Comparison of Fault Detection and Isolation Methods for a Small Unmanned Aircraft

Raghu Venkataraman[a,*], Péter Bauer[b], Peter Seiler[a], Bálint Vanek[b]

[a]*Department of Aerospace Engineering and Mechanics, University of Minnesota, 107 Akerman Hall, 110 Union Street SE, Minneapolis, MN 55455, United States*
[b]*Systems and Control Laboratory, Computer and Automation Research Institute, Hungarian Academy of Sciences, Kende utca 13.-17., Budapest H-1111, Hungary*

Abstract

This paper compares three model-based methods for detecting and isolating stuck control surface faults on a small unmanned aircraft. The first method is parity-space based and compares a sensor measurement against a model-based prediction of the same quantity. The second method is observer-based and involves robust estimation for linear parameter-varying systems. The third method is also observer-based and involves multiple model adaptive estimation. The performance of the three methods are compared using flight data. The comparison shows that the three methods have different detection performance and runtime. The selection of a particular method depends on the application requirements and implementation constraints.

Keywords: Fault detection and isolation, linear parameter-varying systems, robust estimation, multiple model adaptive estimation, small unmanned aircraft systems
2018 MSC: 00-01, 99-00

1. Introduction

Fault detection and isolation (FDI) is one of several technical challenges facing the widespread use of small unmanned aircraft systems (UAS). The traditional approach to this problem involves checking the parity between the outputs of multiple sensors that measure the same quantity. This approach, commonly called as hardware redundancy (Collinson, 2003), is not well-suited for small UAS because they have constraints on their size, weight, and power. The modern approach is to use mathematical models and algorithms to detect faults. This approach, commonly called as analytical redundancy (Isermann and Ballé, 1997), does not require additional hardware and is thus a viable alternative for small UAS.

Textbooks such as Gertler (1998); Chen and Patton (1999); Isermann (2006); Ding (2013, 2014) address the general subject of fault diagnosis using either model-based or data-driven methods. Most fault detection methods make use of the so-called residual to draw inferences regarding the presence of a fault. A typical algorithm consists of a residual generation step, which may be either observer-based or parity space-based, and a residual evaluation step. The generation step produces residuals that, in some cases, are insensitive to noise and model uncertainties and sensitive to the faults under consideration. The evaluation step uses the residual and thresholds to draw inferences regarding the presence of a fault and its characteristics. Isermann (1984); Gertler (1997); Isermann and Ballé (1997); Isermann (2005); Hwang et al. (2010); Gertler (2014) present literature surveys of this area.

Fault diagnosis for aircraft applications has been widely investigated (Patton and Chen, 1994). The EU ADDSAFE program investigated the applicability of advanced model-based fault detection and diagnosis methods to a commercial aircraft benchmark (ADDSAFE, 2012; Goupil and Marcos, 2014). In addition, Goupil (2011) explains the state of practice of fault detection at Airbus and Goupil (2010) provides a specific example of analytical redundancy on-board the A380. The methods considered in the literature include neural networks (Napolitano et al., 1993), H_∞ optimization (Edelmayer et al., 1994; Marcos et al., 2005; Freeman et al., 2011, 2013b; Vanek et al., 2011a), geometric methods (Bokor et al., 2010; Seiler et al., 2011; Vanek et al., 2011b), LPV-based methods (Hecker et al., 2011; Vanek et al., 2014; Varga and Ossmann, 2014), and sliding-mode (Edwards et al., 2000; Alwi and Edwards, 2008). Fault diagnosis using analytical methods takes on additional significance for small UAS, given their payload constraints. The methods considered include multiple-model Kalman filtering (Rago et al., 1998), the superimposition of an excitation signal on the actuator commands (Bateman et al., 2007), H_∞ filtering (Rotstein et al., 2006;

*Corresponding author
Email addresses: venka085@umn.edu (Raghu Venkataraman), bauer.peter@sztaki.mta.hu (Péter Bauer), seile017@umn.edu (Peter Seiler), vanek@sztaki.mta.hu (Bálint Vanek)

Pandita, 2010; Freeman and Balas, 2013; Freeman et al., 2013a), and multiple model adaptive estimation (Cristofaro et al., 2017), (Rotondo et al., 2017), and (Bauer et al., 2018). Some of these methods have also been validated using flight tests, either online or offline.

The effect of the feedback controller is important when designing the FDI methods. For example, a well-designed feedback controller may suppress the fault of interest, thus making it harder to detect. Stoustrup et al. (1997) showed that in the presence of model uncertainty, the designs of the controller and the FDI filter are coupled. However, since this paper compares different FDI methods, such an integrated approach is not feasible. Thus this paper follows a sequential approach wherein each FDI method is designed in consideration of a given nominal controller. The interested reader may refer to Pandita et al. (2011, 2013), and the references therein, for more information on FDI performance under closed-loop control.

From the literature cited thus far, it is clear that several methods have been proposed for fault diagnosis. This paper compares three such model-based methods for detecting and isolating stuck control surface faults on a small UAS. The first method is parity-space based and compares a sensor measurement against a model-based prediction of the same quantity. It is the simplest of the three methods and serves as a baseline. The second method is observer-based and involves the concept of robust estimation for linear parameter-varying (LPV) systems (Venkataraman and Seiler, 2018). The third method is also observer-based and involves the concept of multiple model adaptive estimation (MMAE) which is well established in Hassani et al. (2009a) and Hassani et al. (2009b). The fault detection problem considered in this paper was previously reported in Bauer et al. (2018) using the third method. This paper advances the results of Bauer et al. (2018) by: (1) presenting two additional methods and (2) comparing the industrial relevance of all the methods using flight data.

The paper begins with a description of the aircraft model and the scope of the particular FDI problem considered (Section 2). Then the three methods are presented (Section 3) and compared (Section 4) using a common set of flight data. The conclusions are presented in Section 5. Although the paper discusses one particular UAS, the results are relevant to other UAS configurations.

2. Preliminaries

2.1. Aircraft Model

The aircraft (Figure 1) is called the Vireo and is comprised only of a wing and a fuselage. This aircraft was originally built by Sentera, LLC and is currently maintained and operated by the University of Minnesota. The fully integrated aircraft has a gross mass of 1.28 kg, a wing span of 0.97 m, and a fuselage length of 0.52 m. Control is provided via a pair of independently actuated elevons and a tractor-type electric motor that drives a fixed-pitch propeller. Since the aircraft does not have a rudder, directional control is achieved indirectly via lateral control. Sensing is provided via an inertial measurement unit, a GPS receiver, a magnetometer, and a pitot-static system.

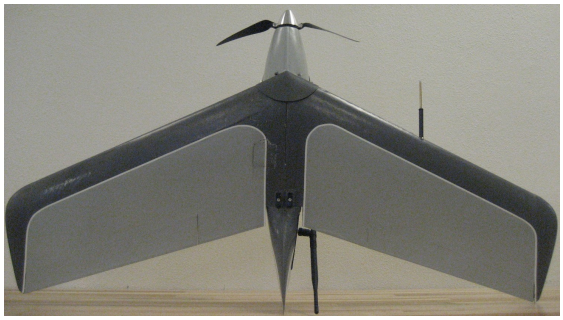

Figure 1: The UAS considered in this paper.

Since this aircraft is assumed to be rigid and the model assumes zero wind, the pertinent states are the Euler angles (ϕ, θ, ψ), the angular velocity in the body axes (p, q, r), the airspeed in the body axes (u, v, w), and the position of the aircraft in a local North-East-Down frame (p_N, p_E, p_D). The models used to design the FDI algorithms make use of only some of these states, as described shortly. The nonlinear equations of motion of rigid, fixed-wing aircraft are documented in several textbooks (Nelson, 1998; Cook, 2007) and are thus not repeated here. The throttle δ_t is normalized to the interval $[0, 1]$. The left δ_l and the right δ_r elevons each have a physical deflection range of $[-30, +20]\,^\circ$, where positive values correspond to trailing-edge down deflections. As such, each elevon excites both the longitudinal and the lateral-directional dynamics. Therefore, for modeling convenience, these dynamics are decoupled by expressing the elevons in terms of the traditional elevator δ_e and the aileron δ_a via the relations $\delta_l = \delta_e - \delta_a$ and $\delta_r = \delta_e + \delta_a$.

In order to design the FDI algorithms, the nonlinear equations of motion are linearized around steady, wings-level, constant altitude, and constant airspeed trim conditions. At any given trim condition, the aircraft dynamics is described by a linear time-invariant (LTI) model:

$$\begin{aligned} \dot{\mathbf{x}}_\star &= \mathbf{A}_\star \mathbf{x}_\star + \mathbf{B}_\star \mathbf{u}_\star, \\ \mathbf{y}_\star &= \mathbf{C}_\star \mathbf{x}_\star + \mathbf{D}_\star \mathbf{u}_\star, \end{aligned} \tag{1}$$

where the subscript $\star$ is replaced with either *lon* or *lat*. The longitudinal model G_{lon} uses: $\mathbf{x_{lon}} = [q, w]^T$, $\mathbf{u_{lon}} = \delta_e$, and $\mathbf{y_{lon}} = q$. G_{lon} only accounts for the short period mode because the phugoid mode of this aircraft is not accurately characterized. The lateral-directional model G_{lat} uses: $\mathbf{x_{lat}} = [v, p, r, \phi]^T$, $\mathbf{u_{lat}} = \delta_a$, and $\mathbf{y_{lat}} = [\phi, p, r]^T$. All the signals are expressed in SI units $(\mathrm{m}, \mathrm{kg}, \mathrm{s})$.

A collection of such LTI models, each of which corresponds to a different flight condition, constitutes a gridded linear parameter-varying (LPV) representation of the aircraft dynamics. This paper considers 20 grid points between the stall speed of 12 $\mathrm{m\,s^{-1}}$ and the high speed limit

of 20 $\mathrm{m\,s^{-1}}$. In addition, past flight data indicates that the rate-of-variation of the airspeed is bounded by $\pm 8\ \mathrm{m\,s^{-2}}$ during typical flight maneuvers. These rate bounds restrict the airspeed trajectories to only those that are realistic. Figure 2 shows the Bode diagrams of the aileron-to-roll rate (left) and the elevator-to-pitch rate (right) responses at each grid point in the airspeed domain. As expected, the frequencies of both the dutch roll mode and the short period mode increase with increasing airspeed. This paper uses LPVTools (a Matlab toolbox) for the modeling, analysis, and synthesis of LPV systems (Balas et al., 2015b; Hjartarson et al., 2015). The Appendix provides the state-space matrices of G_{lon} and G_{lat} at the nominal cruise airspeed of 15.4 $\mathrm{m\,s^{-1}}$.

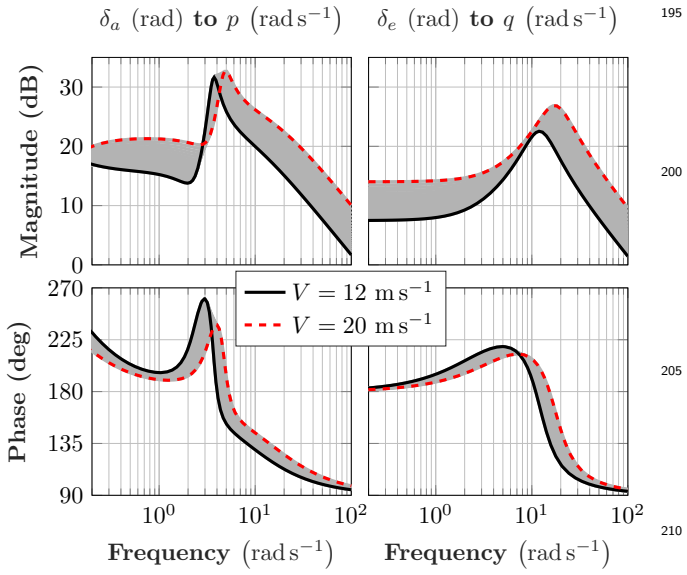

Figure 2: The Bode diagrams of the lateral-directional model (left) and the short period model (right) at each grid point.

The elevon actuator dynamics are given by $G_a(s) = \frac{\omega_a^2}{s^2+2\zeta_a\omega_a s+\omega_a^2}$, where $\zeta_a = 0.77$ and $\omega_a = 62.8\ \mathrm{rad\,s^{-1}}$ are estimated using system identification experiments. The experiments also quantify the closed-loop time delay as $\tau_f = 0.05$ s, which encompasses delays in the actuators, the flight computer, and the sensors. In this paper, all of this delay is grouped at the input to the actuator and modeled using Padé approximations. The delay-free approximation of $G_a e^{-\tau_f s}$ is denoted by G_a^L and is given in the Appendix.

2.2. Scope of Current Work

Each component on a small UAS can fail in a number of different ways, thereby contributing to the failure rate of the overall aircraft (Freeman, 2014). The scope of this paper is limited to the detection and isolation of stuck faults in either of the elevons of the aircraft. The reasons for this are twofold. First, the servomotors that actuate the elevons have a high failure rate, e.g. three servomotors have failed over the course of 35 flights on the UAS considered in this paper. Second, the stuck failure mode poses the greatest risk according to a failure modes and effects analysis (FMEA) of hobby-grade servomotors. The reader is referred to Appendix A of Freeman (2014) and Appendix 1 of Amos et al. (2013) for the FMEA.

Stuck faults impose constraints on the flight envelope of the aircraft. Section III of Venkataraman et al. (2017) shows one example of the impact of stuck faults on the allowable flight envelope of a fixed wing aircraft. In particular, there may be fault magnitudes where it is not possible to trim the aircraft in the conventional sense, e.g. if an elevon gets stuck at one of its extreme positions (hard-over). Therefore, this paper only considers stuck faults for which a steady, wings-level, constant altitude, and constant airspeed trim condition exists. The subset of tolerable stuck faults is centered at the nominal elevon trim position and has the range $[-7, +5]^\circ$. The reader is referred to Venkataraman (2018) for the trim analysis. When one of the elevons of the aircraft experiences a stuck fault, the FDI algorithm: (1) detects that the fault has occurred and (2) isolates whether the failure is in the left or right elevon.

2.3. Requirements

The fault detection and isolation is followed by a reconfiguration of the flight control law (Bauer et al., 2018) or a manual takeover by a human pilot. When a fault occurs, the aircraft will deviate from its trim point. If the fault is not detected in a timely manner, the aircraft may not be recoverable. Thus the FDI algorithms are designed to detect faults within the so-called maximum allowable detection time $\bar{t}_{det}$. This requirement is specified by invoking the work of Wilborn and Foster (2004) in the area of loss-of-control (LOC). This paper makes use of three of the five flight envelopes that Wilborn and Foster (2004) proposed: the unusual attitude, the dynamic pitch control, and the dynamic roll control envelopes. In particular, the fault must be detected before the aircraft state exits any one of the three envelopes. In general, this depends on the magnitude of the stuck fault. From nonlinear simulations, $\bar{t}_{det}$ is obtained as 9 s for faults that are within $\pm 1^\circ$ of the nominal trim elevon position. The reader is referred to Chapter 6 of Venkataraman (2018) for additional details.

3. Methods

This section presents three methods for designing the FDI algorithm. All the methods use some subset of the controller reference commands, the sensor measurements, and the actuator commands. The first two methods (A & B) generate a residual e_* and detect its threshold crossings. In particular, a fault is declared if $|e_*| \geq T_*$. Although the thresholds may be state-dependent or time-varying, this paper uses the constant thresholds T_*. The third method generates residuals, compares them with a threshold to filter out noise variations and then calculates model probabilities based-on the high enough residuals. A threshold

of 50% is applied for model probabilities to select the actually valid aircraft model. The thresholds are important parameters that control the trade-off between the rates of false alarms and missed detections. They are selected to ensure that the FDI algorithms do not declare false alarms when applied to unfaulted flight data (see Section 6.8.2 of Venkataraman (2018) for additional details).

3.1. Method A: Baseline

Method A is pictured in Figure 3. The F_A block comprises the LTI model G_{lat} at an airspeed of 15.4 $\mathrm{m\,s^{-1}}$ and the actuator model G_a. It uses the measured roll rate p and the aileron command δ_{ac} to generate a roll rate residual $\bar{e}_p = \hat{p} - p$, where $\hat{p}$ is the model-predicted roll rate. The roll rate is used since it is the most sensitive to elevon deflections (see the state-space matrices in the Appendix.) The actuator model G_a incorporates not only the second-order transfer function, but also the servo position limits of $[-30, +20]^\circ$, the rate limits of $\pm 338^\circ\,\mathrm{s^{-1}}$, and the time delay of 0.05 s. Thus the output $\bar{\delta}_a$ of G_a is a prediction of the virtual aileron position, based purely on the aileron command. The R_A block filters $\bar{e}_p$ to produce the final residual e_p. R_A is selected as a fifth-order, low-pass Bessel filter with a bandwidth of 2 $\mathrm{rad\,s^{-1}}$ and the thresholds are selected as $\pm 12.5^\circ\,\mathrm{s^{-1}}$.

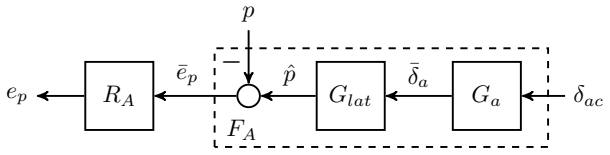

Figure 3: The architecture of Method A of the FDI algorithm.

3.2. Method B: Robust LPV

Method B is observer-based and generates two sets of estimates of the left and right elevon positions (δ_l, δ_r). In the absence of a fault, these two sets of estimates are nearly equal and differ only at high frequencies, i.e. due to noise. In the presence of a fault, one set continues to be a reliable estimate of the actual elevon positions, while the other set exhibits a low frequency divergence. The difference between the two sets of estimates serves as the residual.

Method B is pictured in Figure 4. The F_B block comprises the observer F and the elevon actuator model G_a. The inputs to $\mathrm{diag}\,(G_a, G_a)$ are the left and right elevon deflection commands $(\delta_{lc}, \delta_{rc})$ generated by the nominal controller. The outputs $\left(\bar{\delta}_l, \bar{\delta}_r\right)$ of $\mathrm{diag}\,(G_a, G_a)$ are the first set of estimates of the left and right elevon positions. Since they account only for the actuator model, they are reliable only in the absence of faults.

The observer F uses the commanded roll attitude ϕ_{cmd}, the estimated roll attitude ϕ, and the angular rates p and q to generate the second set of estimates of the left and right elevon positions $\left(\hat{\delta}_l, \hat{\delta}_r\right)$. Since the observer accounts

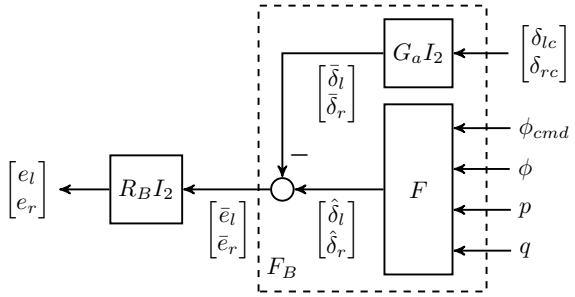

Figure 4: The architecture of Method B of the FDI algorithm.

for the closed-loop aircraft dynamics (explained shortly), $\hat{\delta}_l$ and $\hat{\delta}_r$ are reliable even in the presence of a stuck elevon fault. The output $(\bar{e}_l, \bar{e}_r)$ of F_B is the difference between the two sets of estimates. The $\mathrm{diag}\,(R_B, R_B)$ block filters $(\bar{e}_l, \bar{e}_r)$ to produce the final residual (e_l, e_r). R_B is selected as a fifth-order, low-pass Bessel filter with a bandwidth of 10 $\mathrm{rad\,s^{-1}}$ and the thresholds are selected as $\pm 5^\circ$.

The observer F comprises two other filters (Figure 5). F_{lat} uses $[\phi_{cmd}, \phi, p]^T$ to estimate the position of the virtual aileron $\hat{\delta}_a$. It is designed using the lateral-directional model G_{lat}. F_{long} uses q to estimate the position of the virtual elevator $\hat{\delta}_e$. It is designed using the short period model G_{lon}. The transformation block $T_{lr\leftarrow ea} = \left[\begin{smallmatrix}1 & -1\\ 1 & 1\end{smallmatrix}\right]$ converts $\hat{\delta}_a$ and $\hat{\delta}_e$ into the elevon position estimates $\left(\hat{\delta}_l, \hat{\delta}_r\right)$. The exclusion of the phugoid modal dynamics in designing F_{long} slightly impacts the accuracy of the elevon position estimates $\left(\hat{\delta}_l, \hat{\delta}_r\right)$. As such, this is a shortcoming of the current design that is managed within the leeway afforded by the thresholds. It may be remedied by modeling the phugoid modal dynamics to a higher degree of accuracy.

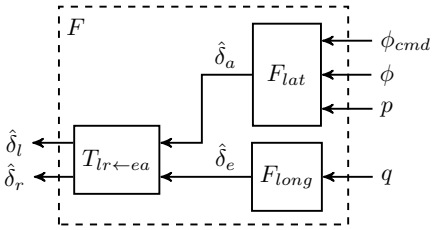

Figure 5: The observer F comprises two other observers.

3.2.1. Synthesis Framework

F_{lat} and F_{long} are designed using the generic block diagram shown in Figure 6, which corresponds to the robust output estimation problem discussed in Venkataraman and Seiler (2018). The specific block diagrams for the synthesis of F_{lat} and F_{long} are obtained by replacing $\star$ in Figure 6 with the appropriate subscript, as explained shortly. $H_\star$ is a nominal LPV plant that includes either G_{lat} or G_{lon}, the actuator model, the nominal controller, and any synthesis weighting functions. These weighting functions are used to

trade-off competing performance objectives. Δ is a block-structured perturbation that includes any uncertainties in G_{lat} or G_{lon}. Its input-output behavior is described using integral quadratic constraints (IQC) (Megretski and Rantzer, 1997). Further, d denotes the generalized disturbances, y denotes the measurements sent to the filter (not to be confused with the plant output), $\delta_\star$ denotes the actual aileron or elevator position, and $\hat{\delta}_\star$ denotes the estimated aileron or elevator position. $W_\star$ filters the estimation error $e_\star = \hat{\delta}_\star - \delta_\star$ over a desired frequency range.

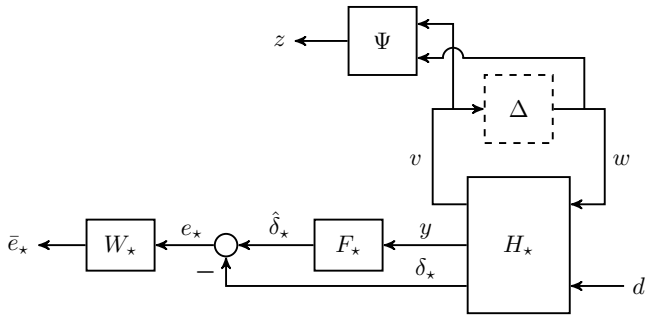

Figure 6: The block diagram for synthesizing F_{lat} and F_{long}.

The synthesis objective is to find the filter $F_\star$ that yields the smallest possible upper bound on the worst-case gain from d to $\bar{e}_\star$. Four design models are considered for $\mathcal{F}_u(H_\star, \Delta)$ by placing restrictions on the model uncertainty and/or the airspeed domain (see Table 1). The nominal filters ($\Delta = 0$) are designed using H_∞ and LPV syntheses, respectively. The robust filters ($\Delta \neq 0$) are designed using Theorem 2 of Venkataraman and Seiler (2018). Sections 3.2.4 demonstrates that, by virtue of making the fewest assumptions, the robust-LPV filter outperforms the other filters. The particular choices of $H_\star$ and Δ for the uncertain, LPV design model (fourth row in Table 1) are explained next.

Table 1: The four design models for synthesizing F_{lat} and F_{long}.

Aircraft model	Uncertainty	Filter
LTI at 15.4 $\mathrm{m\,s^{-1}}$	$\Delta = 0$	Nominal-LTI
LTI at 15.4 $\mathrm{m\,s^{-1}}$	$\Delta \neq 0$	Robust-LTI
LPV: $[12, 20]$ $\mathrm{m\,s^{-1}}$	$\Delta = 0$	Nominal-LPV
LPV: $[12, 20]$ $\mathrm{m\,s^{-1}}$	$\Delta \neq 0$	Robust-LPV

3.2.2. Lateral-Directional Filter F_{lat}

To synthesize the F_{lat}, Figure 6 is used with the systems H_{lat}, F_{lat}, and W_{lat}, and the signals δ_a, $\hat{\delta}_a$, e_a, and $\bar{e}_a$. The measurement signal is $y = [\phi_{cmd}, \phi, p]^T$. Figure 7 shows the generalized plant $\mathcal{F}_u(H_{lat}, \Delta)$, which includes the nominal LPV lateral-directional aircraft model G_{lat}, the actuator model G_a^L, and the roll attitude controller K_A taken as $\left(-0.34 - \frac{0.086}{s}\right)(\phi_{cmd} - \phi) + 0.06p$. The nominal plant G_{lat} is affected by the norm-bounded, LTI, multiplicative uncertainty $\|\Delta\|_\infty \leq 1$ at its input. The uncertain plant is thus given by $G_{lat}(1 + \Delta W_\Delta)$. The input δ_a to the uncertain plant is the quantity to be estimated. The feedback loop involving G_a^L, G_{lat}, and K_A represents the closed-loop lateral-directional aircraft dynamics. This feedback loop is affected by: the disturbance $\tilde{d}_u$ at the input of G_a^L, the disturbance $\tilde{d}_y$ at the output of G_{lat}, and the reference command ϕ_{cmd}. The performance weights W_u, W_y, and W_r relate the disturbances $\tilde{d}_u$, $\tilde{d}_y$, and ϕ_{cmd} to their respective normalized counterparts d_u, d_y, and d_r.

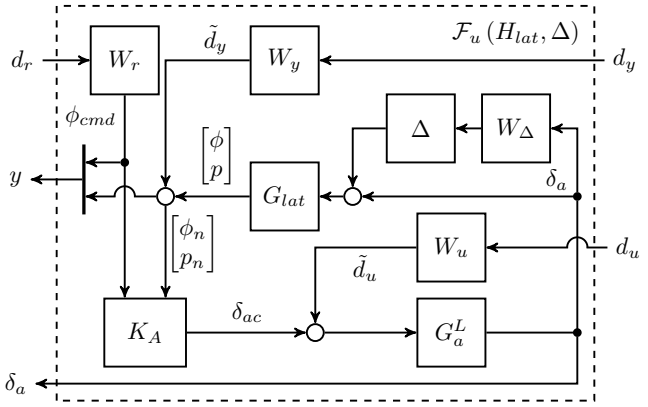

Figure 7: The generalized plant that is used for synthesizing F_{lat}.

The weighted, uncertain closed-loop shown in Figure 6 has the input $d = (d_u, d_y, d_r)$ and the output $\bar{e}_a$. The synthesis is an iterative process that involves weight selection and tuning. Table 2 lists the final values of all the weights, along with their interpretations. For instance, the weights W_u, W_y, and W_r are selected as the typical disturbances expected in the signals $\tilde{d}_u$, $\tilde{d}_y$, and ϕ_{cmd}, respectively.

Table 2: The final weights selected for synthesizing F_{lat}.

Weight	Final value	Weight interpretation:
W_u	$3\,(\pi/180)$	Typical aileron disturbance.
W_y	$\left[\begin{smallmatrix}6 & 0\\ 0 & 6\end{smallmatrix}\right](\pi/180)$	Typical disturbances in the roll angle and the roll rate.
W_r	$30\,(\pi/180)$	Typical roll command.
W_Δ	$\frac{s+3.924}{s+392.4}$	Shapes the uncertainty in G_{lat} across frequency.
W_{lat}	$\frac{s+4}{1.122s+0.07113}$	Inverse of the desired sensitivity $-\delta_a \to e_a$.

The weight W_Δ shapes the uncertainty in G_{lat} across frequency. In general, the model uncertainty is low at low frequencies and high at high frequencies. For this problem, it is assumed that the uncertainty in G_{lat} is 1% at frequencies below the dutch roll mode $\left(4.1\ \mathrm{rad\,s^{-1}}\right)$ and increases to 100% at high frequencies. Thus W_Δ is selected as shown in Figure 8, where the numbers within the parentheses specify the particular levels of uncertainty

at the natural frequencies of the dutch roll mode, the roll subsidence mode, and the actuator.

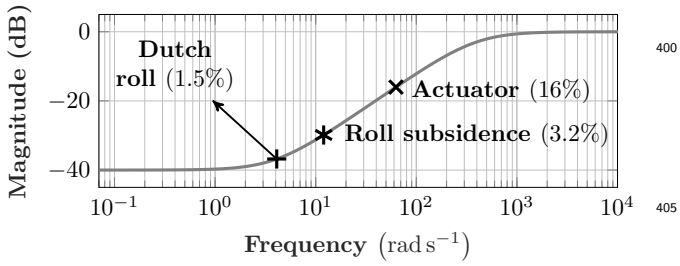

Figure 8: The Bode diagram of the weight W_Δ.

In order to select W_{lat} (the filter on e_a in Figure 6), let S_a and $\bar{S}_a$ denote the sensitivity function $-\delta_a \to e_a$ and its upper bound, respectively. It is more important to minimize e_a at low frequencies as compared to high frequencies. Thus $\bar{S}_a$ is specified as a first-order transfer function with a DC gain of -35 dB, a high-frequency gain of 1 dB, and a bandwidth of 4 $\mathrm{rad\,s^{-1}}$. The bandwidth here refers to the -3 dB point with respect to the high frequency gain and corresponds to the natural frequency of the dutch roll mode. W_{lat} is then selected as $\bar{S}_a^{-1}$.

Since Δ is a norm-bounded LTI uncertainty, it satisfies all IQCs defined by multipliers of the form: $\Pi(j\omega) = \begin{bmatrix} x(j\omega) & 0 \\ 0 & -x(j\omega) \end{bmatrix}$, where $x(j\omega) \geq 0$ is a bounded measurable function (Megretski and Rantzer, 1997). To obtain time-domain IQCs, $x(j\omega)$ is factorized as $\psi_x(j\omega)^\sim M_x \psi_x(j\omega)$, where $\psi_x(j\omega)$ is taken as $\left[1, \frac{1}{j\omega+0.031}\right]^T$ and M_x is a symmetric matrix that is determined during the optimization. The pole in $\psi_x(j\omega)$ is selected through trial and error.

F_{lat} is a quadratically stable LPV system that is scheduled by the airspeed V and its derivative $\dot{V}$. To assess its performance, the nominal, unweighted closed-loop is constructed by setting all the weights to unity and Δ to zero in Figures 6 and 7. Figure 9 shows the Bode diagrams of the sensitivity functions S_a from $-\delta_a$ to e_a at each grid point in the airspeed domain along with the sensitivity bound $\bar{S}_a$. The plot indicates that F_{lat} satisfies the desired sensitivity response at each point in the domain.

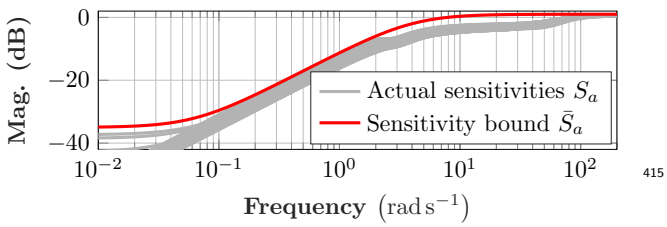

Figure 9: The Bode diagrams of the S_a and $\bar{S}_a$.

3.2.3. Longitudinal Filter F_{long}

To synthesize F_{long}, Figure 6 is used with the systems H_{long}, F_{long}, and W_{long}, and the signals δ_e, $\hat{\delta}_e$, e_e, and $\bar{e}_e$. The measurement signal is $y = q$. Figure 10 shows the generalized plant $\mathcal{F}_u(H_{long}, \Delta)$, which includes the nominal LPV short period model G_{lon}, the actuator model G_a^L, and the nominal pitch damper $K_{PD} = -0.05q$. G_{lon} is affected by the norm-bounded, LTI, multiplicative uncertainty $\|\Delta\|_\infty \leq 1$ at its input. The uncertain plant is thus given by $G_{lon}(1 + \Delta W_\Delta)$. The input δ_e to the uncertain plant is the quantity to be estimated. The feedback loop involving G_a^L, G_{lon}, and K_{PD} represents the closed-loop short period modal dynamics. This feedback loop is affected by: the disturbance $\tilde{d}_u$ at the input of G_a^L and the disturbance $\tilde{d}_y$ at the output of G_{lon}. The performance weights W_u and W_y relate the disturbances $\tilde{d}_u$ and $\tilde{d}_y$ to their respective normalized counterparts d_u and d_y. The

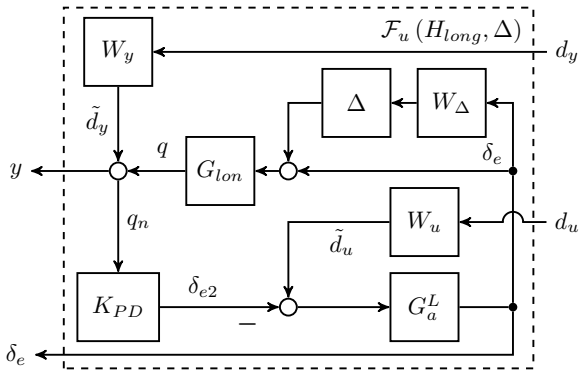

Figure 10: The generalized plant that is used for synthesizing F_{long}.

weighted, uncertain closed-loop shown in Figure 6 has the input $d = (d_u, d_y)$ and the output $\bar{e}_e$. Table 3 lists the final value of all the weights, along with their interpretations.

Table 3: The final weights selected for synthesizing F_{long}.

Weight	Final value	Weight interpretation:
W_u	$4\,(\pi/180)$	Typical elevator disturbance.
W_y	$10\,(\pi/180)$	Typical pitch rate disturbance.
W_Δ	$\frac{s^2+53.43s+213.5}{s^2+436s+427}$	Shapes the uncertainty in G_{lon} across frequency.
W_{long}	$\frac{s+14.5}{1.122s+9.149}$	Inverse of the desired sensitivity $-\delta_e \to e_e$.

The weight W_Δ shapes the uncertainty in G_{lon} across frequency. In general, the model uncertainty is low at low frequencies and high at high frequencies. However, since the phugoid mode is not accurately characterized, it is assumed that the uncertainty in G_{lon} is 50% at frequencies below the phugoid mode $(0.87\ \mathrm{rad\,s^{-1}})$, decreases to around 12% near the short period mode $(14.5\ \mathrm{rad\,s^{-1}})$,

and increases to 100% at high frequencies. Thus W_Δ is selected as shown in Figure 11, where the numbers within the parentheses specify the particular levels of uncertainty at the natural frequencies of the phugoid mode, the short period mode, and the actuator.

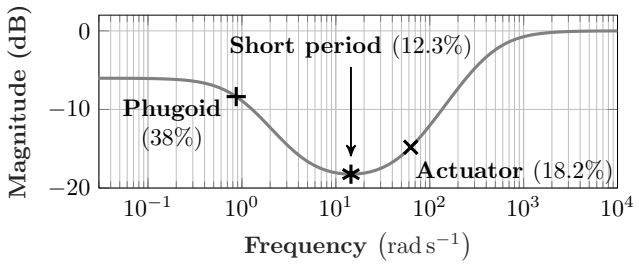

Figure 11: The Bode diagram of the weight W_Δ.

In order to select W_{long} (the filter on e_e in Figure 6), let S_e and $\bar{S}_e$ denote the sensitivity function $-\delta_e \to e_e$ and its upper bound, respectively. Overall, it is more important to minimize e_e at low frequencies as compared to high frequencies. Thus $\bar{S}_e$ is specified as a first-order transfer function with a DC gain of -4 dB, a high-frequency gain of 1 dB, and a bandwidth of 14.5 $\mathrm{rad\,s^{-1}}$. The bandwidth here refers to the -3 dB point with respect to the high frequency gain and corresponds to the natural frequency of the short period mode. W_{long} is then selected as $\bar{S}_e^{-1}$.

F_{long} is a quadratically stable LPV system that is scheduled by V and $\dot{V}$. The nominal, unweighted closed-loop is constructed by setting all the weights to unity and Δ to zero in Figures 6 and 10. Figure 12 shows the Bode diagrams of the sensitivity functions S_e from $-\delta_e$ to e_e at each grid point in the airspeed domain along with the sensitivity bound $\bar{S}_e$. The plot indicates that F_{long} satisfies the desired sensitivity response throughout the domain.

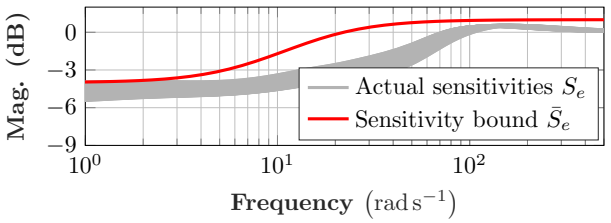

Figure 12: The Bode diagrams of S_e and $\bar{S}_e$.

3.2.4. Worst-Case Analysis

A worst-case analysis is conducted for the closed-loop formed using F_{lat}. Four filters are synthesized using the design models listed in Table 1. Each filter is then interconnected with the uncertain, LPV design model, such that the final system consists of $G_{lat}\,(1 + \Delta W_\Delta)$, K_A, G_a^L, and F_{lat}. The worst-case performance of each filter is analyzed by computing bounds on the worst-case gain from $-\delta_a$ to e_a. Recall that the channel $-\delta_a \to e_a$ quantifies the sensitivity of the estimation error to the true aileron position and is only one of the channels considered during the synthesis. Upper bounds, which account for all allowable airspeed trajectories, are computed by conducting a LPV worst-case gain analysis using Theorem 2 of Pfifer and Seiler (2016). Lower bounds, which account for constant airspeed trajectories, are computed by conducting a LTI worst-case gain analysis at each fixed airspeed in the domain and then choosing the largest such gain.

Figure 13 (left) shows the upper bound on the worst-case gain as a function of the upper bound on $\|\Delta\|_\infty$. As expected, larger uncertainty norm bounds result in larger worst-case gains across all four filter types. However, the increase in the worst-case gain is markedly less pronounced for the robust-LPV filter because its design model accounts for the airspeed variations and the model uncertainty. In particular, the upper bound on the worst-case gain for $\|\Delta\|_\infty \leq 1$ is around 2.1 for the nominal-LTI and the robust-LTI filters, around 1.6 for the nominal-LPV filter, and around 1.2 for the robust-LPV filter. The rapid performance degradations seen in the nominal-LTI and the robust-LTI filters are due to the fact that their respective design models are LTI, whereas this analysis considers all allowable airspeed trajectories.

Figure 13 (right) shows the lower bound on the worst-case gain as a function of airspeed for $\|\Delta\|_\infty \leq 1$. Each point on this plot represents a lower bound on the worst-case gain of the uncertain, LTI system at the corresponding fixed airspeed. These bounds are computed using the command `wcgain` of Matlab's Robust Control Toolbox (Balas et al., 2015a), which not only returns the lower-bound on the worst-case gain, but also the worst-case uncertainty that achieves this gain. Thus the lower bounds shown in Figure 13 are *true* lower bounds at the corresponding fixed airspeed. The largest such value across all constant airspeeds is thus a lower bound on the worst-case gain of the uncertain, LPV system. The dashed rectangles indicate the largest lower bounds and the corresponding worst-case airspeeds. The robust-LPV filter has the smallest (and least variant) lower bound across all airspeeds.

Further, the worst-case uncertainty satisfying $\|\Delta\|_\infty \leq 1$ is computed at 15.78 $\mathrm{m\,s^{-1}}$ for the robust-LPV case and at 20 $\mathrm{m\,s^{-1}}$ for the other three cases. The performances of the four filters, with Δ substituted by their respective worst-case uncertainties, are evaluated using step responses as shown in Figure 14. The responses of the nominal-LTI, the robust-LTI, and the nominal-LPV filters clearly degrade, with overshoot and/or transients, at their worst-case airspeed of 20 $\mathrm{m\,s^{-1}}$. In contrast, the response of the robust-LPV filter at its worst-case airspeed of 15.8 $\mathrm{m\,s^{-1}}$ is largely similar to its response at 20 $\mathrm{m\,s^{-1}}$, indicating that the robust-LPV filter has more consistent worst-case performance.

3.3. Method C: Multiple-Model Adaptive Estimation

The so-called Multiple Model Adaptive Estimation (MMAE) framework can also be used to detect and isolate

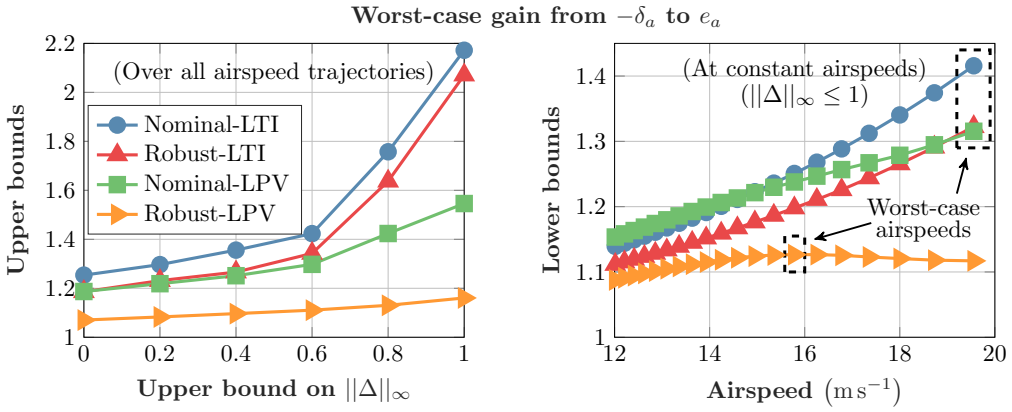

Figure 13: The upper (left) and lower (right) bounds on the worst-case gain from $-\delta_a$ to e_a.

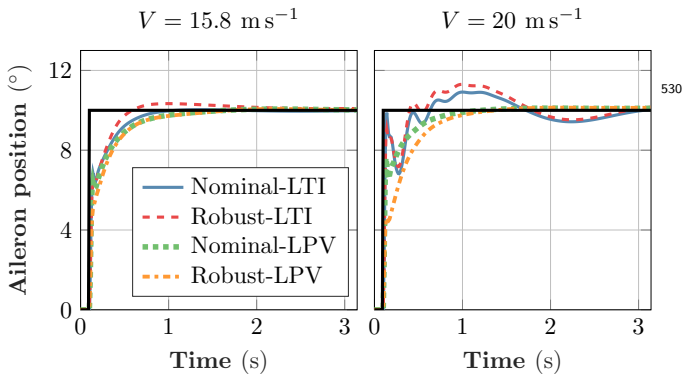

Figure 14: The method B filter responses to a 10° step change in the aileron.

the stuck fault of each of the elevons. MMAE is introduced in detail for example in Hassani et al. (2009a) and Hassani et al. (2009b) here only the basic concept is summarized. Consider an LTI plant (G) with multiple (N) different system models characterized by an i parameter (2). The different models can represent different trim points or fault states of the system. By designing LTI state observers for these models it is possible to estimate the states of the plant and the actual i parameter and so trim point or fault state. Figure 15 shows the structure of the MMAE architecture.

The fixed parameter Multiple-Input-Multiple-Output LTI system models can be characterized by the following discrete time (DT) equations:

$$\begin{aligned} \mathbf{x_i(t+1)} &= \mathbf{A_i x_i(t)} + \mathbf{B_i u(t)} + \mathbf{W_i w(t)} \\ \mathbf{y_i(t)} &= \mathbf{C_i x_i(t)} + \mathbf{D_i u(t)} + \mathbf{V_i v(t)} \end{aligned} \tag{2}$$

where $\mathbf{x_i(t)} \in \mathbb{R}^n$ denotes the state of the system, $\mathbf{u(t)} \in \mathbb{R}^m$ its control input, $\mathbf{y_i(t)} \in \mathbb{R}^p$ its measured noisy output, $\mathbf{w(t)} \in \mathbb{R}^r$ is the state noise, and $\mathbf{v(t)} \in \mathbb{R}^q$ is the measurement noise. Vectors $\mathbf{w(t)}$ and $\mathbf{v(t)}$ are assumed to be zero-mean white Gaussian sequences, mutually uncorrelated with covariances $E\left[\mathbf{w}(\tau)\mathbf{w}(\tau)^T\right] = \mathbf{Q}_\tau$ and $E\left[\mathbf{v}(\tau)\mathbf{v}(\tau)^T\right] = \mathbf{R}_\tau$. The initial condition $\mathbf{x(0)}$ of (2) is a Gaussian random vector with mean and covariance given by $E\{\mathbf{x_i(0)}\} = \mathbf{x_{i_o}}$ and $E\{\mathbf{x_i(0)x_i^T(0)}\} = \mathbf{P_i(0)}$. Matrices $\mathbf{A_i}$, $\mathbf{B_i}$, $\mathbf{W_i}$, $\mathbf{C_i}$, $\mathbf{D_i}$, and $\mathbf{V_i}$ depend on the parameter i ($i = 1 \ldots N$). $\mathbf{W_i} = I$ and $\mathbf{V_i} = I$ is assumed in this case. t and $t+1$ denote consecutive discrete time steps.

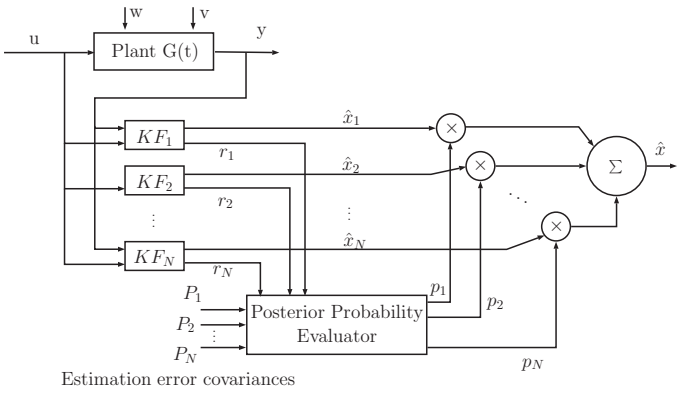

Figure 15: The MMAE architecture

One possible solution of the observer design is to obtain steady state Kalman Filters (KFs) which give state estimates $\hat{\mathbf{x}}_\mathbf{i}(\mathbf{t})$ where $i = 1 \ldots N$. As Figure 15 shows the final state estimate ($\hat{\mathbf{x}}(\mathbf{t})$) is given by (3), as the weighted sum of the $\hat{\mathbf{x}}_\mathbf{i}(\mathbf{t})$ estimates provided by the observers.

$$\hat{\mathbf{x}}(\mathbf{t}) = \sum_{i=1}^{N} p_i(t)\hat{\mathbf{x}}_\mathbf{i}(t) \tag{3}$$

The $p_i(t)$ $i = 1 \ldots N$ weights are calculated inside the Posterior Probability Evaluator (PPE) block. As Figure 15 shows this block receives the output residuals $\mathbf{r_i(t)}$ $i = 1 \ldots N$

$$\mathbf{r_i(t)} = \mathbf{y(t+1)} - \bar{\mathbf{y}}_\mathbf{i}(\mathbf{t+1|t})$$

where

$$\bar{\mathbf{y}}_\mathbf{i}(\mathbf{t+1|t}) = \mathbf{C_i}\bar{\mathbf{x}}_\mathbf{i}(\mathbf{t+1}) + \mathbf{D_i u(t+1)}$$

and the estimation error covariances $\mathbf{P_i}$ from every filter. Here $\bar{\mathbf{x}}_\mathbf{i}(\mathbf{t+1})$ is the predicted state of the i^{th} KF given

$$\bar{\mathbf{x}}_\mathbf{i}(\mathbf{t+1}) = \mathbf{A_i}\hat{\mathbf{x}}_\mathbf{i}(\mathbf{t}) + \mathbf{B_i}\mathbf{u}(\mathbf{t})$$

In Hassani et al. (2009a) the dynamic weights are calculated by the recursive formula:

$$p_i(t+1) = \frac{\beta_i e^{-E_i(t+1)}}{\sum_{j=1}^{N} p_j(t)\beta_j e^{-E_j(t+1)}} p_i(t) \tag{4}$$

where $p_i(t)$ are the a-priori model probabilities (initialized usually as $p_i(0) = 1/N$) and $E_i(t)$ and β_i are defined as

$$E_i(t+1) = \mathbf{r_i(t+1)}^\mathbf{T}\hat{\mathbf{P}}_\mathbf{i}^{-\mathbf{1}}\mathbf{r_i(t+1)} \tag{5}$$

$$\beta_i = \frac{1}{(2\pi)^{\frac{p}{2}}\sqrt{\left|\hat{\mathbf{P}}_\mathbf{i}\right|}} \tag{6}$$

where p is the dimension of $\mathbf{y(t)}$ and $\hat{\mathbf{P}}_\mathbf{i}$ is the steady state covariance matrix of residuals in i^{th} KF given by

$$\hat{\mathbf{P}}_\mathbf{i} = \mathbf{C_i}\mathbf{P_i}\mathbf{C_i^T} + \mathbf{R_i} \tag{7}$$

here $\mathbf{P_i}$ is the steady state estimation error covariance matrix of the i^{th} KF obtained from the related Riccati equation. $\beta_i e^{-E_i(t+1)}$ gives a multivariable Gaussian probability density function. In Hassani et al. (2009a) the authors prove that the conditional probability of the observer which is closest to the actual working mode of the plant will converge to one while all the other probabilities will converge to zero.

3.3.1. MMAE design models for the Vireo aircraft

In this work the set of possible elevon faults is restricted to stuck fault of one elevon keeping in mind that different fault scenarios would possibly need different fault detection strategies. The basic idea of stuck fault detection design is that the aircraft lateral dynamics should be more sensitive to elevon stuck fault than the longitudinal (note that the highest gain in the B matrix in (15) and (16) is in B_{lat} from aileron δ_a to roll rate p). Considering the linearized lateral dynamical model G_{lat} from (16) of the aircraft it includes the following states:

$$\mathbf{x} = \begin{bmatrix} v & p & r & \phi \end{bmatrix}^T$$

Though (16) includes the aileron effect as a single input, for fault detection of the left and right elevons the associated inputs should be included considering the transformation $\delta_a = \begin{bmatrix} -1/2 & 1/2 \end{bmatrix} \begin{bmatrix} \delta_l \\ \delta_r \end{bmatrix}$. This way the input of the considered model will be $u_{lat_2} = \begin{bmatrix} \delta_l & \delta_r \end{bmatrix}^T$. The possible measurable outputs of the lateral dynamics can be the roll rate p, the yaw rate r, the roll angle ϕ and the side acceleration a_y. In MMAE the KF residuals are used to drive the PPE system. Simulation experiments show that residuals of states included in the KF measurement update as measured outputs are usually small after the convergence of the filter. That's why it is advantageous not to include the p roll rate into the measured outputs but calculate and consider its residual in the selection of the best filter. Finally, the selected output vector consists of the yaw rate and the roll angle (note that the roll angle is estimated on-board, but from a fault detection point of view it can be considered as known and measurable):

$$\mathbf{y_{lat_2}} = \begin{bmatrix} r & \phi \end{bmatrix}^T \tag{8}$$

The model (presented in the Appendix in (17)) with the given state and output vectors is observable. Further issues to be handled are the system time delay and actuator dynamics.

Handling of system time delay. The overall time delay in the closed loop controlled Vireo system is about $0.05s$ (see Subsection 2.1). To design MMAE this delay is assumed to be present at the system output as a pure measurement delay. As the implementation of the estimators is in DT, in contrast with the continuous time models presented in the Appendix, there are two possibilities to model the time delay. The first is to make a Padé approximation and discretize that transfer function, the second is to add a chain of delay states to the DT model. The sampling frequency is about 92 Hz in the system, which means about $0.0109s$ sampling time. With this sampling time at least four delay state per variable should be added to approximate the measurement delay. On the contrary, a fourth or fifth degree Padé approximation is advisable to be applied. This requires four or five additional states per variable but gives only an approximation of the delay.

Finally, the chain of delay states was applied to the r yaw rate and ϕ roll angle outputs. The original and augmented state space equations of the discrete time system model are shown below:

$$\begin{aligned} \mathbf{x_{k+1}} &= \mathbf{A}\mathbf{x_k} + \mathbf{B}\mathbf{u_k} \\ \mathbf{y_{k+1}} &= \mathbf{C}\mathbf{x_{k+1}} \end{aligned} \tag{9}$$

$$\underbrace{\begin{bmatrix} \mathbf{x_{k+1}} \\ \mathbf{x^1_{k+1}} \\ \mathbf{x^2_{k+1}} \\ \mathbf{x^3_{k+1}} \\ \mathbf{x^4_{k+1}} \end{bmatrix}}_{\mathbf{x^a_{k+1}}} = \underbrace{\begin{bmatrix} \mathbf{A} & 0 & 0 & 0 & 0 \\ \begin{bmatrix} 0 & I_2 \end{bmatrix} & 0 & 0 & 0 & 0 \\ 0 & I_2 & 0 & 0 & 0 \\ 0 & 0 & I_2 & 0 & 0 \\ 0 & 0 & 0 & I_2 & 0 \end{bmatrix}}_{\mathbf{A_a}} \underbrace{\begin{bmatrix} \mathbf{x_k} \\ \mathbf{x^1_k} \\ \mathbf{x^2_k} \\ \mathbf{x^3_k} \\ \mathbf{x^4_k} \end{bmatrix}}_{\mathbf{x^a_k}} + \underbrace{\begin{bmatrix} \mathbf{B} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{B_a}} \mathbf{u_k}$$

$$\mathbf{y_{k+1}} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & I_2 \end{bmatrix}}_{\mathbf{C_a}} \underbrace{\begin{bmatrix} \mathbf{x_{k+1}} \\ \mathbf{x^1_{k+1}} \\ \mathbf{x^2_{k+1}} \\ \mathbf{x^3_{k+1}} \\ \mathbf{x^4_{k+1}} \end{bmatrix}}_{\mathbf{x^a_{k+1}}} \tag{10}$$

Here, $dim(\mathbf{x^1_k}) = dim(\mathbf{x^2_k}) = dim(\mathbf{x^3_k}) = dim(\mathbf{x^4_k}) = 2$ are the delay states for r and ϕ and I_2 is a two dimensional unit matrix. The augmented system $(\mathbf{A_a}, \mathbf{B_a}, \mathbf{C_a})$ is also observable.

Actuator dynamics. In the estimation, one can use only the commanded inputs δ_{lc}, δ_{rc} of the system because the control surface deflections are not measured on the Vireo. This means that actuator dynamics $G_a(s) = \frac{3943.84}{s^2+48.356s+3943.84}$ will cause a model mismatch. The transfer function model of the actuator dynamics $G_a(s)$ can be transformed into discrete time and included in the augmented system at the input or simply the commanded input can be 'filtered through' the DT actuator dynamics transfer function. In the first case the augmented system is not observable, so the second method is applied and so the filtered commanded inputs are the inputs of the estimators.

Models for stuck elevons. The nominal lateral model of the system is the augmented one in (10) with two inputs (δ_l left and δ_r right elevon deflections). In case of a stuck fault either the left or the right elevon goes into a fixed position. This gives the idea to use two additional lateral models with fixed left or right elevons to model the possibly faulty system. From these three models the one giving the lowest residuals shows the actual fault state of the aircraft (nominal, left elevon stuck, right elevon stuck).

The faulty system models can be formulated by considering the stuck surface as a constant state of the system and reorganizing the model matrices accordingly:

$$\underbrace{\begin{bmatrix}\mathbf{x_{k+1}}\\ \mathbf{x^1_{k+1}}\\ \mathbf{x^2_{k+1}}\\ \mathbf{x^3_{k+1}}\\ \mathbf{x^4_{k+1}}\\ u(l)\end{bmatrix}}_{\mathbf{x^F_{k+1}}} = \underbrace{\begin{bmatrix}\mathbf{A} & 0 & 0 & 0 & 0 & \mathbf{B(:,l)}\\ [0 \quad I_2] & 0 & 0 & 0 & 0 & 0\\ 0 & I_2 & 0 & 0 & 0 & 0\\ 0 & 0 & I_2 & 0 & 0 & 0\\ 0 & 0 & 0 & I_2 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}}_{\mathbf{A_{aF}}} \underbrace{\begin{bmatrix}\mathbf{x_k}\\ \mathbf{x^1_k}\\ \mathbf{x^2_k}\\ \mathbf{x^3_k}\\ \mathbf{x^4_k}\\ u(l)\end{bmatrix}}_{\mathbf{x^F_k}} + \underbrace{\begin{bmatrix}\mathbf{B(:,j)}\\ 0\\ 0\\ 0\\ 0\\ 0\end{bmatrix}}_{\mathbf{B_{aF}}} u(j)_k \tag{11}$$

$$\mathbf{y_{k+1}} = \underbrace{[0 \quad 0 \quad 0 \quad 0 \quad I_2 \quad 0]}_{\mathbf{C_{aF}}} \underbrace{\begin{bmatrix}\mathbf{x_{k+1}}\\ \mathbf{x^1_{k+1}}\\ \mathbf{x^2_{k+1}}\\ \mathbf{x^3_{k+1}}\\ \mathbf{x^4_{k+1}}\\ u(l)\end{bmatrix}}_{\mathbf{x^F_{k+1}}}$$

Here, $\mathbf{B(:,l)}$ is the *lth* column of the $\mathbf{B}$ matrix ($l \in \{1, 2\}$). j is the other column ($j \neq l$). $u(l)$ is the fixed, unknown input, while $u(j)_k$ is the time varying known one (that's why the constant $u(l)$ does not have a time index). If the estimators for the faulty models are accurate enough then the state estimate will give us also the faulty stuck position of the actuator which can be used in the reconfiguration of the system.

3.3.2. MMAE design and application

Denote by $<>$ a diagonal matrix and by $0_{i\times j}$ an $i \times j$ matrix of zeros. KFs for the nominal and the faulty (F) models were designed by assuming reasonable system and measurement noise and making discretization at 92Hz which is the frequency of data logging. The nominal system noise covariance matrix is selected as:

$$\mathbf{Q_N} =< 0.7^2 \quad (2\pi/180)^2 \quad (2\pi/180)^2 \quad (2\pi/180)^2 \quad 0_{1\times 8} >$$

for the faulty system models the additional stuck state noise is 10^{-6} making it possible to have a slowly changing stuck position (note that the stuck state dynamics is a Markov chain driven by random noise where small noise intensity makes any change slow) in the filter and so converge to the real stuck position. The covariance matrix is:

$$\mathbf{Q_F} =< 0.7^2 \quad (2\pi/180)^2 \quad (2\pi/180)^2 \quad (2\pi/180)^2 \quad 0_{1\times 8} \quad 10^{-6} >$$

The matrices show that $0.7m/s$ standard deviation is considered for the lateral velocity and $2^\circ/s$ and 2° for the angular rates and angles respectively. The measurement noise covariance matrix is:

$$\mathbf{R} =< (0.2\pi/180)^2 \quad (0.2\pi/180)^2 >$$

which shows that the tuned (by trial and error) standard deviations of the yaw rate and roll angle are $0.2^\circ/s$ and 0.2° respectively. The considered noisy state equations in the KF design were:

$$\begin{aligned}\mathbf{x^{a/F}_{k+1}} &= \mathbf{A_{a(F)}x^{a/F}_k} + \mathbf{B_{a(F)}u_k} + \mathbf{w_k}\\ \mathbf{y_{k+1}} &= \mathbf{C_{a(F)}x^{a/F}_{k+1}} + \mathbf{v_{k+1}}\end{aligned} \tag{12}$$

Here, $\mathbf{A_a}$ is the matrix of the nominal augmented system, while $\mathbf{A_{a(F)}} = \mathbf{A_{aF}}$ is the matrix of the faulty augmented system so the (F) term is included only for the faulty system models. This is the same for the B and C matrices. On the other hand $\mathbf{x^a_{..}}$ is the state of the augmented non-faulty system, while $\mathbf{x^F_{..}}$ is the state of the faulty system so either a or F superscript should be used depending on the actual system model. The designed KFs will give the predictions and estimates of the augmented state vector. Note that the actual predicted $\bar{p}_{k+1}$ roll rate is not delayed by the model system that is why it should be compared to the real value in the next step (p_{k+1}) so the estimate should be delayed with one step. All residuals were filtered through a fifth order Bessel filter with 5 rad/s bandwidth. Running the MMAE on real flight data has shown that the yaw rate (r) residuals are almost the same for the three filters even in case of a fault that's why finally only the roll rate and angle residuals are considered. However, the roll angle (ϕ) residuals are usually much smaller than the roll rate (p) ones. In the MMAE originally the residuals are scaled by the inverse of their covariance matrix in a quadratic form. However, in the present concept there is no covariance for the p part as it is not treated as a measured output. That's why simple diagonal scaling was applied for the residuals of all filter

which increases the magnitude of the roll angle parts and removes the yaw rate component:

$$RES = \begin{bmatrix} p_{meas} - \bar{p}, & r_{meas} - \bar{r}, & \phi_{meas} - \bar{\phi} \end{bmatrix} \mathbf{W} \begin{bmatrix} p_{meas} - \bar{p} \\ r_{meas} - \bar{r} \\ \phi_{meas} - \bar{\phi} \end{bmatrix}$$
$$\mathbf{W} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 36 \end{bmatrix} \tag{13}$$

These RES values are tested for threshold violation with a threshold of 0.08 to remove false alarms from noise effects in the system. If the absolute RES values of at least two filters are above the threshold then a modified PPE updates the model probabilities in the following way:

$$p_i(t+1) = \frac{e^{-0.04 \cdot RES_i(t+1)}}{\Sigma_{j=1}^{3} e^{-0.04 \cdot RES_j(t+1)} p_j(t)} p_i(t) \tag{14}$$

The sum of these probabilities is guaranteed to be 1 and the largest one should show the actual valid model. Assuming that the system is fault free at initialization the initial probabilities are selected as $p_{No} = 0.98, p_l = p_r = 0.01$ in contrast to the default rule presented as $p_i = \frac{1}{N}$. Here, No is the probability of the nominal working mode, l and r are the probabilities for the left and right stuck modes. None of the initial values should be zero as in this case they will remain zero all the time (see (14)) that's why the lower limits are set as 0.01.

Finally, the model which probability is above 50% (0.5) is selected as valid in every time step. The left and right stuck filters are able to also approximately estimate the stuck position of the surface as will be shown in the next section.

After introducing all three methods their comparison through off-line application to real flight data is presented as follows.

4. Comparison Using Flight Data

The three FDI methods presented in the previous section are validated and compared using prerecorded flight data, i.e. in an offline setting. The FDI algorithms operate in open-loop until they detect a fault and reconfigure the controller. Thus evaluating the FDI algorithms offline is equivalent to evaluating them online. Two flights are conducted during which the UAS tracks a rectangular flight path at constant altitude. Both flights consist of a number of test points wherein stuck faults are injected in the right elevon. To avoid over-speeding, only trailing edge up (negative) stuck faults are injected. Table 4 lists the fault magnitude in each test point along with the detection times obtained using the three methods.

Table 4: The faults injected in each test point of the two flights along with the detection times obtained using the three methods.

Test point	Fault	Detection time (s) A	B	C
FLT32				
T1	−5°	8.51	6.93	12
T2	−5°	3.56	3.03	5
T3	−5°	–	–	7.8
T4	−5°	12.9	8.26	14.7
T5	−5°	2.54	1.89	4
FLT33				
T1	−5°	8.78	8.23	13.5
T2	−5°	5.34	5.06	7.34
T3	−5°	5.22	4.5	7
T4	−6°	2.78	1.84	7.27
T5	−7°	2.48	1.57	3
T6	−7°	2.20	1.26	2.7

Figures 16 and 17 show the residuals generated in all the test points, except FLT32-T3, using Methods A and B, respectively. The timestamps of all the test points are shifted such that zero seconds corresponds to the fault injection time (vertical line). The figures indicate that the residuals are contained within the thresholds (dashed lines) prior to the fault and cross the thresholds shortly after the fault. There are no false alarms or missed detections in any of the test points. Although the fault magnitudes are similar in all the test points (see Table 4), the detection time varies depending on the maneuver being performed immediately prior to the fault. In particular, recall that the UAS tracks a rectangular flight path at constant altitude in both the flights. The aircraft flies straight and level when tracking the four sides of this rectangle and banks sharply when navigating the corners. It takes longer to detect a fault that is injected during the straight and level segments as compared to one that is injected during the banked turns.

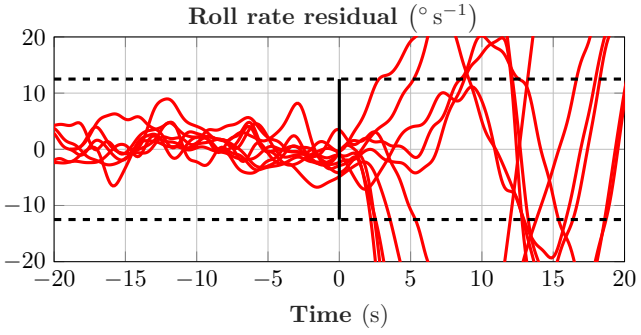

Figure 16: The residuals generated using Method A.

Furthermore, Method A only performs fault detection since it generates only one residual. Method B, on the

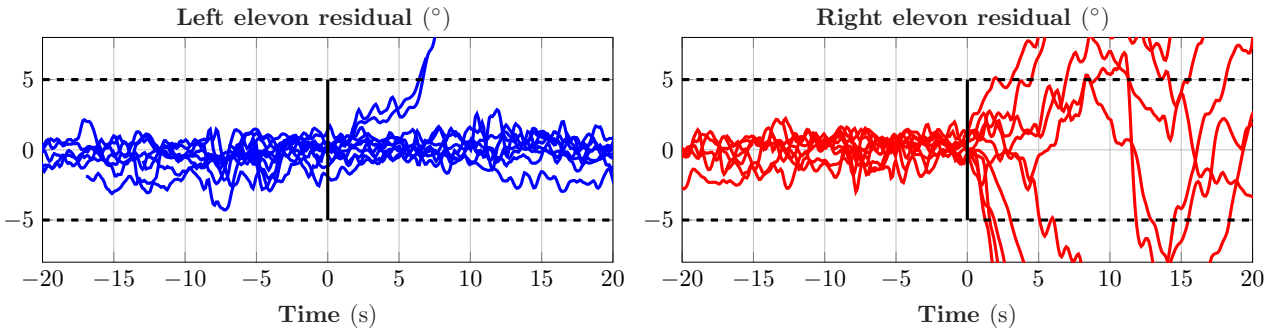

Figure 17: The residuals generated using Method B.

other hand, generates two residuals and thus also isolates the fault. Given that the fault is injected in the right elevon, Figure 17 indicates that the left elevon residual does not cross the thresholds in all except two test points. However, these two threshold crossings occur after the corresponding right elevon residual has already crossed the threshold. Therefore, the fault is correctly isolated in all the test points.

Figures 18 to 21 show the model probabilities and selected models before and after the fault occurrence for flight 32 and 33 data in case of method C. The timestamps of all the test points are again shifted such that zero seconds corresponds to the fault injection time (vertical line). The right stuck fault was successfully detected by the MMAE algorithm even in FLT32-T3 case. The decrease of nominal model probability under 50% and the increase of right stuck probability above 50% can be well seen in Figures 18 and 20 the crossing of probabilities also shows approximately the times of fault detection where the selected model number changes in Figures 19 and 21. There is no false left stuck detection in any of the cases as the highest left stuck probability is only 30% in Figure 20.

Considering Table 4 method C is always slower in the detection of the fault. It is 0.5-4.7s slower than method A and 1.4-6.4s slower than method B. This can be the price of detecting every fault successfully as usually there is a design trade-off between detection time and false alarms.

An advantage of method C can be the determination of the approximate position of the stuck surface which can be useful in the reconfiguration control as pointed out in Bauer et al. (2018) and Venkataraman (2018). This is shown in Figures 22 and 23. For FLT 32 the real stuck positions are $5-5.2°$ while the estimated ones are $5.2-7.7°$ which means an estimation error of $0.2-2.7°$. For FLT 33 T1-T4 the real stuck deflections are $5.2-5.7°$ and the estimation errors are $1-2.8°$. For cases T5-T6 the real values are $6.8°$ and the estimates diverge with maximum $3.2-5.2°$ errors but it should be considered that the time lengths of fault presence are very short in these cases. It would require new flight tests to test if this is always the case for larger stuck positions even with fault presence for longer time.

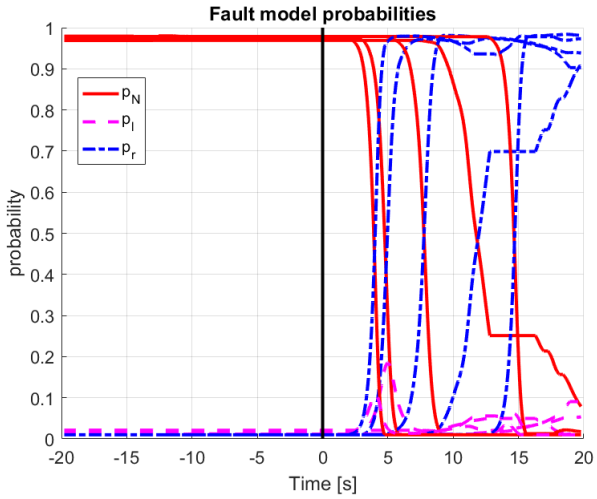

Figure 18: Fault probabilities for FLT 32 data with Method C

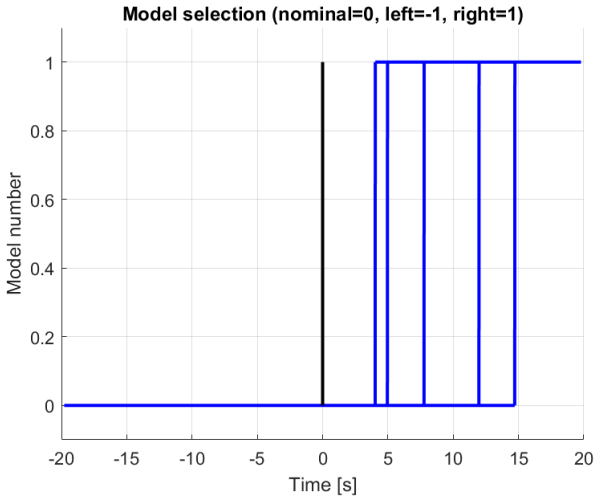

Figure 19: Selected model for FLT 32 data with Method C

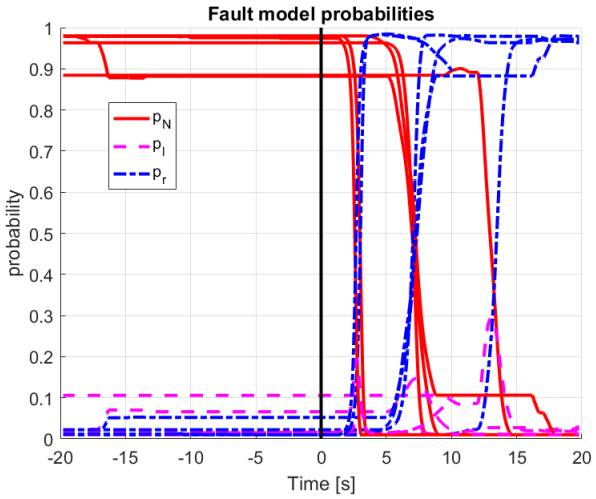

Figure 20: Fault probabilities for FLT 33 data with Method C

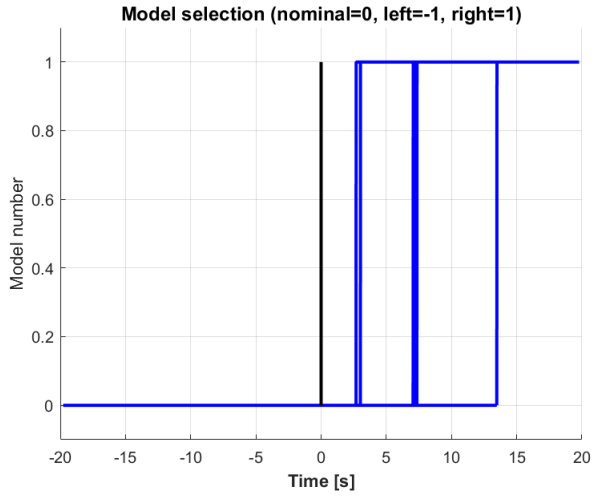

Figure 21: Selected model for FLT 33 data with Method C

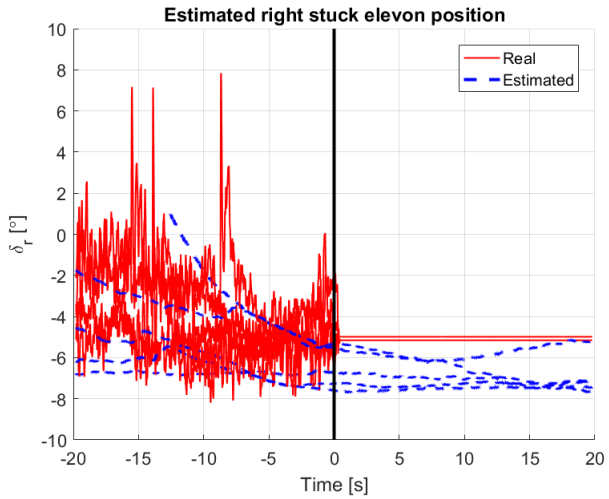

Figure 22: Real and estimated right stuck elevon position for FLT 32 data with Method C

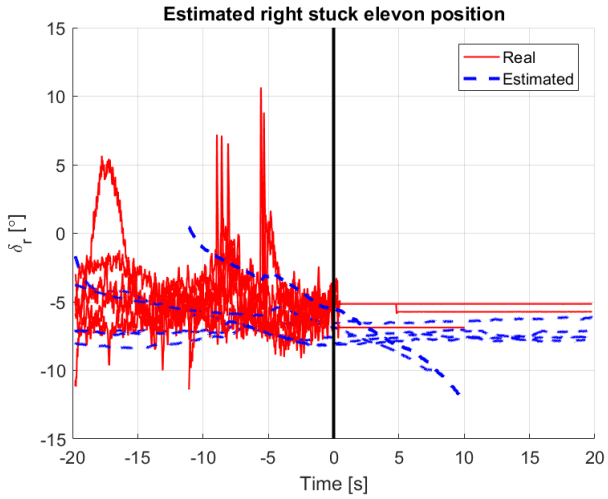

Figure 23: Real and estimated right stuck elevon position for FLT 33 data with Method C

Another way to compare the different filters is to consider their state dimension which also gives a hint for onboard runtime. Model A has 6 states, model B has 32 states (15 states lateral, 13 states longitudinal, 2x2 states actuator model) and model C has 43 states (13 states by filter (3x13=39) and 2x2 states actuator model). So model A has the lowest state dimension and so it would have the shortest runtime, the second is model B and then model C. On the other hand model A is only able to show that there is a fault, model B shows also in which elevon it is and model C also estimates the approximate stuck position. Down sampling method C to 46Hz can result in only 31 states as it decreases the required number of delay states so this is a possibility to make it faster. Summarizing, selection of the best filter depends on several aspects.

5. Conclusions

This paper presented three candidate methods for the diagnosis of stuck elevon faults on a small unmanned aircraft. The first (method A) is a baseline method that estimates the residual of the roll rate by using a transfer function aircraft model and the commanded aileron deflection. The second (method B) is a robust LPV method that estimates the left and right elevon deflections and compares them to the commanded values. The third (method C) is a Multiple Model Adaptive Estimator including a nominal, a left stuck and a right stuck system model and selecting between them based-on the residuals of some measured parameters.

The three methods were compared using flight data recorded on the unmanned aircraft. Method A has the fewest states (6) and method C has the most states (43). However, the order of method C can be decreased to 31 which is close to the 32 states of method B. With regard to the fault detection time, method B is the fastest and

method C is the slowest. Further, method C has the advantage of no missed detections whereas methods A and B missed the fault in one of the test points. Finally, method A can detect the presence of the fault, method B can isolate whether it is the left or the right elevon that is stuck, and method C can also estimate the fault magnitude.

The selection of a particular method should be based on the application requirements, such as runtime, fault detection time, isolation requirement, and the requirement to know the fault magnitude. Method B should be used if it is sufficient to detect and isolate the fault. Method C should be used if the reconfigured flight controller also requires an estimate of the fault magnitude, and the performance deterioration from the flight mission after the fault is slow. Since method A only detects the fault, it may be used with a simple aircraft recovery system like a safety parachute. Finally, additional flight tests with other fault modes will be considered as future work. This will enable the comparisons presented in this paper to be extended to a broader class of faults.

Appendix

The matrices listed here consider all the signals in SI units. The state-space matrices of G_{lon} are (with state $\mathbf{x_{lon}} = [q, w]^T$, input $\mathbf{u_{lon}} = \delta_e$, and output $\mathbf{y_{lon}} = q$):

$$\mathbf{A_{lon}} = \begin{bmatrix} -5.487 & -12.86 \\ 13.85 & -5.688 \end{bmatrix}, \ \mathbf{B_{lon}} = \begin{bmatrix} -185.7 \\ -24.52 \end{bmatrix},$$

$$\mathbf{C_{lon}} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \text{ and } \mathbf{D_{lon}} = 0. \tag{15}$$

The state-space matrices of G_{lat} are (with state $\mathbf{x_{lat}} = [v, p, r, \phi]^T$, input $\mathbf{u_{lat}} = \delta_a$, and output $\mathbf{y_{lat}} = [\phi, p, r]^T$):

$$\mathbf{A_{lat}} = \begin{bmatrix} -0.574 & 1.12 & -15.3 & 9.78 \\ -3.99 & -11.3 & 2.5 & 0 \\ 0.311 & -1.49 & -0.944 & 0 \\ 0 & 1 & 0.0683 & 0 \end{bmatrix}, \ \mathbf{B_{lat}} = \begin{bmatrix} -0.49 \\ -201 \\ -9.61 \\ 0 \end{bmatrix},$$

$$\mathbf{C_{lat}} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \ \mathbf{D_{lat}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{16}$$

The reformulated G_{lat_2} model for MMAE design with state $\mathbf{x_{lat}} = [v, p, r, \phi]^T$, input $\mathbf{u_{lat_2}} = [\delta_l, \delta_r]^T$, and output $\mathbf{y_{lat_2}} = [r, \phi]^T$):

$$\mathbf{A_{lat_2}} = \begin{bmatrix} -0.574 & 1.12 & -15.3 & 9.78 \\ -3.99 & -11.3 & 2.5 & 0 \\ 0.311 & -1.49 & -0.944 & 0 \\ 0 & 1 & 0.0683 & 0 \end{bmatrix}, \ \mathbf{B_{lat_2}} = \begin{bmatrix} 0.245 & -0.245 \\ 100.5 & -100.5 \\ 4.805 & -4.805 \\ 0 & 0 \end{bmatrix},$$

$$\mathbf{C_{lat_2}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \ \mathbf{D_{lat_2}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{17}$$

The delay in $G_a e^{-\tau_f s}$ is first replaced with a fifth-order Padé approximation. A balanced residualization of two states yields G_a^L with the state-space realization:

$$\mathbf{A_a} = \begin{bmatrix} -2.409 & 26.09 & 7.284 & -7.204 & 12.42 \\ -26.09 & -7.319 & -49.17 & 8.948 & -32.95 \\ 7.284 & 49.17 & -26.57 & 60.33 & -56.4 \\ 7.204 & 8.948 & -60.33 & -15.56 & 150.4 \\ 12.42 & 32.95 & -56.4 & -150.4 & -184.1 \end{bmatrix}, \ \mathbf{B_a} = \begin{bmatrix} -2.097 \\ -3.114 \\ 4.485 \\ 2.465 \\ 5.982 \end{bmatrix},$$

$$\mathbf{C_a} = \begin{bmatrix} -2.097 & 3.114 & 4.485 & -2.465 & 5.982 \end{bmatrix}, \ \mathbf{D_a} = -0.06135. \tag{18}$$

The reader is referred to Venkataraman (2018) for additional details on the aircraft model.

Acknowledgments

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 690811 and the Japan New Energy and Industrial Technology Development Organization under grant agreement No. 062600 as a part of the EU/Japan joint research project entitled 'Validation of Integrated Safety-enhanced Intelligent flight cONtrol (VISION)'. This research was also funded by the National Science Foundation under Grant No. NSF/CNS-1329390 entitled "CPS: Breakthrough: Collaborative Research: Managing Uncertainty in the Design of Safety-Critical Aviation Systems". The first author acknowledges financial support from the University of Minnesota through the 2017-2018 Doctoral Dissertation Fellowship. The authors thank the following individuals: T. Colten of Sentera for donating the Vireo aircraft; C. Olson for the flight software, controller implementation, autoland trajectory, head-up display, and piloting; N. Carter, R. Condron, L. Heide, A. Mahon, C. Regan, and B. Taylor for aircraft integration and testing.

References

ADDSAFE, 2012. Advanced fault diagnosis for sustainable flight guidance and control, European 7th Framework Program. http://addsafe.deimos-space.com.

Alwi, H., Edwards, C., 2008. Fault detection and fault-tolerant control of a civil aircraft using a sliding-mode-based scheme. IEEE Transactions on Control Systems Technology 16, 499–510.

Amos, J., Bergquist, E., Cole, J., Phillips, J., Reimann, S., Shuster, S., 2013. UAV for Reliability. Technical Report. University of Minnesota.

Balas, G., Chiang, R., Packard, A., Safonov, M., 2015a. Robust Control Toolbox: Users Guide. Technical Report. The MathWorks Inc.

Balas, G., Hjartarson, A., Packard, A., Seiler, P., 2015b. LPVTools: A Toolbox for Modeling, Analysis, and Synthesis of Parameter Varying Control Systems. Technical Report. Musyn, Inc.

Bateman, F., Noura, H., Ouladsine, M., 2007. Actuators fault diagnosis and tolerant control for an unmanned aerial vehicle, in: IEEE International Conference on Control Applications, pp. 1061–1066. doi:10.1109/CCA.2007.4389374.

Bauer, P., Venkataraman, R., Vanek, B., Seiler, P., Bokor, J., 2018. Fault detection and basic in-flight reconfiguration of a small UAV equipped with elevons, in: 10th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS.

Bokor, J., Seiler, P., Vanek, B., Balas, G., 2010. Robust model matching for geometric fault detection filters: A commercial aircraft example, in: 8th European Workshop on Advanced Control and Diagnosis, pp. 223–228.

Chen, J., Patton, R.J., 1999. Robust Model-Based Fault Diagnosis for Dynamic Systems. Springer, Boston, MA.

Collinson, R., 2003. Introduction to Avionics Systems. 2nd ed., Springer.

Cook, M.V., 2007. Flight Dynamics Principles. Second ed., Elsevier.

Cristofaro, A., Johansen, T.A., Aguiar, A.P., 2017. Icing detection and identification for unmanned aerial vehicles using adaptive nested multiple models. International Journal of Adaptive Control and Signal Processing 31, 1584–1607. doi:10.1002/acs.2787.

Ding, S.X., 2013. Model-Based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools. Springer-Verlag.

Ding, S.X., 2014. Data-driven Design of Fault Diagnosis and Fault-tolerant Control Systems. Springer-Verlag.

Edelmayer, A., Bokor, J., Keviczky, L., 1994. An H_∞ filtering approach to robust detection of failures in dynamical systems, in: IEEE Conference on Decision and Control, pp. 3037–3039. doi:10.1109/CDC.1994.411321.

Edwards, C., Spurgeon, S.K., Patton, R.J., 2000. Sliding mode observers for fault detection and isolation. Automatica 36, 541–553.

Freeman, P., 2014. Reliability Assessment for Low-cost Unmanned Aerial Vehicles. Ph.D. thesis. University of Minnesota.

Freeman, P., Balas, G., 2013. Analytical fault detection for a small UAV, in: AIAA Infotech@Aerospace Conference, Boston, MA, USA.

Freeman, P., Pandita, R., Srivastava, N., Balas, G., 2013a. Model-based and data-driven fault detection performance for a small UAV. IEEE Transactions on Mechatronics 18, 1300–1309.

Freeman, P., Seiler, P., Balas, G., 2011. Robust fault detection for commercial transport air data probes, in: IFAC Proceedings Volumes, pp. 13723–13728.

Freeman, P., Seiler, P., Balas, G., 2013b. Air data system fault modeling and detection. Control Engineering Practice 21, 1290–1301.

Gertler, 1997. Fault detection and isolation using parity relations. Control Engineering Practice 5, 653–661.

Gertler, J.J., 1998. Fault Detection and Diagnosis in Engineering Systems. First ed., Routledge.

Gertler, J.J., 2014. Fault Detection and Diagnosis. John Wiley & Sons. URL: http://dx.doi.org/10.1002/9781118445112.stat03644, doi:10.1002/9781118445112.stat03644.

Goupil, P., 2010. Oscillatory failure case detection in the A380 electrical flight control system by analytical redundancy. Control Engineering Practice 18, 1110–1119.

Goupil, P., 2011. AIRBUS state of the art and practices on FDI and FTC in flight control system. Control Engineering Practice 19, 524 – 539.

Goupil, P., Marcos, A., 2014. The European ADDSAFE project: Industrial and academic efforts towards advanced fault diagnosis. Control Engineering Practice 31, 109–125.

Hassani, V., Aguiar, A.P., Athans, M., Pascoal, A.M., 2009a. Multiple model adaptive estimation and model identification using a minimum energy criterion, in: American Control Conference (ACC) 2009.

Hassani, V., Aguiar, A.P., Pascoal, A.M., Athans, M., 2009b. A performance based modelset-design strategy for multiple model adaptive estimation, in: European Control Conference (ECC) 2009.

Hecker, S., Varga, A., Ossmann, D., 2011. Diagnosis of actuator faults using LPV-gain scheduling techniques, in: AIAA Guidance, Navigation, and Control Conference. AIAA Paper 2011-6680.

Hjartarson, A., Packard, A., Seiler, P., 2015. LPVTools: A Toolbox for Modeling, Analysis, and Synthesis of Parameter Varying Control Systems, in: First IFAC Workshop on Linear Parameter Varying Systems.

Hwang, I., Kim, S., Kim, Y., Seah, C.E., 2010. A survey of fault detection, isolation, and reconfiguration methods. IEEE Transactions on Control Systems Technology 18, 636–653.

Isermann, R., 1984. Process fault detection based on modeling and estimation methods - a survey. Automatica 20, 387–404.

Isermann, R., 2005. Model-based fault-detection and diagnosis status and applications. Annual Reviews in Control 29, 71–85.

Isermann, R., 2006. Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance. Springer, Germany.

Isermann, R., Ballé, P., 1997. Trends in the application of model-based fault detection and diagnosis of technical processes. Control Engineering Practice 5, 709–719.

Marcos, A., Ganguli, S., Balas, G.J., 2005. An application of H_∞ fault detection and isolation to a transport aircraft. Control Engineering Practice 13, 105–119.

Megretski, A., Rantzer, A., 1997. System analysis via integral quadratic constraints. IEEE Transactions on Automatic Control 42, 819–830.

Napolitano, M.R., Chen, C.I., Naylor, S., 1993. Aircraft failure detection and identification using neural networks. Journal of Guidance, Control, and Dynamics 16, 999–1009.

Nelson, R.C., 1998. Flight Stability and Automatic Control. McGraw-Hill.

Pandita, R., 2010. Dynamic flight envelope assessment with flight safety applications. Ph.D. thesis. University of Minnesota.

Pandita, R., Bokor, J., Balas, G., 2011. Performance metrics for fault detection and isolation filters, in: American Control Conference, pp. 1390–1395.

Pandita, R., Bokor, J., Balas, G., 2013. Closed-loop performance metrics for fault detection and isolation filter and controller interaction. International Journal of Robust and Nonlinear Control 23, 419–438. doi:10.1002/rnc.1838.

Patton, R.J., Chen, J., 1994. A review of parity space approaches to fault diagnosis for aerospace systems. Journal of Guidance, Control, and Dynamics 17, 278–285.

Pfifer, H., Seiler, P., 2016. Less conservative robustness analysis of linear parameter varying systems using integral quadratic constraints. International Journal of Robust and Nonlinear Control 26, 3580–3594.

Rago, C., Prasanth, R., Mehra, R.K., Fortenbaugh, R., 1998. Failure detection and identification and fault tolerant control using the IMM-KF with applications to the Eagle-Eye UAV, in: IEEE Conference on Decision and Control, pp. 4208–4213. doi:10.1109/CDC.1998.761963.

Rotondo, D., Cristofaro, A., Hassani, V., Johansen, T.A., 2017. Icing diagnosis in unmanned aerial vehicles using an lpv multiple model estimator. IFAC-PapersOnLine 50, 5238–5243. doi:10.1016/j.ifacol.2017.08.462.

Rotstein, H., Ingvalson, R., Keviczky, T., Balas, G., 2006. Fault-detection design for uninhabited aerial vehicles. Journal of Guidance, Control, and Dynamics 29, 1051–1060.

Seiler, P., Bokor, J., Vanek, B., Balas, G., 2011. Robust model matching for geometric fault detection filters, in: American Control Conference.

Stoustrup, J., Grimble, M.J., Niemann, H., 1997. Design of integrated systems for the control and detection of actuator/sensor faults. Sensor Review .

Vanek, B., Edelmayer, A., Szabo, Z., Bokor, J., 2014. Bridging the gap between theory and practice in LPV fault detection for flight control actuators. Control Engineering Practice 31, 171–182.

Vanek, B., Seiler, P., Bokor, J., Balas, G., 2011a. Robust fault detection filter design for commercial aircraft, in: European Guidance Navigation and Control Conference.

Vanek, B., Seiler, P., Bokor, J., Balas, G., 2011b. Robust model matching for geometric fault detection filters: A commercial aircraft example, in: IFAC World Congress.

Varga, A., Ossmann, D., 2014. LPV model-based robust diagnosis of flight actuator faults. Control Engineering Practice 31, 135–147. doi:10.1016/j.conengprac.2013.11.004.

Venkataraman, R., 2018. Fault-Tolerant Flight Control Using One Aerodynamic Control Surface. Ph.D. thesis. University of Minnesota.

Venkataraman, R., Seiler, P., 2018. Convex LPV synthesis of estimators and feedforwards using duality and integral quadratic constraints. International Journal of Robust and Nonlinear Control 28, 953–975. URL: http://dx.doi.org/10.1002/rnc.3913, doi:10.1002/rnc.3913.

Venkataraman, R., Seiler, P., Lukátsi, M., Vanek, B., 2017. Reliability assessment of actuator architectures for unmanned aircraft. Journal of Aircraft 54, 955–966.

Wilborn, J.E., Foster, J.V., 2004. Defining commercial transport loss-of-control: A quantitative approach, in: AIAA Atmospheric Flight Mechanics Conference.