time t is assumed to enter to the control surface actuation model described above, where

$$\delta_{er_f}(t) = \begin{cases} \delta_{e_c}(t_f), & \forall t \geq t_f \text{ :freezing failure} \\ -0.5\alpha(t), & \forall t \geq t_f \text{ :floating failure} \end{cases} \tag{2}$$

where t_f is the time of failure and $\alpha(t)$ is the AoA at time t (see Gümüşboğa et al. [42, 37] for details).

2.3 Linear design model

In a healthy F-16 aircraft, the longitudinal motion and the lateral/directional motion are decoupled [31, 34]. To utilise this decoupling, in the linear model, the states are ordered as first the longitudinal states (V, α, θ, q) and then the lateral/directional states (β, ϕ, p, r). The other states (N_d, E_d, ψ, h) are not used in the linear model, since they have either no or negligible effect on the dynamic behaviour of the aircraft. The N_d, E_d and ψ state derivatives are functions of other states, but these states themselves are not coupled back into the state equations (and thus have no effect on the other states). Also, the altitude state, h, has negligible coupling to other states [31]. The linearisation is done around an operating point, that is, a flight condition. The calculated equilibrium values of the state and input variables at a given flight condition are called trim values [41]. The state vector, $\mathbf{x}$, and the input vector, $\mathbf{u}$, used in the linear model are defined as

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} \Delta V & \Delta\alpha & \Delta\theta & \Delta q & \Delta\beta & \Delta\phi & \Delta p & \Delta r \end{bmatrix}^T \\ \mathbf{u} &= \begin{bmatrix} \Delta\delta_e & \Delta\delta_a & \Delta\delta_r \end{bmatrix}^T \end{aligned} \tag{3}$$

where Δ indicates the deviation of the actual value from the trim value. Details of the linearisation procedure can be found in work by Gümüşboğa et al. [37], and details of the calculation of the trim values can be found in work by Gümüşboğa et al. [41]. In the present study, as the flight condition, we consider steady wings-level flight at $V = 100\text{m/s}$ and $h = 1,000\text{m}$. The trim values of the states and the inputs for this flight condition are as follows (in SI units): $\mathbf{x}_{\text{trim}} = [100 \;\; 0.123 \;\; 0.123 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]^T$ and $\mathbf{u}_{\text{trim}} = [-0.024 \;\; 0 \;\; 0]^T$.

The linearised model for the aircraft dynamics (apart from the control surface actuators) is represented as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \tag{4}$$

where $\mathbf{A}$ and $\mathbf{B}$ are the system dynamics and the input matrices, respectively. For the above-mentioned flight condition, these matrices are obtained by Gümüşboğa et al. [37] as (in SI units)

$$\mathbf{A} = \begin{bmatrix} -0.022 & -1.395 & -9.828 & -0.672 & 0 & 0 & 0 & 0 \\ -0.002 & -0.582 & 0 & 0.908 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 3\times10^{-7} & 0.324 & 0 & -0.708 & 0 & 0 & 0 & 0 \\ 3\times10^{-9} & 0 & 0 & 0 & 0.182 & 0.097 & 0.143 & -0.996 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.145 \\ 5\times10^{-7} & -4\times10^{-6} & 0 & 0 & -19.291 & 0 & -2.047 & 0.716 \\ 3\times10^{-7} & -8\times10^{-6} & 0 & 0 & 5.364 & 0 & -0.041 & -0.337 \end{bmatrix} \tag{5}$$

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

and

$$\mathbf{B} = \begin{bmatrix} -1.139 & 0 & 0 \\ -0.072 & 0 & 0 \\ 0 & 0 & 0 \\ -4.301 & 0 & 0 \\ 0 & -0.010 & -0.028 \\ 0 & 0 & 0 \\ 0 & -15.980 & 2.470 \\ 0 & -0.667 & -1.304 \end{bmatrix} \tag{6}$$

The decoupling mentioned above is evident from the above matrices, where the elevator deflection affects the longitudinal states while the aileron and rudder deflections affect the lateral/directional states.

The design models for each of the control surface actuators, on the other hand, are obtained as the negative of the linear lag dynamics, given by equation (1) [31]. Thus, for each control surface, the transfer function from its command to its (negative) deflection is assumed to be

$$P(s) = \frac{-20.2}{s + 20.2}. \tag{7}$$

The negative sign is due to the fact that positive control surface deflections cause negative related aerodynamic moments.

3.0 Classical and Robust Fault-Tolerant Controllers

In this section, the classical controller and the robust FTC to be used in the proposed integrated FTC, shown in Fig. 1, are briefly presented and then compared.

3.1 The classical controller

The classical controller consists of two components: a pitch-rate Control Augmentation System (CAS) and a roll-axis Stability Augmentation System (SAS), both of which have been studied extensively in the literature and are currently used in most aircraft [34, 35, 31]. The pitch-rate CAS is used for pitch-rate tracking. The roll-axis SAS, on the other hand, is used to reject any roll-axis disturbances. This is necessary in the present study to enable a fair comparison between the classical controller and the robust FTC, since asymmetric elevator failure produces serious roll-axis disturbance. The overall classical controller is implemented as shown in Fig. 2, where the pitch-rate CAS and the roll-axis SAS components are clearly indicated. Since the design of the control system is based on the linear model, the deviations of the measured signals around their trim values are also shown in this figure. Furthermore, $\Delta q_r = q_r - q_{\text{trim}}$, where q_r denotes the pitch-rate reference signal. Next, we briefly present the design procedures for each component of the classical controller, which were originally given in the work by Gümüşboğa et al. [37] in more detail.

3.1.1 The pitch-rate CAS

The pitch-rate CAS uses the pitch-rate (q) and AoA (α) measurements and produces the elevator command (δ_{e_c}), as shown in Fig. 2. The design of the pitch-rate CAS is based on the linearised longitudinal

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

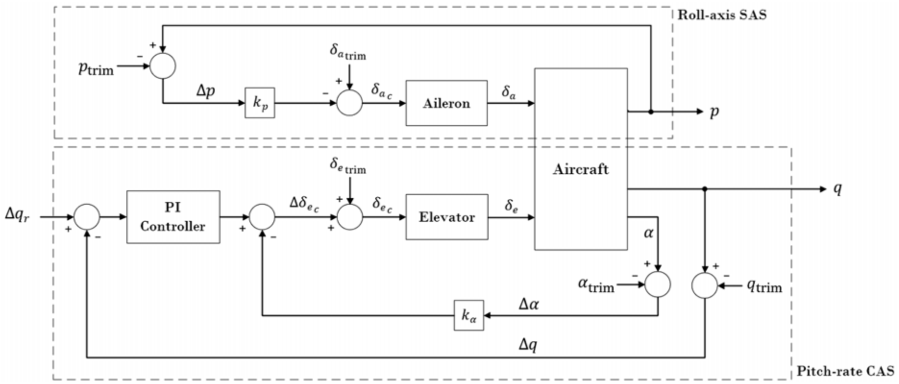

Figure 2. *The classical controller.*

sub-model, which is obtained from the linearised model (4) by taking the first four states and the first input. Furthermore, (7) is used as the elevator model. The modes of the open-loop linearised longitudinal sub-model are

$$\underbrace{-0.1256 \pm 0.1507j,}_{\text{short period modes}} \underbrace{-1.1950, 0.1351}_{\text{phugoid modes}}$$

In addition, there is also an open-loop mode at −20.2 due to the actuator dynamics. This clearly shows that the aircraft is unstable in the longitudinal direction, since one of the so-called phugoid modes lies in the right half of the complex plane. To stabilise the system, $\Delta\alpha$ feedback is used in the inner loop as shown in Fig. 2. The k_α gain is determined by the root-locus method as $k_\alpha = 0.08$. When the inner loop is closed by this feedback, the eigenvalues related to the phugoid mode (before the outer loop is closed) are obtained as $\{-0.0069 \pm 0.0290j\}$, with only negligible displacements in the other modes. Thus, this feedback stabilises the longitudinal dynamics. After the inner loop is closed, the PI controller, with transfer function

$$C_{PI}(s) = k_{\text{prop}} + \frac{k_i}{s} \tag{8}$$

is designed, to achieve the desired tracking of the pitch-rate reference signal. The coefficients k_{prop} and k_i are determined using the PID Tuner tool in Simulink software as follows: $k_{\text{prop}} = 1$ and $k_i = 0.75$.

3.1.2 The roll-axis SAS

The roll-axis SAS uses the roll-rate (p) measurement and produces the aileron command (δ_{a_c}), as shown in Fig. 2. The design of the roll-axis SAS is based on the linearised lateral/directional sub-model, which is obtained from the linearised model (4) by taking the last four states and the second input (the rudder command is not used since the only aim is to stabilise the roll dynamics while no directional tracking is performed). Furthermore, (7) is used as the aileron model. The modes of the open-loop linearised lateral/directional sub-model are

$$\underbrace{-0.2013 \pm 2.7585j,}_{\text{Dutch-roll modes}} \underbrace{-1.7926,}_{\text{roll mode}} \underbrace{-0.0067}_{\text{spiral mode}}$$

The task of the roll-axis SAS is to increase the stability of the roll mode for a better roll motion response. For this, Δp feedback is used, as shown in Fig. 2. The k_p gain is determined by the root-locus method as $k_p = 0.1$, which places the roll mode at $s = -3.68$, which is about twice as fast as the open-loop value.

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

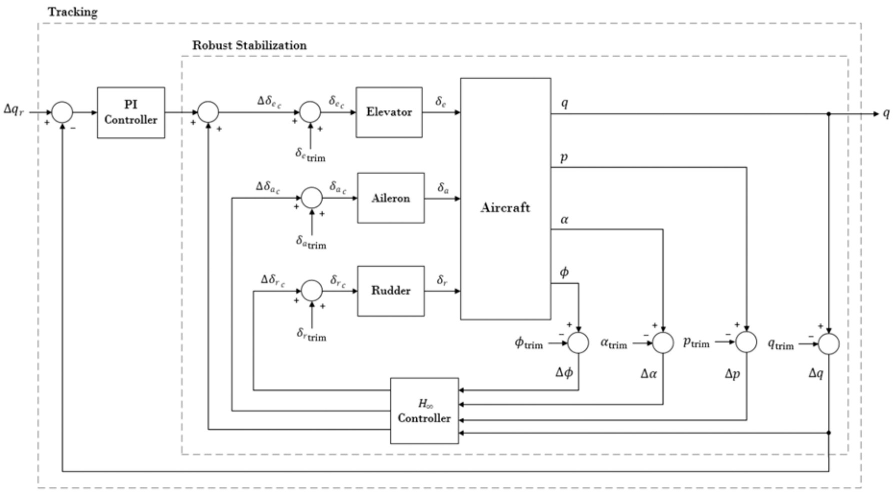

Figure 3. *The robust FTC.*

3.2 The robust FTC

In the work by Gümüşboğa et al. [37], it was shown that the classical controller achieves stability and desired tracking of pitch-rate reference signals in the case of a healthy aircraft. However, it cannot cope with asymmetric elevator failure. Therefore, a robust FTC was designed in the work by Gümüşboğa et al. [37] to achieve stability and pitch-rate tracking despite such failures. This controller has a two-loop structure, as shown in Fig. 3.

The inner loop aims for robust stabilisation of the aircraft despite asymmetric elevator failure. The outer loop aims for tracking of pitch-rate reference signals. The inner loop is designed using the H_∞ control approach [47]. This controller is designed based on an extended model described as

$$\begin{aligned}\dot{\mathbf{x}}_{ext} &= \mathbf{A}_{ext}\mathbf{x}_{ext} + \mathbf{B}_1\mathbf{w} + \mathbf{B}_2\mathbf{u}_c \\ \mathbf{z} &= \mathbf{C}_1\mathbf{x}_{ext} + \mathbf{D}_{11}\mathbf{w} + \mathbf{D}_{12}\mathbf{u}_c \\ \mathbf{y} &= \mathbf{C}_2\mathbf{x}_{ext} + \mathbf{D}_{21}\mathbf{w} + \mathbf{D}_{22}\mathbf{u}_c\end{aligned} \tag{9}$$

where $\mathbf{x}_{ext} = [\Delta V\ \Delta\alpha\ \Delta\theta\ \Delta q\ \Delta\beta\ \Delta\phi\ \Delta p\ \Delta r\ \Delta\delta_e\ \Delta\delta_a\ \Delta\delta_r]^T$ and $\mathbf{u}_c$ consists of control commands, $\mathbf{u}_c = [\Delta\delta_{e_c}\ \Delta\delta_{a_c}\ \Delta\delta_{r_c}]^T$. In this control problem, the measurement output vector is $\mathbf{y} = [\Delta p\ \ \Delta q\ \ \Delta\alpha\ \ \Delta\phi]^T$. It is assumed that these variables can be measured directly. The H_∞ control method minimises the effect of the exogenous disturbance input, $\mathbf{w}$, on the performance output, $\mathbf{z}$. Thus, these vectors should be selected in accordance with the asymmetric elevator failure case. The works by Gümüşboğa et al. [42, 37] show that asymmetric elevator failure leads to disturbances in the pitch and roll moments. The other forces and moments are indirectly induced with coupling effects as the symmetry of the aircraft is impaired due to asymmetric elevator failure. Thus, the exogenous disturbance input $\mathbf{w}$ is chosen as the moment and force disturbances, $\mathbf{w} = [\delta L\ \ \delta M\ \ \delta N\ \ \delta X\ \ \delta Y\ \ \delta Z]^T$. Since the aim is to minimise the effect of these disturbances on the roll and pitch rates, as well as on the roll angle, the deviations of these variables from their trim values must be chosen as part of the $\mathbf{z}$ vector. Furthermore, because the control surface deflections and their rates must satisfy certain limits, the deviations of these variables from their trim values must also be chosen as part of the $\mathbf{z}$ vector. Thus, the performance output vector is chosen as $\mathbf{z} = \mathbf{Q}[\Delta p\ \Delta q\ \Delta\phi\ \Delta\delta_e\ \ \Delta\delta_a\ \Delta\delta_r\ \Delta\dot{\delta}_e\ \Delta\dot{\delta}_a\ \Delta\dot{\delta}_r]^T$, where $\mathbf{Q}$ is a constant diagonal weight matrix with positive diagonal elements, which are all taken as unity in

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

the present application. The matrices for this extended F-16 model are then obtained as follows (in SI units):

$$\mathbf{A}_{\text{ext}} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0}_{3\times8} & -20.2\mathbf{I}_3 \end{bmatrix}, \mathbf{B}_1 = \begin{bmatrix} \widetilde{\mathbf{B}}_1 \\ \mathbf{0}_{3\times6} \end{bmatrix}, \mathbf{B}_2 = \begin{bmatrix} \mathbf{0}_{8\times3} \\ -20.2\mathbf{I}_3 \end{bmatrix}, \mathbf{C}_1 = \mathbf{Q}\begin{bmatrix} \widetilde{\mathbf{C}}_1 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times8} & \mathbf{I}_3 \\ \mathbf{0}_{3\times8} & -20.2\mathbf{I}_3 \end{bmatrix}$$

$$\mathbf{D}_{11} = \mathbf{0}_{9\times6}, \mathbf{D}_{12} = \mathbf{Q}\begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{I}_3 \\ -20.2\mathbf{I}_3 \end{bmatrix}, \mathbf{C}_2 = \begin{bmatrix} \widetilde{\mathbf{C}}_2 & \mathbf{0}_{4\times3} \end{bmatrix}, \mathbf{D}_{21} = \mathbf{0}_{4\times6}, D_{22} = \mathbf{0}_{4\times3} \tag{10}$$

where

$$\widetilde{\mathbf{B}}_1 = \begin{bmatrix} 0 & 0 & 0 & 1\times10^{-4} & 0 & 1\times10^{-5} \\ 0 & 0 & 0 & -1\times10^{-6} & 0 & 8\times10^{-6} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1\times10^{-5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1\times10^{-6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 8\times10^{-5} & 0 & 1\times10^{-6} & 0 & 0 & 0 \\ 1\times10^{-6} & 0 & 1\times10^{-5} & 0 & 0 & 0 \end{bmatrix},$$

$$\widetilde{\mathbf{C}}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \text{ and } \widetilde{\mathbf{C}}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Here, **A** and **B** are given in (5) and (6), respectively. Additionally, **0** and **I** denotes the zero and identity matrix with the indicated dimension, respectively.

The H_∞ controller is then designed in the MATLAB environment using the H_∞ control design toolbox. The design uses the extended F-16 model given above. The state-space realisation of the robust controller designed is as follows (in SI units):

$$\begin{aligned} \dot{\hat{\mathbf{x}}} &= \hat{\mathbf{A}}\hat{\mathbf{x}} + \hat{\mathbf{B}}\mathbf{y} \\ \mathbf{u}_c &= \hat{\mathbf{C}}\hat{\mathbf{x}} + \hat{\mathbf{D}}\mathbf{y}, \end{aligned} \tag{11}$$

where

$$\hat{\mathbf{A}} =$$

$$\begin{bmatrix} -0.022 & 52.987 & -9.828 & 6.984 & 0 & 0.003 & -0.005 & 0 & -1.140 & 0 & 0 \\ -0.002 & -1.092 & 0 & 0.736 & 0 & 6\times10^{-6} & 1\times10^{-5} & 0 & -0.072 & 0 & 0 \\ 0 & -0.834 & 0 & 0.690 & 0 & 2\times10^{-5} & 4\times10^{-5} & 0 & 0 & 0 & 0 \\ 2\times10^{-7} & 0.095 & 0 & -0.939 & 0 & 1\times10^{-6} & 2\times10^{-6} & 0 & -4.301 & 0 & 0 \\ 3\times10^{-9} & -3\times10^{-6} & 0 & -5\times10^{-7} & 0.182 & 0.081 & 0.254 & -0.996 & 0 & -0.010 & -0.028 \\ 0 & 9\times10^{-6} & 0 & 1\times10^{-6} & 0 & -0.872 & 0.514 & 0.145 & 0 & 0 & 0 \\ 5\times10^{-7} & 1\times10^{-5} & 0 & 2\times10^{-6} & -19.291 & -0.486 & -5.292 & 0.716 & 0 & -15.980 & 2.469 \\ 3\times10^{-7} & -3\times10^{-6} & 0 & 9\times10^{-7} & 5.364 & 0.085 & -0.024 & -0.337 & 0 & -0.667 & -1.304 \\ -1\times10^{-4} & 0.1462 & 4\times10^{-4} & 0.710 & 4\times10^{-5} & -2\times10^{-5} & -8\times10^{-6} & -4\times10^{-6} & -2.481 & 3\times10^{-5} & -4\times10^{-6} \\ 7\times10^{-7} & -8\times10^{-6} & -4\times10^{-8} & -1\times10^{-5} & -33.445 & 12.954 & 6.508 & 5.509 & 3\times10^{-5} & -26.499 & 4.394 \\ -3\times10^{-9} & -2\times10^{-6} & -7\times10^{-7} & 1\times10^{-6} & 8.259 & -2.265 & -1.168 & -0.887 & -4\times10^{-6} & 4.394 & -1.3527 \end{bmatrix}$$

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

$$\hat{\mathbf{B}} = \begin{bmatrix} 1.534 & -2\times10^{3} & -1\times10^{4} & 0.896 \\ -0.004 & 57.778 & 129.159 & -0.002 \\ -0.013 & 104.164 & 211.074 & -0.007 \\ -8\times10^{-4} & 77.677 & 57.778 & -3\times10^{-4} \\ -37.403 & 1\times10^{-4} & 7\times10^{-4} & 5.387 \\ 162.994 & -3\times10^{-4} & -0.002 & 292.588 \\ 1\times10^{3} & -8\times10^{-4} & -0.004 & 162.994 \\ -5.838 & -3\times10^{-4} & -0.001 & 28.411 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\hat{\mathbf{C}} = \begin{bmatrix} 3\times10^{-8} & -3\times10^{-5} & -8\times10^{-8} & -1\times10^{-4} & -8\times10^{-9} & 3\times10^{-9} & 2\times10^{-9} & 0 & -0.004 & -6\times10^{-9} & 0 \\ 0 & 2\times10^{-9} & 0 & 3\times10^{-9} & 0.007 & -0.003 & -0.002 & -0.001 & -6\times10^{-9} & 0.001 & -9\times10^{-4} \\ 0 & 0 & 0 & 0 & -0.002 & 4\times10^{-4} & 2\times10^{-4} & 2\times10^{-4} & 0 & -8\times10^{-4} & -0.004 \end{bmatrix}$$

$$\hat{\mathbf{D}} = \mathbf{0}_{3\times4}$$

In this calculation, values smaller than 1×10^{-9} (in absolute value) are directly represented as zero because they do not contribute to the accuracy of the calculation.

After the inner loop is closed by this controller, the PI controller in the outer loop, which has the transfer function (8), is designed to achieve the desired tracking of the pitch-rate reference signal. The coefficients k_{prop} and k_i are determined using the PID Tuner tool in Simulink software as follows: $k_{\text{prop}} = 1.5$ and $k_i = 1.1$. More details on the design procedure can be found in the work by Gümüşboğa et al. [37].

3.3 Comparison of the classical controller and the robust FTC

The performance of the classical controller and the robust FTC was examined in detail by Gümüşboğa et al. [37], both for the healthy aircraft and in the case of asymmetrical elevator failure. Furthermore, in Section 5 of the present paper, the performance of these controllers is also examined together with the proposed integrated FTC.

The healthy condition simulations presented in Section 5.1 (Fig. 8) show that both controllers work very well with very similar performance when there is no elevator failure. However, the failure condition simulations in Sections 5.2 and 5.3 (Figs 11 and 14) show that the classical controller cannot cope with asymmetric elevator failure, as the flight safety is impaired and the aircraft crashes into the ground during the simulation period of 20s. On the other hand, the robust FTC continues to fly the aircraft in a safe manner and the pitch-rate tracking is reasonably good despite the occurrence of either of the two failures (Figs 11 and 14).

This analysis shows that, regarding the performance in the case of no failure, either of the two controllers could be used. Therefore, since the robust FTC is the only choice in the case of asymmetric elevator failure, one might be tempted to use it in both the no-failure and failure conditions. However, one must also consider the computational complexity.

The robust FTC represented in Fig. 3 includes an H_∞ controller in the inner loop and one PI controller in the outer loop. The H_∞ controller is an 11-dimensional, 4-input, 3-output linear time-invariant system. The discrete-time implementation of the overall robust FTC requires 200 floating-point multiplications and 188 floating-point additions in each cycle. The classical controller, on the other hand, includes one PI controller and two static feedback gains (k_α and k_p). Therefore, the discrete-time implementation of this controller requires four floating-point multiplications and four floating-point additions in each cycle. Therefore, the complexity of the robust FTC is about 50 times that of the classical controller.

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

Table 1. *Controllers' calculation times*

Controller	Mean calculation time
Classical	0.2684μs
Robust FTC	3.1823μs

This means that, in a dedicated computer, the calculation times required for the robust FTC may be as much as 50 times longer than those required for the classical controller. This ratio may, of course, be smaller in a non-dedicated computer. To obtain an initial idea on what this ratio might be in a non-dedicated computer, we timed the calculations done to compute the controller commands during the non-linear simulations. These simulations were carried out using MATLAB R2018b software on a basic laptop with an Intel i7 8550U 2GHz processor and 8GB RAM. The calculation times were measured by using the 'tic-toc' method in MATLAB, and only the time it takes for the control system to generate controller outputs within one iteration was measured. These calculation times are, of course, influenced by factors such as instantaneous processor performance changes, operating system background processes and so on. To minimise such effects, the simulations (done with an integration step size of 10ms) were run for as long as 2,000s, that is, 200,000 cycles, and the averages of the calculation times over 200,000 cycles obtained for each control structure. The obtained mean calculation times are presented in Table 1. These values indicate that the calculation time required for the robust FTC is as much as 12 times longer than that required for the classical controller. Considering that the flight computer runs many different tasks simultaneously, it may be important to economise on the calculation times of each task. Therefore, unless there is a need, such as the occurrence of a failure, it would be better to use the classical control structure. This motivates the use of the integrated control structure proposed herein. To use this structure, however, one also needs a failure detection system. The design of this system is presented in the next section.

4.0 Failure Detection System

In this section, the failure detection system to be used as part of the proposed integrated FTC, shown in Fig. 1, is presented. In preliminary work, a model-based failure detection system was designed by Gümüşboğa et al. [48]. Although that design works successfully for control surface failure detection, it creates delays in the detection when aircraft pitch-rate manoeuvres are involved. In the present work, this problem is solved by modifying this structure, and this modified failure detection system is then used to detect right elevator failures. Before presenting the specific design, however, we provide an introduction to the failure detection strategy employed here.

4.1 Failure detection: an overview

A failure detection system design includes a robust residual generation process followed by a decision-making process [49, 50]. The failure detection system has the basic structure shown in Fig. 4 for a particular set of hypothesised failures. The residual generation process uses the information obtained from the sensors and the dynamic model of the system. While the information about the actual condition of the system comes from the sensors, the dynamic model of the system contains information about how the healthy system should behave. The process of obtaining the deviation between these information robustly is called residual generation. Intuitively, the residuals refer to the deviation between the sensor data and the expected behaviour of the healthy system. When there is no failure in the system, the residuals should be unbiased, indicating agreement between the observed and the expected, normal behaviour of the system. On the contrary, a failure signature often takes the form of residual biases that are characteristic of the failure.

In the decision-making process, these residuals are examined for the presence of failure signatures. Decision functions or statistics are first calculated using the residuals. This includes processes that

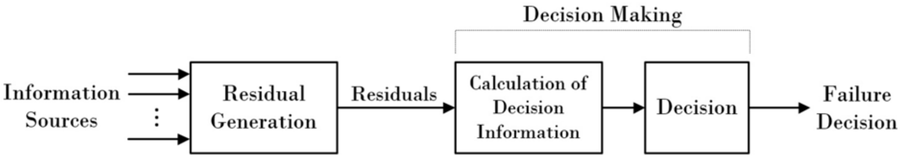

Figure 4. *General structure of a failure detection system.*

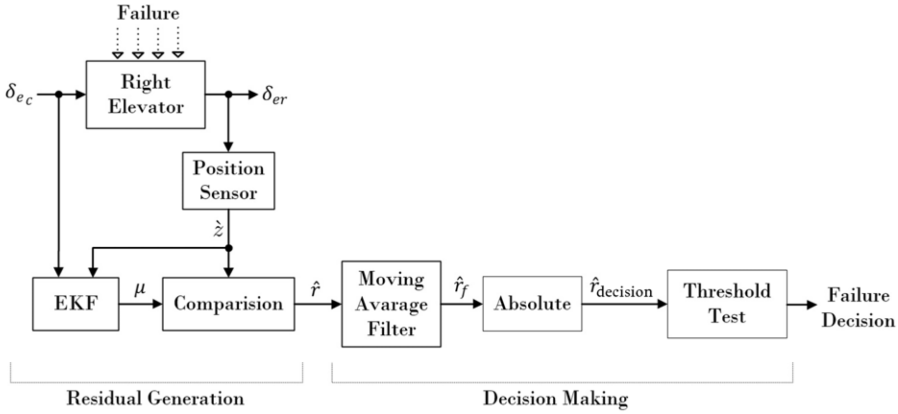

Figure 5. *Failure detection system.*

increase the accuracy of information such as fusing residues, filtering and so on. After that, a decision rule is applied to the decision functions to determine whether any failures have occurred in the system. The decision process generally use a simple threshold test on the instantaneous values of the residuals, but there are also more sophisticated algorithms based on statistical decision theory [51].

4.2 Failure detection system design for right elevator failure

In this subsection, a failure detection system, based on the general framework given above, is presented for the right elevator failure. When a freezing or floating failure occurs on the right elevator surface, the angle of deflection of the right elevator clearly deviates from the elevator command produced by the controller. This deviation indicates a faulty condition. The failure detection system is shown in Fig. 5. This system contains a residual generation process followed by a decision-making process.

4.2.1 Residual generation

The residual generation process is shown in Fig. 5. This process uses the deflection angle of the right elevator, δ_{er}, which is measured by a rotary absolute encoder, and the elevator control input, δ_{e_c}, generated by the control system. To generate a residual, the sensor data obtained from the actual right elevator is compared with the expected deflection angle according to the healthy elevator dynamics model. For this, an estimation filter is used for greater robustness against measurement noise and modelling uncertainties. The Extended Kalman Filter (EKF) is used because the right elevator dynamics are non-linear.

Since the failure detection system is to be implemented in discrete time, here it is directly designed in discrete time, where the sampling period is chosen as the cycle period of the non-linear F-16 simulation, which is $T_d = 0.01$s. The measurement at the discrete-time instant k of the deflection of the right elevator, δ_{er}, which is measured by a rotary absolute encoder, is denoted by $\grave{\mathbf{z}}(k)$. The residual signal is obtained as

$$\hat{\mathbf{r}}(k) = \grave{\mathbf{z}}(k) - \boldsymbol{\mu}(k)\,, \tag{12}$$

where $\boldsymbol{\mu}(k)$ is the expected value of the right elevator deflection, calculated by the EKF.

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

The EKF assumes a process model of the form [52, 53, 54]

$$\begin{aligned}\grave{\mathbf{x}}(k+1) &= g(\grave{\mathbf{x}}(k), \grave{\mathbf{u}}(k)) + \boldsymbol{\varepsilon}(k)\\ \grave{\mathbf{z}}(k) &= h(\grave{\mathbf{x}}(k)) + \boldsymbol{\nu}(k),\end{aligned} \tag{13}$$

where $\grave{\mathbf{x}}(k)$, $\grave{\mathbf{u}}(k)$ and $\grave{\mathbf{z}}(k)$ are, respectively, the state, the control input and the measurement, at time k. The functions g and h are the process and measurement model, respectively, and will be derived below. The discrete random zero-mean process noise $\boldsymbol{\varepsilon}(k)$ is used to describe uncertainties in the process model, while $\boldsymbol{\nu}(k)$ is used for the representation of the measurement noise. Both stochastic processes, $\boldsymbol{\varepsilon}(k)$ and $\boldsymbol{\nu}(k)$, are assumed to be uncorrelated in time and to have a Gaussian distribution with zero mean at each k. The variance of $\boldsymbol{\varepsilon}(k)$ is assumed to be $\mathbf{Q}(k)$, while the variance of $\boldsymbol{\nu}(k)$ is assumed to be $\mathbf{R}(k)$.

The EKF considers the non-linear elevator model as the process model, g. The non-linear elevator model consists of the actuator's linear lag dynamics, given by (1), followed by the rate limit and the deflection limit, which are given in Section 2.2. A state-space model in continuous time for the linear lag dynamics can be obtained as

$$\begin{aligned}\dot{\mathbf{x}}_L(t) &= -20.2\mathbf{x}_L(t) + 20.2\breve{\mathbf{u}}(t)\\ \boldsymbol{\delta}_L(t) &= \mathbf{x}_L(t),\end{aligned} \tag{14}$$

where $\mathbf{x}_L(t)$ is the state variable, $\boldsymbol{\delta}_L(t)$ is the output of the linear lag dynamics and $\breve{\mathbf{u}}(t) = \delta_{e_c}(t)$ is the elevator control input (all in radians) at time t (in seconds). Since the EKF is realised in discrete time, this model is discretised, where the discretisation period is $T_d = 0.01$s. The zero-order hold method is used for discretisation. The discrete-time model is obtained as

$$\begin{aligned}\mathbf{x}_l(k+1) &= a_d\mathbf{x}_l(k) + b_d\grave{\mathbf{u}}(k)\\ \boldsymbol{\delta}_l(k) &= \mathbf{x}_l(k),\end{aligned} \tag{15}$$

where

$$\begin{aligned}a_d &= e^{-20.2T_d} = 0.8171\\ b_d &= \int_0^{T_d} e^{-20.2\tau}20.2d\tau = -(e^{-20.2T_d} - 1) = 1 - e^{-20.2T_d} = 0.1829\end{aligned} \tag{16}$$

Eventually, the overall model for the right elevator surface dynamics is obtained by incorporating the rate and deflection limits of the right elevator, as follows:

$$\begin{aligned}\grave{\mathbf{x}}(k+1) &= g(\grave{\mathbf{x}}(k), \grave{\mathbf{u}}(k))\\ \boldsymbol{\delta}(k) &= \grave{\mathbf{x}}(k),\end{aligned} \tag{17}$$

where

$$g(\grave{\mathbf{x}}, \grave{\mathbf{u}}) = sat(sat(a_d\grave{\mathbf{x}} + b_d\grave{\mathbf{u}}, \grave{\mathbf{x}} - \bar{\rho}_d, \grave{\mathbf{x}} + \bar{\rho}_d), -\bar{\delta}, \bar{\delta}) \tag{18}$$

Here, $sat(\cdot)$ represents the saturation function and is defined as

$$sat(a, b, c) = \begin{cases} a, & b < a < c\\ b, & a \leq b\\ c, & a \geq c\end{cases} \tag{19}$$

Furthermore, $\bar{\delta} = 0.4363$rad is the deflection limit in absolute value, and

$$\bar{\rho}_d = \bar{\rho}\, T_d = 0.0105\text{rad}, \tag{20}$$

where $\bar{\rho} = 1.0472$rad/s is the rate limit in absolute value.

The right elevator deflection angle measurement is assumed to be obtained by a rotary absolute encoder, which produces a discrete output with a certain resolution as it counts the incremental number of steps for the elevator movement. The resolution of the encoder is assumed to be high enough that the

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

discretisation effect can be represented as a noise, which is included in $\boldsymbol{\nu}(k)$, on the output (in the simulations presented in the next section, we assume that this resolution is 0.001rad). Thus, the measurement model, h, is simply expressed as

$$h(\grave{\mathbf{x}}(k)) = \grave{\mathbf{x}}(k) \tag{21}$$

The EKF equations consists of two consecutive sets as the measurement-update phase and the time-update phase [53, 54]. The measurement-update phase corrects the state information with the measurement data from the sensor. The time-update phase progresses the state with the information obtained from the healthy aircraft and elevator model.

(i) Measurement-update phase:

$$\mathbf{K}(k) = \bar{\boldsymbol{\Sigma}}(k)\mathbf{H}(k)^T[\mathbf{H}(k)\bar{\boldsymbol{\Sigma}}(k)\mathbf{H}(k)^T + \mathbf{R}(k)]^{-1} \tag{22}$$

$$\boldsymbol{\mu}(k) = \bar{\boldsymbol{\mu}}(k) + \ \mathbf{K}(k)[\grave{\mathbf{z}}(k) - h(\bar{\boldsymbol{\mu}}(k))] \tag{23}$$

$$\boldsymbol{\Sigma}(k) = [\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)]\bar{\boldsymbol{\Sigma}}(k) \tag{24}$$

(ii) Time-update phase:

$$\bar{\boldsymbol{\mu}}(k+1) = g(\boldsymbol{\mu}(k), \grave{\mathbf{u}}(k)) \tag{25}$$

$$\bar{\boldsymbol{\Sigma}}(k+1) = \mathbf{G}(k)\boldsymbol{\Sigma}(k)\mathbf{G}(k)^T + \mathbf{Q}(k) \tag{26}$$

where $\bar{\boldsymbol{\mu}}(k)$ and $\boldsymbol{\mu}(k)$ denote, respectively, the uncorrected and corrected (by the measurement $\grave{\mathbf{z}}(k)$) estimates of the state variable $\grave{\mathbf{x}}(k)$. In a similar way, $\bar{\boldsymbol{\Sigma}}(k)$ and $\boldsymbol{\Sigma}(k)$ denote the uncorrected and corrected error variances of the state variable $\grave{\mathbf{x}}(k)$. The matrix $\mathbf{K}(k)$, computed in (22), is the Kalman gain. It specifies the degree to which the measurement is incorporated into the new state estimate. The corrected state estimate $\boldsymbol{\mu}(k)$, computed in (23), is the sum of the uncorrected state estimate, $\bar{\boldsymbol{\mu}}(k)$, plus the Kalman gain times a residual comprising the difference between the measurement vector $\grave{\mathbf{z}}(k)$ and estimated measurement $h(\bar{\boldsymbol{\mu}}(k))$. After that, the state error variance is updated in (24), where the matrix $\mathbf{H}(k)$ represent the linear term in a Taylor series expansion of the function $h(\grave{\mathbf{x}})$ around $\grave{\mathbf{x}} = \bar{\boldsymbol{\mu}}(k)$. The matrix $\mathbf{H}(k)$ is calculated as

$$\mathbf{H}(k) = \left.\frac{\partial h(\grave{\mathbf{x}})}{\partial \grave{\mathbf{x}}}\right|_{\grave{\mathbf{x}}=\bar{\boldsymbol{\mu}}(k)} \tag{27}$$

In the present case, it is assumed that the deflection angle of the right elevator can be measured directly, as shown in (21). Thus, in our case, $\mathbf{H}(k) = 1$ is constant.

In the time update, the forward propagation of the state estimate, $\bar{\boldsymbol{\mu}}(k+1)$, is calculated directly by a non-linear process model, g, at each discrete-time instant k using (17). In (26), the state error variance is propagated.

The $\mathbf{G}(k)$ matrix used here represents the linear term in a Taylor series expansion of the function $g(\grave{\mathbf{x}}, \grave{\mathbf{u}})$ around $\grave{\mathbf{x}} = \boldsymbol{\mu}(k)$ and $\grave{\mathbf{u}} = \grave{\mathbf{u}}(k)$. The matrix $\mathbf{G}(k)$ is calculated as

$$\mathbf{G}(k) = \left.\frac{\partial g(\grave{\mathbf{x}}, \grave{\mathbf{u}})}{\partial \grave{\mathbf{x}}}\right|_{\grave{\mathbf{x}}=\boldsymbol{\mu}(k),\, \grave{\mathbf{u}}=\grave{\mathbf{u}}(k)} \tag{28}$$

Even if the $\mathbf{H}(k)$ matrix is considered constant, $\mathbf{G}(k)$ is obtained by linearising the process model, g, around the corrected estimates of the state variable $\grave{\mathbf{x}}$. The partial derivative term given in (28) is calculated numerically according to the corrected estimates of the state, $\boldsymbol{\mu}(k)$, at each iteration, as follows:

$$\left.\frac{\partial g(\grave{\mathbf{x}}, \grave{\mathbf{u}})}{\partial \grave{\mathbf{x}}}\right|_{\grave{\mathbf{x}}=\boldsymbol{\mu}(k),\, \grave{\mathbf{u}}=\grave{\mathbf{u}}(k)} \approx \frac{g(\boldsymbol{\mu}(k) + \gamma, \grave{\mathbf{u}}(k)) - g(\boldsymbol{\mu}(k), \grave{\mathbf{u}}(k))}{\gamma} \tag{29}$$

where γ represents a small change in the state. The value of γ is selected as 10^{-4} in the simulations presented in the next section.

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

Table 2. *The EKF parameters*

Parameter	Symbol	Value
Standard deviation of the process noise for elevator dynamics	σ_q	0.002rad
Process noise covariance matrix	$\mathbf{Q}$	σ_q^2
Standard deviation of the measurement noise of position sensor	σ_r	0.005rad
Measurement noise covariance matrix	$\mathbf{R}$	σ_r^2
Initial mean of state (trim value of the elevator is considered)	$\bar{\boldsymbol{\mu}}(0)$	−0.024rad
Initial state variance	$\bar{\boldsymbol{\Sigma}}(0)$	0.1rad^2

Eventually, to generate the residual signal, $\hat{\mathbf{r}}$, the estimation of the state $\boldsymbol{\mu}(k)$ (the output of the EKF) is compared with the measurement read from the sensor, as shown in (12).

4.2.2 Decision-making

The residual signal is first filtered by a Moving-Average Filter (MAF), as shown in Fig. 5, to reduce the disturbing effect of noise. The mathematical expression for the MAF is [55]

$$\hat{\mathbf{r}}_f(k) = \begin{cases} \frac{1}{k+1}\sum_{l=0}^{k}\hat{\mathbf{r}}(l), & k < N_w \\ \frac{1}{N_w}\sum_{l=k-N_w+1}^{k}\hat{\mathbf{r}}(l), & k \geq N_w \end{cases} \tag{30}$$

where N_w is the number of points used in the moving average. The value of N_w is selected as 10 in the simulations presented in the next section. With this filtering process, a smoother signal, $\hat{\mathbf{r}}_f$, is obtained. Then, a decision signal, $\hat{\mathbf{r}}_{\text{decision}}$, is obtained as

$$\hat{\mathbf{r}}_{\text{decision}}(k) = |\ \hat{\mathbf{r}}_f(k)\ | \tag{31}$$

Finally, this decision signal is subjected to a simple decision logic to determine the presence of a failure. The failure detection system uses a threshold-based logic for the final decision regarding failure. The statistics of the noises considered in the fault detection system are assumed to be in accordance with military-grade high-precision sensors. These values are presented in Table 2. By using these error levels, the fixed threshold value is determined by simulation-based experiments as proposed by Zolghadri et al. [56]. The experiments intend to represent a trade-off between keeping the false alarm rate small and detecting small amplitudes. In this way, the threshold value to be used in the fault detection system is determined empirically according to simulation outputs as 0.01rad. In flight simulations, if the signal $\hat{\mathbf{r}}_{\text{decision}}$ exceeds this threshold value, the failure detection system indicates that there is a failure of the right elevator.

5.0 Simulation Studies

In this section, the performance of the integrated fault-tolerant control structure shown in Fig. 1 is analysed by non-linear flight simulations. The non-linear flight simulations run for this purpose are coded in a discrete iterative structure in the MATLAB environment. The Euler integration method is used to propagate the states of the aircraft with non-linear dynamics, and the cycle period is chosen as $T_d = 0.01$s. Since the failure detection system is already designed in discrete time, it can be integrated directly into the F-16 non-linear flight simulation. However, the controllers given in Section 4 are continuous-time systems. To integrate the control systems into the discrete non-linear simulation running in the MATLAB environment, the controllers are discretised. The zero-order hold method is used for discretisation, and 0.01s is selected as the sampling period. Finally, these discretised controllers are integrated into the complete non-linear model. Thus, the structure shown in Fig. 1 is constituted.

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

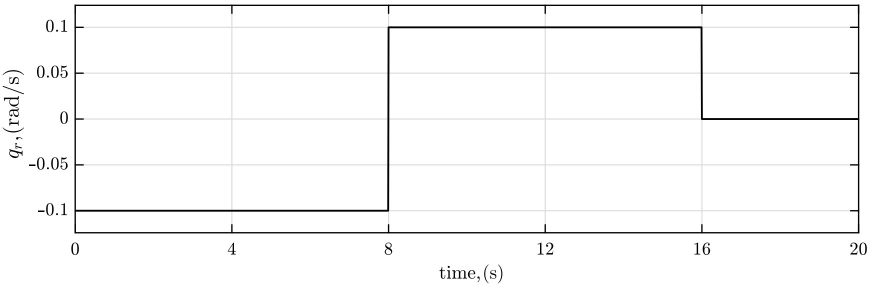

Figure 6. *Pitch-rate reference signal.*

In all the simulations presented below, the pitch-rate reference signal shown in Fig. 6 is used as the reference signal. All the non-linear simulations are initiated at the trim values given in Section 2.3. The parameters of the EKF, used in the residue generation process, are selected as shown in Table 2. The standard deviations given in this table are assumed by considering the worst-case noise margin considering the F-16 aircraft and military-grade sensor (i.e. high-precision absolute encoder) noise. The measurement value, $\grave{\mathbf{z}}$, used as an input to the failure detection system, is obtained by adding an artificial measurement noise, $\boldsymbol{\nu}$, to the right elevator deflection angle, δ_{er}. This measurement noise is generated artificially in MATLAB as a Gaussian noise with zero mean and standard deviation σ_r, which is given in Table 2. This measurement is used by the EKF to generate the expected value, $\boldsymbol{\mu}$, of the deflection angle, as explained in the previous section. The residual signal, $\hat{\mathbf{r}}$, is then obtained as shown in (12) and filtered as shown in (30), to obtain the filtered residual signal, $\hat{\mathbf{r}}_f$. Finally, the decision signal, $\hat{\mathbf{r}}_{\text{decision}}$, is obtained as shown in (31), which is subjected to the threshold test to decide whether a fault has occurred. The threshold value is set to 0.01rad as explained at the end of Section 4.2.2.

Firstly, the condition of healthy flight (no failure on the aircraft) is examined. It is clear from the structure in Fig. 1 that, if there is no failure, the classical controller should be active throughout the flight. In the first subsection below, it is shown that this is indeed the case. Thus, the simulations for the healthy condition allow us to analyse and compare the performance of both the classical controller and the robust FTC. After that, the performance of the proposed integrated control strategy is compared with other controllers, in the case of freezing failure and floating failure separately.

5.1 Healthy condition

In this subsection, the healthy aircraft simulation is performed using the integrated FTC and the robust FTC. We show that no false alarm is produced by the failure detection system during the complete simulations in the healthy condition. Thus, in the case of integrated FTC, the classical controller is active during the whole flight. Therefore, the results obtained for the integrated FTC are also valid for the classical controller.

The right elevator deflection measurement, $\grave{\mathbf{z}}$, used as an input to the failure detection system, and the expected value of the right elevator deflection (output of the EKF) are shown in the first row of Fig. 7. Additionally, the residual signal and the filtered residual signal are shown in the first plot in the second row of Fig. 7. As a preliminary assessment, the fact that the residual signal is unbiased indicates that there is no failure. The final decision, however, is made by a threshold test on the decision signal, $\hat{\mathbf{r}}_{\text{decision}}$, which is shown in the second plot in the second row of Fig. 7, together with the threshold level. The decision signal remains below the threshold throughout the simulation, thus no failure decision is taken.

The states of the aircraft under integrated FTC (or equivalently the classical controller) and robust FTC are obtained as shown in Fig. 8. Additionally, the elevator commands, deflections and deflection rates produced within this simulation are shown in Fig. 9. Since there is no failure or disruptive effect on the aircraft, the lateral/directional motion is not disturbed. Therefore, the aileron and rudder deflections are practically zero.

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

Table 3. Controllers' tracking performance comparison (IAE index)

Controller	Healthy condition	Freezing failure	Floating failure
Classical	0.0906	0.6054	0.6793
Robust FTC	0.0873	0.2545	0.2652
Integrated FTC	0.0906	0.2221	0.2600

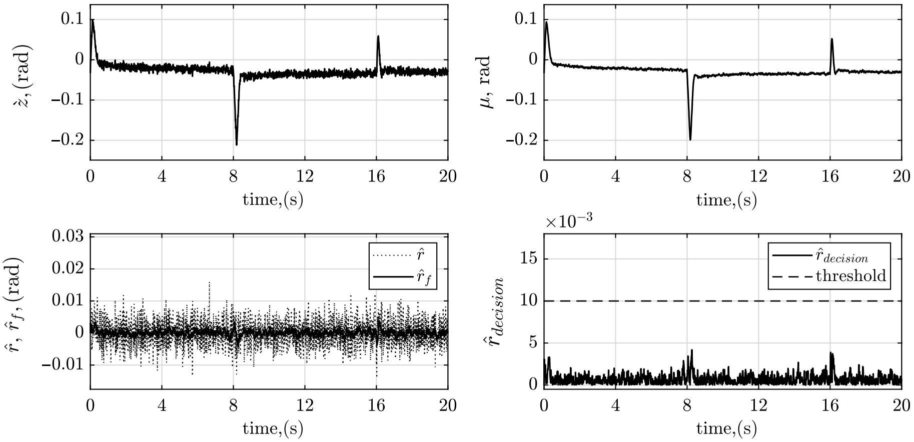

***Figure 7.** The measurement and the outputs of the failure detection system in healthy condition.*

After the simulation starts, the aircraft moves to the dive, and after the eighth second, starts to nose upwards according to the reference pitch-rate command given in Fig. 6. The results show that both controllers perform very well in case of no failure. Except for the 'east' state, all graphs are almost the same for both controllers. The robust FTC disturbs the 'east' state less than the classical controller. The disruption, however, is negligible (less than 2cm in the west direction as the aircraft travels more than 2km in the north direction). Additionally, the pitch-rate state indicates that both controllers work well and perform the tracking function very quickly. The pitch-rate tracking performance of the controllers is presented in Table 3 based on the Integral of the Absolute magnitude of the Error (IAE) performance index, defined as

$$\text{IAE} = \int_0^{T_s} |\, q_r(t) - q(t)\, |\, dt, \tag{32}$$

where T_s denotes the simulation time length and the integral is calculated numerically within the flight simulations. The IAE index shows that the tracking performance of the controllers is very close to each other. In addition, the pitch angle, AoA and altitude states also change accordingly. The increase in the velocity is also expected due to the dive motion of the aircraft. Furthermore, the lateral/directional states (roll and yaw angles and their rates, and the side-slip angle) remain almost zero. Besides, the control surface deflections and rates given in Fig. 9 show that the designed control system operates within the deflection and rate limits given in Section 2.2. As a result, these simulations show that both controllers (the integrated FTC (classical controller) and the robust FTC) work very well in the healthy condition and their performance is very similar.

5.2 Freezing failure

In this subsection, the condition of freezing failure is examined. Immediately after the eighth second of the 20s non-linear simulation, when the control system moves the elevator to apply the reverse pitch-rate

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

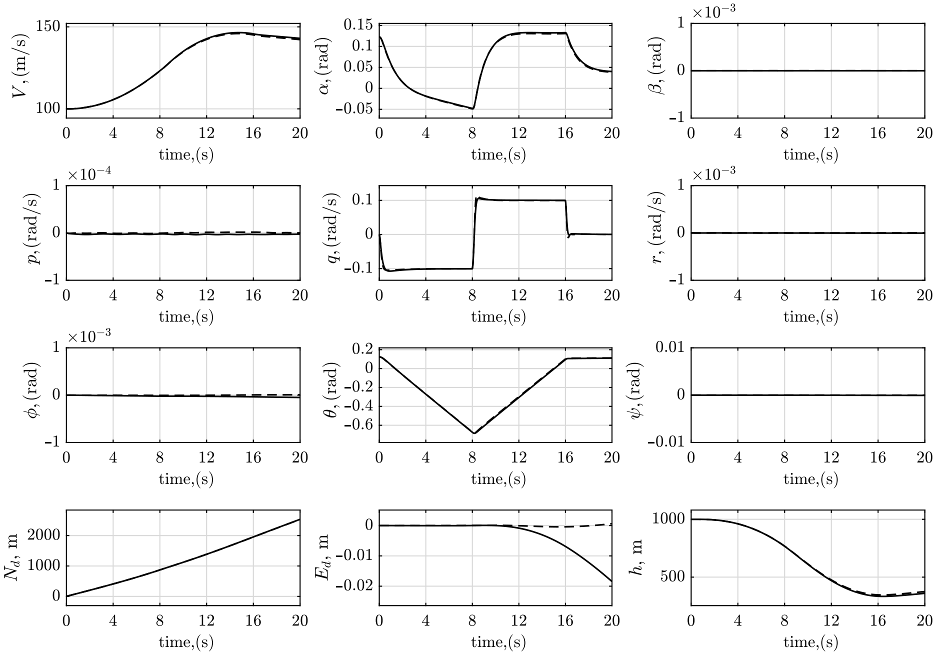

***Figure 8.** The states of the healthy F-16 aircraft (solid line: integrated FTC (equivalently classical controller); dashed line: robust FTC).*

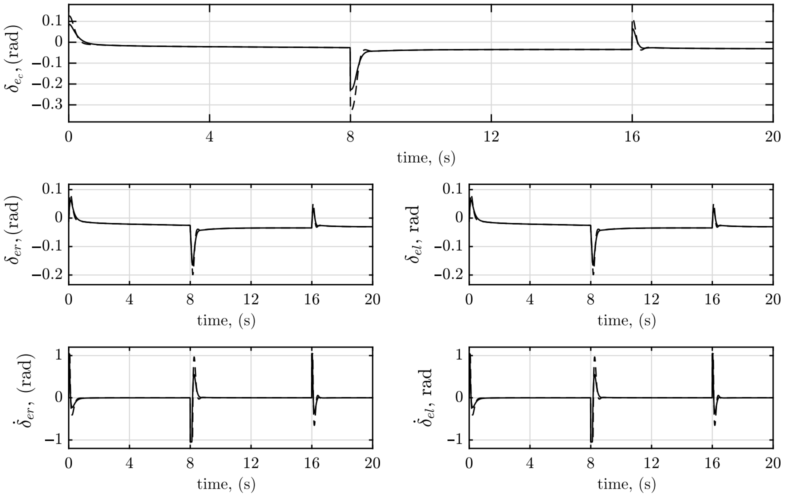

***Figure 9.** The elevator commands, deflections and deflection rates (solid line: integrated FTC (equivalently classical controller); dashed line: robust FTC) in the case of a healthy aircraft.*

command, as shown in Fig. 6, the right elevator becomes stuck in its extreme position. In this example, the right elevator is stuck at −0.1981rad (≈ −11.35deg) after 8.17s.

When the non-linear flight simulation is run for the above conditions, the failure is detected by the failure detection system at 8.25s of the simulation, and the system switches from the classical controller to the robust FTC. The right elevator deflection measurement taken from the sensor by the failure detection system and the expected value of the state (output of the EKF) are shown in the first row of Fig. 10.

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

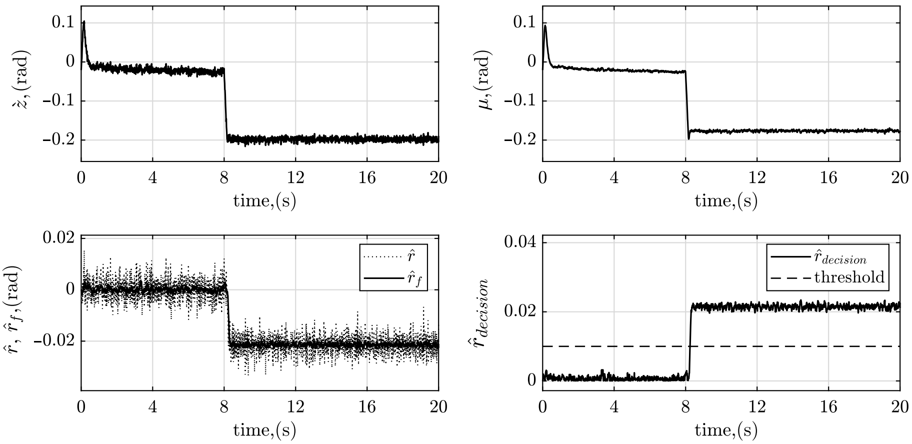

Figure 10. *The measurement and the outputs of the failure detection system in the case of freezing failure.*

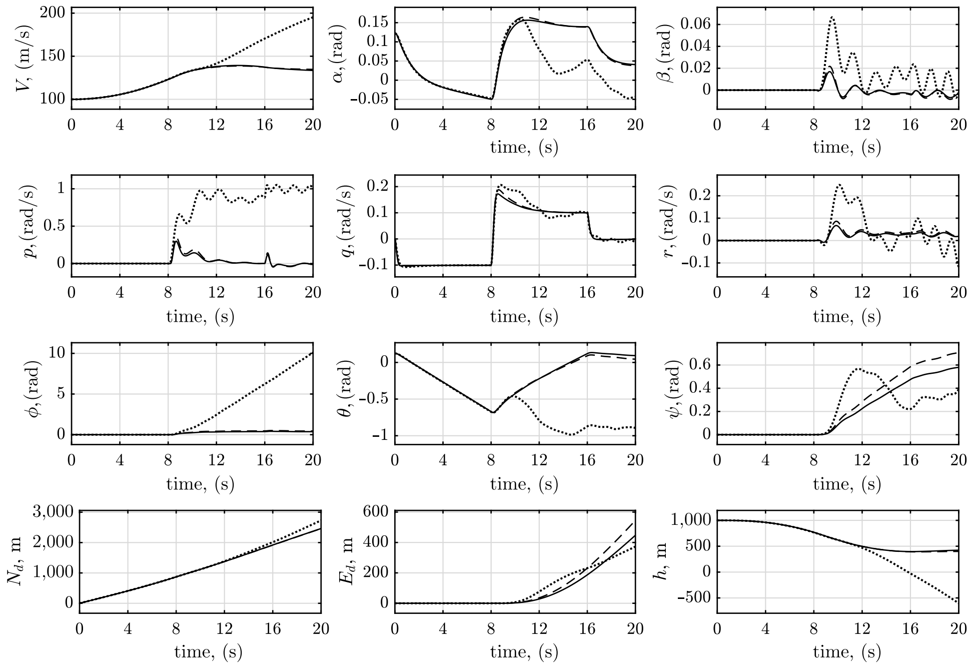

Figure 11. *The states of the F-16 aircraft in the case of freezing failure (solid line: integrated FTC; dashed line: robust FTC; dotted line: classical controller).*

Additionally, the residual signal and the filtered residual signal are shown in the bottom left plot of Fig. 10. In the bottom right plot of Fig. 10, the decision signal is shown together with the threshold level. The decision signal exceeds the threshold level at 8.25s, which is only 0.08s, i.e. eight sampling cycles, after the occurrence of the fault. Thus, the failure detection system detects the failure in only 80ms.

The classical controller is active from the beginning of the simulation until the failure is detected. The failure detection system detects the failure at 8.25s, and the control authority switches to the robust FTC. The states of the F-16 aircraft are shown in Fig. 11, in the cases of the integrated FTC, robust FTC and the classical controller. It is seen that the performance of the integrated FTC and the robust FTC are almost the same. Both controllers can continue to fly the aircraft safely despite the failure. Furthermore,

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

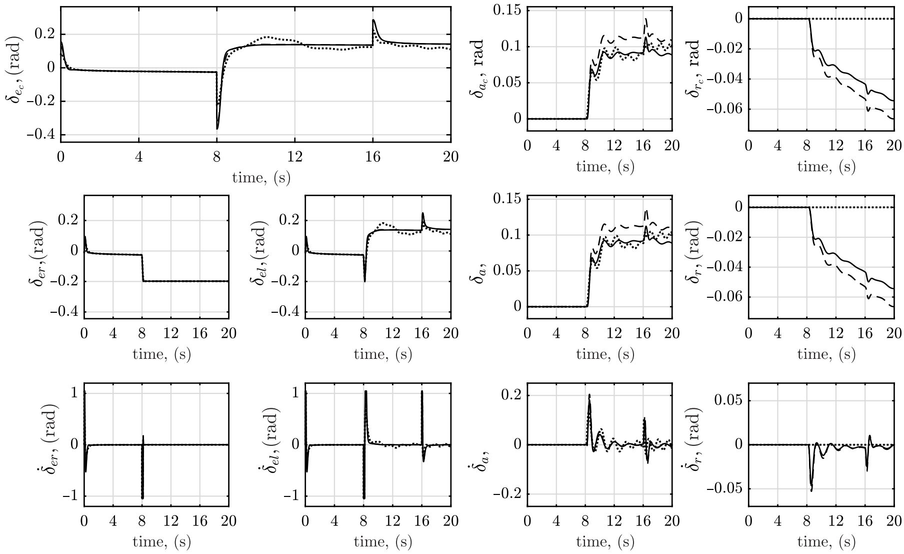

Figure 12. *The control surface commands, deflections and deflection rates in the case of freezing failure (solid line: integrated FTC; dashed line: robust FTC; dotted line: classical controller).*

the pitch rate tracking is also good considering the adverse effect of the failure. The pitch-rate tracking performance of the controllers is presented in Table 3. It is clear from this table that the performance of the proposed controller is slightly better than that of the robust FTC in the case of freezing failure. Although this difference is in favour of the proposed controller, it is not a significant difference in terms of the pitch-rate tracking performance. The reason why the tracking performance index of the integrated controller is slightly better is that the classical controller performs slightly better in the first 8s of the simulation, even though it has lower performance than the robust FTC throughout the simulation. There is such a small advantage as the integrated FTC uses the classical controller at the beginning of the simulation and the robust FTC after the failure occurs. Therefore, since this is a case study-specific factor, the performance of the proposed integrated FTC and the robust FTC is considered to be similar, which is a desirable situation, because this study aims for a structure that reduces the computational load without sacrificing performance. On the other hand, the performance of the classical controller is relatively inadequate compared with the other two controllers. In the classical controller, beyond the tracking performance, it cannot cope with the rolling moment disturbance caused by the asymmetrical right elevator failure. The roll rate increases consistently and reaches a high magnitude. The aircraft quickly turns around the body x-axis and crashes into the ground before the end of the simulation period. Hence, the classical controller cannot cope with freezing failure.

The control surface commands, deflections and deflection rates produced by each control system are shown in Fig. 12. It is clearly seen that the right elevator is jammed after 8.17s. The control surface deflections and rates shown in Fig. 12 indicate that the deflection and rate limits given in Section 2.2 are not violated when using any of the controllers.

The analysis for this section concerned the elevator freezing at its widest deflection during the pitch-rate tracking manoeuvre shown in Fig. 6. In addition, the margin of extreme jamming failures at which this proposed control approach is effective is also analysed as follows. As an extreme scenario, the freezing of the right elevator at the deflection limit (i.e. hard-over failure) creates a rolling moment of 24,160Nm when other states and control surfaces are at their trim values. However, under the same conditions, the ailerons can produce a rolling moment of 76,859Nm. Therefore, the ailerons can compensate for the disruptive rolling moment, since the rolling moment production capability is much higher than that caused by the asymmetrical hard-over elevator failure. On the other hand, the situation is slightly different for the decreasing pitching moment authority. The F-16 is designed using a Relaxed Static

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

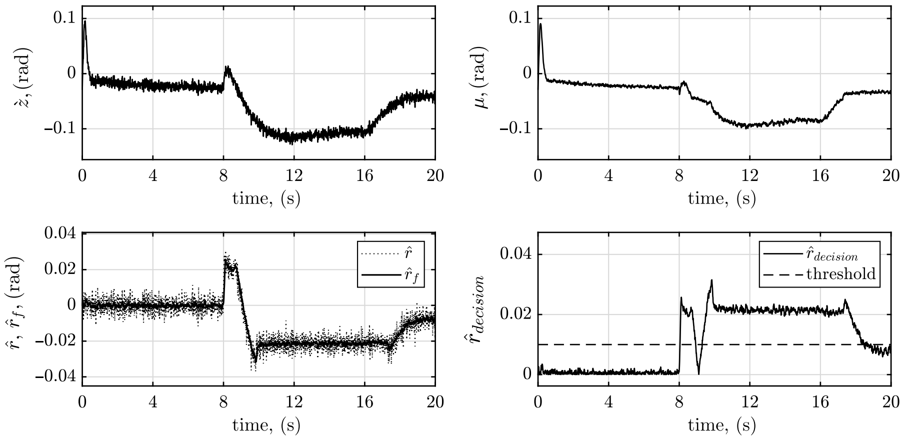

Figure 13. *The measurement and the outputs of the failure detection system in the case of floating failure.*

Stability (RSS) approach. Therefore, even in stable, level flight, the elevator surfaces must work continuously. The hard-over failure of the right elevator leaves no margin of deviation for the left elevator for continuous moment corrections due to RSS. Experiments on the non-linear F-16 simulation determined that the proposed controller can cope with the maximum deviation of -0.37rad of the right elevator for the flight scenario examined in this paper. Thus, the integrated FTC can recover all freezing failures greater than this value up to the limit of deviation of the right elevator in the positive direction. In summary, due to the longitudinal dynamics of the aircraft with RSS, the proposed controller naturally cannot operate in the most extreme failure condition (negative deflection limit for right elevator) but operates across a very wide freezing failure margin.

5.3 Floating failure

In this subsection, the condition of floating failure is examined. In the eighth second of the 20s nonlinear simulation, when the control system moves the elevator to apply the reverse pitch-rate command, the right elevator starts to float. In this example, the right elevator floats according to the assumption $\delta_{erf} = -0.5\alpha$ after 8s. When the non-linear flight simulation is run for the above conditions, the measurement and the outputs of the failure detection system are shown in Fig. 13. The decision signal exceeds the threshold level at 8.07s, which is only 0.07s, i.e. seven sampling cycles, after the occurrence of the failure. Thus, the failure detection system detects the failure in only 70ms, and the system switches from the classical controller to the robust FTC. The integrated fault-tolerant controller switches only once, as stated above. The floating failure is detected at 8.07s into the flight simulation. After the failure is detected, the robust FTC runs until the simulation is finished. Therefore, the drop of the decision signal to below the threshold value at around 9s, as seen in Fig. 13, does not result in switching.

The states of the F-16 aircraft are shown in Fig. 14, in the cases of the integrated FTC, robust FTC and classical controller. As in the case of freezing failure, the performance of the integrated FTC and the robust FTC is almost the same. Both controllers can continue to fly the aircraft safely despite the failure. Furthermore, the pitch-rate tracking is also good considering the adverse effect of the failure. The pitch-rate tracking performance of the controllers is presented in Table 3. It is clear from this table that the performance of the proposed controller is very similar to that with the robust FTC in the case of floating failure. On the other hand, the pitch-rate tracking performance of the classical controller is quite inadequate compared with the other two controllers. The rolling moment disturbance caused by the asymmetrical elevator failure disrupts the other states of the classical controlled aircraft further. The roll

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

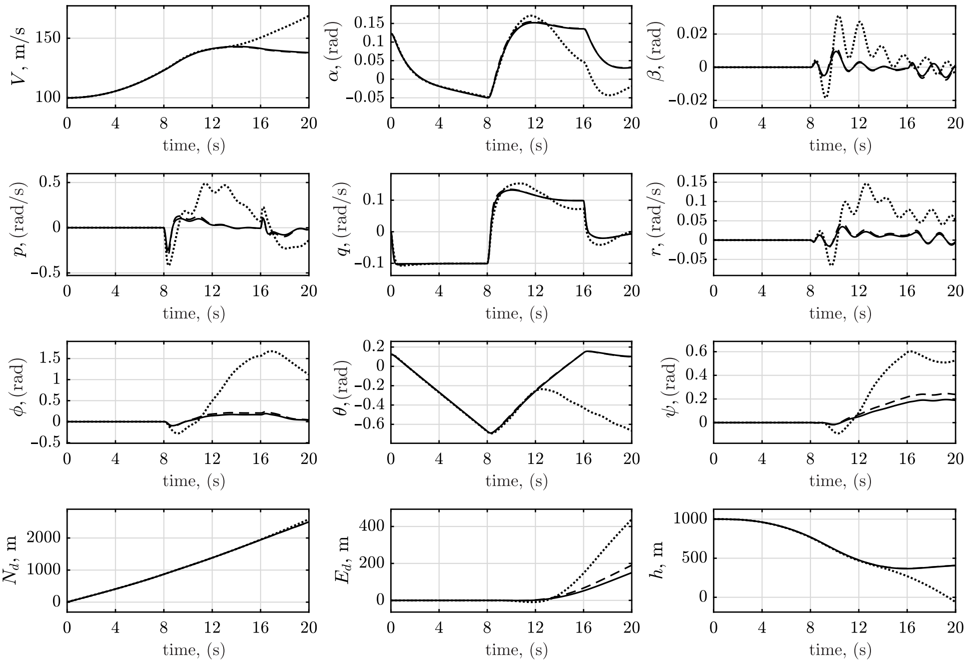

Figure 14. The states of the F-16 aircraft in the case of floating failure (solid line: integrated FTC; dashed line: robust FTC; dotted line: classical controller).

rate is completely out of control, changes direction and reaches high magnitudes. The aircraft crashes into the ground by the end of the simulation. Therefore, the classical controller also cannot cope with floating failure.

In the floating failure, the right elevator deflects depending on the aircraft's AoA as shown in (2). The aerodynamic model used in this study considers a low angle-of-attack flight regime [57] (i.e. $-0.1745\text{rad} \leq \alpha \leq 0.7854\text{rad}$). Furthermore, since the CAS controls the pitch rate of the aircraft, the AoA always remains limited because the control system implicitly controls the AoA by providing the pitch-rate manoeuvre, thus reducing the effects of the floating failure. Therefore, in this study, simulations can only be carried out within these limits, and it is shown that this failure can be recovered.

The control surface commands, deflections and deflection rates produced by each control system are shown in Fig. 15. It is clearly seen that the right elevator starts to float after 8s. The control surface deflections and rates shown in Fig. 15 indicate that the deflection and rate limits given in Section 2.2 are not violated when using any of the controllers.

6.0 Conclusions

Asymmetric elevator failures are fatal malfunctions that disrupt flight safety. The primary purpose of this work is to develop an FTC strategy such that the aircraft can continue to fly safely and complete its mission in the case of such failures. In addition to these objectives, the designed controller was expected to be applicable to real systems. Applicability to real systems also requires consideration of the computational burden. Given the actual systems, a flight control computer running numerous control systems has limited computing capacity. Since such failures are unlikely to occur, it is unnecessary to use controllers that require long computation times. Thus, in this work, an integrated FTC system approach that can be used for combat aircraft by enabling switching between the classical and robust FTC systems according to the occurrence of a failure is proposed. The proposed structure is also verified by non-linear F-16 flight simulations. These simulations show that the proposed controller produces a good response

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

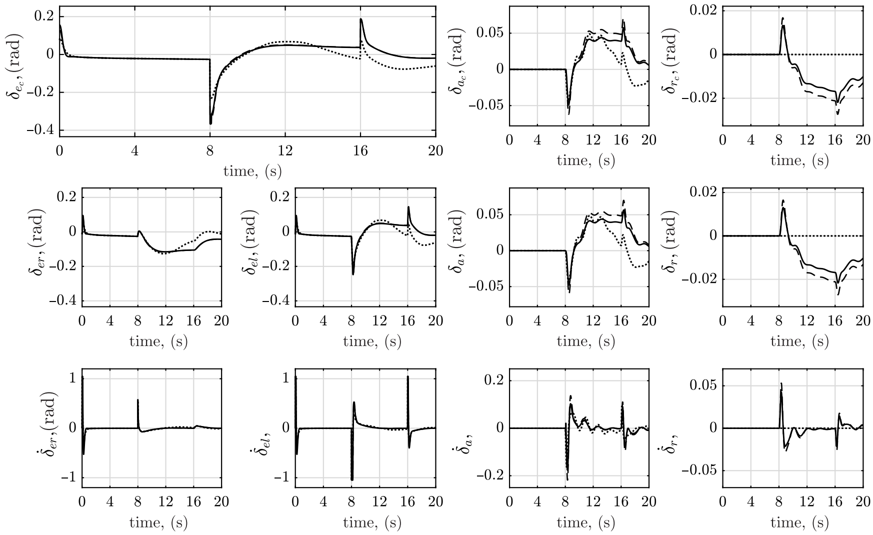

Figure 15. *The control surface commands, deflections and deflection rates in the case of floating failure (solid line: integrated FTC; dashed line: robust FTC; dotted line: classical controller).*

in both non-faulty and faulty conditions. The proposed method also reduces the load on the flight control computer by using the fast classical controller under normal conditions. However, in case of failure, it quickly switches to the robust FTC, which continues to fly the aircraft safely and completes the required manoeuvre.

In the proposed control approach, once a failure is detected, the robust FTC runs continuously. Hence, the integrated control system switches only once. Therefore, the problem of instability caused by multiple switchings between different controllers [58] does not arise when using the proposed approach.

The simulations performed within the scope of this study show that the integrated FTC performs quite similarly to the robust FTC structure proposed previously in the work of Gümüşboğa et al. [37]. Moreover, with this proposed integrated FTC approach, a structure with much less computational burden is presented without compromising control performance.

Within the scope of this paper, the aerodynamic redundancy approach is used effectively. The aerodynamic redundancy approach uses the remaining (healthy) control surfaces when one fails. In this approach, the main logic is to generate the moments of the failed control surface by using other control surfaces. For this reason, fault tolerance in aerodynamic redundancy is limited by the moment-generating capacities of the healthy control surfaces. These natural limits determine the operating limits of the designed integrated fault-tolerant control system. As a result, the proposed controller can deal with different and multiple failures within these natural limits. However, when these limits are exceeded, auxiliary control surfaces or hardware redundancy structures may be considered.

Acknowledgements. The authors received financial support from the Turkish Aerospace Industries Inc. through grant TM3021 for the research of this article.

References

[1] Argha, A., Su, S.W., Savkin, A. and Celler, B.C. Novel frameworks for the design of faulttolerant control using optimal slidingmode control, *Int. J. Robust Nonlin. Control*, 2018, **28**, (8), pp 3015–3032.

[2] Li, B., Hu, Q., Ma, G. and Yang, Y. Fault tolerant attitude stabilization incorporating closed-loop control allocation under actuator failure, *IEEE Trans. Aerosp. Electr. Syst.*, 2018, 55, (4), pp 1989–2000.

[3] Saied, M., Knaiber, M., Mazeh, H., Shraim, H. and Francis, C. BFA fuzzy logic based control allocation for fault-tolerant control of multirotor UAVs, *Aeronaut. J.*, 2019, **123**, (1267), pp 1356–1373.

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

[4] Ahmed-Zaid, F., Ioannou, P., Gousman, K. and Rooney, R. Accommodation of failures in the F-16 aircraft using adaptive control, *IEEE Control Syst. Mag.*, 1991, 11, (1), pp 73–78.

[5] Bai, L., Gao, Z., Qian, M. and Han, B. Fault tolerant scheme design for the flight control systems using adaptive control allocation technique, Proceedings of the 2018 Chinese Control and Decision Conference (CCDC), 2018, Shenyang, China, pp 2247–2251.

[6] Zhang, Y. and Jiang, J. Bibliographical review on reconfigurable fault-tolerant control systems, *IFAC Proc.*, 2003, 36, (5), pp 257–268.

[7] Muenchhof, M., Beck, M. and Isermann, R. Fault-tolerant actuators and drives Structures, fault detection principles and applications, *Annu. Rev. Control*, 2009, 33, (2), pp 136–148.

[8] Goupil, P., Boada-Bauxell, J., Marcos, A., Cortet, E., Kerr, M. and Costa, H. AIRBUS efforts towards advanced real-time fault diagnosis and fault tolerant control, *IFAC Proc.*, 2014, 47, (3), pp 3471–3476.

[9] Kanev, S.K. Robust fault-tolerant control, Doctoral dissertation, University of Twente, 2004.

[10] Rotondo, D. Background on fault tolerant control, in *Advances in Gain-Scheduling and Fault Tolerant Control Techniques*, Springer, 2018, Cham, pp 129–145.

[11] Saied, M., Lussier, B., Fantoni, I., Shraim, H. and Francis, C. Active versus passive fault-tolerant control of a redundant multirotor UAV, *Aeronaut. J.*, 2020, **124**, (1273), pp 385–408.

[12] Zhu, P., Jiang, J. and Yu, C. Fault-tolerant control of hypersonic vehicles based on fast fault observer under actuator gain loss fault or stuck fault, *Aeronaut. J.*, 2020, 124, (1278), pp 1190–1207.

[13] Liu, X., Liang, Y., Pan, Q. and Zhang, H. On robust control of missile control system, Proceedings of the 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Elsevier, vol. 39, 2006, pp 354–359.

[14] Zang, Q., Ye, S., Li, Y. and Wang, X. An enhanced LMI approach for mixed H_2/H_∞ flight tracking control, *Chin. J. Aeronaut.*, 2010, 24, (3), pp 324–328.

[15] Chadli, M., Aouaouda, S., Karimi, H.R. and Shi, P. Robust fault tolerant tracking controller design for a VTOL aircraft, *J. Franklin Inst.*, 2012, **350**, pp 2627–2645.

[16] Vural, S.Y., Dasdemir, J. and Hajiyev, C. Passive fault tolerant lateral controller design for an UAV, *IFAC-PapersOnLine*, 2018, **51**, (30), pp 446–451.

[17] Sobel, K. Eigenstructure assignment for fault tolerant flight control, Tech Rep, NASA Technical Report Server (NTRS), 2002.

[18] Zhang, Y. and Jiang, J. Active fault-tolerant control system against partial actuator failures, *IEE Proc. Control Theory Appl.*, 2002, 149, (1), pp 95–104.

[19] Gao, Z., Zhou, Z., Jiang, G., Qian, M. and Lin, J. Active fault tolerant control scheme for satellite attitude systems: Multiple actuator faults case, *Int. J. Control Autom. Syst.*, 2018, **16**, (4), pp 1794–1804.

[20] Yu, Y., Wang, H. and Li, N. Fault-tolerant control for over-actuated hypersonic reentry vehicle subject to multiple disturbances and actuator faults, *Aerosp. Sci. Technol.*, 2019, 87, pp 230–243.

[21] Fekih, A. and Chowdhury, F.N. A robust fault tolerant control strategy for a class of nonlinear systems, Proceedings of the American Control Conference, Minneapolis, MN, USA, 2006.

[22] Jiang, J. and Yu, X. Fault-tolerant control systems: A comparative study between active and passive approaches, *Annu. Rev. Control*, 2012, 36, (1), pp 60–72.

[23] Gao, Z., Cecati, C. and Ding, S.X. A survey of fault diagnosis and fault-tolerant techniques-Part I: Fault diagnosis with model-based and signal-based approaches, *IEEE Trans. Ind. Electr.*, 2015, 62, (6), pp 3757–3767.

[24] Tsourdos, A. *Flight Control Systems: Practical Issues in Design and Implementation*, The Institution of Electrical Engineers, Cornwall, UK, 2000.

[25] Tewari, A. *Advanced Control of Aircraft, Spacecraft and Rockets*, John Wiley & Sons, West Sussex, UK, 2011.

[26] Durham, W., Bordignon, K.A. and Beck, R. *Aircraft Control Allocation*, John Wiley & Sons, West Sussex, UK, 2017.

[27] Norouzi, R., Kosari, A. and Sabour, M.H. Evaluating the effects of lateral control surfaces failure on the generic transport model: A case study, *Aeronaut. J.*, 2020, **124**, (1277), pp 1016–1054.

[28] Ju, H. and Tsai, C. Longitudinal axis flight control law design by adaptive backstepping, *IEEE Trans. Aerosp. Electr. Syst.*, 2007, 43, (1), pp 311–329.

[29] Alwi, H. and Edwards, C. Fault tolerant longitudinal aircraft control using non-linear integral sliding mode, *IET Control Theory Appl.*, 2014, **8**, (17), pp 1803–1814.

[30] Chen, F., Niu, J. and Jiang, G. Nonlinear fault-tolerant control for hypersonic flight vehicle with multi-sensor faults, *IEEE Access*, 2018, **6**, pp 25427–25436.

[31] Stevens, B.L., Lewis, F.L. and Johnson, E.N. *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, John Wiley & Sons, New Jersey, USA, 2015.

[32] Hess, J.A., Morelli, E.A. and Murri, D.G. Flight-test techniques for quantifying pitch rate and angle-of-attack rate dependencies, *J. Aircr.*, 2017, 54, (6), pp 2367–2377.

[33] Hess, R.A. and Peng, C. Design for robust aircraft flight control, *J. Aircr.*, 2018, **55**, (2), pp 875–886.

[34] Etkin, B. and Reid, L.D. *Dynamics of Flight: Stability and Control*, Wiley, 1996, New York.

[35] Nelson, R.C. *Flight Stability and Automatic Control*, WCB/McGraw Hill, 1998, New York.

[36] Roskam, J. *Airplane Flight Dynamics and Automatic Flight Controls*, DARcorporation, Kansas, USA, 1998.

[37] Gümüşboğa, İ.and İftar, A. Fault-tolerant control augmentation system design for asymmetric elevator failures in a combat plane, *Kybernetika*, 2020, 56, (4), pp 767–793.

[38] Hajiyev, C. and Caliskan, F. Sensor and control surface/actuator failure detection and isolation applied to F-16 flight dynamic, *Aircr. Eng. Aerosp. Technol.*, 2005, 77, pp 152–160.

https://doi.org/10.1017/aer.2021.66 Published online by Cambridge University Press

[39] Adams, R.J., Buffington, J.M., Sparks, A.G. and Banda, S.S. *Robust Multivariable Flight Control*, Springer Science & Business Media, London, UK, 2012.

[40] Knapp, M., Berger, T., Tischler, M., Cotting, M. and Marcus, A. Development of a full flight envelope F-16 VISTA simulation model from closed-loop flight data, Proceedings of the AIAA SciTech Forum and Exposition, Kissimmee, FL, USA, 2018.

[41] Gümüşboğa, İ.and İftar, A. Aircraft trim analysis by particle swarm optimization, *J. Aeronaut. Space Technol.*, 2019, 12, (2), pp 185–196.

[42] Gümüşboğa, İ.and İftar, A. Modeling of asymmetrical elevator failures in the F-16 aircraft, Proceedings of the AIAA SciTech Forum and Exposition, San Diego, CA, USA, 2019.

[43] Huo, Y. Model of F-16 fighter aircraft, Tech *Rep*, University of Southern California, 2007.

[44] Nguyen, L.T., Ogburn, M.E., Gilbert, W.P., Kibler, K.S., Brown, P.W. and Deal, P.L. Simulation study of stall/post-stall characteristics of a fighter airplane with relaxed static stability, Tech Rep, NASA-TP-1538, NASA Langley Research Center, Hampton, VA, 1979.

[45] Fielding, C. and Flux, P.K. Non-linearities in flight control systems, *Aeronaut. J.*, 2003, 107, (1077), pp 673–686.

[46] Boskovic, J.D. and Raman, K.M. A decentralized fault-tolerant control system for accommodation of failures in higher-order flight control actuators, *IEEE Trans.* Control Syst. *Technol.*, 2009, 18, (5), pp 1103–1115.

[47] Zhou, K., Doyle, J.C. and Glover, K. *Robust and Optimal Control*, Prentice Hall, 1996, New Jersey.

[48] Gümüşboğa, İ.and İftar, A. Failure detection system design for asymmetric elevator failures in the F-16 aircraft, Proceedings of the 2019 International Symposium on Electrical and Electronics Engineering (ISEE), Ho Chi Minh, Vietnam, 2019.

[49] Chow, E.Y. A Failure Detection System Design Methodology, Doctoral Dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 1980.

[50] Chow, E.Y. and Willsky, A. Analytical redundancy and the design of robust failure detection systems, *IEEE Trans. Autom. Control*, 1984, 29, (7), pp 603–614.

[51] Wald, A. *Sequential Analysis*, Courier Corporation, New York, USA, 1973.

[52] Anderson, B.D.O. and Moore, J.B. *Optimal Filtering*, Prentice-Hall, 1979, Englewood Cliffs.

[53] Thrun, S., Burgard, W. and Fox, D. *Probabilistic Robotics*, MIT Press, Massachusetts, USA, 2005.

[54] Ducard, G. and Geering, H.P. Efficient nonlinear actuator fault detection and isolation system for unmanned aerial vehicles, *J. Guid. Control Dyn.*, 2008, **31**, (1), pp 225–237.

[55] Smith, S.W. *The Scientist and Engineer's Guide to Digital Signal Processing*, Analog Devices, California, USA, 1997.

[56] Zolghadri, A., Henry, D., Cieslak, J., Efimov, D. and Goupil, P. *Fault Diagnosis and Fault-Tolerant Control and Guidance for Aerospace Vehicles*, Springer, London, UK, 2014.

[57] Morelli, E.A. Global nonlinear parametric modelling with application to F-16 aerodynamics, Proceedings of the American Control Conference, Philadelphia, PA, USA, 1998.

[58] Hespanha, J.P. and Morse, A.S. Switching between stabilizing controllers, *Automatica*, 2002, **38**, (11), pp 1905–1917.

Cite this article: Gümüşboğa İ. and İftar A. (2022). An integrated fault-tolerant control strategy for control surface failure in a fighter aircraft. *The Aeronautical Journal*, **126**, 1568–1592. https://doi.org/10.1017/aer.2021.66