Model-Based Detection and Isolation of Rudder Faults for a Small UAS

Raghu Venkataraman* and Peter Seiler†
Department of Aerospace Engineering & Mechanics
University of Minnesota, Minneapolis, MN, 55455, USA

New regulatory safety standards will soon require unmanned aircraft systems to meet high levels of reliability. There is potential to increase the reliability of such systems without necessarily increasing the number of hardware components. This paper motivates a mix of physical and analytical redundancy in order to increase the system-level reliability of a small unmanned aircraft. The aircraft discussed in this paper has a split rudder for fault-tolerant control. Hardware faults, such as a stuck rudder, need to be detected and isolated in real-time in order for the controller to be reconfigured. In this paper, flight dynamics principles are used to design a model-based filter for detecting and isolating stuck faults in the split rudder of the aircraft. A classical controller is developed in order to make the aircraft robust to stuck rudder faults. The performance and robustness of the filter is evaluated, in closed-loop, through high fidelity simulations. The results in this paper highlight the potential for increasing the reliability of safety-critical aviation systems through analytical redundancy.

Nomenclature

V	Airspeed [m/s]
h	Altitude [m]
α	Angle of attack [deg]
β	Angle of sideslip [deg]
ϕ	Roll attitude [deg]
θ	Pitch attitude [deg]
ψ	Heading angle [deg]
p	Roll rate [deg/s]
q	Pitch rate [deg/s]
r	Yaw rate [deg/s]
τ	Throttle setting [unitless]
δ^t_{rud}	Deflection of top rudder [deg]
δ^b_{rud}	Deflection of bottom rudder [deg]
δ^r_{ail}	Deflection of right aileron [deg]
δ^l_{ail}	Deflection of left aileron [deg]
δ_{ele}	Deflection of elevator [deg]

Subscripts

FDI	Denotes a simulated signal within the FDI algorithm.
cmd	Denotes a commanded signal.
$real$	Denotes a real signal or system.
ref	Denotes a reference signal.
$trim$	Denotes the trim value of a signal.

*Graduate Student, venka085@umn.edu
†Assistant Professor, seile017@umn.edu

I. Introduction

This paper describes a model-based fault detection and isolation (FDI) filter designed to detect rudder faults on a small unmanned aircraft system (UAS). Recently, UASs have found increasing civilian applications, such as law enforcement, search & rescue, and precision agriculture. While UASs are projected to operate increasingly in airspace typically reserved for manned aircraft, their current reliability metrics do not meet the certification standards set by the Federal Aviation Administration (FAA) for manned aircraft. In 2012, the United States Congress passed H.R.658 [1] - the FAA Modernization and Reform Act - in order to facilitate the safe integration of UASs into the national airspace. In particular, section 332 of H.R.658 mandates the FAA to "provide for the safe integration of civil unmanned aircraft systems into the national airspace system as soon as practicable, but not later than September 30, 2015." While the FAA works on creating new certification standards to include the operation of UASs in the national airspace, aircraft designers will need to work towards increasing their reliability. Model-based detection and isolation of faults has the potential to increase the system-level reliability of UASs while operating within the limits of their typical design constraints.

To put this challenge in perspective, consider the current safety standards set by the FAA for manned commercial aircraft: in order for a commercial aircraft to be certified, there should be no more than one catastrophic failure per one billion hours of flight operation. Airframe manufacturers, such as Boeing, meet the 10^{-9} failures-per-flight-hour standard by utilizing hardware redundancy in their designs. For example, the Boeing 777 has 14 spoilers each with its own actuator; two actuators each for the outboard ailerons, left & right elevators, and flaperons; and three actuators for the single rudder [2]. In addition, the computing platform, electrical and hydraulic power lines, and communication paths have triple layer redundancy. On the other hand, most civil UASs have reliabilities that are orders of magnitude below the 10^{-9} level required for manned commercial aircraft. For instance, the UAV Research Group at the University of Minnesota (UMN) [3] operates an *Ultra Stick 120* aircraft (described further in section II.A) with single-string, off-the-shelf components. A comprehensive fault tree analysis yielded a failure rate of 2.2×10^{-2} failures-per-flight-hour [a] for this aircraft [4]. UASs have such low reliability because most, if not all, of their on-board components are single-string, i.e. there are single points of failure on the UAS that can lead to a system-level catastrophic failure. Hardware redundancy is required to improve UAS reliability but must be used judiciously due to design constraints on size, weight, and power. Methods that provide analytical redundancy, such as the FDI filter discussed in this paper, have the potential to bridge the gap between commercial aircraft, that almost entirely use hardware redundancy, and current UASs, that are almost entirely single-string designs.

Some new commercial aircraft, such as the Airbus A380, come equipped with a limited degree of analytical redundancy [5]. For example, a model-based fault detection algorithm is used to detect oscillatory failure modes in the electrical flight control system of the A380 [6]. In addition to model-based fault detection techniques, several data-driven approaches exist. Detailed descriptions of the various model-based and data-driven fault detection methods can be found in existing literature [7–10]. The performance of model-based and data-driven fault detection algorithms are compared in [11, 12]. A detailed survey of various fault detection, isolation, and reconfiguration methods is presented in [13]. In addition, the performance of an FDI filter depends on whether the system is in closed-loop or open-loop control. Signal-based methods are applied to synthesize FDI filters and their performance is analyzed under closed-loop control in [14].

It is worth emphasizing that analytical redundancy is not a panacea for increasing the reliability of UASs. After a fault has been detected and isolated, there is still a need to reconfigure the controller in order to prevent loss of aircraft (LOA). Often, a successful reconfiguration can only be achieved with hardware redundancy. For example, if a stuck control surface on a UAS would normally lead to LOA, no degree of analytical redundancy can change that outcome. An attempt is made in this paper to reach a middle ground by including both hardware and analytical redundancy on a small UAS. Specifically, hardware redundancy is provided by splitting the rudder of the UAS into two pieces in order to ensure some limited yaw control authority even if there is a fault in one of the rudders. Analytical redundancy is provided through a model-based FDI filter that detects and isolates faults in the split rudder. The experimental platform, the simulation environment used to evaluate the FDI filter, and the flight control law of the UAS are described in section II. In contrast to the some of the literature reviewed above, a physics-based approach is followed in order to characterize the fault modes and their effects. Although more advanced signal-based methods exist for

[a] This analysis provides a theoretical estimate of the reliability and no loss of aircraft has occurred to date.

synthesizing FDI filters, understanding the physics of the fault is critical in order to effectively apply the more advanced methods. In particular, the principles of flight dynamics [15, 16] are used to understand rudder faults and are used as guidelines to architect the FDI filter in section III. Finally, the performance and robustness of the FDI filter is assessed, in simulation, in section IV.

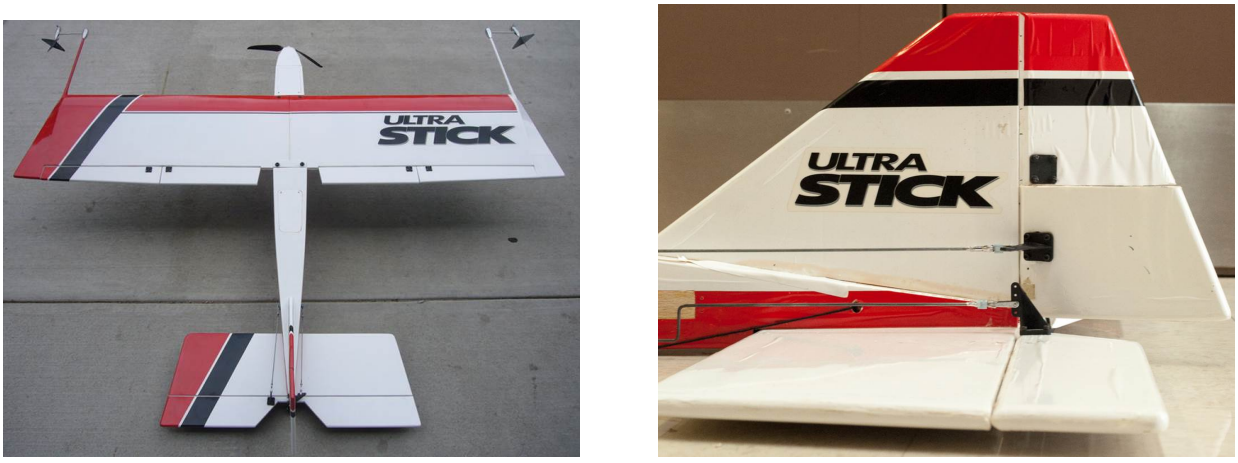
II. Infrastructure for Simulation and Flight Tests

II.A. Experimental Platform

The airframe is a commercial, off-the-shelf, radio-controlled aircraft called the Ultra Stick 120 [17], shown in Figure 1(a). The Ultra Stick 120 has a wingspan of 1.92 m and a mass of about 7.4 kg. The UMN UAV Research Group has retrofitted the airframe with custom avionics [3, 18, 19] for enabling research in the areas of real-time control, guidance, navigation, and fault detection. The avionics include a sensor suite, a flight control computer, and a telemetry radio. The airframe comes equipped with the standard suite of aerodynamic control surfaces - flaps, ailerons, elevator, and rudder - each actuated by its own servo motor.

A comprehensive reliability analysis was performed to identify the critical components on the Ultra Stick 120 [4]. In particular, two standard reliability analyses were performed: fault tree analysis (FTA) and failure modes & effects analysis (FMEA). These analyses identified the most critical components on the aircraft that should be supplemented with hardware redundancy. Through simulation, it was concluded that a stuck rudder and/or a stuck elevator would result either in a loss of mission (LOM) or LOA depending on the fault level, airspeed, and altitude. In order to mitigate the degradation in performance during LOM and prevent LOA, the airframe was modified by splitting the rudder and elevator into two parts, each actuated by its own servo motor [20]. It was reasoned that if one of the two rudders got stuck in flight, the other rudder would be able to provide some limited yaw control authority, thereby allowing for the reconfiguration of the surfaces and effectively increasing the reliability of the airframe. A similar reasoning can be made for the split elevator. The split rudder is shown in Figure 1(b). The rudder was split in such a way that the top and bottom pieces have equal side force and yawing moment derivatives.

Including the split tail surfaces, this aircraft has a total of eight aerodynamic control surfaces. While each surface is independently actuated, the flight software allows for them to be coupled symmetrically (such as the elevators) or anti-symmetrically (such as the ailerons). In addition, these redundant surfaces allow for the testing and validation of reconfigurable control laws after a fault has been detected in the surfaces. From an infrastructure standpoint, this aircraft serves as the test platform for all the safety-critical reliability research that is being undertaken by the UMN UAV Research Group. The focus of this paper is restricted to the detection and isolation of stuck faults in either of the rudders. Consequently, the commanded maneuvers, controller outputs, and plant outputs considered in this paper were chosen based on their effect on the lateral-directional aircraft dynamics.



(a) Baseline Ultra Stick 120 (b) Modified aircraft with split rudder

Figure 1: The baseline and modified Ultra Stick 120 aircraft.

II.B. Simulation Environment

The UMN UAV Research Group has developed a high-fidelity simulation environment for the Ultra Stick 120 with extensive documentation [3]. This simulation environment was built using Matlab/Simulink and contains models for the aircraft subsystems. The rigid body dynamics are implemented using the standard six degree-of-freedom, nonlinear aircraft equations of motion [21]. The aerodynamic stability and control derivatives were identified from wind tunnel experiments [22, 23]. The simulation models the forces & moments and the propwash generated by the electric motor and propeller pair. The simulation also includes first-order, rate and position limited actuator models for the servo motors. The sensor models for the inertial measurement unit, air data probes, and magnetometer include band-limited white noise for each measurement. The simulation environment also contains subsystems that model environmental effects, such as wind gusts, atmospheric turbulence, and the Earth's gravitational & magnetic fields. In particular, the Discrete Wind Gust Model and the Discrete Dryden Wind Turbulence Model are added from Matlab's Aerospace Blockset. Finally, closed-loop flight control laws and navigation & guidance filters are also included.

The nonlinear aircraft model can be trimmed and linearized at any flight condition within the flight envelope of the aircraft. The simulation environment and the flight control computer allow for extensive software-in-the-loop and hardware-in-the-loop simulations of the aircraft model. The entire simulation environment, details about the aircraft fleet, components, wiring, and data from numerous flight tests have been made open-source and can be freely downloaded from the website of the UMN UAV research group [3].

In Section III, a model-based FDI filter is developed that, when implemented on the experimental platform, would compare the measured response of the real aircraft with the simulated response of the aircraft model. When no faults are injected, the measured and simulated responses of the aircraft would not perfectly match because of several unmodeled effects. The aircraft, actuator, and sensor models have model uncertainty. The first-principles-based aircraft equations of motion do not completely capture all the dynamics of the aircraft. Several parameters of the aircraft, such as the inertia, geometry, and aerodynamic coefficients, also have some degree of uncertainty. In flight, the aircraft is subjected to several sources of exogenous disturbances, such as steady winds, wind gusts, and atmospheric turbulence. In addition, all measurements obtained through flight tests are corrupted with sensor noise.

II.C. Flight Control Law

A classical flight control law has been designed and validated by the UMN UAV Research Group. This control law serves as the baseline for any flight test involving closed-loop control. The control law has a standard two-tiered structure that consists of an outer loop for guidance and an inner loop for attitude control. The outer loop tracks desired airspeed (V_{ref}), altitude (h_{ref}), and heading angle (ψ_{ref}) and generates the following commands: desired throttle (τ_{cmd}), desired pitch attitude (θ_{ref}), and desired roll attitude (ϕ_{ref}). While τ_{cmd} is sent to the throttle actuator, θ_{ref} and ϕ_{ref} are tracked separately by the inner loop. A longitudinal dynamics inner loop tracks θ_{ref} and generates the elevator deflection command ($\delta_{ele,cmd}$). A lateral-directional dynamics inner loop tracks ϕ_{ref} and generates aileron ($\delta_{ail,cmd}$) and rudder ($\delta_{rud,cmd}$) deflection commands. A positive control surface deflection is associated with: a trailing-edge down deflection of the elevator; a trailing-edge down deflection of the right aileron, coupled with a trailing-edge up deflection of the left aileron; and a trailing-edge left deflection of the rudder. Specifically for the ailerons, $\delta_{ail,cmd} = +\delta^r_{ail,cmd} = -\delta^l_{ail,cmd}$. More details about the baseline flight control architecture can be found in [18, 24].

From the closed-loop aircraft response with the baseline controller, it was observed that stuck faults injected at the rudder resulted in a nonzero sideslip angle in steady-state. Since nonzero sideslip is almost never desirable, the lateral-directional dynamics inner loop was modified in order to make the controller robust to rudder faults. Thus, only this particular loop and the modifications made to it will be elaborated in this section. Figure 2 shows the modified lateral-directional dynamics inner loop. The bottom part of the figure 2 shows the roll attitude (ϕ_{ref}) tracker implemented as a proportional-integral (PI) control law in the block K_ϕ. The error between ϕ_{ref} and ϕ is the input to the PI law. A separate loop tracks p with a proportional gain K_p. The output of the roll attitude tracker is $\delta_{ail,cmd}$. The feedback of $\delta_{ail,cmd}$ to K_ϕ is used for integrator anti-windup in the PI controller K_ϕ.

The top part of figure 2 shows that additive faults are injected at the input to the plant. One of the key control objectives is to have zero sideslip in steady-state under healthy and faulty conditions. In order for the controller to be robust to faults in either of the two rudders, integral action on β is required. A β tracker is implemented as a PI law with anti-windup protection in block K_β. Since a nonzero sideslip angle

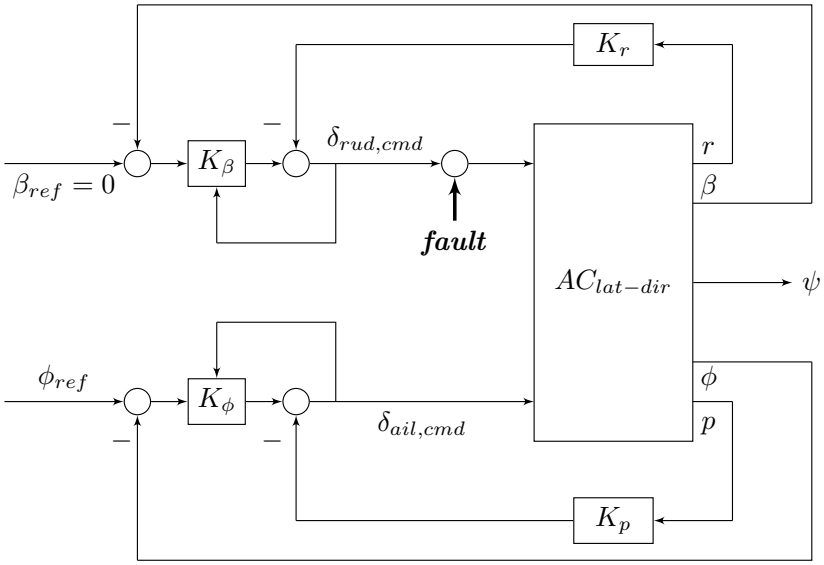


Figure 2: The lateral-directional dynamics inner loop.

is almost never desirable in flight, β_{ref} is set identically equal to zero. The yaw rate (r) is tracked with a proportional gain K_r. It should be noted that the modified flight control law does not treat the split rudders as separate control surfaces, i.e. the same rudder deflection command is sent to the actuators of both the top and bottom rudders. The robustness of the modified controller to faults injected at either of the two rudders is evident in the results presented in section IV. In general, it should be noted that while making the controller robust to faults is desirable, it also makes the detection and isolation of those faults more difficult because the controller masks the faults in the closed-loop response. The challenges associated with detecting and isolating faults when the aircraft is in closed-loop control are discussed in Section III.

III. Fault Detection and Isolation Filter

The objective of this research is the real-time detection, isolation, and estimation of faults at either the top or the bottom rudder of the Ultra Stick 120. Each component of the FDI filter (detection, isolation, and estimation) requires a different output variable or control command to be compared with that generated by the model. The following sections discuss the implementation of the FDI filter, models for the rudder fault modes, and the architecture of the FDI filter. The challenges associated with detecting and isolating faults with the aircraft in closed-loop control are also discussed.

III.A. FDI Filter Implementation

The model-based FDI filter compares measured outputs and control commands with their simulated counterparts. Figure 3 is a block diagram representation of how the FDI filter is implemented for real-time operation. The blocks P_{real} and P_{FDI} represent the real and simulated aircraft dynamics. These dynamics are depicted as generalized blocks for simplicity. The generalized plant contains the aircraft, actuator, and sensor dynamics as well as the flight control law. In other words, figure 2 is condensed into the P blocks in figure 3. Both P_{real} and P_{FDI} take in the same vector-valued reference signal (ref) as an input, where $ref = [\beta_{ref}, \phi_{ref}]^T$. Since P_{real} and P_{FDI} share the same flight control law (within each P block), they would respond similarly to the reference commands ref. However, P_{real} has model uncertainty, represented by the Δ block, and is affected by wind gusts & turbulence (d), sensor noise (n), and fault injections (f). None of these unmodeled effects (Δ, d, n) or faults (f) enter the P_{FDI} block. It is worth mentioning here that the FDI filter needs to be robust to the unmodeled effects (Δ, d, n) so that false alarms are not declared frequently. In addition, the FDI filter should be responsive to the faults (f) so that there are no missed detections. The generalized outputs of the P blocks are the closed-loop plant measurements (y) and control commands (u). With reference to figure 2, $y = [\beta, \phi, \psi, p, r]^T$ and $u = [\delta_{rud,cmd}, \delta_{ail,cmd}]^T$. The model-based

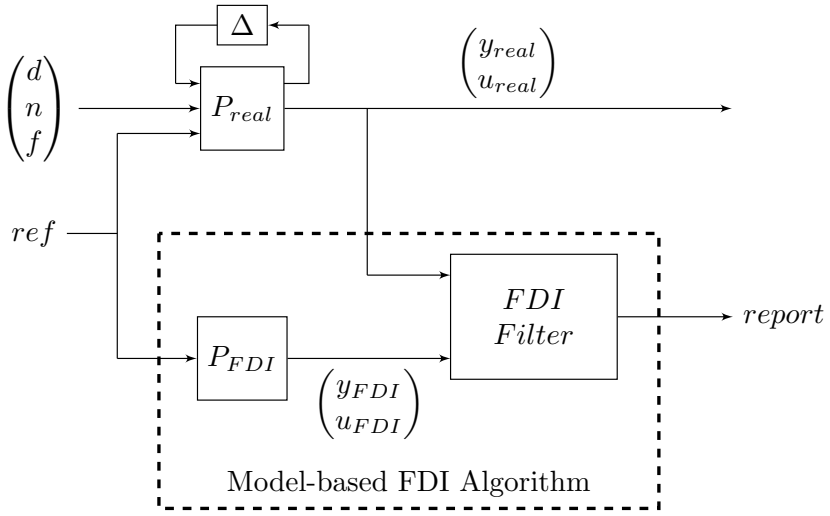


Figure 3: Model-Based FDI Filter Implementation.

FDI algorithm consists of the generalized plant model (P_{FDI}) and the FDI filter, and is enclosed by the dashed box. At a high level, the FDI filter works by comparing the y and u signals coming from each P block and is designed to be sensitive only to the fault signals (f). Ideally, the FDI filter should reject the effects of Δ, d and n.

III.B. Fault Modeling

In this research, only stuck faults are injected at the top and bottom rudders of the real aircraft (P_{real}). The faults are injected after a certain preset time has elapsed, but the controller has no a priori knowledge of the fault injection. To gain a better understanding of the flight dynamics, P_{real} is initially simulated using the nonlinear, high-fidelity model and P_{FDI} uses a linear model obtained by linearization at one flight condition. The difference between the high-fidelity nonlinear model and the lower fidelity linear model approximately captures the effect of model uncertainty (Δ). Some relevant results are presented in this section to demonstrate the closed-loop response of the real aircraft to the injected faults. These results will help motivate the architecture of the FDI filter in the next section. In the following results, sensor noise and turbulence effects are added, but steady winds and wind gusts are not.

The aircraft is trimmed at an altitude of 100 m and an airspeed of 23 m/s, and is commanded to fly straight and level along a heading reference of 155°. The trim conditions of the aircraft are: $\beta_{trim} = \phi_{trim} = p_{trim} = r_{trim} = 0$, $\psi_{trim} = 155°$, $\delta_{rud,trim} = 0$, and $\delta_{ail,trim} = 0.3°$. Figure 4 shows the response of P_{real} after a $+25°$ (positive saturation limit) stuck fault is injected, in simulation, at the top rudder at $t = 5s$. The fault injection time step is marked by a vertical dashed line. The signals shown are $y_{real} = [\beta, \phi, \psi, p, r]^T$ and $u_{real} = [\delta_{rud,cmd}, \delta_{ail,cmd}]^T$. Along with $\delta_{rud,cmd}$, the actual surface deflection (δ^t_{rud}) is also shown. The response of δ^t_{rud} shows that the top rudder is stuck at $+25°$ after $t = 5s$. For the first five seconds of the simulation, all the signals in y_{real} and u_{real} are at their respective trim values. The high frequency oscillations seen on all the signals are due to the effects of sensor noise and atmospheric turbulence.

From the six subplots shown in figure 4, it can be seen that all the signals in y_{real} and u_{real} depart from their respective trim values immediately after the fault is injected at $t = 5s$. All the signals show some distinct transient properties. With the top rudder stuck positively (trailing edge deflected left), a positive side force is generated on the vertical stabilizer. This positive side force results in a positive rolling moment and a negative yawing moment. As a result, the aircraft immediately yaws to the left ($r < 0$) and rolls to the right ($p > 0$). As previously mentioned in section II.C, the lateral-directional dynamics controller has proportional gains on p and r. Consequently, the yaw rate and roll rate transients show up as spikes and subside quickly. The spikes in the body angular rates lead to slower changes in the Euler angles, with ϕ increasing and ψ decreasing from their respective trim values. The ϕ and ψ signals reach their respective maximum and minimum values within a few seconds. As mentioned previously, the ϕ signal is tracked by a PI control law. Although the PI law results in the error decay being sluggish, it guarantees zero steady-

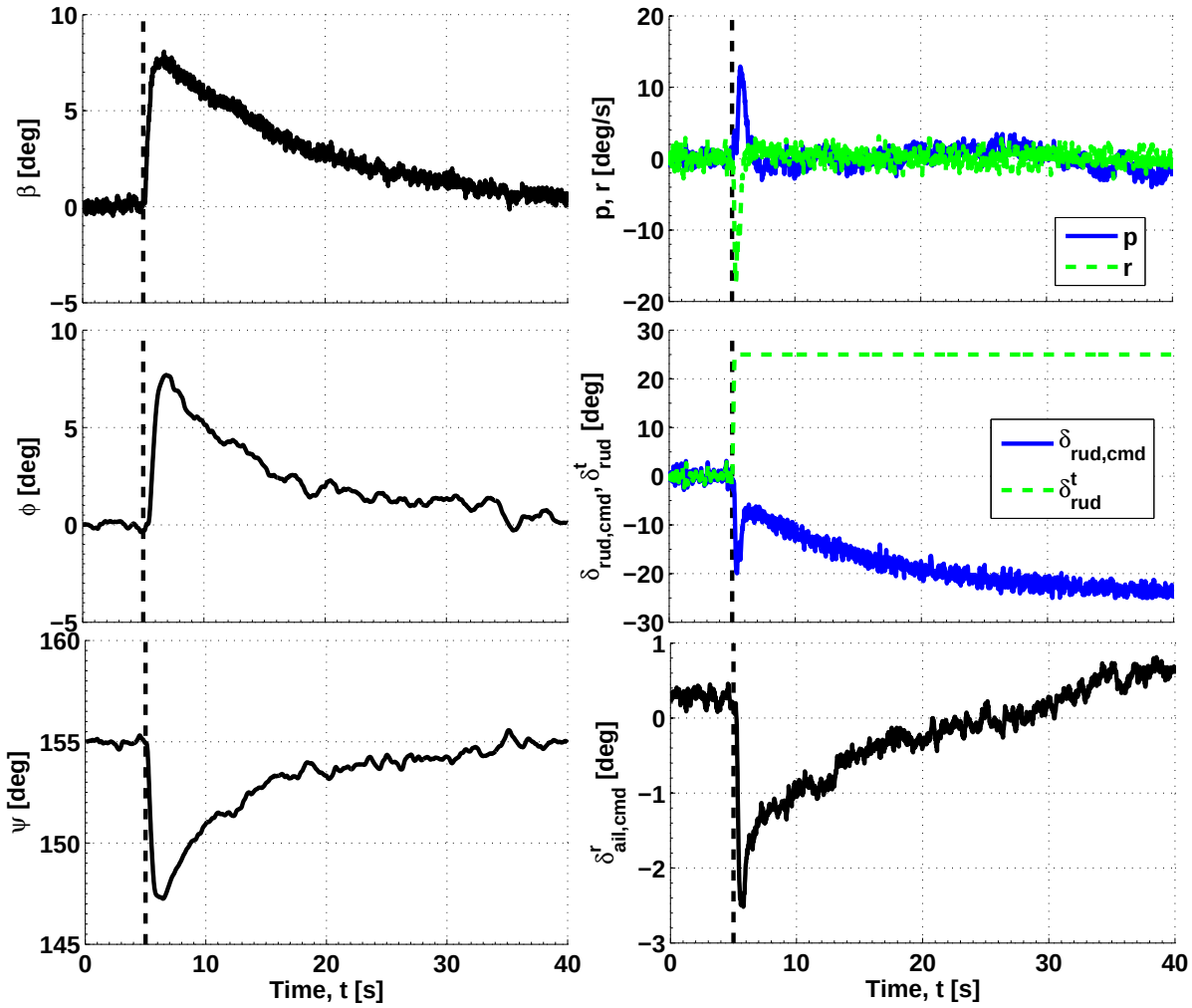


Figure 4: Plant and controller outputs for the top rudder stuck at $+25^\circ$ at $t = 5s$.

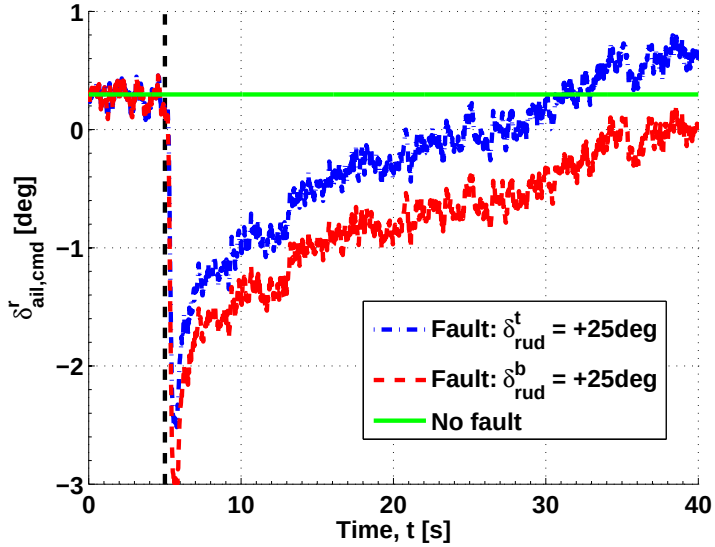


Figure 5: Aileron command for the top and bottom rudders stuck at $+25^\circ$ at $t = 5s$.

state tracking error. The ψ signal is tracked by a proportional gain in the outer loop (not discussed in this paper, see [18]) and has an error decay rate similar to that of ϕ. In addition, the rudder fault results in an immediate buildup of positive sideslip (β). The β tracker, which is also implemented as a PI law, results in an asymptotic convergence of β to $\beta_{ref} = 0$. As a result of the integral control on β and ϕ, all the y_{real} signals return to their respective trim values in steady-state ($t \geq 40s$). A key property of stuck rudder faults is that they cannot be detected simply by monitoring the steady-state response of y_{real}. The detection logic, described in detail in section III.C, uses the transient response of y_{real} to detect rudder faults.

The simulation results show that the closed-loop system is robust to the worst-possible top rudder fault of 25°. The flight control law treats the injected fault as a disturbance and compensates by commanding the bottom rudder to deflect in the opposite direction, as seen in the response of $\delta_{rud,cmd}$. In steady-state, $\delta_{rud,cmd}$ asymptotically converges to $-\delta^t_{rud}$. This equal, but opposite, deflection of the bottom rudder produces a negative side force and a positive yawing moment that counteracts the effect of the top rudder. Some interesting observations can also be made about the aileron deflection command ($\delta_{ail,cmd}$). The buildup of positive sideslip immediately after the onset of the fault produces a negative rolling moment due to the effect of the sideslip on the vertical stabilizer. To compensate for this negative rolling moment, the controller commands the ailerons to deflect in the negative direction, as seen in the response of $\delta_{ail,cmd}$. In steady-state, however, $\delta_{ail,cmd}$ does not converge to its trim value. This phenomenon is explained in greater detail in the following paragraphs.

A similar simulation was performed with the bottom rudder stuck at +25°. It was observed that the response of y_{real} and $\delta_{rud,cmd}$ were almost identical to the case of the top rudder fault. Hence, these plots are not reproduced in this paper. In fact, faults of equal magnitude and direction in the top and bottom rudders result in very similar responses in the y_{real} and $\delta_{rud,cmd}$ signals. The difference is so small that the source of the fault cannot be identified from either the transient or steady-state response of y_{real} and $\delta_{rud,cmd}$. The only signal from which equal faults in the top and bottom rudders can be differentiated is $\delta_{ail,cmd}$. Hence, only the $\delta_{ail,cmd}$ signal is reproduced for the bottom rudder fault.

Figure 5 shows the response of $\delta_{ail,cmd}$ to both top and bottom rudder faults of +25°. Although the transient response of $\delta_{ail,cmd}$ for top and bottom rudder faults are quite similar, a clear separation can be seen as steady-state is approached ($t \geq 40s$). In steady-state, the integral control in the β tracker drives the sideslip angle to zero by deflecting the healthy rudder in a direction opposite to the faulty rudder. Since the top and bottom rudders have slightly different rolling moment derivatives (due to their different moment arms), their net contribution to the rolling moment is non-zero. Since the top rudder has a larger rolling moment derivative than the bottom rudder, the net rolling moment contribution depends on the direction in which the top rudder is deflected in steady-state. If the top rudder is deflected positively in steady-state, the net rolling moment would be positive, and the controller would compensate by deflecting the ailerons positively, i.e. $\delta_{ail,cmd} > \delta_{ail,trim}$. This phenomenon can be seen in figure 5: for a top rudder fault of +25°, the steady-state value of $\delta_{ail,cmd}$ is greater than its trim value.

On the other hand, if the top rudder is deflected negatively in steady-state, the net rolling moment would be negative, and the controller would compensate by deflecting the ailerons negatively, i.e. $\delta_{ail,cmd} < \delta_{ail,trim}$. For a bottom rudder fault of +25°, the top rudder is deflected negatively in steady-state and results in a net negative rolling moment. As shown in figure 5, the controller compensates by deflecting the ailerons negatively in steady-state. The isolation & estimation filter, described in detail in section III.C, makes use of this phenomenon to isolate and estimate rudder faults.

III.C. FDI Filter Architecture

The FDI filter, that was enclosed by the dashed box in figure 3, is shown in greater detail in figure 6. The FDI filter takes in four vector-valued inputs: the real & simulated plant outputs (y_{real}, y_{FDI}) and the real & simulated controller commands (u_{real}, u_{FDI}). These four signals are processed in real-time by the filter and a report is generated. The filtering of the fault is a three-stage process involving detection, isolation, and estimation. The isolation and estimation stages are combined into a single block in figure 6. The performance of each stage can be quantified using appropriate metrics. The detection stage detects the occurrence of a rudder fault and has three main performance metrics: detection time, probability of missed detection, and probability of false alarm [25, 26]. The isolation stage pinpoints the source of the fault, i.e. it determines if the fault was injected at the top or the bottom rudder. A boolean flag is used to quantify the correctness of isolation. The flag is set to 1 if the source of the fault is isolated correctly and is set to 0 otherwise. The estimation stage generates an estimate of the fault magnitude and direction. The estimation error serves as

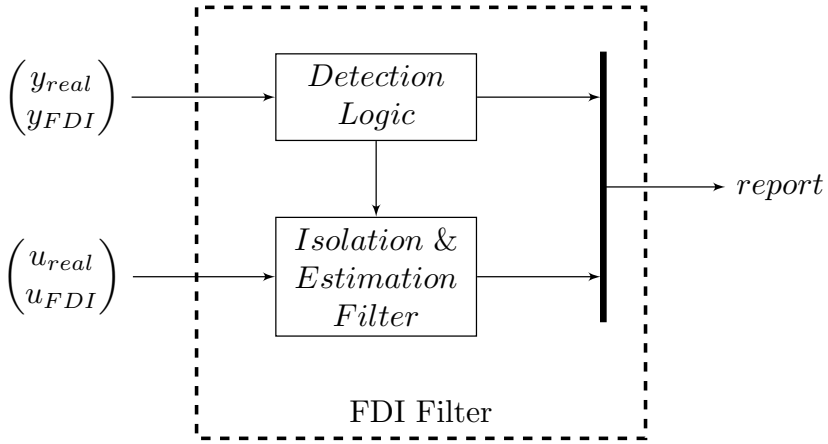


Figure 6: FDI Filter Architecture

a performance metric for the estimation stage. The arrow connecting the detection block to the isolation & estimation block in figure 6 indicates that the isolation & estimation filter is activated only if a fault is first detected by the detection logic. In section III.B, the principles of flight dynamics were invoked to analyze the fault effects shown in figures 4 and 5. In this section, the understanding of the fault effects is used alongside traditional linear analysis tools to construct both the detection logic and the isolation & estimation filter.

Detection Logic

It was concluded in section III.B that rudder faults would need to be detected based on the transient response of y_{real}. Specifically, the difference between the transient responses of y_{real} and y_{FDI} is used to detect rudder faults. The detection logic takes in y_{real} and y_{FDI} as inputs and generates a vector-valued residual signal (e_y) by subtracting each signal in y_{FDI} from its respective counterpart in y_{real}. Mathematically, $e_y = y_{real} - y_{FDI} = [\Delta\beta, \Delta\phi, \Delta\psi, \Delta p, \Delta r]^T$, where Δ denotes a difference between the *real* signal and the simulated FDI signal. As mentioned previously in section III.A, y_{real} and y_{FDI} will be similar because they share the same flight reference commands. Consequently, the mean of e_y will be small in the absence of a fault and under nominal conditions. Conversely, in the presence of a rudder fault, some components of this residual vector will be nonzero in transient and/or steady state. In addition, e_y will contain high-frequency components due to the effects of sensor noise and atmospheric turbulence, and lower frequency components from wind gusts and model uncertainty. The detection logic analyzes the transient response of e_y and is designed to be robust to model uncertainty, wind gusts, atmospheric turbulence, and sensor noise, but sensitive to the injected faults. This is possible because stuck rudder faults have a *unique and detectable signature* compared to wind gusts and maneuvers.

As mentioned previously, detection time is a standard metric to assess the performance of the detection logic. The detection time is the time that elapses between the injection of a fault and its successful detection. Faults that are injected at the rudder show up after some time in the y_{real} and u_{real} signals because of time lags inherent in the closed-loop aircraft dynamics. A deeper analysis of the results presented in section III.B reveal that rudder faults show up first in the body angular rates p & r and only later in β, ϕ and ψ. This makes physical sense because of the presence of integrators between p & r and β, ϕ & ψ. In order to detect faults quickly, the transient response of the residuals Δp and Δr are analyzed. In order to make the detection logic more reliable, the residual $\Delta\beta$ is also analyzed, along with Δp and Δr. In this detection logic, measurements from the airdata sensor (β) and the inertial measurement unit (IMU) (p & r) are used in fault detection. By analyzing residuals from two different sensors, actuator faults can be detected with higher confidence levels. Simultaneous faults in both sensors systems that mimic a rudder fault is very unlikely.

A standard technique in fault detection [8] is to raise a flag when the residual crosses a specified threshold. If the threshold is set too low, false alarms may be declared frequently. Conversely, if the threshold is set too high, there may be frequent missed detections. There is literature that shows how the threshold can be set analytically in order to balance the probabilities of false alarm and missed detection [27, 28]. However, in this paper, the thresholds for each residual are simply set based on the characterization of the sensor noise. As previously mentioned, the sensor noises are modeled as band-limited white noise derived

Table 1: Standard deviations and thresholds of residuals based on sensor noise characterization.

Source	Signal	σ	Threshold
Airdata sensor	$\Delta\beta$	0.1812°	$T_\beta = \pm 3\sigma_\beta$
IMU	Δp	0.4051°	$T_p = \pm 4\sigma_p$
IMU	Δr	0.4051°	$T_r = \pm 7\sigma_r$

from independent and identically distributed (iid) zero-mean Gaussian distributions [3]. The thresholds for each residual (T_β, T_p, T_r) are set equal to some multiple of their respective standard deviation, based on simulation results. The standard deviations (σ) of the Gaussian distributions for each of the sensors and the corresponding thresholds are shown in Table 1.

Within the detection logic, three separate flags (F_i, $i \in \{\beta, p, r\}$) are maintained for the residuals $\Delta\beta$, Δp, and Δr. The residuals $\Delta\beta$, Δp, and Δr are monitored at 50 Hz - the same sample rate used by the flight control law. Each flag is set equal to zero if the corresponding residual is within the limits defined by its threshold. The flags are set equal to +1 if the residual exceeds the positive threshold and -1 if the residual drops below the negative threshold. In summary, for $i \in \{\beta, p, r\}$, and at each sample time k,

$$F_i(k) = \begin{cases} +1 & if\ e_{y,i}(k) \geq +T_i \\ 0 & if\ |e_{y,i}(k)| < T_i \\ -1 & if\ e_{y,i}(k) \leq -T_i \end{cases} \tag{1}$$

The results presented in section III.B show that rudder faults result in a unique and detectable combination of transients in β, p, and r. For example, a positive rudder fault (irrespective of whether it is injected in the top or bottom rudder) results in β increasing, p increasing, and r decreasing from their respective trim values. Conversely, a negative rudder fault (irrespective of whether it is injected in the top or bottom rudder) results in β decreasing, p decreasing, and r increasing from their respective trim values. This pattern also shows up in the transient response of the e_y signals and, by extension, the flag variables (F_i, $i \in \{\beta, p, r\}$). In particular, positive rudder faults result in the following flag variable pattern: $[F_\beta, F_p, F_r] = [+1, +1, -1]$. Conversely, negative rudder faults result in the pattern, $[F_\beta, F_p, F_r] = [-1, -1, +1]$.

The detection logic monitors the three flags at each sample time for either of these two patterns. Further, a global detection flag variable is maintained in the detection logic with a default value of zero. The global flag is set equal to +1 if the $[+1, +1, -1]$ pattern is observed, and to -1 if the $[-1, -1, +1]$ pattern is observed, for five consecutive sample times. After performing extensive simulations, it was observed that rudder faults, depending on their sign, either produce the $[+1, +1, -1]$ or the $[-1, -1, +1]$ pattern over several sample times. This is in contrast to the effects of turbulence and sensor noise that may produce the patterns for one or two sample times. By checking for consistency in the pattern over five consecutive sample times, the logic is made robust and false alarms are avoided. This logical check over five consecutive sample times corresponds to a special case of an up/down counter that is commonly used in commercial avionics to avoid false alarms. In conclusion, a global flag of +1 indicates a positive rudder fault and -1 indicates a negative rudder fault.

Embedded in this unique sign pattern of the flag variables is some phase characteristics of the signals β, p, and r. Linear analysis tools can be exploited to understand the uniqueness of these phase characteristics. Specifically, the frequency responses of β, p, and r due to injected rudder faults can be compared with those due to wind gusts. Nominally, the aircraft is trimmed at an altitude of 100 m and an airspeed of 23 m/s. A linear closed-loop model is obtained at this nominal trim point. Figure 7 shows the Bode magnitude and phase plots of the closed-loop frequency response of β and p, at the nominal trim point. The lines marked δ_{rud} represent rudder fault injections and those marked $Wind(y)$ represent wind gusts directed along the body y-axis of the aircraft. In order to draw proper conclusions, the transfer functions that are used to generate these Bode plots are normalized. The normalization is done such that the transfer functions related to β match at a frequency of 1 rad/s. This normalization only affects the Bode magnitude plot and does not affect the phase plot.

It is seen in Figure 7 that the gain variations with frequency, in the plotted frequency range, are similar between rudder faults and wind gusts. The main takeaway from Figure 7 is that there is a significant phase

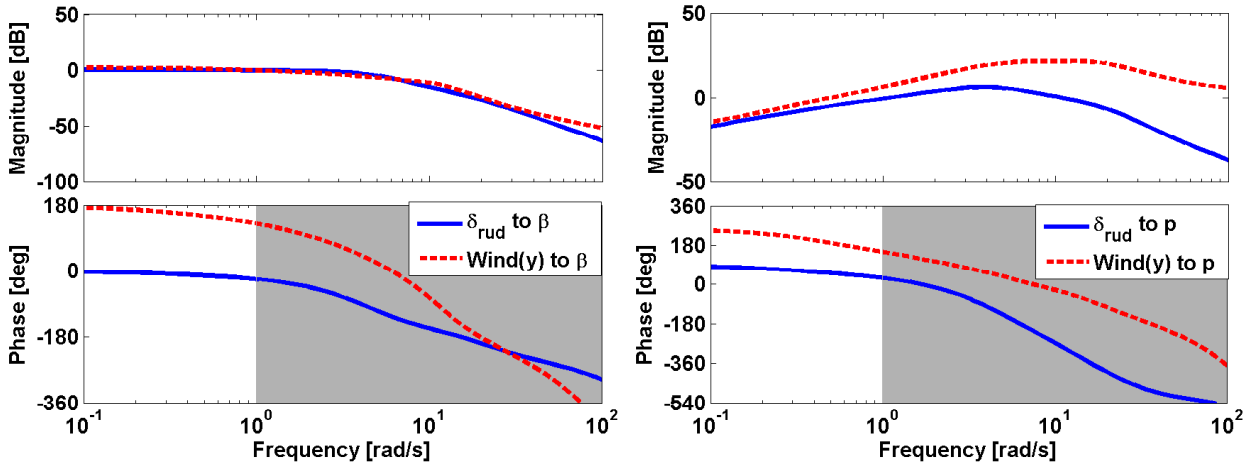


Figure 7: Bode magnitude and phase plots comparing the closed-loop frequency responses of β and p due to rudder faults with wind gusts. Trim altitude is 100 m and airspeed is 23 m/s.

difference between the responses induced by rudder faults and wind gusts. The bandwidth of the actuators that control the rudders is 8 Hz (50 rad/s). For this analysis, wind gusts between 1 m/s and 15 m/s, that persist over a distance of 1 m, are considered. This corresponds to a frequency range of 6 rad/s to 90 rad/s. The overall frequency range of interest is 1 rad/s to 100 rad/s and is highlighted by the gray boxes in the phase plots. More specifically, in the frequency response of p, it is observed that there is a phase difference of at least 180° between δ_{rud} and $Wind(y)$, over the frequency range of interest. A phase difference of approximately 180° is also observed in the frequency response of β, but only near 1 rad/s. A similar phase difference is also seen in the frequency response of r, but is omitted from this paper. For frequencies where there is only a small phase difference in any one signal among β, p, and r, a sufficiently large phase difference can be found in at least one of the other two signals. By using all three signals for fault detection, it is ensured that the filter is sensitive to rudder faults and insensitive to external aerodynamic disturbances.

In applying the detection logic to the residuals, it might be desirable to filter out the high frequency components by using a low-pass filter. However, this has the drawback of introducing a phase lag between the filtered residual and the raw residual and, thereby, delaying the detection. In order to be able to detect faults as soon as possible, the raw residuals are directly fed to the detection logic. The global flag and the detection time stamp are included in the report generated by the FDI filter.

Isolation & Estimation Filter

After a fault is detected, the isolation & estimation filter pinpoints the source of the fault and generates an estimate of the fault level. The isolation & estimation filter takes in u_{real} and u_{FDI} as inputs and generates a vector-valued residual signal (e_u) by subtracting each signal in u_{FDI} from its respective counterpart in u_{real}. Mathematically, $e_u = u_{real} - u_{FDI} = [\Delta\delta_{rud,cmd}, \Delta\delta_{ail,cmd}]^T$, where Δ denotes a difference between the *real* signal and the simulated FDI signal. As mentioned previously, u_{real} and u_{FDI} will be similar because they share the same flight reference commands. Consequently, the mean of e_u will be small in the absence of a fault and under nominal conditions. Conversely, in the presence of a rudder fault, some components of e_u will be nonzero in transient and/or steady state. It was concluded in section III.B that the only signal from which equal faults in the top and bottom rudders can be differentiated is $\delta_{ail,cmd}$. As shown in figure 5, the transient response of $\delta_{ail,cmd}$ to top and bottom rudder faults are similar. However, the steady-state value of $\delta_{ail,cmd}$ depends on the source of the fault. The isolation filter monitors the steady-state behavior of the aileron command residual ($\Delta\delta_{ail,cmd}$) and identifies the source of the fault.

The control command residual e_u also contains high frequency components due to the effects of sensor noise and atmospheric turbulence. In addition, e_u contains lower frequency components from wind gusts and model uncertainty. The isolation & estimation filter analyzes the steady-state response of e_u and is designed to be robust to model uncertainty, wind gusts, atmospheric turbulence, and sensor noise, but sensitive to the injected faults. The aileron command residual is computed as: $\Delta\delta_{ail,cmd} = \delta_{ail,cmd} - \delta_{ail,trim}$. However, it is seen in figure 5 that the steady-state value of $\delta_{ail,cmd}$ for top and bottom rudder faults is very close to the trim value. This implies that the signal-to-noise ratio (SNR) of $\Delta\delta_{ail,cmd}$ will be very small in steady-state.

In order to properly detect the steady-state value of the aileron command residual, $\Delta\delta_{ail,cmd}$ would need to have a higher SNR. In order to boost the steady-state SNR of $\Delta\delta_{ail,cmd}$, the high frequency components of the residual need to be removed through a low-pass filter. Although the low-pass filter would introduce a phase lag, the magnitude of the phase lag would not be too large because the mean of the residual only has low frequency components as steady-state is approached.

The low pass filter is chosen to be a first-order lag: $H(s) = \frac{1}{2s+1}$. The time constant of $2s$ implies that frequencies above 0.08 Hz are filtered out by H. The aileron command residual ($\Delta\delta_{ail,cmd}$) is filtered using $H(s)$ and is analyzed at each sample time by the isolation filter. At each time step, the preceding fifty time steps are analyzed in order to check if the residual has reached steady-state. The residual ($\Delta\delta_{ail,cmd}$) is declared to be in steady-state only if the preceding fifty time steps satisfy the following statistical constraints: i) mean < 0.35, ii) range < 0.015, and iii) standard deviation < 0.005. In addition, it is concluded from simulation that the lateral-directional dynamics of the closed-loop plant (the P blocks in figure 3) has a time constant of 12 seconds. This implies that steady-state is reached roughly $36s$ after the fault is detected. This information is also used in the isolation filter to ensure that steady-state is not declared earlier than expected. Once steady-state has been declared for $\Delta\delta_{ail,cmd}$, the estimation filter is activated.

The estimation filter generates an estimate of the magnitude and direction of the injected fault. As mentioned previously in section III.B, after a fault is injected, the controller responds by deflecting the healthy rudder in the opposite direction. Consequently, a direct measure of the fault level is $-\delta_{rud,cmd}$ after steady-state is reached. It is observed that $\delta_{rud,cmd}$ reaches steady-state at approximately the same time as $\Delta\delta_{ail,cmd}$. As shown in figure 4, $\delta_{rud,cmd}$ also contains high frequency components. Since the fault level is estimated near steady-state, $\delta_{rud,cmd}$ is also filtered without the penalty of phase lag. At this point, estimates are available for the fault level and for $\Delta\delta_{ail,cmd}$. Using the signs of these two estimates, the source of the fault and its direction can be isolated. This isolation can be summarized into an isolation matrix, as shown in Table 2. The isolation matrix is a one-to-one mapping between the causes (fault modes) and the effects (output responses). If the set of output responses is restricted to only those shown in the isolation matrix, the mapping becomes one-to-one & onto and can, hence, be inverted. For any entry in the matrix, a positive sign implies an increase from its trim value and a negative sign implies a decrease from its trim value. In the results presented in section IV, several difference fault levels are considered, including the case of the rudder stuck at 0°. As an example, consider the case where $\delta_{ail,cmd} < \delta_{ail,trim}$ and the steady-state value of $\delta_{rud,cmd}$ is positive (row 3 in table 2). This combination of effects has a unique cause: a negative fault in the top rudder, i.e. $\delta^t_{rud} < 0$. Thus, the isolation & estimation filter simultaneously isolates the source of the fault and generates an estimate of the fault magnitude and direction. In the following section, simulation results are presented for different rudder fault levels and the performance and robustness of the FDI filter is assessed.

Table 2: Fault Isolation Matrix

Fault mode	Control commands (steady-state response)		Plant outputs (transient response)				
	$\delta_{rud,cmd}$	$\delta_{ail,cmd}$	β	ϕ	ψ	p	r
$\delta^t_{rud} > 0$	−	+	+	+	−	+	−
$\delta^b_{rud} < 0$	+	+	−	−	+	−	+
$\delta^t_{rud} < 0$	+	−	−	−	+	−	+
$\delta^b_{rud} > 0$	−	−	+	+	−	+	−

IV. Results

The FDI filter, that was developed in section III.C, is applied to the aircraft model described in section II.B. With reference to figure 3, during flight tests, P_{real} represents the actual aircraft dynamics and P_{FDI} represents the analytical model used by the FDI filter. In order to simulate faults, P_{real} is initially represented by the nonlinear, high-fidelity model and P_{FDI} uses a linear model obtained by linearization at one flight condition. Three different sets of plots are presented in this section to illustrate the performance and robustness of the FDI filter. In all three sets of plots, P_{real} is affected by atmospheric turbulence and

sensor noise. In particular, the first set of plots illustrate the robustness of the filter to wind gusts. The second set of plots illustrate the robustness of the filter to model uncertainty and commanded maneuvers. The third set of plots show two performance metrics of the FDI filter - detection time and fault estimate - as a function of the injected fault level. All the results presented in this section are simulated.

Robustness to wind gusts

For the first set of plots, the aircraft is trimmed at an altitude of 100 m and an airspeed of 23 m/s, and is commanded to fly straight and level along a heading reference of 155°. The trim conditions of the aircraft are: $\beta_{trim} = \phi_{trim} = p_{trim} = r_{trim} = 0$, $\psi_{trim} = 155°$, $\delta_{rud,trim} = 0$, and $\delta_{ail,trim} = 0.3°$. It should be noted here that both P_{real} and P_{FDI} use the same trim conditions. The first set of plots that will be discussed here are shown in figure 8. Figure 8 contains five subplots that all share the same horizontal time axis from 0 to $60s$. The top three subplots show the time history of the e_y residuals, specifically, $\Delta\beta$, Δp, and Δr. The bottom two subplots show the time histories of the e_u residuals, specifically, $\Delta\delta_{rud,cmd}$ and $\Delta\delta_{ail,cmd}$.

A wind gust of length $[dx, dy, dz] = [1, 1, 1]$ and amplitude $[du, dv, dw] = [0, 1.5, 0]$ is injected at $t = 5s$. The wind gust length $[dx, dy, dz]$ indicates the distance, measured in the Earth-fixed reference frame, along which the wind gust affects the aircraft. The wind gust amplitude $[du, dv, dw] = [0, 1.5, 0]$ indicates the perturbation velocities, measured along the body-fixed reference frame, induced by the wind gust. Physically, this models a wind gust striking the aircraft on its starboard side and directed toward its port side. This wind gust direction was chosen because it excites the lateral-directional dynamics of the aircraft - the same dynamics excited by rudder faults - and thus tests the robustness of the FDI filter. The Discrete Wind Gust Model, imported from Matlab's Aerospace Blockset, is used to apply this wind gust in simulation. More details about this wind gust model and its parameters can be found in the Matlab documentation. In addition, a fault of +10° is injected at the bottom rudder at $t = 20s$. The wind gust injection time is marked by the green tab at $t = 5s$ and the rudder fault injection time is marked by the maroon tab at $t = 20s$ on the horizontal axis.

The residuals e_y are shown in blue color in the top three subplots in figure 8. Overlaid on top are the flag variables (F_β, F_p, F_r) of the detection logic, described in section III.C. The variation of the flag variables between the values -1, 0, and +1 can be seen in the plots. Starting at $t = 0$, all the e_y residuals have zero mean because y_{real} and y_{FDI} are very similar. As seen in the plots, the e_y signals contain high frequency components due to the effects of atmospheric turbulence and sensor noise. At $t = 5s$, a wind gust is injected in simulation that affects only the *real* aircraft P_{real}. The model simulated within the FDI algorithm (P_{FDI}) does not see the effect of the wind gust. Consequently, the residual signals in e_y diverge from zero, as seen in the plots. The sideslip angle increases immediately due to the increased lateral velocity induced by the starboard side wind gust. This increase in the sideslip angle shows up as a spike in the $\Delta\beta$ residual which triggers the flag F_β to a value of +1. The wind gust produces a large negative side force on the vertical stabilizer, which translates to a negative rolling moment and a positive yawing moment. As a result, the aircraft rolls to the left ($p < 0$) and yaws to the right ($r > 0$). These perturbations show up as spikes in the Δp and Δr residuals, with Δp peaking negatively and Δr peaking positively. This, in turn, triggers their respective flags as: $F_p = -1$ and $F_r = +1$. As seen from the plots of the three flags, overlaid on the e_y residuals, the wind gust results in the following flag pattern: $[F_\beta, F_p, F_r] = [+1, -1, +1]$. Since this pattern does not match with either of the patterns in the detection logic, the global flag variable is not triggered and remains at its default value of 0 and a fault is not triggered. This fact is indicated by the green color of the plots of the three flags. Thus, the wind gust is successfully rejected by the detection logic as a true negative.

The wind gust subsides by $t = 15s$ and the means of all the e_y residuals return to zero. At $t = 20s$, a fault of +10° is injected in the bottom rudder. Subsequently, all the e_y residuals diverge from zero. The positive bottom rudder fault produces a positive sideslip in the aircraft, along with a positive roll rate, and a negative yaw rate. The closed-loop aircraft response to a positive bottom rudder fault is quite similar to that of the positive top rudder fault, that was explained in detail in section III.B. As a result, the $\Delta\beta$ residual increases, triggering its flag to +1. The Δp residual also increases, resulting in $F_p = +1$. In addition, the Δr residual decreases, resulting in $F_r = -1$. The flag pattern of $[F_\beta, F_p, F_r] = [+1, +1, -1]$ is detected by the detection logic and the global flag is set to +1. The fault is detected at $t = 20.44s$, implying a detection time of $0.44s$. The fact that the global flag turns +1, is indicated by the red color of the individual flags after $t = 20.44s$. Once the global flag reaches a nonzero value for five consecutive sample times, it is held at that value for all future times, as seen in figure 8.

After the fault is detected, the detection logic triggers the isolation & estimation filter. This filter makes use of the e_u residuals, shown by the blue colored plots in the bottom two subplots of figure 8. The injected

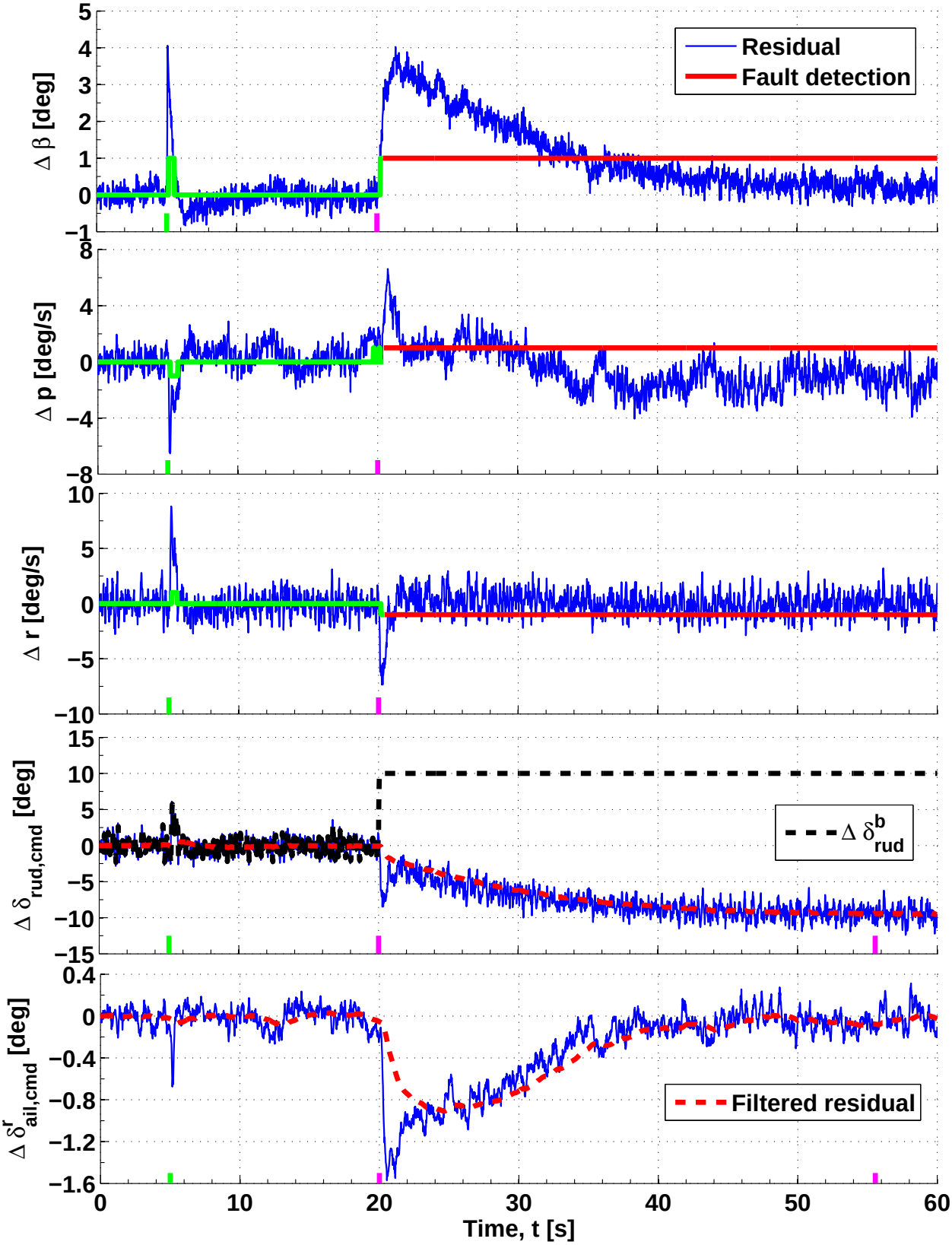


Figure 8: Residuals e_y and e_u for straight and level flight at the nominal flight condition. A wind gust is injected at $t = 5s$ and a bottom rudder stuck fault of $+10°$ is injected at $t = 20s$.

fault is seen in the actual deflection of the bottom rudder: δ^b_{rud} is stuck at $+10^\circ$ $\forall t = 20s$. The flight control law responds to this fault by commanding the top rudder to deflect in the opposite direction, as seen in the decrement of $\Delta\delta_{rud,cmd}$, and by commanding the ailerons to initially deflect negatively, as seen in the negative values of $\Delta\delta_{ail,cmd}$. The filtered e_u residuals are shown by the dashed red lines overlaid on the residuals. The isolation filter monitors the steady-state value of the filtered $\Delta\delta_{ail,cmd}$ residual. It is observed that $\Delta\delta_{aim,cmd}$ reaches steady state at about $t = 55.56s$ (marked with a maroon tab). This implies an isolation time of approximately $35s$, the same as the settling time of the closed-loop plant. The generated estimate of the fault is $+9.4^\circ$, implying an estimation error of 0.6°. Overall, it is seen that the FDI filter is robust to the wind gust, but responsive to the actual fault injection.

Robustness to model uncertainty and commanded maneuvers

In the seconds set of plots, the robustness of the FDI filter is assessed in the presence of a greater level of model uncertainty and with commanded maneuvers. One way of introducing additional model difference is by setting different trim airspeeds for P_{real} and P_{FDI}. The linear model that is used in P_{FDI} is generated at a nominal airspeed of 23 m/s. However, the nonlinear model that is used in P_{real} is trimmed at an off-nominal airspeed of 25 m/s. In addition to the off-nominal flight condition, a doublet heading maneuver is commanded. The combination of the off-nominal flight condition and the doublet maneuver tests the robustness of the FDI filter to conditions commonly experienced during flight tests.

While P_{real} and P_{FDI} have different trim airspeeds, they have the same trim altitude of 100 m. Although the different trim airspeeds result in different longitudinal trim states, the lateral-directional trim states are unaffected. Under trim conditions, the aircraft is commanded to fly straight and level along $\psi_{ref} = 155^\circ$.

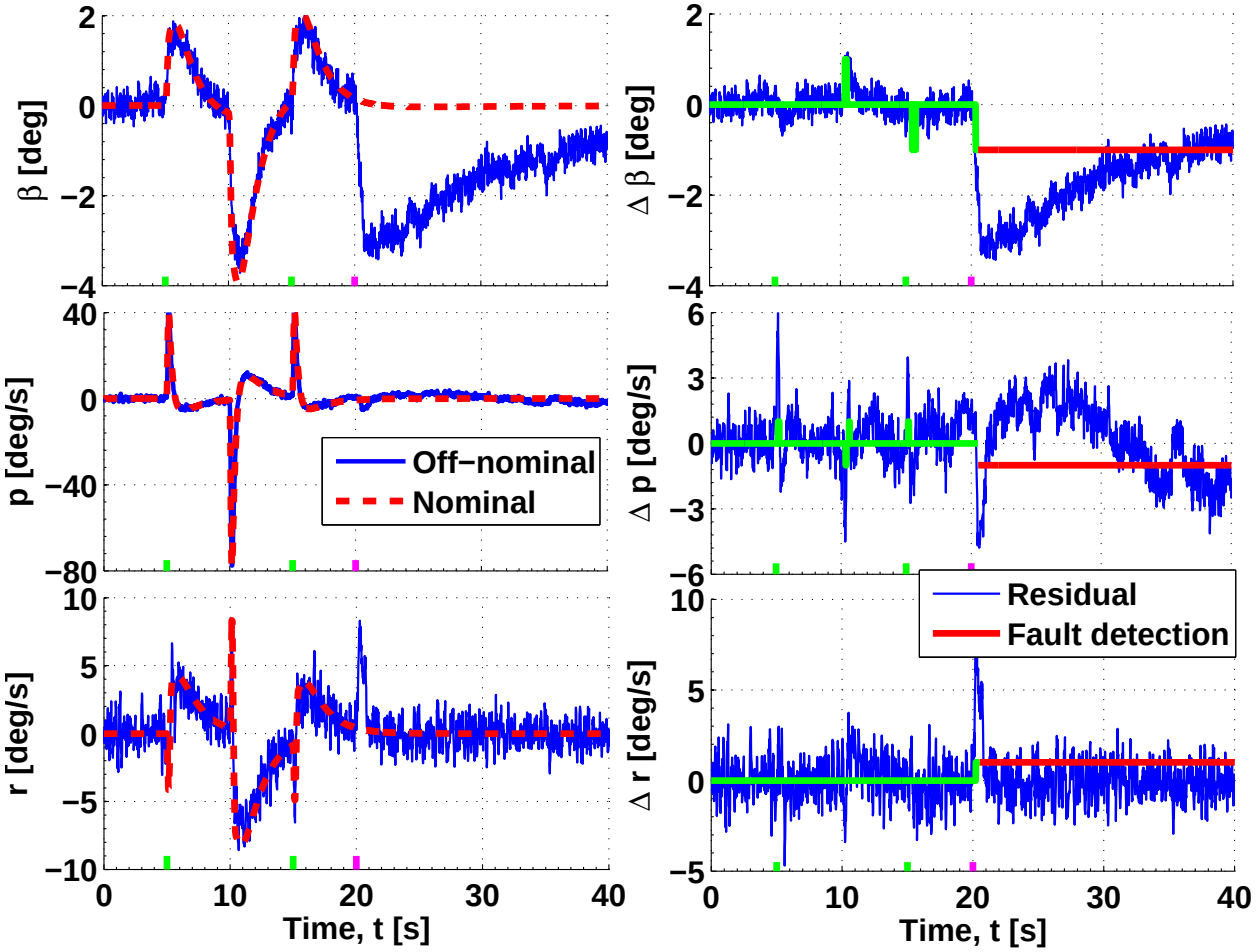


Figure 9: Signals y_{real} & y_{FDI} and residuals e_y for a maneuver at an off-nominal flight condition. A ψ doublet of $\pm 10^\circ$ is injected for $t \in [5s, 15s]$ and a top rudder stuck fault of -10° is injected at $t = 20s$.

Figure 9 contains six subplots that all share the same horizontal time axis from 0 to $40s$. The three subplots on the left show the responses of y_{real} (solid blue) and y_{FDI} (dashed red). Specifically, the variables used in the detection logic, β, p, and r, are plotted. The three subplots on the right show the time histories of the e_y residuals ($\Delta\beta$, Δp, Δr) in solid blue. Their respective flags are shown either in green or in red depending on whether a fault is detected. At $t = 5s$, a doublet heading maneuver is commanded: $\psi_{ref} = \psi_{trim} + 10^\circ$ for $t \in [5s, 10s]$, $\psi_{ref} = \psi_{trim} - 10^\circ$ for $t \in [10s, 15s]$, and $\psi_{ref} = \psi_{trim}$ for $t > 15s$. The duration of the ψ doublet maneuver is marked using the green tabs at $t = 5s$ and $t = 15s$. The ψ doublet maneuver excites the lateral-directional dynamics of the aircraft. In addition, a fault of -10° is injected in the top rudder at $t = 20s$. The rudder fault injection is marked using a maroon tab at $t = 20s$. As explained in the following, the FDI filter does not respond to the off-nominal ψ doublet, but responds to the injected fault.

Since both P_{real} and P_{FDI} are fed the same reference commands, they both respond similarly to the ψ doublet. Consequently, the mean variations in y_{real} and y_{FDI} are similar and the e_y residuals have close to zero mean even during the ψ doublet. The ψ doublet does cause two of the residuals, $\Delta\beta$ and Δp to exceed their respective thresholds and this triggers their flags, as seen in figure 9. However, since F_r does not get triggered, the global flag remains at its default value of zero. After the fault is injected, the three left subplots show the y_{real} signals diverging from their y_{FDI} counterparts. The negative fault in the top rudder produces a negative sideslip, a negative roll rate, and a positive yaw rate. These variations can be seen in the plots of the y_{real} as well as the e_y signals. The $\Delta\beta$ and Δp signals become negative and the Δr signal becomes positive. This results in the individual flags becoming: $[F_\beta, F_p, F_r] = [-1, -1, +1]$. This flag pattern is detected, the global flag is set to -1, and a fault is detected at $20.54s$. The fault detection is indicated by the red color of the individual flags $\forall t > 20.54s$. The performance of the isolation & estimation filter is not shown in figure 9. However, from simulation, it is observed that the isolation time is $35s$ after the fault is detected, and the estimation error is 0.4°.

Performance metrics of FDI filter

The variation of two specific performance metrics of the FDI filter - detection time and fault estimate - as a function of the injected fault for the top and bottom rudders are shown in figure 10. In generating these two plots, both P_{real} and P_{FDI} were trimmed at the nominal airspeed of 23 m/s and commanded to straight and level flight. In addition, P_{real} was subject to atmospheric turbulence and sensor noise, but not wind gusts. The top and bottom rudders on the Ultra Stick 120 have a range of motion of $\pm 25^\circ$, as shown on the horizontal axis. It is seen in the left subplot that faults of larger magnitudes have lower detection times as compared to faults of smaller magnitudes. This makes physical sense because a larger fault will induce

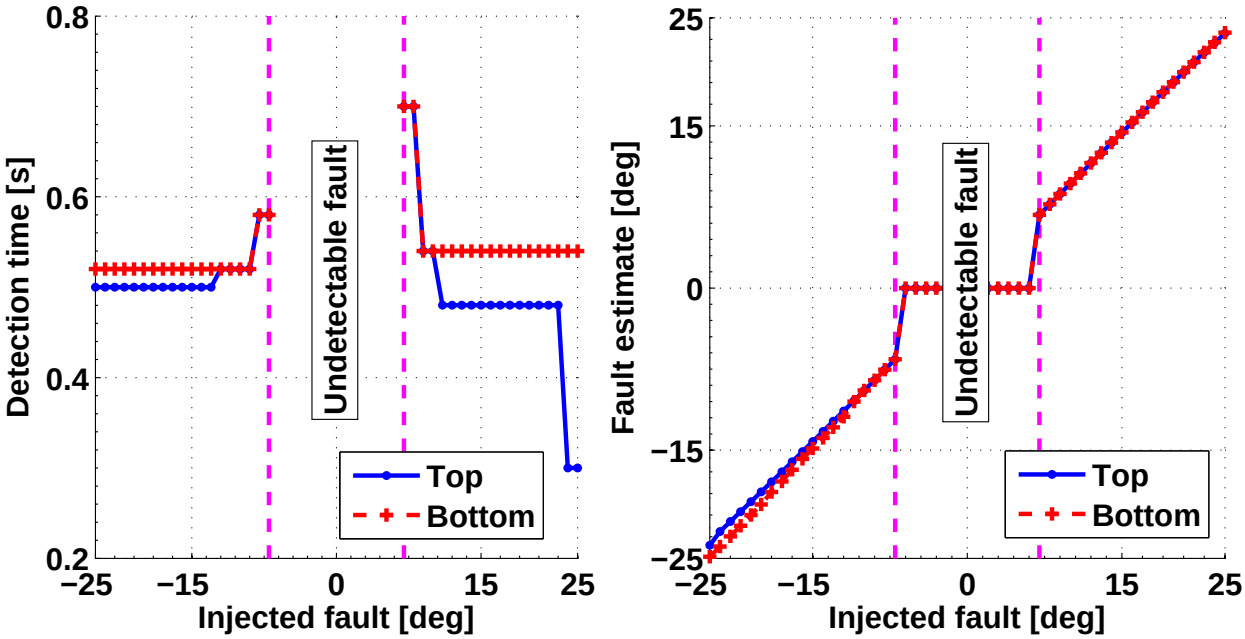


Figure 10: Variation of detection time and fault estimate with the injected fault level.

larger transients in the residual vector e_y. Larger transients in e_y will result in the residuals crossing their respective thresholds sooner and triggering the global flag faster. Another interesting observation is that top rudder faults are detected faster than bottom rudder faults over a large range of the fault level. At lower fault levels, the detection time for top rudder faults is no worse than the detection time for bottom rudder faults. This observation, too, makes physical sense. Since the top rudder has a larger rolling moment arm as compared to the bottom rudder, a top rudder fault will cause larger transients in Δp and lead to faster detection.

It is also seen that faults between -7° and $+7^\circ$ are undetectable because they do not trigger the global flag. This is because the flag patterns encoded in the logic are not observed for at least five consecutive sample times. The detection logic has four unique parameters that govern its sensitivity to rudder faults: the three thresholds (T_β, T_p, T_r) and the number of consecutive frames over which the flag pattern needs to be observed. Presumably, these undetectable faults may become detectable if the thresholds are made smaller and/or the number of frames reduced. However, the improved performance of the FDI filter will come with the price of reduced robustness. By making the FDI filter sensitive to smaller fault levels, it is likely that wind gusts, turbulence, and sensor noise may trigger false alarms. This trade-off between performance and robustness translates to balancing the probabilities of false alarm and missed detection. An FDI filter that is very robust and that hardly ever generates false alarms is also likely to have a larger number of missed detections, and vice-versa.

The variation of the fault estimate with the fault level is shown in the right subplot of figure 10. For a perfect isolation filter, this plot would correspond to a straight line with unit slope passing through the origin. However, the isolation filter developed declares steady state at a finite time, and hence results in some nonzero estimation error. Consequently, the plot is slightly skewed from the ideal. For positive rudder faults, the estimates generated by the filter for top and bottom rudder faults are very close and their respective plots almost lie on top of each other. For negative rudder faults, the estimates generated by the filter for top and bottom rudder faults are slightly different: top rudder faults appear to have worse estimation errors than bottom rudder faults. Since fault levels between -7° and $+7^\circ$ are undetectable, the isolation & estimation filter is never triggered and the fault estimate is zero in that range.

It is worth noting here that the problem of fault detection deals with naturally stochastic processes, such as turbulence and sensor noise. Therefore, while presenting results, it is not sufficient to simply present the results of a single simulation run. In order to account for the stochastic nature of the residuals, Monte Carlo simulations would need to be performed. Future work will involve these Monte Carlo simulations in order to better characterize the performance and robustness of the FDI filter.

V. Conclusion

The aircraft discussed has a split rudder for fault-tolerant control. A detection and isolation problem was formulated for stuck faults injected in one of the two rudders. The proposed flight control law was shown to be robust to simulated stuck rudder faults. In addition, a model-based fault detection, isolation, and estimation filter was designed using flight dynamics principles. Through simulation, the model-based filter was shown to be sensitive to stuck rudder faults, yet robust to off-nominal flight conditions, wind gusts, atmospheric turbulence, and sensor noise. The filter was evaluated on the basis of two performance metrics: detection time and estimation error. It was observed that the detection time for stuck faults decreased as the fault magnitude increased. In addition, the estimation error decreased as the estimation time increased. A certain range of stuck rudder faults was observed to be undetectable. Future work will involve validating the simulation results through flight tests and expanding the scope of the filter to cover other aircraft fault modes.

Acknowledgments

The authors would like to thank Brian Taylor for providing support with the simulation environment, experimental data, and flight tests. The authors acknowledge the members of the senior design team that performed the reliability analysis for the Ultra Stick 120: Jeremy Amos, Erik Bergquist, Jay Cole, Justin Phillips, Shawn Reimann, and Simon Shuster. This work was supported 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". Any opinions, findings, and conclusions or

recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF.

References

[1]United States Congress, "House Resolution 658: FAA Modernization and Reform Act of 2012," 2012, Section 332: Integration of Civil Unmanned Aircraft Systems into National Airspace System.

[2]Yeh, Y. C., "Triple-triple redundant 777 primary flight computer," *Proceedings of the 1996 IEEE Aerospace Applications Conference*, Aspen, CO, USA, 1996, pp. 293–307.

[3]University of Minnesota, "UAV Research Group," `www.uav.aem.umn.edu`, 2014.

[4]Amos, J., Bergquist, E., Cole, J., Phillips, J., Reimann, S., and Shuster, S., "UAV for Reliability," `www.aem.umn.edu/∼SeilerControl/SafetyCritical.shtml`, December 2013.

[5]Goupil, P., "AIRBUS state of the art and practices on FDI and FTC in flight control system," *Control Engineering Practice*, Vol. 19, No. 6, June 2011, pp. 524 – 539.

[6]Goupil, P., "Oscillatory failure case detection in the A380 electrical flight control system by analytical redundancy," *Control Engineering Practice*, Vol. 18, No. 9, 2010, pp. 1110–1119.

[7]Gertler, J. J., *Fault detection and diagnosis in engineering systems*, Marcel Dekker, 1st ed., 1998.

[8]Isermann, R., *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*, Springer-Verlag, Germany, 2006.

[9]Ding, S. X., *Model-Based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*, Springer-Verlag, Germany, 1st ed., 2008.

[10]Chen, J. and Patton, R., *Robust Model-Based Fault Diagnosis for Dynamic Systems*, Kluwer, Boston, MA, 1999.

[11]Freeman, P., Pandita, R., Srivastava, N., and Balas, G., "Model-Based and Data-Driven Fault Detection Performance for a Small UAV," *IEEE Transactions on Mechatronics*, Vol. 18, No. 4, 2013, pp. 1300–1309.

[12]Freeman, P. and Balas, G., "Analytical Fault Detection for a Small UAV," *AIAA Infotech@Aerospace Conference*, Boston, MA, USA, 2013.

[13]Hwang, I., Kim, S., Kim, Y., and Seah, C. E., "A Survey of Fault Detection, Isolation, and Reconfiguration Methods," *IEEE Transactions on Control Systems Technology*, Vol. 18, 2010, pp. 636–653.

[14]Pandita, R., Bokor, J., and Balas, G., "Closed-loop performance metrics for fault detection and isolation filter and controller interaction," *International Journal of Robust and Nonlinear Control*, Vol. 23, December 2011, pp. 419–438.

[15]Cook, M. V., *Flight Dynamics Principles*, Elsevier, 2nd ed., 2007.

[16]Nelson, R. C., *Flight Stability and Automatic Control*, McGraw-Hill, 2nd ed., 1998.

[17]Murch, A. M., Paw, Y. C., Pandita, R., Li, Z., and Balas, G. J., "A Low Cost Small UAV Flight Research Facility," *Advances in Aerospace Guidance, Navigation and Control*, edited by F. Holzapfel and S. Theil, Springer-Verlag Berlin Heidelberg, 2011, pp. 29–40.

[18]Dorobantu, A., Johnson, W., Lie, F. A., Taylor, B., Murch, A., Paw, Y. C., Gebre-Egziabher, D., and Balas, G., "An Airborne Experimental Test Platform: From Theory to Flight," *American Control Conference*, June 2013, pp. 659–673.

[19]Lie, F. A., Dorbantu, A., Taylor, B., Gebre-Egziabher, D., Seiler, P., and Balas, G., "An Airborne Experimental Test Platform: From Theory to Flight (Part 1)," *InsideGNSS*, April 2014, pp. 44–58.

[20]Amos, J., Bergquist, E., Cole, J., Phillips, J., Reimann, S., and Shuster, S., "UAV for Reliability Build," `www.aem.umn.edu/∼SeilerControl/SafetyCritical.shtml`, May 2014.

[21]Stevens, B. L. and Lewis, F. L., *Aircraft Control and Simulation*, John Wiley & Sons, 1992.

[22]Hoe, G., Owens, D., and Denham, C., "Forced Oscillation Wind Tunnel Testing for FASER Flight Research Aircraft," *AIAA Atmospheric Flight Mechanics Conference*, AIAA, Minneapolis, MN, August 2012.

[23]Owens, D., Cox, D. E., and Morelli, E. A., "Development of a Low-Cost Sub-Scale Aircraft for Flight Research: The FASER Project," *25th AIAA Aerodynamic Measurement Technology and Ground Testing Conference*, AIAA, San Francisco, CA, June 2006.

[24]Paw, Y. C., *Synthesis and Validation of Flight Control for UAV*, Ph.D. thesis, University of Minnesota-Twin Cities, 2009.

[25]Hu, B. and Seiler, P., "A probabilistic method for certification of analytically redundant systems," *Proceedings of the 2nd International Conference of Control and Fault-Tolerant Systems*, Nice, France, 2013, pp. 13–18.

[26]Hu, B. and Seiler, P., "Probability Bounds for False Alarm Analysis of Fault Detection Systems," *Proceedings of the 51st Allerton Conference on Communication, Control, and Computing*, Champaign, IL, USA, 2013, pp. 989–995.

[27]Sturza, M. A., "Navigation System Integrity Monitoring Using Redundant Measurements," *Navigation, Journal of The Institute of Navigation*, Vol. 35, No. 4, Winter 1988-1989, pp. 483–502.

[28]Hu, B. and Seiler, P., "Certification Analysis for a Model-Based UAV Fault Detection System," *AIAA Guidance, Navigation, and Control Conference*, National Harbor, MD, USA, 2014.