Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Fault-tolerant Optimal Control Scheme for Satellite Micro-launchers

Alexandra Reitu *, **Andrei Sperilă** *, **Bogdan D. Ciubotaru** *

* *Automatic Control and Systems Engineering Department, University Politehnica of Bucharest (e-mail: alexandra.reitu@stud.acs.pub.ro, {andrei.sperila, bogdan.ciubotaru}@acse.pub.ro).*

Abstract: An integration of Fault Detection, Isolation and Recovery (FDIR) with the Linear Quadratic Gaussian (LQG) technique is presented, which achieves fault tolerance while maintaining control optimality. The FDIR scheme is tested on a space micro-launcher model, and simulation results show successful accommodation of both sensor and actuator faults.

Keywords: Aerospace control, LQG control, Fault detection, Fault diagnosis, Fault tolerant systems.

1. INTRODUCTION

The concept of FDIR is being studied widely in the literature since the beginning of the 1970s and rapidly became a mandatory component in most aerospace applications. Initially based around physical redundancy for both detection and recovery, most FDIR schemes at present are model-based and use analytical redundancy. A thorough analysis of the subject can be found in Chen and Patton (2012), Blanke et al. (2016) and Isermann (2011). Other valuable case studies are also presented in Frank (1990) and Lou et al. (1986), as well as in Patton et al. (2000).

During the same period, LQG control became the preeminent technique employed in the aerospace industry due to its tractable tuning procedure for nominal performance and its effectiveness in minimizing sensor noise. A complete analysis on the subject is presented in Kwakernaak and Sivan (1972), as well as in Anderson and Moore (2007) and Zhou et al. (1995).

The procedure introduced in this paper is developed for a satellite micro-launcher prototype, presented in Diaconescu et al. (2019) [1], consisting of a new recovery technique that brings together FDIR algorithms as well as LQG control: instead of switching between separate controllers for nominal and faulty functioning, the optimal controller is adapted by only recomputing and replacing its control and filtering feedback matrices, thus maintaining optimality and avoiding command signal shocks that usually result from controller switching.

The paper is organized as follows. Section 2 presents the theoretical concepts pertaining to LQG control and FDI. Section 3 focuses on the implementation details of the novel recovery procedure. Section 4 showcases the effectiveness of the proposed technique by simulating two typical in-flight fault scenarios. Section 5 concludes with a set of closing remarks.

[1] This prototype is under analysis in the European Space Agency contract, number 4000119953/17/F/JLVs, Advanced Control Techniques for Future Launchers.

2. SYSTEM MODEL AND CONTROL DESIGN

2.1 System model

Accounting for the strong non-linear characteristics of the micro-launcher's model, multiple linear approximations are needed in order to capture the launcher's dynamics. These may vary greatly depending on the micro-launcher's location on its prescribed trajectory, therefore each linear approximation is valid only for a short amount of time, as trajectory phases change along with variations in atmospheric conditions.

The solid-body dynamics of the micro-launcher are fully detailed in Chelaru (2018), and may be given in state-space form as

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ a_{\dot{\theta}}^{\theta} & 0 & a_{\dot{\theta}}^{\dot{z}} \\ a_{\dot{z}}^{\theta} & 0 & a_{\dot{z}}^{\dot{z}} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ b_{\dot{\theta}}^{\delta} & a_{\dot{\theta}}^{\alpha_\omega} \\ b_{\dot{z}}^{\delta} & a_{\dot{z}}^{\alpha_\omega} \end{bmatrix} \begin{bmatrix} \delta_n \\ \alpha_\omega \end{bmatrix}, \qquad (1)$$

where $a_{\dot{\theta}}^{\theta}$, $a_{\dot{\theta}}^{\dot{z}}$, $a_{\dot{z}}^{\theta}$, $a_{\dot{z}}^{\dot{z}}$, $b_{\dot{\theta}}^{\delta}$, $a_{\dot{\theta}}^{\alpha_\omega}$, $b_{\dot{z}}^{\delta}$, $a_{\dot{z}}^{\alpha_\omega}$ denote atmospheric coefficients that vary according to linearization conditions. The states of the system are:

- θ - pitch angle (deg),
- $\dot{\theta}$ - pitch rate (deg/sec),
- $\dot{z}$ - lateral drift speed (m/sec),

and the inputs are:

- δ_n - TVC nozzle deflection (deg),
- α_ω - wind incidence (deg),

where TVC abbreviates for "Thrust Vector Control".

For the application presented in this paper, a discrete-time linear approximation of the micro-launcher dynamics was used. The considered time frame for which the approximation is valid is from 0 to 17 seconds after the launch. Wind incidence dynamics have been ignored, and the system has been discretized with a sample time of $T_s = 0.1$ seconds. Moreover, in order to implement a fault-recovery module, it was considered that the system's first two states, θ and $\dot{\theta}$, are being measured redundantly by two sensors each. In general, the likelihood that two sensors which measure the same signal may both fail is considered to be negligible. The final discrete-time state-space realization

is given numerically by

$$A_{ML} = \begin{bmatrix} 1.0004 & 0.1000 & -0.0000 \\ 0.0077 & 1.0004 & -0.0002 \\ 3.2642 & 0.1651 & 0.9318 \end{bmatrix}, B_{ML} = \begin{bmatrix} 0.1479 \\ 2.9577 \\ -3.0868 \end{bmatrix},$$

$$C_{ML} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \qquad D_{ML} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The TVC actuator model acts as a low-pass filter and its continuous-time model is given by

$$G_{TVC}(s) = K\frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}e^{-s\tau}, \tag{2}$$

where $\tau = 0.015$ seconds. Although the chosen sampling time for discretization is greater than the input delay, the former was chosen as the smallest quantity which preserves the stabilizability of the system after discretization. The discrete-time state-space realization for the TVC actuator, discretized in the same manner as the rigid body, is then as follows

$$A_{TVC} = \begin{bmatrix} 4.127e-05 & 0.0003686 & -0.0005781 \\ -0.000173 & -0.0002871 & 0.0001213 \\ -2.428e-05 & -4.076e-06 & -4.767e-05 \end{bmatrix},$$

$$B_{TVC} = \begin{bmatrix} 1.109 \\ -0.1052 \\ -0.2693 \end{bmatrix}, \; C_{TVC} = \begin{bmatrix} 0.243 \\ -0.001977 \\ 0.01177 \end{bmatrix}^T,$$

$$D_{TVC} = \begin{bmatrix} 0.7334 \end{bmatrix}.$$

The output of the TVC actuator, $u_p[k]$, will be measured by two sensors in order to ensure that input-fault detection is both possible and reliable. However, as its values are not employed in the control feedback loop, the two TVC actuator measurements will not be explicitly included in the final state-space model.

2.2 The discrete-time Riccati equation

The discrete-time Riccati equation and a series of computationally-efficient algorithms designed to solve it are extensively covered in Laub (1979). The general expression of the equation is given by

$$A^TXA - X - (A^TXB + S)(B^TXB + R)^{-1}\times \\ \times(B^TXA + S^T) + Q = 0. \tag{3}$$

For most practical applications, the matrix S which quantifies state-input interaction is taken 0. Sufficient conditions for the existence of a unique stabilizing solution may be taken as:

- (A,B) is stabilizable;
- $\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} > 0$ or, if $S = 0$, simply $Q > 0$ and $R > 0$.

The discrete-time Riccati equation is solved by computing a basis U for the stable deflation subspace of the symplectic matrix

$$Z = \begin{bmatrix} A + BR^{-1}B^TA^{-1}Q & -BR^{-1}B^TA^{-T} \\ -A^{-T}Q & A^{-T} \end{bmatrix}. \tag{4}$$

If we write U as

$$U = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \in \mathbb{R}^{2n\times n}, \tag{5}$$

then the solution to the Riccati equation is computed as

$$X = U_2U_1^{-1}. \tag{6}$$

For a comprehensive treatment of this subject, one can consult Bittanti et al. (1991) and Ionescu et al. (1999).

2.3 Controller synthesis

In order to minimize measurement noise and disturbance effects on the system's sensors, a Linear Quadratic Gaussian (LQG) control algorithm will be implemented. Sensor fault detection algorithms are usually heavily influenced by measurement noise, therefore LQG control ensures that we use the optimal controller for a given noise level. The procedure for implementing LQG control algorithms is given in detail in Maciejowski (2007) and Skogestad and Postlethwaite (2007).

The micro-launcher linear model is unstable but fully controllable and observable, while the TVC actuator model is stable. For controller synthesis, ensuring closed-loop stability, good stability margins and reference tracking is vital. A schematic of the control loop is given in Fig. 1.

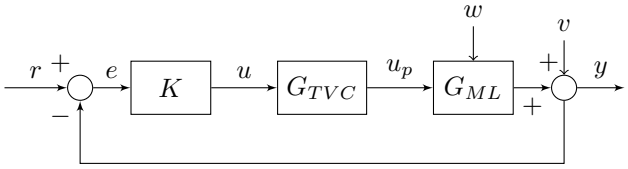

Fig. 1. Control loop signals and transfers

For reference tracking, an integrator was included in the open-loop transfer, in accordance with the Internal Model Principle. The augmented open-loop transfer for negative output feedback now becomes

$$P_a = -G_{ML}G_{TVC}\frac{T_s}{z-1} = \left[\begin{array}{c|c} A_a & B_a \\ \hline C_a & D_a \end{array}\right]. \tag{7}$$

The stabilizing LQG controller's realization is then

$$K_s = \left[\begin{array}{c|c} A_a + B_aF + LC_a + LD_aF & -L \\ \hline F & 0 \end{array}\right], \tag{8}$$

where the matrices L, F are the stabilizing solutions of the discrete-time filtering,

$$A_aYA_a^T - Y - A_aYC_a^T(C_aYC_a^T + R_L)^{-1}C_aYA_a^T + Q_L = 0, \tag{9}$$

and control,

$$A_a^TXA_a - X - A_a^TXB_a(B_a^TXB_a + R_F)^{-1}B_a^TXA_a + Q_F = 0, \tag{10}$$

Riccati equations, along with the feedback matrices

$$L^T = -(C_aYC_a^T + R_L)^{-1}C_aYA_a^T, \tag{11}$$

$$F = -(B_a^TXB_a + R_F)^{-1}B_a^TXA_a, \tag{12}$$

for the filtering and control cases, respectively. For the control equation, the weighting matrices,

$$R_F = 1,$$
$$Q_F = diag\{1, 10, 1, 1, 1, 1, 1, 1, 1\},$$

are chosen empirically, by tuning so as to obtain the best compromise between overshoot and settling time. For the filtering equations, the weighting matrices Q_L and R_L are chosen to ensure that a stabilizing solution to the filtering Riccati equation exists and that the filtering is optimal with regard to measurement noise, respectively

$$R_L = E\{v[n]v^T[n]\} = diag\{\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2\}, \tag{13}$$

$$Q_L = \epsilon I_n, \; \epsilon \to 0, \; \epsilon > 0. \tag{14}$$

Here, Q_L is the covariance matrix of the internal process noise, $Q_L = E\{w[n]w^T[n]\}$. The internal noise is considered to be zero, but in order to ensure that the filtering Riccati equation has a stabilizing solution, Q_L was defined

as a positive definite matrix with minute eigenvalues. For most spaceflight applications, a measurement noise with a standard deviation of around 1% of usual signal amplitudes is considered moderately-high. We shall employ this level of noise and, since simulations in this paper have unit-level amplitudes for most measured signals, we will consider all measurement noise to have variances of $\sigma_{(\cdot)} = 10^{-4}$. Finally, the integrator is folded into the stabilization block and the LQG controller is given as

$$K = \frac{T_s}{z-1} K_s. \tag{15}$$

2.4 Fault detection and isolation

Fault Detection and Isolation (FDI) schemes are usually based on the generation of residual signals. A residual has the purpose of monitoring one or more values of interest for the system, and to indicate when a non-typical, faulty behavior occurs. For discrete-time systems, model based residual generators use parity relations to verify the consistency of a value. This method is thoroughly analyzed in Chen and Patton (2012). Next, we present a residual generation method based on temporal redundancy. We consider the following discrete-time linear model

$$\begin{cases} x[k+1] = Ax[k] + Bu[k] + R_1 f[k] \\ y[k] = Cx[k] + Du[k] + R_2 f[k] \end{cases} \tag{16}$$

where $x \in \mathbb{R}^n$ is the state vector, $y \in \mathbb{R}^m$ is the output vector, $u \in \mathbb{R}^r$ is the input vector and $f \in \mathbb{R}^g$ is the fault vector.

The temporal-redundancy parity relations are constructed by using the model equations from time sample $k-s$ until the present time sample, k. As given in Chen and Patton (2012), these are written as follows

$$\begin{bmatrix} y[k-s] \\ y[k-s+1] \\ \vdots \\ y[k] \end{bmatrix} - H \begin{bmatrix} u[k-s] \\ u[k-s+1] \\ \vdots \\ u[k] \end{bmatrix} = Wx[k-s] + M \begin{bmatrix} f[k-s] \\ f[k-s+1] \\ \vdots \\ f[k] \end{bmatrix} \tag{17}$$

or with simplified notation

$$Y[k] - HU[k] = Wx[k-s] + MF[k], \tag{18}$$

where we define the matrices H, W, M as

$$H = \begin{bmatrix} D & 0 & \dots & 0 \\ CB & D & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{s-1}B & CA^{s-2}B & \dots & D \end{bmatrix} \in \mathbb{R}^{(s+1)\times(s+1)r}, \tag{19}$$

$$W = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^s \end{bmatrix} \in \mathbb{R}^{(s+1)m\times n}, \tag{20}$$

$$M = \begin{bmatrix} R_2 & 0 & \dots & 0 \\ CR_1 & R_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{s-1}R_1 & CA^{s-2}R_1 & \dots & R_2 \end{bmatrix} \in \mathbb{R}^{(s+1)\times(s+1)r}. \tag{21}$$

Matrix M cannot be computed, as the faults dynamics cannot be readily known.

Considering the previous parity relation, a residual signal can be defined as

$$r[k] = V(Y[k] - HU[k]) = VWx[k-s] + VMF[k], \tag{22}$$

where V is computed such that the residual will not be affected by the nominal dynamics of the process, but will respond to stimulation by fault signals. Therefore, the matrix V must ensure that the following

$$VW = 0 \Rightarrow V^T \in \ker(W^T), \text{ while } VM \neq 0. \tag{23}$$

In order for V to exist, the number of time samples s used for time redundancy must be large enough to ensure that W has more rows than columns, so that it is guaranteed to have a left null space with V taken as its basis. Finally, the residual signal in extended form is

$$r[k] = V \left(\begin{bmatrix} y[k-s] \\ y[k-s+1] \\ \vdots \\ y[k] \end{bmatrix} - H \begin{bmatrix} u[k-s] \\ u[k-s+1] \\ \vdots \\ u[k] \end{bmatrix} \right). \tag{24}$$

In addition, to help with the detection of slowly-varying fault signals, the residuals are further filtered as follows

$$r_f[k] = \frac{z+0.5}{z-1} r[k].$$

3. FAULT ISOLATION AND RECOVERY SCHEME

3.1 Isolation and recovery for sensor fault

The detection procedure for the plant's outputs was implemented with a separate residual generator for each of the four sensors. This approach ensures the possibility of isolating a fault in any of the sensors, whereas implementing a single residual generator for all four sensors would result in the residual signals interacting with each other in a way that would make fault isolation very difficult to achieve. The dynamics of each output was used separately to implement the four residual generators. The relations are as follows:

$$G = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{c|c} A & B \\ \hline C_1 & D_1 \\ C_2 & D_2 \\ C_3 & D_3 \\ C_4 & D_4 \end{array}\right] = G_{ML}\ G_{TVC}, \tag{25}$$

$$G_i = \left[\begin{array}{c|c} A & B \\ \hline C_i & D_i \end{array}\right],\ i = \overline{1:4}, \tag{26}$$

$$r_i[k] = V_i \left(\begin{bmatrix} \hat{y}_i[k-s] \\ \hat{y}_i[k-s+1] \\ \vdots \\ \hat{y}_i[k] \end{bmatrix} - H_i \begin{bmatrix} u[k-s] \\ u[k-s+1] \\ \vdots \\ u[k] \end{bmatrix} \right), \tag{27}$$

$$r_i[k] = V_i(\hat{Y}_i[k] - H_i U[k]),\ i = \overline{1:4}\ . \tag{28}$$

Taking into account the effect of measurement noise on individual residual signals, optimal Kalman filters were used in order to estimate the real value of each measured output. The estimated values were then used for residual generation, therefore minimizing the effect of the measurement noise on the residual signals.

The estimated outputs were computed as follows

$$\begin{cases} \dot{\hat{x}}_i = (A + L_i C_i)\hat{x}_i + [B + L_i D_i \quad -L_i] \begin{bmatrix} u \\ y_i \end{bmatrix}, \\ \hat{y}_i = C_i \hat{x}_i + D_i u, \qquad\qquad i = \overline{1:4}, \end{cases} \tag{29}$$

where $\hat{x}_i$ is the estimated state vector of the i-th observer and the matrix L_i is computed using the solution of the discrete-time Riccati equation

$$AY_iA^T - Y_i - AY_iC_i^T(C_iY_iC_i^T + R_i)^{-1}C_iY_iA^T + Q_i = 0, \tag{30}$$

$$L_i^T = -(C_iY_iC_i^T + R_i)^{-1}C_iY_iA^T, \tag{31}$$

where $Q_i = \epsilon I_n$ and $R_i = \sigma_i^2$, $i = \overline{1:4}$. The recovery module for sensor faults consists in adapting the controller

by modifying its state matrices. As seen in section II, using LQG controllers the noise effect can be penalized by choosing the diagonal values of R_L as the value of the noise variance. When there are no faults, the matrix R_L is defined as in (13).

If there is a fault on one of the sensors, then a controller that strongly penalizes its value can be computed by increasing the sensor's associated noise variance: $\sigma_f \to \infty$. Considering all cases for sensor faults, and assuming that for each of the plant's outputs (pitch angle and pitch rate) only one of the two corresponding sensors can be faulty at any given time, 9 different matrices R_{L_i} were used. For the nominal case, the matrix given in (13) was used, while for the other 8 cases it was modified by replacing the value of $\sigma_k,\ k = \overline{1:4}$, with σ_f, depending on which of the four sensors became faulty. In order to consider each case of sensor fault, 9 different matrices L_i were computed. For each matrix L_i a discrete-time Riccati equation was solved, respectively

$$A_a Y_i A_a^T - Y_i - A_a Y_i C_a^T (C_a Y_i C_a^T + R_{L_i})^{-1} \times \\ \times C_a Y_i A_a^T + Q_L = 0, \tag{32}$$

$$L_i^T = -(C_a Y_i C_a^T + R_{L_i})^{-1} C_a Y_i A_a^T,\ i = \overline{1:9}. \tag{33}$$

Thus, 9 different controllers were computed for the sensor's recovery module. For each controller, only the matrix L would change. The recovery module ensures that when a fault is detected and isolated, the proper L_i is used. The stabilizing controller is as follows

$$K_{s_i} = \left[\begin{array}{c|c} A_a + B_a F + L_i C_a + L_i D_a F & -L_i \\ \hline F & 0 \end{array}\right],\ i = \overline{1:9}. \tag{34}$$

3.2 Isolation and recovery for actuator fault

In order to isolate actuator faults, two sensors measuring the TVC nozzle's deflection, $u_p[k]$, were used.

For each of them a residual generator was implemented

$$r_i[k] = V_{TVC}\left(\begin{bmatrix} u_{p_i}[k-s] \\ u_{p_i}[k-s+1] \\ \vdots \\ u_{p_i}[k] \end{bmatrix} - H_{TVC}\begin{bmatrix} u[k-s] \\ u[k-s+1] \\ \vdots \\ u[k] \end{bmatrix}\right), \tag{35}$$

$$r_i[k] = V_{TVC}(U_{p_i}[k] - H_{TVC}U[k]),\ i = \overline{1:2}. \tag{36}$$

For detection, a fault of the actuator is declared only if both $u_p[k]$ sensors measure similar values and if a significant change in the residual signals is observed, in order to rule out sensor malfunction. For the scenario investigated in this paper, the fault was modeled as a saturation on the commanded nozzle deflection, $u[k]$, which corresponds to actuator loss of effectiveness.

In order to ensure that the closed-loop system remains stable when the plant's command is saturated, the controller K_s is modified to produce a command signal that is smaller in amplitude by recomputing the F matrix. This was achieved by modifying its weighting matrices, R_F and Q_F, and increasing their values to further penalize both state and command signal dynamics. A pair of appropriate retuning matrices Q_{F_f} and R_{F_f} was found, ensuring a command amplitude that is low enough to accommodate the actuator fault. As is customary, these weighting matrices for the control Riccati equation were chosen in an empirical manner.These are

$$R_{F_f} = 10^5,$$
$$Q_{F_f} = diag\{10^4, 10^4, 1, 1, 1, 1, 1, 1, 1\}.$$

Then, F_f is computed as a solution to the discrete-time Riccati equation defined by the new weighting matrices

$$A_a^T X_i A_a - X_i - A_a^T X_i B_a (B_a^T X_i B_a + R_{F_f})^{-1} \times \\ \times B_a^T X_i A_a + Q_{F_f} = 0, \tag{37}$$

$$F_f = -(B_a^T X_a B + R_{F_f})^{-1} B_a^T X A_a. \tag{38}$$

For the purpose of actuator fault recovery, only two cases are possible. When there is no fault, the controller K_s uses the nominal matrix computed for F. When there is a fault, the value of F inside the controller's state space equations is switched to F_f by the recovery module, which ensures the command is lower in amplitude and thus less prone to be affected by saturation. With both recovery modules considered, the stabilizing controller has the following expression

$$K_{s_{ij}} = \left[\begin{array}{c|c} A + BF_j + L_i C + L_i D F_j & -L_i \\ \hline F_j & 0 \end{array}\right], \tag{39}$$

where $i = \overline{1:9}$ and $j = \overline{1:2}$. As always, the optimal stabilizing block $K_{s_{ij}}$ has a discretized integrator attached to its output in order to form the desired controller

$$K = \frac{T_s}{z-1} K_{s_{ij}}. \tag{40}$$

4. MICRO-LAUNCHER APPLICATION

4.1 Sensor FDIR scenario

In order to illustrate the performance of the proposed FDIR scheme, a series of faults have been applied to the plant's sensors. As stated above, each of the two outputs (pitch angle and pitch rate) have a set of two corresponding sensors each, in order to have the possibility of operating function recovery when one of the two fails.

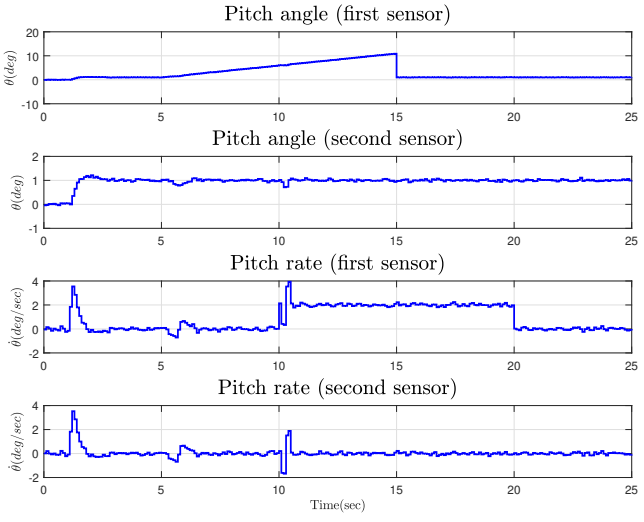

Fig. 2. Measured values during sensor failures

Fig. 2 showcases the values measured by the four sensors of the micro-launcher, when the reference is given as

$$r[k] = \mathbf{1}[k-10]\ ,$$

with two faults placed upon the first and the third sensor:

$$f_1[k] = (k-50)(\mathbf{1}[k-50] - \mathbf{1}[k-150]), \qquad y_{f_1} = y_1 + f_1\ ,$$
$$f_3[k] = 2(\mathbf{1}[k-100] - \mathbf{1}[k-200]), \qquad y_{f_3} = y_3 + f_3\ .$$

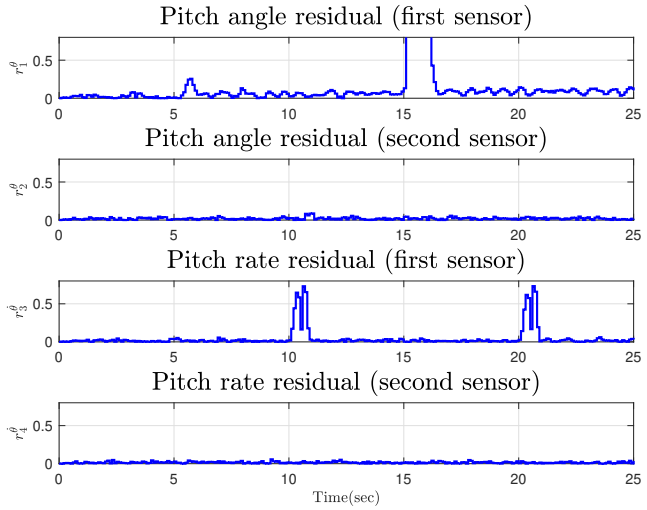

Fig. 3. Residual values during sensor failures

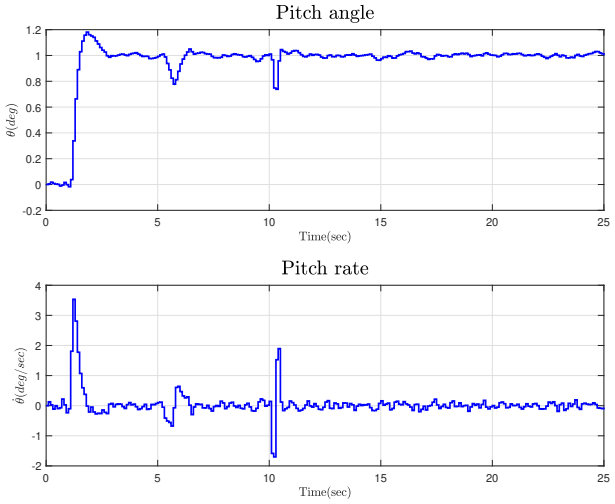

Fig. 4. Reconfiguration effects for faulty sensors

In Fig. 3, the associated residual signals are illustrated. The first fault is detected at approximately $t = 6$ *sec*, when the residual associated to the first sensor exceeds the fault threshold, and the values measured by the first and second sensors begin diverging, respectively

$$r_1[k] \geq 0.15 \ , \quad \|Y_1 - Y_2\|_2 \geq 0.5,$$
$$Y_i = [y_i[k-s] \ \dots \ y_i[k-1] \ y_i[k]] \ , \ i = \overline{1:4} \ .$$

Note that the fault threshold was chosen large enough so that a fault scenario is not triggered by mere measurement noises, but also small enough to detect the ramp-like fault acting upon the first pitch angle sensor at $t = 5$ *sec*. The delay induced in the detection process is directly influenced by the number of samples s used to generate the residual. Here, a delay of approximately 0.6 *sec* was considered a good compromise between response time and reliability. The first sensor becomes functional again at approximately $t = 17$ *sec*, after another spike in the associated residual has been detected, indicating that the fault's effect has disappeared and the values measured by the first and second sensors are similar, respectively

$$r_1[k] < 0.15 \ , \quad \|Y_1 - Y_2\|_2 < 0.5 \ .$$

The second fault is detected at $t = 10$ *sec*, when the following relations are satisfied:

$$r_3[k] \geq 0.1 \ , \quad \|Y_3 - Y_4\|_2 \geq 0.5 \ .$$

The third sensor is declared functional again at $t = 21$ *sec*, when the following relations are satisfied:

$$r_3[k] < 0.1 \ , \quad \|Y_3 - Y_4\|_2 < 0.5 \ .$$

Before $t = 6$ *sec*, the nominal matrix L is used:

$$L = \begin{bmatrix} 0.4780 & 0.4780 & 0.0690 & 0.0690 \\ 0.0118 & 0.0118 & 0.7747 & 0.7747 \\ 1.5964 & 1.5964 & -0.6832 & -0.6832 \\ 0.0051 & 0.0051 & 0.2279 & 0.2279 \\ -0.0005 & -0.0005 & -0.0216 & -0.0216 \\ -0.0012 & -0.0012 & -0.0553 & -0.0553 \\ 0.0046 & 0.0046 & 0.2055 & 0.2055 \end{bmatrix} .$$

Between $t = 6$ *sec* and $t = 10$ *sec*, the nominal L matrix used to compute the controller is replaced with L_{f_1} in order to accommodate the fault placed upon the first sensor:

$$L_{f_1} = \begin{bmatrix} 0.0000 & 0.9176 & 0.0697 & 0.0697 \\ 0.0000 & 0.0218 & 0.7748 & 0.7748 \\ 0.0000 & 3.1868 & -0.6830 & -0.6830 \\ 0.0000 & 0.0092 & 0.2280 & 0.2280 \\ 0.0000 & -0.0009 & -0.0216 & -0.0216 \\ 0.0000 & -0.0022 & -0.0554 & -0.0554 \\ 0.0000 & 0.0083 & 0.2055 & 0.2055 \end{bmatrix} .$$

It can be noted that the first column which corresponds to the estimation of the values for the first sensor is zero, thus indicating that the first feedback signal is no longer used to compute the command. The following two fault cases that arise until $t = 21$ *sec* are treated in the same manner, with the columns corresponding to faulty sensors being full of 0 values.

Fig. 4 showcases the effects of the sensors faults on the plant's outputs, when the recovery module is implemented. As it can be seen, after a short transient period, the outputs return to the normal values, and the faults have no further effect.

4.2 Actuator FDIR scenario

For the modeling of the actuator failure, the considered reference is

$$r[k] = \mathbf{1}[k-30] - \mathbf{1}[k-90] + \mathbf{1}[k-150] - \mathbf{1}[k-210] \ ,$$

while the fault is modeled as a saturation, starting at $t = 15$ *sec*:

$$sat(u[k]) = \begin{cases} -0.4 \ , & u[k] \leq -0.4, \\ 0.4 \ , & u[k] \geq 0.4, \\ u[k] \ , & u[k] \in (-0.4, 0.4) \ . \end{cases}$$

Fig. 5 illustrates the command sent to the actuator. Before the saturation is applied at $t = 15$ *sec*, we have

$$\|u\|_\infty = 1.3 \ ,$$

while after $t = 15$ *sec* the command is being saturated.

In Fig. 6, the plant's outputs are illustrated, which clearly indicate that the closed-loop system is now unstable, and the nominal control effort has failed to stabilize the launcher. In order to prevent complete mission failure, the recovery module modifies the controller's state space equations, ensuring the closed-loop can withstand strong saturation levels and still remain stable.

Fig. 7 shows the controller output when the recovery module is implemented and active. At roughly $t = 15$ *sec*, the actuator becomes saturated and the recovery module switches the value of F inside the controller's state-space equations to F_f:

$$F_f = [-0.16 \ -0.17 \ 0 \ -0.13 \ 0 \ 0 \ -0.77] .$$

The nominal value of F, used before $t = 15$ *sec*, was

$$F = [-1.56 \ -0.5 \ -0.07 \ -0.3 \ 0 \ -0.02 \ -0.99] .$$

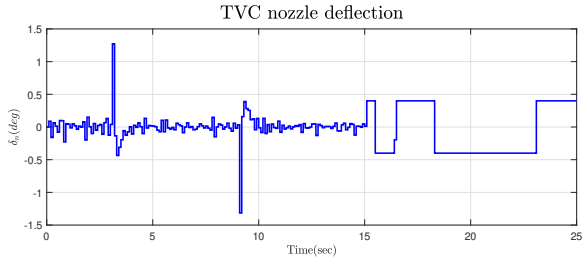

Fig. 5. Saturated controller output without fault recovery

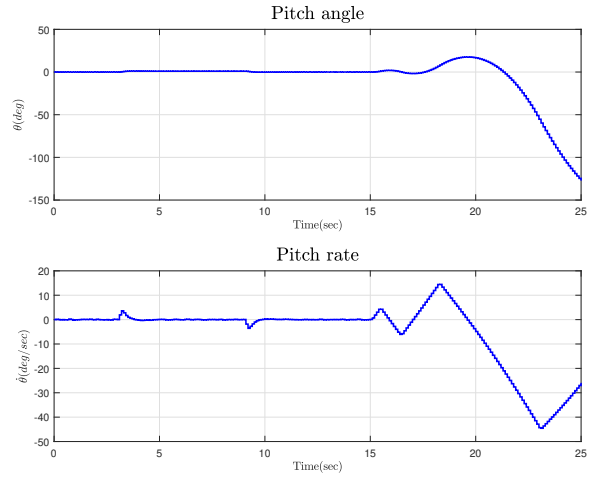

Fig. 6. Flight instability caused by faulty actuator

The matrix F_f ensures a smaller command $u[k]$, as shown by comparing the Frobenius norms of the two matrices:

$$\|F\|_F = 1.938, \quad \|F_f\|_F = 0.814, \quad \|F_f\|_F < \|F\|_F .$$

The effect of the recovery can be seen in Fig. 8. At $t = 15$ *sec*, when the second rectangular impulse is sent to the closed-loop system, a strong oscillation appears on both outputs but, due to the action of the recovery module, the outputs return to normal operating ranges after roughly 5 seconds.

5. CONCLUSIONS

As showcased by the application results in section 4, the proposed recovery method ensures not only fault-tolerance in case of sensor failure, but also provides the means to successfully preserve closed-loop stability in case of actuator failure. Moreover, by using this method to adapt the controller's state equations without switching to a new one, the state vector does not suffer from discontinuities and the control law maintains its optimal quality.

ACKNOWLEDGEMENTS

The authors would like to thank Teodor V. Chelaru and Adrian M. Stoica for providing access to the micro-launcher's dynamical model.

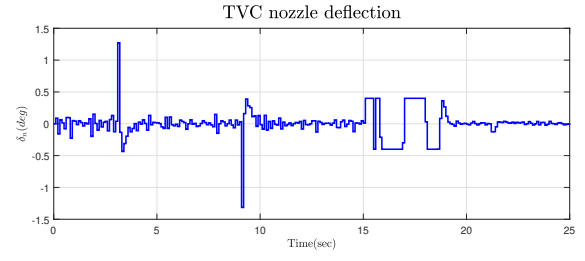

Fig. 7. Saturated controller output with fault recovery

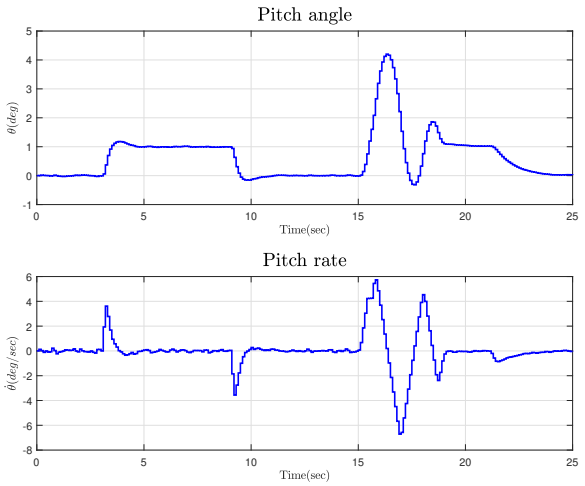

Fig. 8. Reconfiguration effects for faulty actuator

REFERENCES

Anderson, B. and Moore, J. (2007). *Optimal Control, Linear Quadratic Methods, 12th edition.* Dover Publications.

Bittanti, S., Laub, A., and Willems, J. (1991). *The Riccati Equation.* Springer-Verlag.

Blanke, M., Kinnaert, M., Lunze, J., and Staroswiecki, M. (2016). *Diagnosis and Fault-Tolerant Control, 3rd edition.* Springer-Verlag.

Chelaru, T. (2018). Performances Evaluation for Micro-launcher, Mathematical Model. *International Journal of Modeling and Optimization*, 8, 197–201.

Chen, J. and Patton, R. (2012). *Robust Model-Based Fault Diagnosis for Dynamic Systems, 2nd edition.* Springer.

Diaconescu, S., Sperilă, A., Ciubotaru, B., and Stoica, A. (2019). Space Micro-launcher H_∞ control under parametric modeling uncertainties. *Proc. 21st IFAC Symposium on Automatic Control in Aerospace.*

Frank, P. (1990). Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-Based Redundancy - A Survey and Some New Results. *Automatica*, 26, 459–474.

Ionescu, V., Oară, C., and Weiss, M. (1999). *Generalized Riccati Theory and Robust Control: A Popov Function Approach.* John Wiley and Sons.

Isermann, R. (2011). *Fault-Diagnosis Applications.* Springer.

Kwakernaak, H. and Sivan, R. (1972). *Linear Optimal Control Systems.* Wiley-Interscience.

Laub, A. (1979). *A Schur Method for Solving Algebraic Riccati Equations.* IEEE Transactions on Automatic Control.

Lou, X., Willsky, A., and Verghese, G. (1986). Optimally Robust Redundancy Relations for Failure Detection in Uncertain Systems. *Automatica*, 22, 333–344.

Maciejowski, J. (2007). *Multivariable Feedback Design, 2nd edition.* Addison-Wesley.

Patton, R., Frank, P., and Clark, R. (2000). *Issues of Fault Diagnosis for Dynamic Systems.* Springer-Verlag.

Skogestad, S. and Postlethwaite, I. (2007). *Multivariable Feedback Control, 3nd edition.* Wiley.

Zhou, K., Doyle, J., and Glover, K. (1995). *Robust and Optimal Control.* Pearson Education.