Fault Detection and Basic In-Flight Reconfiguration of a Small UAV Equipped with Elevons ⋆

Peter Bauer * **Raghu Venkataraman** ** **Balint Vanek** * **Peter J. Seiler** **
Jozsef Bokor ***

* *Systems and Control Laboratory, Institute for Computer Science and Control, Hungarian Academy of Sciences, H-1111 Budapest Kende utca 13-17., (e-mails: bauer.peter@ sztaki.mta.hu, vanek@ sztaki.mta.hu).*
** *Department of Aerospace Engineering and Mechanics, University of Minnesota, 110 Union Street SE, Minneapolis, MN 55455, (e-mails: venka085@umn.edu, seile017@umn.edu)*
*** *MTA-BME Control Engineering Research Group, (e-mail: bokor@ sztaki.mta.hu)*

Abstract: The paper introduces the in-flight fault detection and basic reconfiguration of a small unmanned aerial vehicle equipped with two elevons and an electric motor. The considered fault scenario is one control surface stuck at a given position during straight and level flight. The fault detection is solved with Multiple Model Adaptive Estimation considering non-faulty and faulty (left or right surface stuck) system models. Basic reconfiguration to stabilize the flight against atmospheric disturbances is done applying the remaining surface in the lateral channel and the total energy control concept to hold the airspeed and altitude between acceptable limits in the longitudinal channel. Promising results are achieved in software-in-the-loop simulation with the fault detection and reconfiguration.

Keywords: Aircraft control, Fault detection, Fault identification, Control reconfiguration, Tracking applications.

1. INTRODUCTION

Aviation authorities in the United States (US) and European Union (EU) are developing long-term frameworks for the integration of Unmanned Aerial Systems (UAS) into their respective national airspaces. The European Aviation Safety Agency (EASA) has been developing the UAS concept of operations using a risk-based approach. In the US, the Federal Aviation Administration (FAA) regulates civilian UAS via Part 107 of the Federal Aviation Regulations. Regulatory hurdles aside, widespread use of UASs is subject to several technical challenges related to safety and reliability. These include obstacle detection, collision avoidance, path planning and routing, automated deconfliction, and on-board fault management. On-board fault management is a particularly interesting case since UAS have constraints on their size, weight, and power. Fault diagnosis and fault tolerance algorithms solve this problem without introducing hardware redundancy.

Textbooks such as Gertler (1998); Chen and Patton (1999); Isermann (2006); Ding (2008) provide details on the various model-based and data-driven fault detection methods. Hwang et al. (2010) provides a detailed survey of various fault detection, isolation, and reconfiguration methods. Fault diagnosis techniques have been applied to commercial aircraft by Goupil (2010, 2011) as well as unmanned aircraft by Freeman et al. (2013); Freeman and Balas (2013). Steinberg (2005) provides a historical overview of reconfigurable flight control. While faults can occur in a variety of places, e.g. sensors, avionics, load-bearing structures, etc., this paper will focus on actuator faults. Fault-tolerant flight control for managing actuator failures are discussed in Crider (2004); Boskovic et al. (2007); Nguyen et al. (2008); Hitachi (2009); Chowdhary et al. (2013).

This paper specifically focuses on detecting and managing actuator faults on a small UAS that is equipped with two elevons and a puller-type electric motor. This boils down to three steps: (1) detecting that a failure has occurred, (2) identifying whether the failure is in the left or the right elevon, and (3) tolerating the failure through control reconfiguration. Venkataraman and Seiler (2016) showed, using flight tests on a different UAS, that a single aerodynamic control surface is sufficient to stabilize the aircraft and execute straight & level flight and banked turns. This was achieved using robust, multi-variable control. This paper extends this key idea by developing a fault diagnosis algorithm (Section 4), a new control reconfiguration approach (Section 5), and validating both via simulations (Section 6). These three sections contain the novelty of this paper. First, however, a basic description of the aircraft and its model (Section 2) and the baseline controller (Section 3) are presented.

2. SYSTEM MODEL

The aircraft is called the Vireo and is pictured in Figure 1. It is comprised only of a wing and a fuselage. This aircraft was

⋆ 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 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".

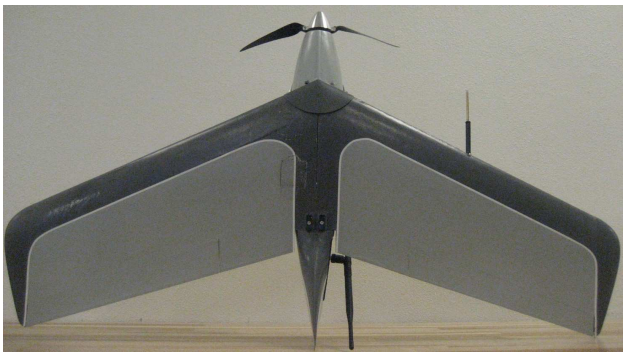

Fig. 1. The aircraft has two elevons and a motor.

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 two elevons and a puller-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. The throttle δ_T, left elevon δ_L, and right elevon δ_R are independently actuated. The deflection range of each elevon is $[-25,+25]^\circ$, where positive values correspond to trailing-edge down deflections. The particular actuators used provide for a maximum rate-of-deflection of $\pm 338^\circ\,\mathrm{s}^{-1}$.

Venkataraman (2018) developed a nonlinear, six degrees-of-freedom model for this aircraft. 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 dynamics are linearized about a steady, wings-level, constant altitude flight condition at a cruise airspeed of 15 $\mathrm{m\,s}^{-1}$. Although each elevon excites both the longitudinal and lateral-directional dynamics, they may be decomposed into the traditional elevator δ_E and the aileron δ_A via the transformation:

$$\begin{bmatrix}\delta_E\\ \delta_A\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1 & 1\\ -1 & 1\end{bmatrix}\begin{bmatrix}\delta_L\\ \delta_R\end{bmatrix}. \tag{1}$$

This allows the dynamics to be decoupled. The longitudinal dynamics G_{lon} are described by:

$$\dot{x}_{lon} = A_{lon}x_{lon} + B_{lon}u_{lon}, \tag{2}$$

where $x_{lon} = [u,w,q,\theta]^T$, $u_{lon} = [\delta_T,\delta_E]^T$,

$$A_{lon} = \begin{bmatrix}-0.15 & 0.75 & -1.02 & -9.8\\ -0.88 & -5.7 & 13.9 & -0.67\\ 0.88 & -13 & -5.5 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}, \text{ and } B_{lon} = \begin{bmatrix}6.5 & 0.15\\ 0 & -25\\ 0 & -186\\ 0 & 0\end{bmatrix}. \tag{3}$$

The lateral-directional dynamics G_{lat} are described by:

$$\dot{x}_{lat} = A_{lat}x_{lat} + B_{lat}\delta_A, \tag{4}$$

where $x_{lat} = [v,p,r,\phi]^T$,

$$A_{lat} = \begin{bmatrix}-0.42 & 1.12 & -15.3 & 9.8\\ -5.16 & -11.3 & 2.1 & 0\\ 1.4 & -1.5 & -0.66 & 0\\ 0 & 1 & 0.068 & 0\end{bmatrix}, \text{ and } B_{lat} = \begin{bmatrix}-0.49\\ -283\\ -16.7\\ 0\end{bmatrix}. \tag{5}$$

The linearized states in the models above use SI units, i.e. distances are in meters and angles are in radians. In addition, note that the evolution of ψ, p_N, p_E, and p_D are not listed above for brevity. They may be obtained from a flight dynamics textbook, such as Cook (2007).

Finally, the dynamics of the elevon actuators are described by

$$G_{act} = \frac{3940}{s^2+97s+3940} \tag{6}$$

Since (1) is a linear transformation, G_{act} may be treated as describing the dynamics of the virtual elevator actuator and the virtual aileron actuator. Later in Section 4 this actuator dynamics will be applied separately to the left and right elevons. The linear models described in this section are used to design the baseline and fault-tolerant controllers as well the fault diagnosis algorithm.

3. BASELINE CONTROLLER

Separate classical, loop-at-a-time controllers are designed for the longitudinal (Figure 2) and lateral-directional (Figure 3) dynamics. Figure 2 shows that the innermost loop implements a pitch damper by feeding back the pitch rate q. Wrapped around this is a pitch tracker that tracks a commanded pitch angle θ_c. Similarly, the innermost loop of Figure 3 implements a roll damper by feeding back the roll rate p. Wrapped around this is a roll tracker that tracks a commanded roll angle ϕ_c. Both dampers use proportional-only gains and both trackers use proportional-integral gains. Actuator dynamics are considered in both cases using G_{act} in series with the plant.

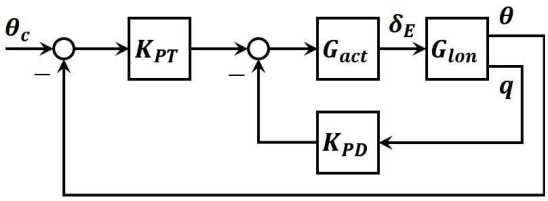

Fig. 2. The baseline longitudinal controller.

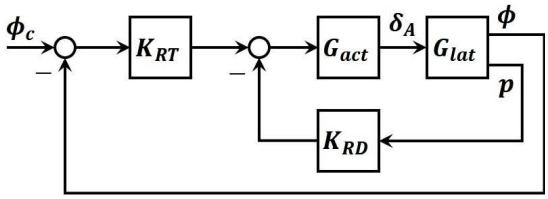

Fig. 3. The baseline lateral-directional controller.

The gains are initially designed using the models G_{lon} and G_{lat} and then iteratively updated using flight tests. The final gains are: $K_{PD} = -0.05$, $K_{PT} = -0.4 - \frac{0.2}{s}$, $K_{RD} = -0.06$, and $K_{RT} = -0.34 - \frac{0.086}{s}$. These yield a bandwidth of approximately 1.8 $\mathrm{rad\,s}^{-1}$ in tracking both θ_c and ϕ_c. The pitch command θ_c is generated by a proportional-integral controller $K_{VT} = -0.048 - \frac{0.004}{s}$ that tracks the barometric airspeed, which is defined as $V = \sqrt{u^2+v^2+w^2}$.

In addition, note that Figure 2 depicts only the elevator controller for G_{lon}. The throttle commands are generated by a proportional-integral controller $K_{AT} = 0.01 + \frac{6.66\times10^{-4}}{s}$ that tracks the altitude $h = -p_D$. Finally, outer loop guidance controllers are implemented for waypoint and flight path tracking. The continuous-time controllers described above are discretized and implemented on the flight computer of the Vireo at a sampling rate of 100 Hz. The total closed-loop time delay is calculated from flight tests to be approximately equal to 0.05 s. This encompasses delays in the actuators, the flight computer (numerical implementation of the controller), and the sensors.

4. FAULT DETECTION AND IDENTIFICATION

The so-called Multiple Model Adaptive Estimation (MMAE) framework is used to detect and isolate 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 (7). 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 4 shows the structure of the MMAE architecture.

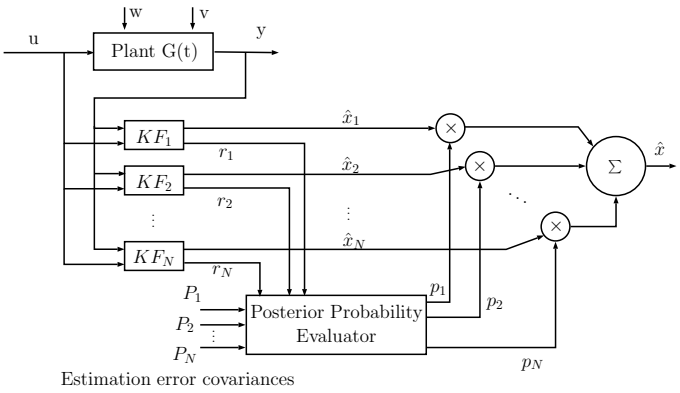

Fig. 4. The MMAE architecture

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

$$\begin{aligned} x_i(t+1) &= A_i x_i(t) + B_i u(t) + W_i w(t) \\ y_i(t) &= C_i x_i(t) + D_i u(t) + V_i v(t) \end{aligned} \tag{7}$$

where $x_i(t) \in \mathbb{R}^n$ denotes the state of the system, $u(t) \in \mathbb{R}^m$ its control input, $y_i(t) \in \mathbb{R}^p$ its measured noisy output, $w(t) \in \mathbb{R}^r$ is the state noise, and $v(t) \in \mathbb{R}^q$ is the measurement noise. Vectors $w(t)$ and $v(t)$ are zero-mean white Gaussian sequences, mutually uncorrelated with covariances $E[w(t);w(\tau)] = Q_{t\tau}$ and $E[v(t);v(\tau)] = R_{t\tau}$. The initial condition $x(0)$ of (7) is a Gaussian random vector with mean and covariance given by $E\{x_i(0)\} = x_{i_0}$ and $E\{x_i(0)x_i^T(0)\} = P_i(0)$. Matrices A_i, B_i, W_i, C_i, D_i, and V_i depend on the parameter i ($i = 1 \ldots N$). $W_i = I$ and $V_i = I$ is assumed in our case. t and $t+1$ denote consecutive discrete time steps.

One possible solution of the observer design is to obtain steady state Kalman Filters (KFs) which give state estimates $\hat{x}_i(t)$ where $i = 1 \ldots N$. As Figure 4 shows the final state estimate ($\hat{x}(t)$) is given by (8), as the weighted sum of the $\hat{x}_i(t)$ estimates provided by the observers.

$$\hat{x}(t) = \sum_{i=1}^{N} p_i(t)\hat{x}_i(t) \tag{8}$$

The $p_i(t)$ $i = 1 \ldots N$ weights are calculated inside the Posterior Probability Evaluator (PPE) block. As Figure 4 shows this block receives the output residuals $r_i(t)$ $i = 1 \ldots N$

$$r_i(t) = y(t+1) - \hat{y}_i(t+1|t)$$

where

$$\hat{y}_i(t+1|t) = C_i \bar{x}_i(t+1) + D_i u(t+1)$$

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

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

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

$$p_i(t+1) = \frac{\beta_i e^{-E_i(t+1)}}{\sum_{j=1}^{N} p_j(t)\beta_j e^{-E_j(t+1)}} p_i(t) \tag{9}$$

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

$$E_i(t+1) = r_i(t+1)^T \hat{P}_i^{-1} r_i(t+1) \tag{10}$$

$$\beta_i = \frac{1}{(2\pi)^{\frac{p}{2}} \sqrt{|\hat{P}_i|}} \tag{11}$$

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

$$\hat{P}_i = C_i P_i C_i^T + R_i \tag{12}$$

here 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.

4.1 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 (3) and (5) is in B_{lat} from aileron δ_a to roll rate p). Considering the linearized lateral dynamical model G_{lat} from (5) of the aircraft it includes the following states:

$$x = [v \;\; p \;\; r \;\; \phi]^T$$

Meanwhile (5) 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 in (1). This way the input of the considered model will be $u = [\delta_L \;\; \delta_R]^T$. 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 should be estimated on-board, but from a fault detection point of view it can be considered as known and measurable):

$$y = [r \ \phi]^T \tag{13}$$

The model 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 Section 3). 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 to the continuous time models presented in Section 2) there are two possibilities to model the time delay. The first is to make a Pade approximation and discretize that transfer function, the second is to add a chain of delay states to the DT model. The control frequency is set to 100 Hz in the system, however it is enough to run the estimators at 50 Hz which gives $0.02s$ sampling time. With this sampling time two delay state per variable should be added to approximate the measurement delay. On the contrary, at least second degree Pade approximation is required but it is better to apply fourth or fifth degree. This requires at least two, but possibly even four or five additional states per variable.

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:

$$\begin{aligned}
&x_{k+1} = Ax_k + Bu_k \\
&y_{k+1} = Cx_{k+1} \\
&\begin{bmatrix} x_{k+1} \\ x^1_{k+1} \\ x^2_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A & 0 & 0 \\ [0\ I_2] & 0 & 0 \\ 0 & I_2 & 0 \end{bmatrix}}_{A_a} \begin{bmatrix} x_k \\ x^1_k \\ x^2_k \end{bmatrix} + \underbrace{\begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix}}_{B_a} u_k \\
&y_{k+1} = \underbrace{[0\ 0\ I_2]}_{C_a} \begin{bmatrix} x_{k+1} \\ x^1_{k+1} \\ x^2_{k+1} \end{bmatrix}
\end{aligned} \tag{14}$$

$$\begin{aligned}
A &= \begin{bmatrix} 0.984 & 0.024 & -0.29 & 0.194 \\ -0.08 & 0.8 & 0.038 & -0.008 \\ 0.008 & -0.026 & 0.99 & 0.0007 \\ -0.0008 & 0.018 & 0.0016 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0.014 & -0.014 \\ 1.827 & -1.827 \\ 0.068 & -0.068 \\ 0.019 & -0.019 \end{bmatrix} \\
C &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{15}$$

Here, $dim(x^1_k) = dim(x^2_k) = 2$ are the delay states for r and ϕ and I_2 is a two dimensional unit matrix and the 50 Hz discrete time state space matrices are shown in (15). The augmented system (A_a, B_a, C_a) is also observable.

Actuator dynamics In the estimation we can use only the commanded inputs of the system (see Fig. 5) because the control surface deflections are not measured on the Vireo. This means that actuator dynamics (6) will cause a model mismatch. The transfer function model of the actuator dynamics (6) 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 (14) 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:

$$\begin{aligned}
&\begin{bmatrix} x_{k+1} \\ x^1_{k+1} \\ x^2_{k+1} \\ u(l) \end{bmatrix} = \underbrace{\begin{bmatrix} A & 0\ 0 & B(:,l) \\ [0\ I_2] & 0\ 0 & 0 \\ 0 & I_2\ 0 & 0 \\ 0 & 0\ 0 & 1 \end{bmatrix}}_{A_{aF}} \begin{bmatrix} x_k \\ x^1_k \\ x^2_k \\ u(l) \end{bmatrix} + \underbrace{\begin{bmatrix} B(:,j) \\ 0 \\ 0 \end{bmatrix}}_{B_{aF}} u(j)_k \\
&y_{k+1} = \underbrace{[0\ 0\ I_2\ 0]}_{C_{aF}} \begin{bmatrix} x_{k+1} \\ x^1_{k+1} \\ x^2_{k+1} \\ u(l) \end{bmatrix}
\end{aligned} \tag{16}$$

Here, $B(:,l)$ is the lth column of the 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. Simulation tests show that the convergence of the estimated stuck position is slow as it is assumed to be a constant state, but finally it fits well to the real stuck value (see the commanded δ_L signal (which is the estimated stuck position) relative to the real one in Fig. 9). This means that regular updates of the estimated stuck position in the reconfiguration should be done.

4.2 MMAE design and application

KFs for the nominal and the faulty (F) models were designed by assuming reasonable system and measurement noise. The nominal system noise covariance matrix is selected as:

$$Q_N = <0.5^2 \ \ (2\pi/180)^2 \ \ (2\pi/180)^2 \ \ (2\pi/180)^2 \ \ 0 \ \ 0 \ \ 0 \ \ 0>$$

for the faulty system models the additional stuck state noise is 10^{-6} making it possible to have a slowly changing stuck position in the filter and so converge to the real stuck position. The covariance matrix is:

$$Q_F = <0.5^2 \ \ (2\pi/180)^2 \ \ (2\pi/180)^2 \ \ (2\pi/180)^2 \ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 10^{-6}>$$

The matrices show that $0.5m/s$ standard deviation is considered for the lateral velocity and $2^\circ/s$ or 2° for the angular rates and

angles respectively. The measurement noise covariance matrix is:

$$R = < (2\pi/180)^2 \;\; (5\pi/180)^2 >$$

which shows that the standard deviations of the yaw rate and roll angle are $2^\circ/s$ and 5° respectively. Here, $<>$ means a diagonal matrix. The considered noisy state equations in the KF design were:

$$\begin{aligned} x_{k+1} &= A_{a(F)}x_k + B_{a(F)}u_k + w_k \\ y_{k+1} &= C_{a(F)}x_{k+1} + v_{k+1} \end{aligned} \tag{17}$$

Here, A_a is the matrix of the nominal augmented system, while $A_{a(F)} = A_{aF}$ 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. The designed KFs will give the predictions and estimates of the augmented state vector. Note that the actual predicted $\bar{p}$ roll rate is not delayed by the model system that is why it should be delayed with 2 discrete time steps ($0.04s$) before forming a residual from it. Running the MMAE in the software-in-the-loop (SIL) simulation has shown that the yaw rate (r) and roll angle (ϕ) residuals are one order of magnitude 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 a measured output. That's why simple diagonal scaling was applied for the residuals of all filter which increases the magnitude of the yaw rate and roll angle parts

$$\begin{aligned} RES &= \left[p_{meas} - \bar{p},\; r_{meas} - \bar{r},\; \phi_{meas} - \bar{\phi}\right] W \begin{bmatrix} p_{meas} - \bar{p} \\ r_{meas} - \bar{r} \\ \phi_{meas} - \bar{\phi} \end{bmatrix} \\ W &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix} \end{aligned} \tag{18}$$

These residual values are applied in a modified PPE which calculates fictitious probabilities in the following way:

$$p_i(t+1) = \frac{e^{-RES_i(t+1)}}{\Sigma_{j=1}^{3} e^{-RES_j(t+1)} p_j(t)} p_i(t) \tag{19}$$

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 (19)).

5. BASIC RECONFIGURATION

In case of the stuck fault of one surface on the Vireo the remaining actuation possibilities will become very limited. One has one movable surface and the engine to control the whole aircraft. The first step can be to set the trim position of the moving surface to the position of the stuck one to not to generate roll moment in the trim point (that's why estimation of the stuck position through the MMAE is also important). This can only be done if the stuck position does not cause too much pitching moment so it is around the straight and level flight trim point.

Then the roll motion (lateral dynamics) of the aircraft can be controlled by moving the operational surface relative to this new trim point. The roll tracker gains K_{RD}, K_{RT} can be simply doubled ($K_{RD}^{rc} = 2K_{RD}$ and $K_{RT}^{rc} = 2K_{RT}$ where superscript rc means the reconfigured roll gain) to substitute the effect of the other surface's deflection (assuming linear relation between roll moment and surface deflection).

The only remaining actuation possibility is to use the throttle to somehow control the airspeed and altitude (longitudinal dynamics) of the aircraft. Controlling two independent variables with one control input is known to be impossible but a modification of the total energy control (TECS) concept (see Lambregts (1983a), Lambregts (1983b), Beard (2014) and Argyle and Beard (2016)) can possibly keep the airspeed and altitude and so the total energy of the aircraft in an acceptable range by adding or removing energy from the system through the engine thrust. This requires that the longitudinal dynamics should be inherently stable. In case of the Vireo aircraft the longitudinal poles are as follows:

$$-0.105 \pm 0.884i,\; -5.62 \pm 12.7i$$

so the longitudinal modes are all stable. The total energy is the sum of potential and kinetic energy. By controlling its value the system can possibly hold altitude and airspeed within acceptable limits. So, the throttle can be controlled by a PI controller (with $K_{TT} = 10^{-3} + \frac{2 \cdot 10^{-4}}{s}$ gain) relative to its trim value ($\delta_T(0)$) based-on the error between the reference and actual total energies:

$$\begin{aligned} \delta_T &= \delta_T(0) + K_{TT}\Delta E \\ \Delta E &= mg(h_{ref} - h) + \frac{1}{2}m(V_{ref}^2 - V^2) \end{aligned} \tag{20}$$

Here, h_{ref} and V_{ref} are the reference altitude and velocity values, m is the mass of the aircraft and g is the gravitational constant. Examining deeper this control law shows that ΔE can be zero in three cases:

(1) State 1: $h = h_{ref},\; V = V_{ref}$
(2) State 2: $h < h_{ref},\; V > V_{ref}$
(3) State 3: $h > h_{ref},\; V < V_{ref}$

State 1 is the targeted state where the system holds altitude and airspeed. State 2 and 3 can be theoretically arbitrary other states satisfying $V = \sqrt{2g(h_{ref} - h) + V_{ref}^2}$. However, both of them characterizes transient points of the Phugoid motion as in state 2 the aircraft should ascend and decelerate and in state 3 the aircraft should descend and accelerate if the Phugoid motion is stable (see Cook (2007)) approaching state 1 in both cases. In a forthcoming work this stability should be examined in detail through Lyapunov theory for example.

Summing up the things, after reconfiguration the lateral dynamics (trajectory tracking) of the aircraft can be controlled through roll motion with the remaining movable elevon. The longitudinal dynamics (altitude and airspeed) can be regulated by applying the total energy concept to control the throttle of the engine. In the next section the SIL environment and test results will be introduced.

6. SIL SIMULATION RESULTS

The SIL simulation scheme can be seen in Fig. 5.

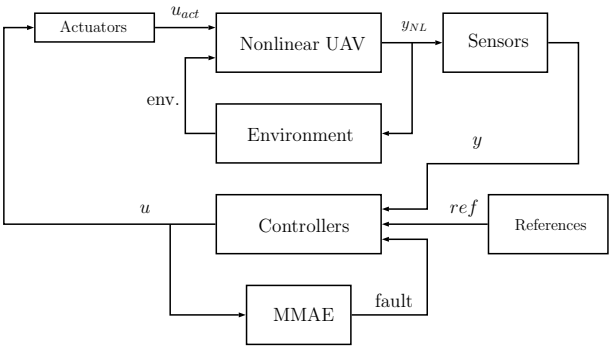

Fig. 5. The scheme of the SIL simulation model

In the scheme the MMAE block includes the actuator model (6) and fault detection and identification algorithm from Section 4, the Controllers block includes the baseline (Section 3) and reconfiguration (Section 5) controllers together with the switching between them. The Environment block includes an Atmosphere model, the Word Magnetic and Gravity models together with deterministic wind, wind gust and turbulence (Dryden wind turbulence) models. The References block provides heading, indicated airspeed (IAS) and altitude references to the tracking controllers.

Test runs are done for straight and level flight (15 m/s IAS, 100 m altitude, 155 degrees heading) of the aircraft applying turbulence disturbances through the Dryden wind turbulence model. Without these disturbances the effect of fault application and reconfiguration could remain hidden for the steady straight and level flight case. The 6 m (20 ft) wind velocity was set to 7.72 m/s modeling light turbulence according to DoD (2004). Three runs were done, one with the nominal system (150 seconds) (Nominal), one with stuck fault of the left surface at 50 seconds without reconfiguration (Faulty) and one with stuck fault and reconfiguration (Reconfigured). The results are summarized in Fig.s 6 to 11.

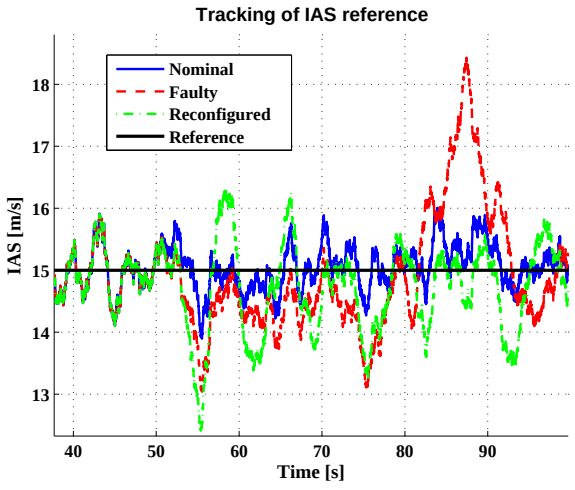

Fig. 6. The tracking of IAS (zoomed for better readability)

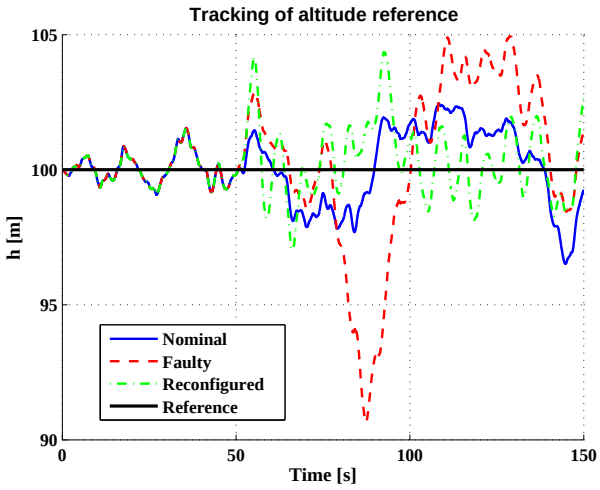

Fig. 7. The tracking of altitude

Fig. 6 presents the IAS tracking results for Nominal, Faulty and Reconfigured cases with 15 m/s constant reference. After the activation of the fault the tracking errors in the faulty case are larger than in the nominal and there is a high peak at about 75 s. In case of the reconfigured faulty flight the errors are larger than in the nominal case and smaller or equal then in the faulty case. There is no large peak with reconfiguration.

Fig. 7 presents the altitude tracking results for Nominal, Faulty and Reconfigured cases with 100 m constant reference. After the activation of the fault the tracking errors in the faulty case are most of the time larger than in the nominal and there is a low peak at about 75 s. In case of the reconfigured faulty flight the errors are most of the time smaller then in the faulty case and even smaller then in the nominal case. There is no large peak with reconfiguration.

Fig.s 8 and 9 show the commanded and real elevon deflections with fault activation at 50 s without (Fig. 8) and with (Fig. 9) reconfiguration. If there is no reconfiguration the left elevon command can increase to about 25 degrees while the real deflection is fixed at about 0.9 degrees. The right elevon command works about normally until the complete turn at about 75 s. Finally, the turning is compensated out and tracking of the given heading is continued as the trajectory parallel with the reference line in Fig. 10 shows. In case of reconfiguration after the detection of the fault at about 53 s the commanded left deflection (which is the output of the left stuck MMAE filter) is close to the stuck value. Neglecting the switching transient the right elevon works normally during the whole time.

Regarding the trajectory tracking (Fig. 10) only the heading reference (155 degrees) is given and tracked no waypoint or straight line tracking is implemented. However, in the Nominal and Reconfigured cases the straight flight is pretty nice there are only small differences from the assumed reference line (not visible in the figure). On the contrary without reconfiguration there is a circular trajectory segment (complete turn) and then there is a large deviation from the line. Of course, in case of waypoint or line tracking these errors would be smaller but turbulence can strongly affect the flight qualities if the aircraft is not reconfigured. The peaks of the IAS and altitude tracking errors are during this complete turn. These trajectory tracking results show the effectiveness of the proposed reconfiguration

strategy which makes the completion of a mission with a reconfigured aircraft possible.

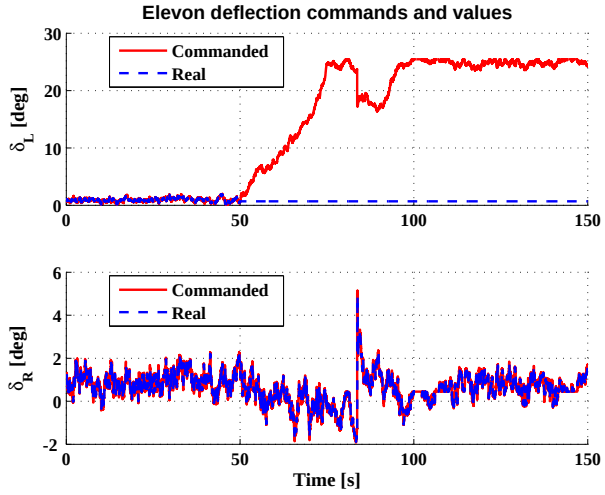

Fig. 8. Elevon commands and deflections without reconfiguration

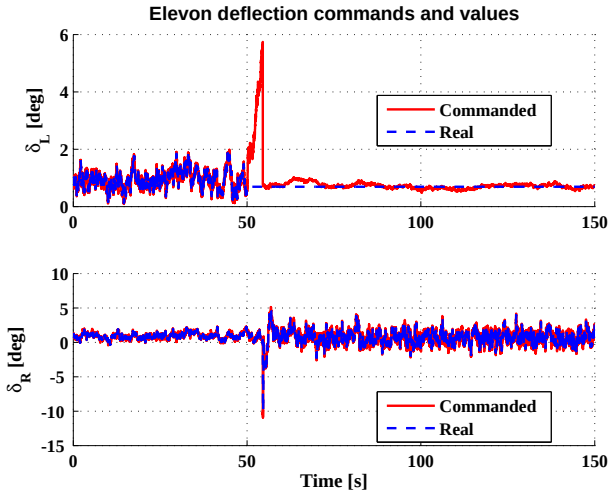

Fig. 9. Elevon commands and deflections with reconfiguration

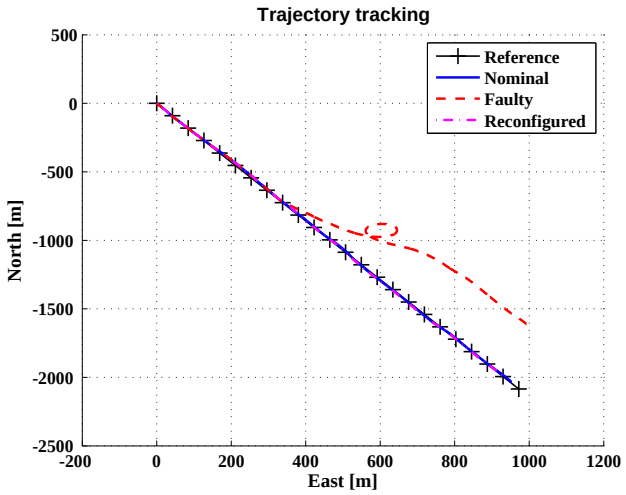

Fig. 10. Tracking of straight trajectory

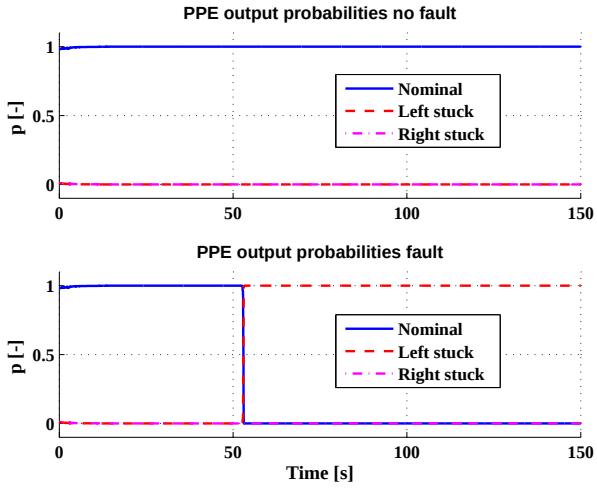

Fig. 11. Resulting probabilities from the PPE

Fig. 11 shows the probability outputs in the Nominal case and in the Faulty case. In the nominal case the probability of nominal working mode is about 1 and the other probabilities are about zero all the time. In the faulty case there is a sudden change around 53 s. The left stuck probability moves to 1 and the nominal moves to 0. This shows really good detection of the fault.

Summarizing the results one can conclude that both fault detection and identification and reconfiguration work well and the aircraft can continue its mission. There is no observable difference in the tracking of the heading angle with the nominal or the reconfigured controller. The IAS tracking error changes from about $\pm 0.5 m/s$ to about $\pm 1.3 m/s$ but this is acceptable if does not cause stall or overload of the structure. The altitude tracking results are slightly better with the reconfigured controller ($\pm 2m$) than with the normal control ($\pm 2.5m$). The fault was activated at 50 s and detected at 53 s which seems to be a long time, but as there is only turbulence disturbance the reconfiguration is able to stabilize the aircraft. In case of larger disturbances the detection filters would react faster (as they would get larger excitation) and so detection and reconfiguration could be done earlier.

7. CONCLUSION

This paper has dealt with the in-flight fault detection, identification and reconfiguration control of a small unmanned aircraft equipped with two elevons and an electric engine.

A nonlinear model of the aircraft was developed. This model was trimmed and linearized at its cruise airspeed, and the resulting linear models are then used to design the controllers and fault diagnosis algorithms.

Stuck fault of one elevon is assumed during the mission of the aircraft. The fault detection and identification is solved with Multiple Model Adaptive Estimation defining three different lateral system models, one with all elevons working, one with left elevon stuck and one with right elevon stuck. LTI Kalman Filters are designed for all models and their residuals are processed by a Posterior Probability Evaluator (PPE). The output of the PPE gives the probabilities with which each model is valid. It is proven in the literature that the probability of

the model closest to the actual working mode of the system converges to one, the others converge to zero.

Reconfiguration of the aircraft flight control system is solved considering the estimated position of the stuck surface. First, all elevon trim points are set to this value to zero out the lateral moments resulting from asymmetric deflection (this can only be applied if the stuck position is small, otherwise there will be too large pitching moment, but in a normal flight mission this should be working). Second, the lateral dynamics of the aircraft is controlled through the remaining working elevon by doubling the roll gains of the controller. Third the IAS and altitude are held between acceptable limits by applying total energy control through the throttle of the engine. This requires that the longitudinal dynamics of the aircraft should be stable.

A software-in-the-loop (SIL) simulation of the aircraft is built including the full nonlinear model together with environmental effects such as wind turbulence. This model is completed by the fault detection and identification and reconfiguration components. Straight and level flight simulations are done with the nominal system, with a left surface stuck fault without reconfiguration and a left surface stuck fault with reconfiguration. Results show that both the fault detection and identification and the reconfiguration methods work well and the aircraft can continue the mission despite the stuck fault.

Future research should include proof of the stability of the total energy control, SIL tests with turns and altitude and IAS reference changes, then implementation and real flight test of fault detection and identification and finally reconfiguration on the Vireo aircraft. Before flight testing, it is advisable to explore the deflection ranges of the elevons during the planned mission and check if the longitudinal channel can be stabilized if the stuck position is the possible maximum during the mission.

ACKNOWLEDGEMENTS

The first author would like to thank Peter J. Seiler, Daniel Ossmann, Raghu Venkataraman, Curtis Olson and Christopher Regan for their support and hospitality during his stay in Summer 2016 and 2017 in Minneapolis which led to the results of this article.

REFERENCES

Argyle, M.E. and Beard, R.W. (2016). Nonlinear Total Energy Control for the Longitudinal dynamics of an aircraft. In *2016 American Control Conference (ACC)*, 6741–6746. doi: 10.1109/ACC.2016.7526733.

Beard, R.W. (2014). UAVBook Supplement Total Energy Control for Longitudinal Autopilot. URL `http://uavbook.byu.edu/-lib/exe/fetch.php?media=shared:-tecs_autopilot.pdf`.

Boskovic, J.D., Prasanth, R., and Mehra, R.K. (2007). Retrofit fault-tolerant flight control design under control effector damage. *Journal of Guidance, Control, and Dynamics*, 30, 703712.

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

Chowdhary, G., Johnson, E.N., Chandramohan, R., Kimbrell, M.S., and Calise, A. (2013). Guidance and control of airplanes under actuator failures and severe structural damage. *Journal of Guidance, Control, and Dynamics*, 36, 1093–1104.

Cook, M.V. (2007). *Flight Dynamics Principles*. Elsevier, second edition.

Crider, L.D. (2004). Control of commercial aircraft with vertical tail loss. In *AIAA 4th Aviation Technology, Integration and Operations (ATIO) Forum*.

Ding, S.X. (2008). *Model-Based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*. Springer-Verlag, Germany, first edition.

DoD (2004). Flying Qualities of Piloted Aircraft.

Freeman, P. and Balas, G. (2013). Analytical fault detection for a small uav. In *AIAA Infotech@Aerospace Conference*. Boston, MA, USA.

Freeman, P., Pandita, R., Srivastava, N., and Balas, G. (2013). Model-based and data-driven fault detection performance for a small UAV. *IEEE Transactions on Mechatronics*, 18(4), 1300–1309.

Gertler, J.J. (1998). *Fault detection and diagnosis in engineering systems*. Marcel Dekker, first edition.

Goupil, P. (2010). Oscillatory failure case detection in the A380 electrical flight control system by analytical redundancy. *Control Engineering Practice*, 18(9), 1110–1119.

Goupil, P. (2011). AIRBUS state of the art and practices on FDI and FTC in flight control system. *Control Engineering Practice*, 19(6), 524 – 539.

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

Hitachi, Y. (2009). *Damage-Tolerant Control System Design for Propulsion-Controlled Aircraft*. Master's thesis, University of Toronto.

Hwang, I., Kim, S., Kim, Y., and Seah, C.E. (2010). A survey of fault detection, isolation, and reconfiguration methods. *IEEE Transactions on Control Systems Technology*, 18(3), 636–653.

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

Lambregts, A.A. (1983a). Integrated system design for flight and propulsion control using total energy principles. In *In Proc. of the Aircraft Design, Systems and Technology Meeting*. AIAA, Forth Worth, TX, USA.

Lambregts, A.A. (1983b). Vertical flight path and speed control autopilot design using total energy principles. In *In Proc. of Guidance and Control Conference*. AIAA, Gatlinburg, TN, USA.

Nguyen, N., Krishnakumar, K., and Kaneshige, J. (2008). Flight dynamics and hybrid adaptive control of damaged aircraft. *Journal of Guidance, Control, and Dynamics*, 31, 751–764.

Steinberg, M. (2005). Historical overview of research in reconfigurable flight control. *Journal of Aerospace Engineering*, 219, 263–275.

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

Venkataraman, R. and Seiler, P.J. (2016). Safe flight using one aerodynamic control surface. In *AIAA Guidance, Navigation, and Control Conference*.