AIAA Guidance, Navigation, and Control (GNC) Conference
August 19-22, 2013, Boston, MA

AIAA 2013-4693

Minimal Modeling Adaptive Control of the NASA Generic Transport Model with Unknown Control-Surface Faults

Ming-Jui Yu [*], Yousaf Rahman [†], Ella M. Atkins [‡],
Ilya V. Kolmanovsky [§], and Dennis S. Bernstein[¶]

University of Michigan, 1320 Beal Ave., Ann Arbor, MI 48109

Sudden or gradual faults in sensors, actuators, and processors can have catastrophic consequences for plants that depend on control technology. This paper applies adaptive control to an aircraft with unknown control-surface faults and investigates the robustness of the controller to uncertainties in the aircraft dynamics. Control-surface faults include magnitude and rate saturation, deadzone, and jams in the ailerons and rudder. Retrospective cost adaptive control (RCAC) is applied in both centralized and constrained architectures with a single tuning for each architecture in order to compare the range of unknown faults that RCAC can compensate for through on-line adaptation. The results show that the centralized controller architecture can generally compensate for a larger range of fault parameters. We then characterize the robustness of the centralized controller architecture with respect to erroneous prior modeling information.

I. Introduction

The use of control as an enabling technology can be potentially disastrous when that technology fails. In particular, sudden or gradual faults in sensors, actuators, and processors can have catastrophic consequences for plants that depend on control technology. By the same token, a control system that can reconfigure and adapt to faults may have the ability to compensate for significantly off-nominal conditions [10]. In this way, control technology can enhance safety rather than decrease reliability.

The most straightforward approach to compensate for faults is to ascertain that a fault has occurred, determine the nature of the fault, and modify the control system based on the fault conditions. The modified control system may be based on prior analysis and synthesis, with verification and validation performed prior to operation.

In practice, however, it may not be possible to identify the onset or nature of a fault. This motivates the need to address faults whose onset and nature are unknown. In addition, unknown faults may require a control system that can adapt to off-nominal conditions that go beyond anticipated plant variations.

In the present paper we address the problem of unknown faults by using the nonlinear Generic Transport Model (GTM) simulation developed by NASA [1, 11] in support of a scaled flight vehicle [9]. The GTM

[*] Graduate Student, Aerospace Engineering Department
[†] Graduate Student, Aerospace Engineering Department
[‡] Associate Professor, Aerospace Engineering Department
[§] Professor, Aerospace Engineering Department
[¶] Professor, Aerospace Engineering Department

simulation includes sensor and actuator models that can be modified to simulate a wide variety of faults, such as jammed control surfaces. Most of the aerodynamic and electromechanical components of the GTM model can be modified to simulate the effects of unknown faults.

To investigate the ability to compensate for unknown, sudden-onset faults, we apply retrospective cost adaptive control (RCAC) to the GTM simulation. RCAC is a direct digital control approach that operates at a fixed sample rate dictated by hardware or chosen to reflect the control bandwidth required by the aircraft dynamics, commands, and disturbances. RCAC is applicable to linear plants that are possibly MIMO, nonminimum phase (NMP), and unstable [2, 3, 6–8, 12]. RCAC relies on knowledge of Markov parameters and, for NMP open-loop-unstable plants, estimates of the NMP zeros. This information can be obtained from either analytical modeling or system identification [5]. RCAC does not require that the plant uncertainty and disturbances be matched with the control inputs, and no information about the disturbance spectrum is needed. RCAC was applied to the NASA GTM model in [4].

As a first step in considering the ability of RCAC to address unknown faults, we consider four types of input nonlinearities to represent unknown control-surface faults. These nonlinearities are magnitude saturation, rate saturation, deadzone, and jammed control surfaces. Although all control surfaces are subject to magnitude and rate saturation, we consider the possibility that the magnitude and rate saturation limits are reduced below the nominal limits. We also consider deadzone nonlinearities, where the control surface remains at its trim deflection until a minimum command is reached. Finally, a jammed control surface remains at a fixed value away from trim, ignoring all commands issued by the control law. All of these faults are assumed to be sudden and unknown to the controller.

For the next step, we further characterize the robustness of the controller under emergency conditions by varying the prior information provided to the controller. Each of the first five Markov parameters of the linearized aircraft dynamics is correctly identified and provided to RCAC as the matrix $\tilde{\mathcal{H}}$, and we compare the performance under unknown jammed actuator faults. We then extend the investigation by testing the controller's sensitivity to uncertainty in the choice of Markov parameters. The matrix $\mathcal{H}$ remains unchanged while we vary the controller's estimate of $\tilde{\mathcal{H}}$. The performance of each instance under unknown jammed actuator faults is then compared.

We consider two implementations of RCAC. One is a centralized version of RCAC, while the other is a constrained architecture. These implementations are described in Section 3.

II. Adaptive Controller Design

II.A. Problem Formulation

Consider the MIMO discrete-time system

$$x(k+1) = Ax(k) + Bu(k) + D_1 w(k), \tag{1}$$

$$y(k) = Cx(k) + D_2 w(k), \tag{2}$$

$$z(k) = E_1 x(k) + E_0 w(k), \tag{3}$$

where $x(k) \in \mathbb{R}^n$, $y(k) \in \mathbb{R}^{l_y}$, $z(k) \in \mathbb{R}^{l_z}$, $u(k) \in \mathbb{R}^{l_u}$, $w(k) \in \mathbb{R}^{l_w}$, and $k \geq 0$. Our goal is to develop an adaptive output feedback controller that minimizes the performance variable z in the presence of the exogenous signal w with minimal modeling information about the dynamics and w. Note that w can represent either a command signal to be followed, an external disturbance to be rejected, or both. The system (1)–(3) can represent a sampled-data application arising from a continuous-time system with sample and hold operations.

If $D_1 = 0$ and $E_0 \neq 0$, then the objective is to have the output $E_1 x$ follow the command signal

$-E_0 w$. On the other hand, if $D_1 \neq 0$ and $E_0 = 0$, then the objective is to reject the disturbance w from the performance measurement $E_1 x$. Furthermore, if $D_1 = \begin{bmatrix} \hat{D}_1 & 0 \end{bmatrix}$, $E_0 = \begin{bmatrix} 0 & \hat{E}_0 \end{bmatrix}$, and $w(k) = \begin{bmatrix} w_1(k)^{\rm T} & w_2(k)^{\rm T} \end{bmatrix}^{\rm T}$, then the objective is to have $E_1 x$ follow the command $-\hat{E}_0 w_2$ while rejecting the disturbance w_1. Lastly, if D_1 and E_0 are empty matrices, then the objective is output stabilization, that is, convergence of z to zero.

II.B. Retrospective Cost

For $i \geq 1$, define the Markov parameter of G_{zu} given by

$$H_i \triangleq E_1 A^{i-1} B. \tag{4}$$

For example, $H_1 = E_1 B$ and $H_2 = E_1 AB$. Let r be a positive integer. Then, for all $k \geq r$,

$$x(k) = A^r x(k-r) + \sum_{i=1}^{r} A^{i-1} B u(k-i) + \sum_{i=1}^{r} A^{i-1} D_1 w(k-i), \tag{5}$$

and thus

$$z(k) = E_1 A^r x(k-r) + \sum_{i=1}^{r} E_1 A^{i-1} D_1 w(k-i) + E_0 w(k) + \bar{H}\bar{U}(k-1), \tag{6}$$

where

$$\bar{H} \triangleq \begin{bmatrix} H_1 & \cdots & H_r \end{bmatrix} \in \mathbb{R}^{l_z \times r l_u}$$

and

$$\bar{U}(k-1) \triangleq \begin{bmatrix} u(k-1) \\ \vdots \\ u(k-r) \end{bmatrix}.$$

Next, we rearrange the columns of $\bar{H}$ and the components of $\bar{U}(k-1)$ and partition the resulting matrix and vector so that

$$\bar{H}\bar{U}(k-1) = \mathcal{H}' U'(k-1) + \mathcal{H} U(k-1), \tag{7}$$

where $\mathcal{H}' \in \mathbb{R}^{l_z \times (r l_u - l_U)}$, $\mathcal{H} \in \mathbb{R}^{l_z \times l_U}$, $U'(k-1) \in \mathbb{R}^{r l_u - l_U}$, and $U(k-1) \in \mathbb{R}^{l_U}$. Then, we can rewrite (6) as

$$z(k) = \mathcal{S}(k) + \mathcal{H} U(k-1), \tag{8}$$

where

$$\mathcal{S}(k) \triangleq E_1 A^r x(k-r) + \sum_{i=1}^{r} E_1 A^{i-1} D_1 w(k-i) + E_0 w(k) + \mathcal{H}' U'(k-1). \tag{9}$$

Next, for $j = 1, \ldots, s$, we rewrite (8) with a delay of k_j time steps, where $0 \leq k_1 \leq k_2 \leq \cdots \leq k_s$, in the form

$$z(k-k_j) = \mathcal{S}_j(k-k_j) + \mathcal{H}_j U_j(k-k_j-1), \tag{10}$$

where (9) becomes

$$\mathcal{S}_j(k-k_j) \triangleq E_1A^r x(k-k_j-r) + \sum_{i=1}^{r} E_1A^{i-1}D_1w(k-k_j-i) + E_0w(k-k_j) + \mathcal{H}'_jU'_j(k-k_j-1) \tag{11}$$

and (7) becomes

$$\bar{H}\bar{U}(k-k_j-1) = \mathcal{H}'_jU'_j(k-k_j-1) + \mathcal{H}_jU_j(k-k_j-1), \tag{12}$$

where $\mathcal{H}'_j \in \mathbb{R}^{l_z \times (rl_u - l_{U_j})}$, $\mathcal{H}_j \in \mathbb{R}^{l_z \times l_{U_j}}$, $U'_j(k-k_j-1) \in \mathbb{R}^{rl_u - l_{U_j}}$, and $U_j(k-k_j-1) \in \mathbb{R}^{l_{U_j}}$. Now, by stacking $z(k-k_1), \ldots, z(k-k_s)$, we define the *extended performance*

$$Z(k) \triangleq \begin{bmatrix} z(k-k_1) \\ \vdots \\ z(k-k_s) \end{bmatrix} \in \mathbb{R}^{sl_z}. \tag{13}$$

Therefore,

$$Z(k) \triangleq \tilde{\mathcal{S}}(k) + \tilde{\mathcal{H}}\tilde{U}(k-1), \tag{14}$$

where

$$\tilde{\mathcal{S}}(k) \triangleq \begin{bmatrix} \mathcal{S}_1(k-k_1) \\ \vdots \\ \mathcal{S}_s(k-k_s) \end{bmatrix} \in \mathbb{R}^{sl_z}, \tag{15}$$

$\tilde{U}(k-1)$ has the form

$$\tilde{U}(k-1) \triangleq \begin{bmatrix} u(k-q_1) \\ \vdots \\ u(k-q_{l_{\tilde{U}}}) \end{bmatrix} \in \mathbb{R}^{l_{\tilde{U}}}, \tag{16}$$

where, for $i = 1, \ldots, l_{\tilde{U}}$, $k_1 \leq q_i \leq k_s + r$, and $\tilde{\mathcal{H}} \in \mathbb{R}^{sl_z \times l_{\tilde{U}}}$ is constructed according to the structure of $\tilde{U}(k-1)$. The vector $\tilde{U}(k-1)$ is formed by stacking $U_1(k-k_1-1), \ldots, U_s(k-k_s-1)$ and removing copies of repeated components.

Next, we define the *retrospective performance*

$$\hat{z}(k-k_j) \triangleq \mathcal{S}_j(k-k_j) + \mathcal{H}_j\hat{U}_j(k-k_j-1), \tag{17}$$

where the past controls $U_j(k-k_j-1)$ in (10) are replaced by the surrogate controls $\hat{U}_j(k-k_j-1)$. In analogy with (13), the *extended retrospective performance* for (17) is defined as

$$\hat{Z}(k) \triangleq \begin{bmatrix} \hat{z}(k-k_1) \\ \vdots \\ \hat{z}(k-k_s) \end{bmatrix} \in \mathbb{R}^{sl_z} \tag{18}$$

and thus is given by

$$\hat{Z}(k) = \tilde{\mathcal{S}}(k) + \tilde{\mathcal{H}}\hat{\tilde{U}}(k-1), \tag{19}$$

where the components of $\hat{\tilde{U}}(k-1) \in \mathbb{R}^{l_{\tilde{U}}}$ are the components of $\hat{U}_1(k-k_1-1), \ldots, \hat{U}_s(k-k_s-1)$ ordered in the same way as the components of $\tilde{U}(k-1)$. Subtracting (14) from (19) yields

$$\hat{Z}(k) = Z(k) - \tilde{\mathcal{H}}\tilde{U}(k-1) + \tilde{\mathcal{H}}\hat{\tilde{U}}(k-1). \tag{20}$$

Finally, we define the *retrospective cost function*

$$J(\hat{\tilde{U}}(k-1),k) \overset{\triangle}{=} \hat{Z}^{\rm T}(k)R(k)\hat{Z}(k), \tag{21}$$

where $R(k) \in \mathbb{R}^{l_z s \times l_z s}$ is a positive-definite performance weighting. The goal is to determine refined controls $\hat{\tilde{U}}(k-1)$ that would have provided better performance than the controls $U(k)$ that were applied to the system. The refined control values $\hat{\tilde{U}}(k-1)$ are subsequently used to update the controller.

II.C. Cost Function Optimization with Adaptive Regularization

To ensure that (21) has a global minimizer, we consider the regularized cost

$$\bar{J}(\hat{\tilde{U}}(k-1),k) \overset{\triangle}{=} \hat{Z}^{\rm T}(k)R(k)\hat{Z}(k) + \eta(k)R_2\hat{\tilde{U}}^{\rm T}(k-1)\hat{\tilde{U}}(k-1), \tag{22}$$

where $\eta(k) \geq 0$, and $R_2 \in \mathbb{R}^{\hat{\tilde{U}}} \geq 0$. Substituting (20) into (22) yields

$$\bar{J}(\hat{\tilde{U}}(k-1),k) = \hat{\tilde{U}}(k-1)^{\rm T}\mathcal{A}(k)\hat{\tilde{U}}(k-1) + \hat{\tilde{U}}^{\rm T}(k-1)\mathcal{B}^{\rm T}(k) + \mathcal{C}(k), \tag{23}$$

where

$$\mathcal{A}(k) \overset{\triangle}{=} \tilde{\mathcal{H}}^{\rm T}R(k)\tilde{\mathcal{H}} + \eta(k)R_2 I_{l_{\tilde{U}}}, \tag{24}$$

$$\mathcal{B}(k) \overset{\triangle}{=} 2\tilde{\mathcal{H}}^{\rm T}R(k)[Z(k) - \tilde{\mathcal{H}}\tilde{U}(k-1)], \tag{25}$$

$$\mathcal{C}(k) \overset{\triangle}{=} Z^{\rm T}(k)R(k)Z(k) - 2Z^{\rm T}(k)R(k)\tilde{\mathcal{H}}\tilde{U}(k-1) + \tilde{U}^{\rm T}(k-1)\tilde{\mathcal{H}}^{\rm T}R(k)\tilde{\mathcal{H}}\tilde{U}(k-1). \tag{26}$$

If either $\tilde{\mathcal{H}}$ has full column rank or $\eta(k) > 0$ and $R_2 > 0$, then $\mathcal{A}(k)$ is positive definite. In this case, $\bar{J}(\hat{\tilde{U}}(k-1),k)$ has the unique global minimizer

$$\hat{\tilde{U}}(k-1) = -\frac{1}{2}\mathcal{A}^{-1}(k)\mathcal{B}(k). \tag{27}$$

II.D. Controller Construction

The control $u(k)$ is given by the strictly proper time-series controller of order $n_{\rm c}$ given by

$$u(k) = \sum_{i=1}^{n_{\rm c}} M_i(k)u(k-i) + \sum_{i=1}^{n_{\rm c}} N_i(k)y(k-i), \tag{28}$$

where, for all $i = 1,\ldots,n_{\rm c}$, $M_i(k) \in \mathbb{R}^{l_u \times l_u}$ and $N_i(k) \in \mathbb{R}^{l_u \times l_y}$. The control (28) can be expressed as

$$u(k) = \theta(k)\phi(k-1), \tag{29}$$

where

$$\theta(k) \overset{\triangle}{=} [M_1(k)\ \cdots\ M_{n_{\rm c}}(k)\ N_1(k)\ \cdots\ N_{n_{\rm c}}(k)] \in \mathbb{R}^{l_u \times n_{\rm c}(l_u+l_z)} \tag{30}$$

and

$$\phi(k-1) \overset{\triangle}{=} \begin{bmatrix} u(k-1) \\ \vdots \\ u(k-n_{\rm c}) \\ y(k-1) \\ \vdots \\ y(k-n_{\rm c}) \end{bmatrix} \in \mathbb{R}^{n_{\rm c}(l_u+l_y)}. \tag{31}$$

II.E. Recursive Least Squares Update of $\theta(k)$

Next, let d be a positive integer such that $\tilde{U}(k-1)$ contains $u(k-d)$ and define the cumulative cost function

$$J_{\rm R}(\theta,k) \triangleq \sum_{i=d+1}^{k} \lambda^{k-i}\|\phi^{\rm T}(i-d-1)\theta^{\rm T}(k)-\hat{u}^{\rm T}(i-d)\|^2+\lambda^k(\theta(k)-\theta_0)P_0^{-1}(\theta(k)-\theta_0)^{\rm T}, \tag{32}$$

where $\|\cdot\|$ is the Euclidean norm, and $\lambda\in(0,1]$ is the forgetting factor. Minimizing (32) yields

$$\begin{aligned}\theta^{\rm T}(k)=\theta^{\rm T}(k-1)+\beta(k)P(k-1)\phi(k-d-1)\cdot[\phi^{\rm T}(k-d)P(k-1)\phi(k-d-1)+\lambda(k)]^{-1}\\ \cdot[\phi^{\rm T}(k-d-1)\theta^{\rm T}(k-1)-\hat{u}^{\rm T}(k-d)],\end{aligned}$$

where $\beta(k)$ is either zero or one. The error covariance is updated by

$$\begin{aligned}P(k)=\beta(k)\lambda^{-1}P(k-1)+[1-\beta(k)]P(k-1)-\beta(k)\lambda^{-1}P(k-1)\phi(k-d-1)\\ \cdot[\phi^{\rm T}(k-d-1)P(k-1)\phi(k-d)+\lambda]^{-1}\quad\cdot\phi^{\rm T}(k-d-1)P(k-1).\end{aligned} \tag{33}$$

We initialize the error covariance matrix as $P(0)=\alpha I_{3n_{\rm c}}$, where $\alpha>0$. Note that when $\beta(k)=0$, $\theta(k)=\theta(k-1)$ and $P(k)=P(k-1)$. Therefore, setting $\beta(k)=0$ switches off the controller adaptation, and thus freezes the control gains. When $\beta(k)=1$, the controller is allowed to adapt.

III. RCAC Architectures

We consider two implementations of RCAC. In the first implementation, a single RCAC controller block uses the performance (error) variables in order to adapt and command the various control surfaces without knowledge of possible actuator degradation. In the second implementation, multiple RCAC subcontroller blocks are used to command the control surfaces, one dedicated to each control surface and without knowledge of possible degradation of the control surface for which it is responsible. These architectures can be viewed as centralized and constrained, respectively. Figure 1 shows block diagrams of both configurations. The constrained architecture is motivated by the fact that having multiple SISO (in the case of a scalar error variable) and MISO (in the case of vector error variable) RCAC control laws allows for each error-to-actuator channel to have a different inital error covariance matrix, allowing for more flexible controller tuning. The question that this section seeks to address is whether the separate RCAC subcontrollers in the constrained architecture are able to adapt effectively during simultaneous operation despite the fact that axis coupling is not modeled. A similar RCAC architecture for structural control is considered in [13].

III.A. Controller Architecture Comparison

In all examples, the controller gains are allowed to converge during initial level straight-line flight. At 200 sec, the aircraft is commanded to follow a 400-sec 90-deg heading command at the same time actuator failure occurs. In all examples, a single tuning is used for the centralized architecture as well as for the constrained architecture. For the centralized controller, the tuning parameters are

$$n_c=16, P_0=0.08, p_c=0.1, \eta_0=0.01 \tag{34}$$

and

$$\tilde{\mathcal{H}}=H_1\in\mathbb{R}^{1\times 3}, \tag{35}$$

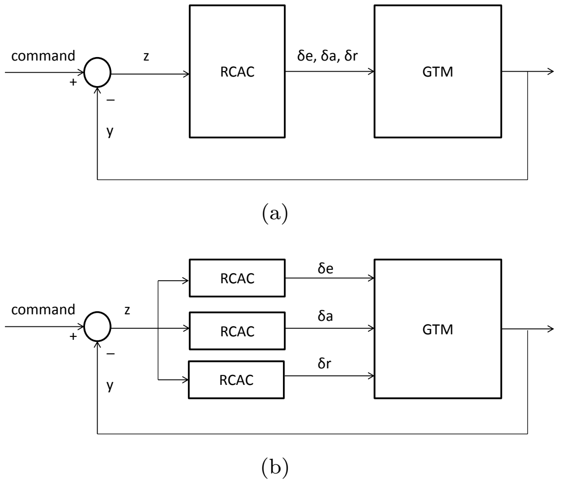


Figure 1. (a) Centralized and (b) constrained RCAC architectures. In the constrained architecture, each RCAC controller is SISO, and the adaptation occurs separately in each block.

The Markov parameters are calculated based on a linearized model of GTM obtained using the Simulink function linmod. For the constrained case, each RCAC subcontroller uses the tuning parameters

$$n_c = 16, P_0 = 0.1, p_c = 0.1, \eta_0 = 0.03 \tag{36}$$

and

$$\tilde{\mathcal{H}} = H_1 \in \mathbb{R}^{1\times 1}, \tag{37}$$

where H_1 may be different for each subcontroller. Note that the centralized and constrained architecture require the same amount of prior modeling information. The centralized architecture uses a $\tilde{\mathcal{H}}$ that is a 1×3 matrix, requiring estimation of three numbers. The constrained architecture uses three scalar matrices $\tilde{\mathcal{H}}$, also requiring estimation of three numbers.

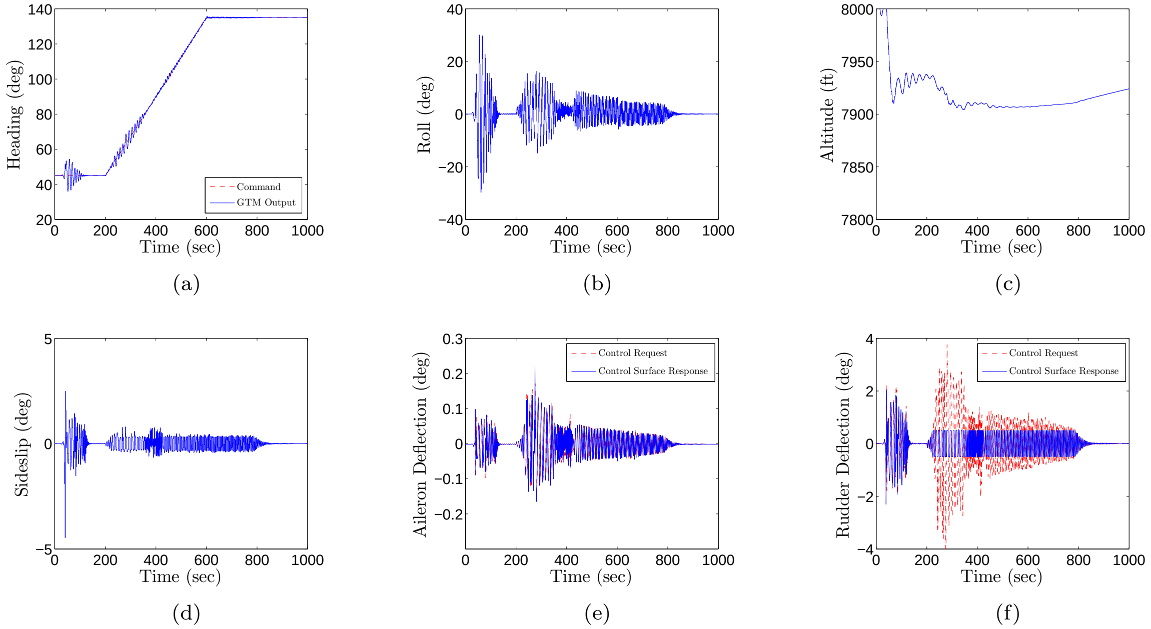
III.A.1. Magnitude Saturation

We compare the performance of the centralized and constrained controllers with unknown magnitude saturation. Figures 2 and 3 show how both controllers perform when the rudder deflection is suddenly saturated within ± 0.5 deg. In all cases, both control architectures are able to follow the command.

Figure 4 shows the evolution of the adaptive gains. While the gains for the individual subcontrollers converge at different times, each is able to compensate for the adaptation of the two other subcontrollers and together minimize the error. Table 1 summarizes the results for magnitude saturation.

III.A.2. Rate Saturation

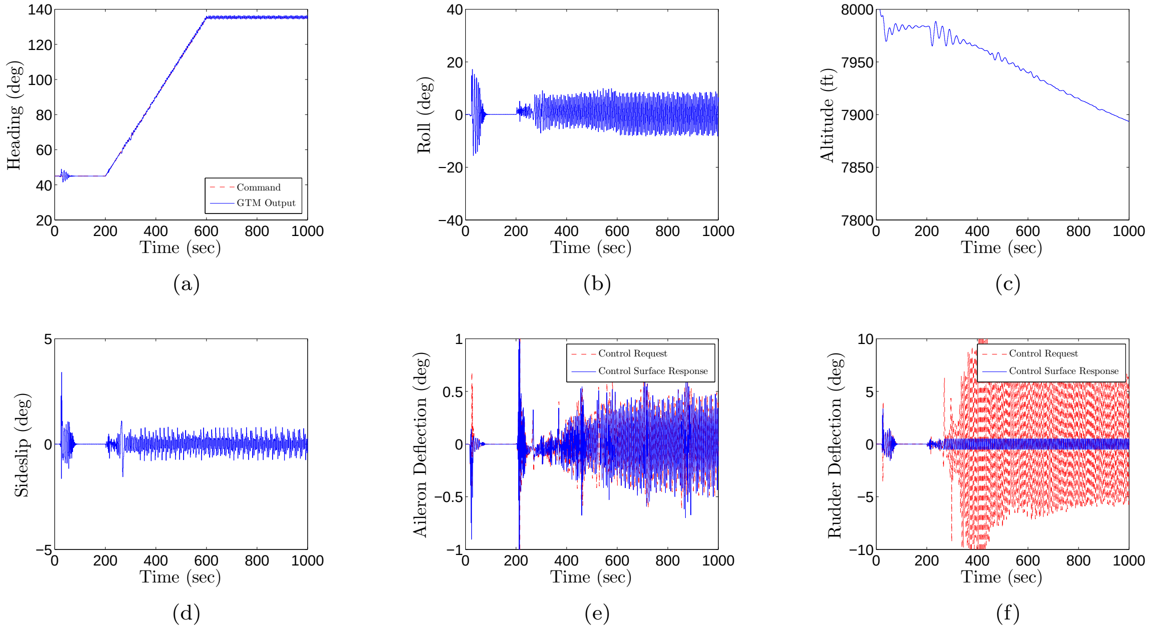
We next compare the performance of the centralized and constrained controllers with unknown rate saturation. Figures 5 and 6 show how both controllers perform when the rudder deflection is rate-saturated within ±0.1 deg/sec. In this example, the constrained configuration is able to follow the command while the centralized RCAC controller is not.



(a) (b) (c)

(d) (e) (f)

Figure 2. Heading command using the centralized architecture, with the rudder deflection magnitude-saturated within ±0.5 deg at 200 sec. (a) Heading, (b) roll, (c) altitude, (d) sideslip, (e) aileron deflection, and (f) rudder deflection. The centralized controller first attempts to command deflections of ±4 deg, and then adapts completely at 800 sec to stay within saturation bounds.



(a) (b) (c)

(d) (e) (f)

Figure 3. Heading command using the constrained architecture, with the rudder deflection magnitude-saturated within ±0.5 deg at 200 sec. (a) Heading, (b) roll, (c) altitude, (d) sideslip, (e) aileron deflection, and (f) rudder deflection. The constrained controller is able to follow the command, but produces an oscillatory response.

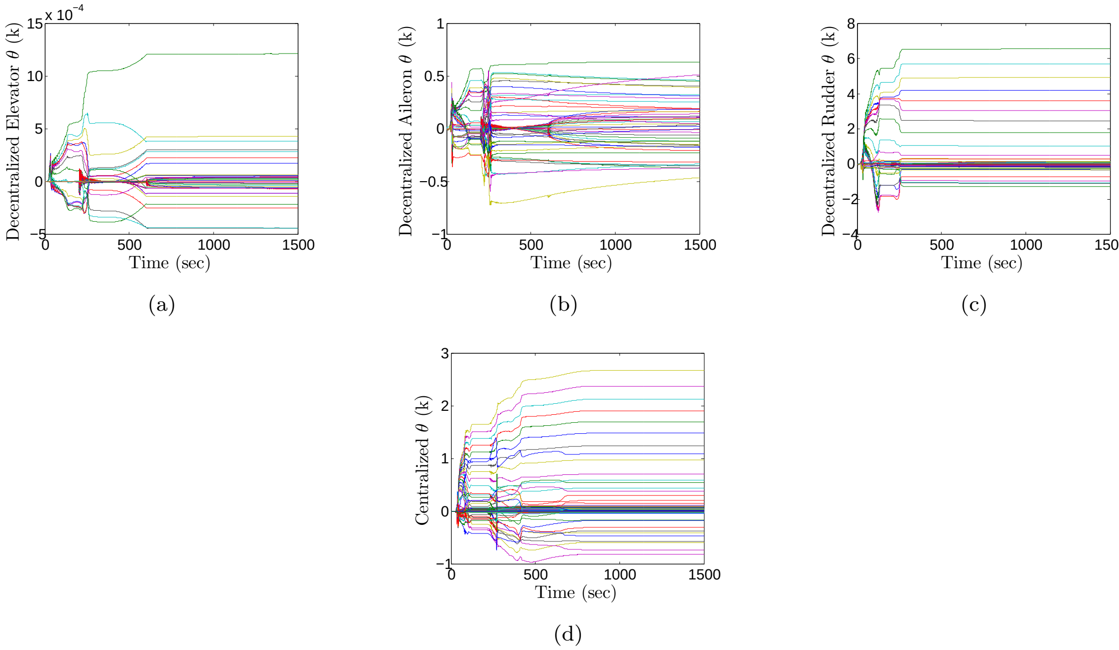
Table 2 summarizes the closed-loop performance for various rate-saturation limits. While both controllers exhibit similar robustness, in one case the constrained architecture was successful while the centralized controller was not.

Saturation Range (deg)	C,A	Co,A	C,R	Co,R
0	Y	Y	Y	Y
1	Y	Y	Y	Y
2	Y	Y	Y	Y
3	Y	Y	Y	Y
4	Y	Y	Y	Y
5	Y	Y	Y	Y
6	Y	Y	Y	Y
7	Y	Y	Y	Y
8	Y	Y	Y	Y
9	Y	Y	Y	Y
10	Y	Y	Y	Y

Table 1. Summary of controller performance for saturated actuators. (C) Centralized RCAC, (Co) constrained RCAC, (A) aileron saturated, and (R) rudder saturated. The notation used is, Y: the controller was able to track the command, and N: the controller was not able to track the command. Both controllers peform similarly when actuator saturation is present.

Maximum Rate (deg/sec)	C,A	Co,A	C,R	Co,R
0.1	Y	Y	N	Y
0.3	Y	Y	Y	Y
0.5	Y	Y	Y	Y
0.7	Y	Y	Y	Y
0.9	Y	Y	Y	Y
1.1	Y	Y	Y	Y
1.3	Y	Y	Y	Y
1.5	Y	Y	Y	Y
1.7	Y	Y	Y	Y
1.9	Y	Y	Y	Y
2.1	Y	Y	Y	Y

Table 2. Summary of controller performance for rate-saturated actuators. (C) Centralized RCAC, (Co) constrained RCAC, (A) rate-saturated aileron, and (R) rate-saturated rudder. The notation used is, Y: the controller was able to track the command, and N: the controller was not able to track the command. The centralized controller is not able to control the aircraft when the rudder is rate saturated at ± 0.1 deg/sec. The constrained controller follows the command for all rate-saturation settings.



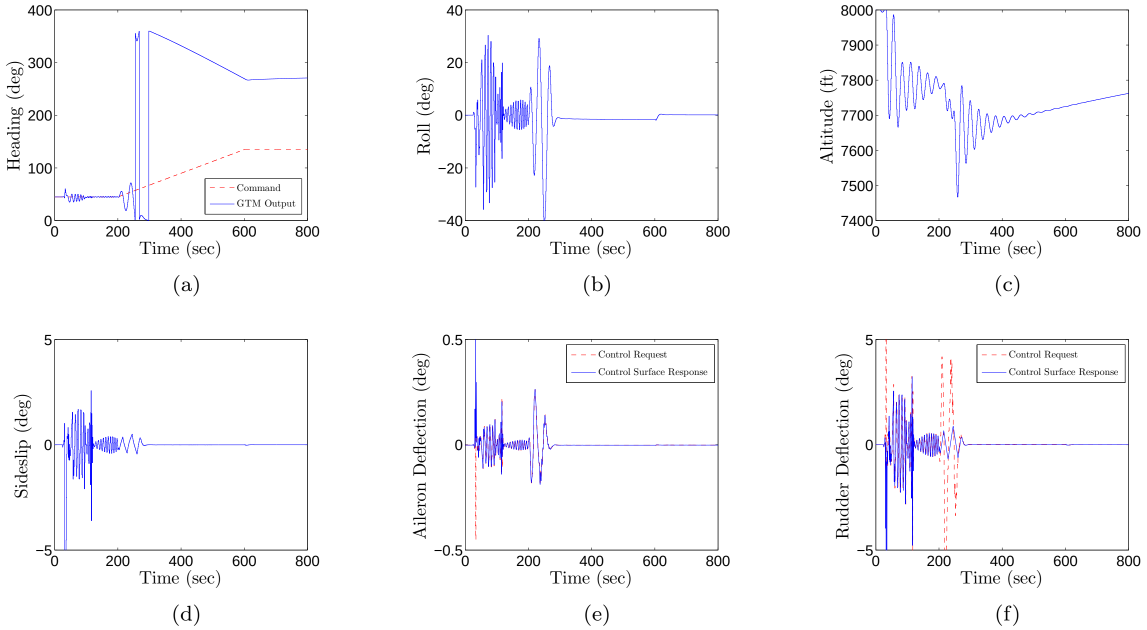
(a)

(b)

(c)

(d)

Figure 4. Evolution of the adaptive gains for the centralized and constrained controllers for the case of rudder magnitude-saturated within ±0.5 deg at 200 sec. (a) Constrained elevator subcontroller, (b) constrained aileron subcontroller, (c) constrained rudder subcontroller, and (d) centralized RCAC. Adaptive gains for the centralized controller converge around 800 sec. Adaptive gains for the elevator, aileron, and rudder subcontrollers converge around 600 sec, 1400, and 270 sec, respectively.



(a)

(b)

(c)

(d)

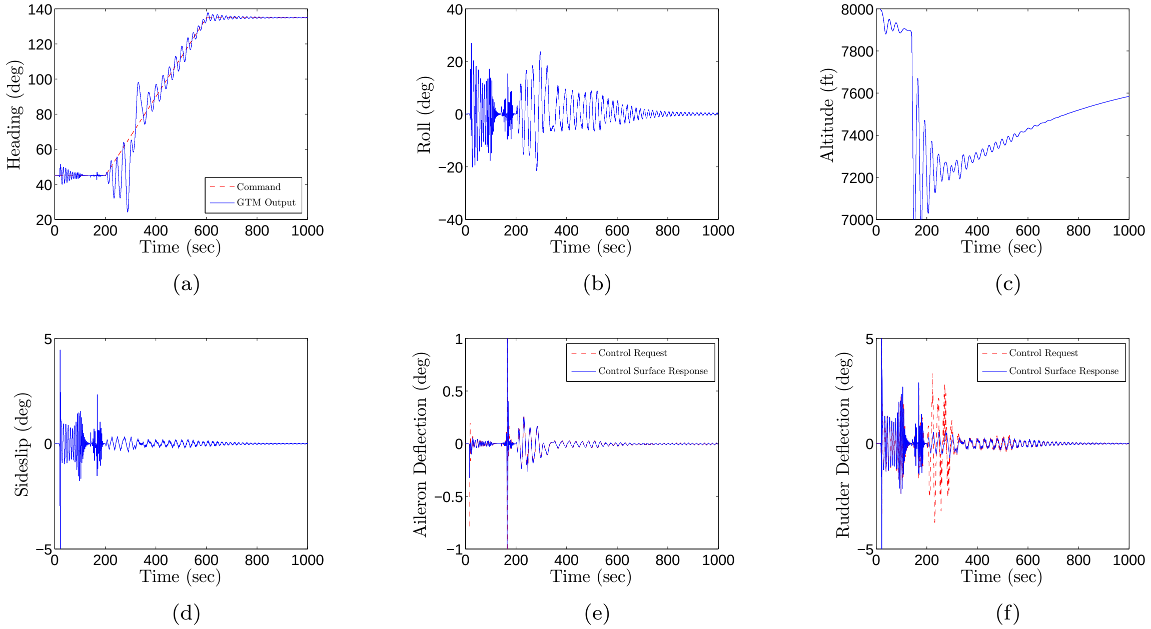
(e)

(f)

Figure 5. Heading command using the centralized architecture, with rudder deflection rate-saturated within±0.1 deg/sec at 200 sec. (a) Heading, (b) roll, (c) altitude, (d) sideslip, (e) aileron deflection, and (f) rudder deflection. At the onset of the fault, the centralized controller drives the aircraft to perform aggressive oscillations in heading, eventually settling on an incorrect heading.

III.A.3. Deadzone

We also compare controller performance with an unknown deadzone. Figures 7 and 8 show how both controllers perform with a rudder deadzone range of ±2.5 deg. In this example, both control architectures

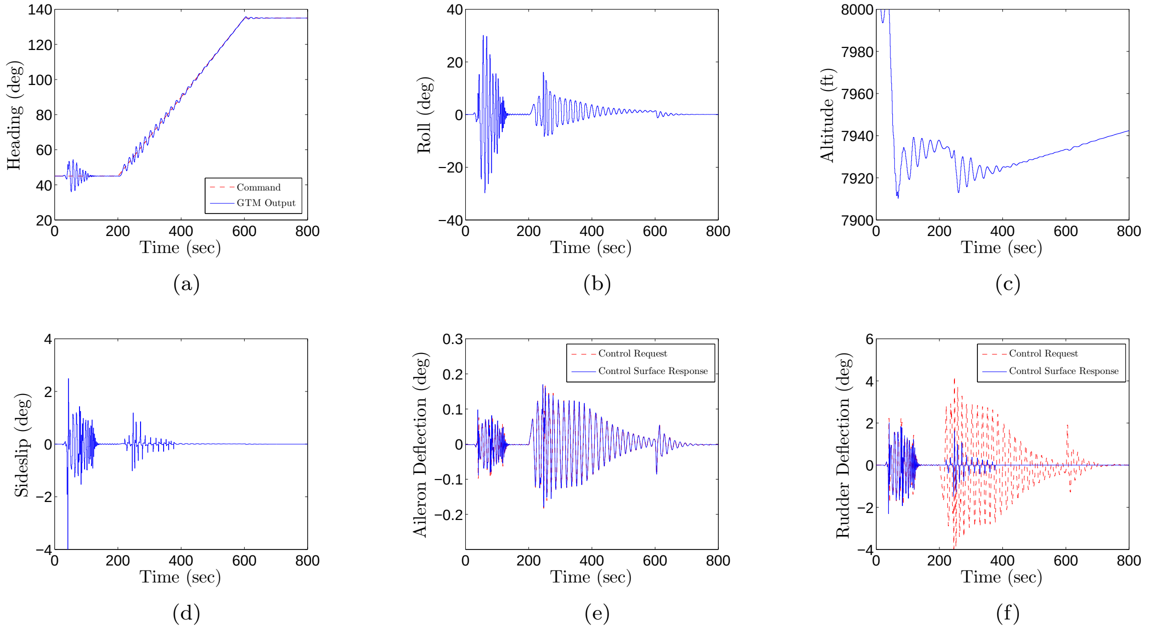


(a) (b) (c)

(d) (e) (f)

Figure 6. Heading command using the constrained architecture, with rudder deflection rate-saturated within ±0.1 **deg/sec at 200 sec. (a) Heading, (b) roll, (c) altitude, (d) sideslip, (e) aileron deflection, and (f) rudder deflection. The three constrained controllers decrease the commanded actuator deflection rate in response to the rate-saturation limit. The aircraft remains in proximity of the commanded heading angle and eventually reaches the commanded heading.**

are able to follow the command.

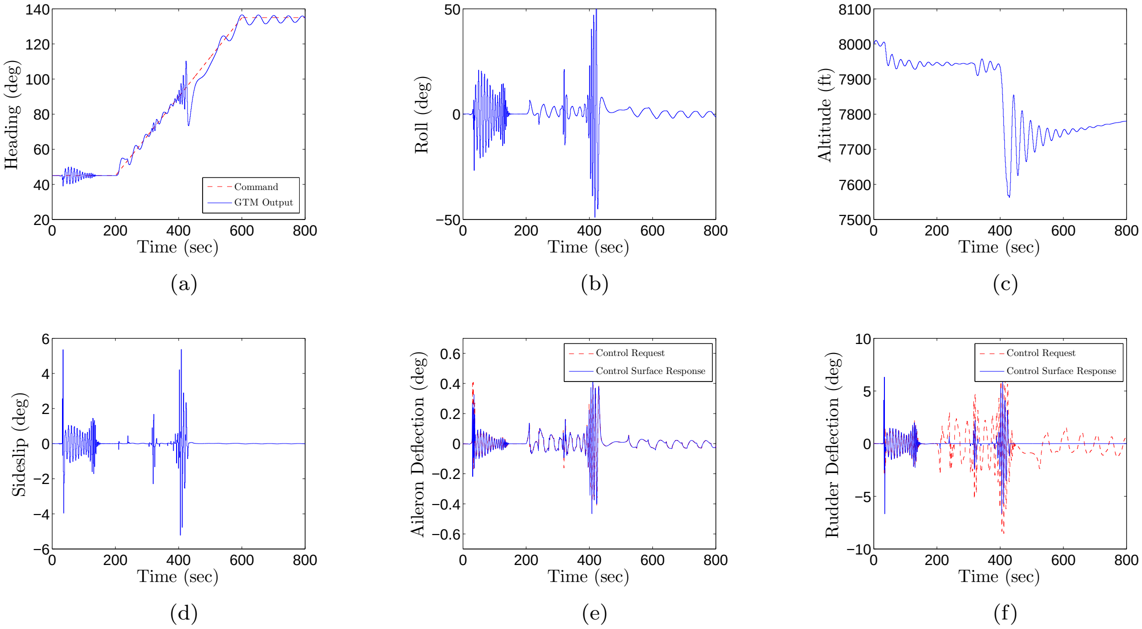


(a) (b) (c)

(d) (e) (f)

Figure 7. Heading command using the centralized architecture, with a rudder deflection deadzone of ±2.5 **deg at 200 sec. (a) Heading, (b) roll, (c) altitude, (d) sideslip, (e) aileron deflection, and (f) rudder deflection. At the onset of the fault, the centralized controller increases aileron activity to compensate for the fault. The commanded rudder deflection is also increased to overcome the deadzone.**

Table 3 summarizes the closed-loop performance for various control surface deadzone widths. Both controllers exhibit high robustness during the occurrence of deadzone faults in one of the actuators.



(a) (b) (c)

(d) (e) (f)

Figure 8. Heading command using the constrained architecture, with a rudder deflection deadzone of ±2.5 deg at 200 sec. (a) Heading, (b) roll, (c) altitude, (d) sideslip, (e) aileron deflection, and (f) rudder deflection. At the onset of the fault, the aileron subcontroller increases aileron activity to compensate for the fault. The rudder subcontroller also increases the commanded rudder deflection to overcome the deadzone.

III.A.4. Jammed Control Surfaces

We compare controller performance when an actuator becomes jammed at an unknown deflection angle. Figure 9 and 10 show how both controller perform with the aileron jammed at a deflection angle of -2 deg. The centralized architecture was able to follow the command while the constrained architecture was not.

Table 4 summarizes the closed-loop performance for actuators jammed at various angles. Overall, the centralized controller is more successful than the constrained controllers.

IV. Robustness under Modeling Error

In the previous section, it was found that the centralized controller architecture is more reliable than the constrained controller architecture for unknown actuator faults. Those test cases are based on accurate modeling information from the linearized aircraft dynamics. In particular, Markov parameters were obtained from the linearized dynamics by

$$H_i \triangleq E_1 A^{i-1} B. \tag{38}$$

We then experimentally determined the Markov parameter choice i^*, where H_{i^*} is the Markov parameter that produces the best performance from RCAC. Note that i^* may vary depending on the system dynamics and the choice of commands and actuators. In reality, it may not be possible to identify i^* or obtain an accurate estimate of H_{i^*}, therefore RCAC must maintain good performance for different choices of i and H_i.

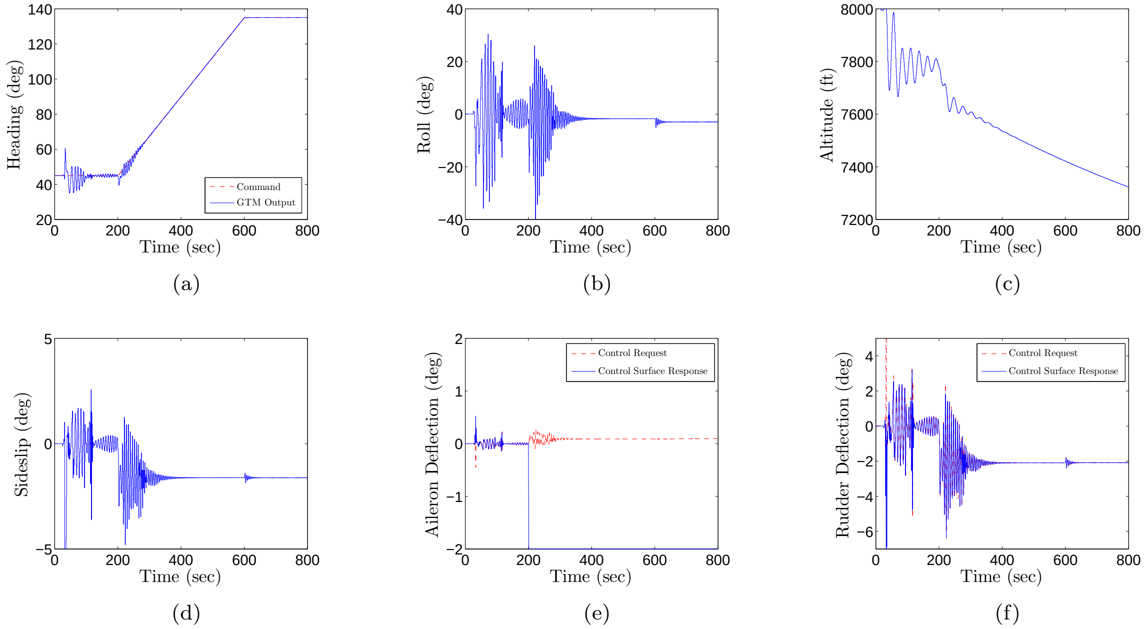
In this section, we specify two cases of inaccurate prior modeling information. The first case assumes

Deadzone Range (deg)	C,A	Co,A	C,R	Co,R
0	Y	Y	Y	Y
1	Y	Y	Y	Y
2	Y	Y	Y	Y
3	Y	Y	Y	Y
4	Y	Y	Y	Y
5	Y	Y	Y	Y
6	Y	Y	Y	Y
7	Y	Y	Y	Y
8	Y	Y	Y	Y
9	Y	Y	Y	Y
10	Y	Y	Y	Y

Table 3. Summary of controller performance for actuator deadzones. (C) Centralized RCAC, (Co) constrained RCAC, (A) aileron deadzone, and (R) rudder deadzone. The notation used is, Y: the controller was able to track the command, and N: the controller was not able to track the command. Both control architectures produced similarly robust performance in the face of deadzone faults.

Jam Angle (deg)	C,A	Co,A	C,R	Co,R
-5	N	N	N	N
-4	N	N	N	N
-3	N	N	N	N
-2	Y	N	N	N
-1	Y	Y	N	N
0	Y	Y	Y	Y
1	Y	Y	N	N
2	Y	N	N	N
3	N	N	N	N
4	N	N	N	N
5	N	N	N	N

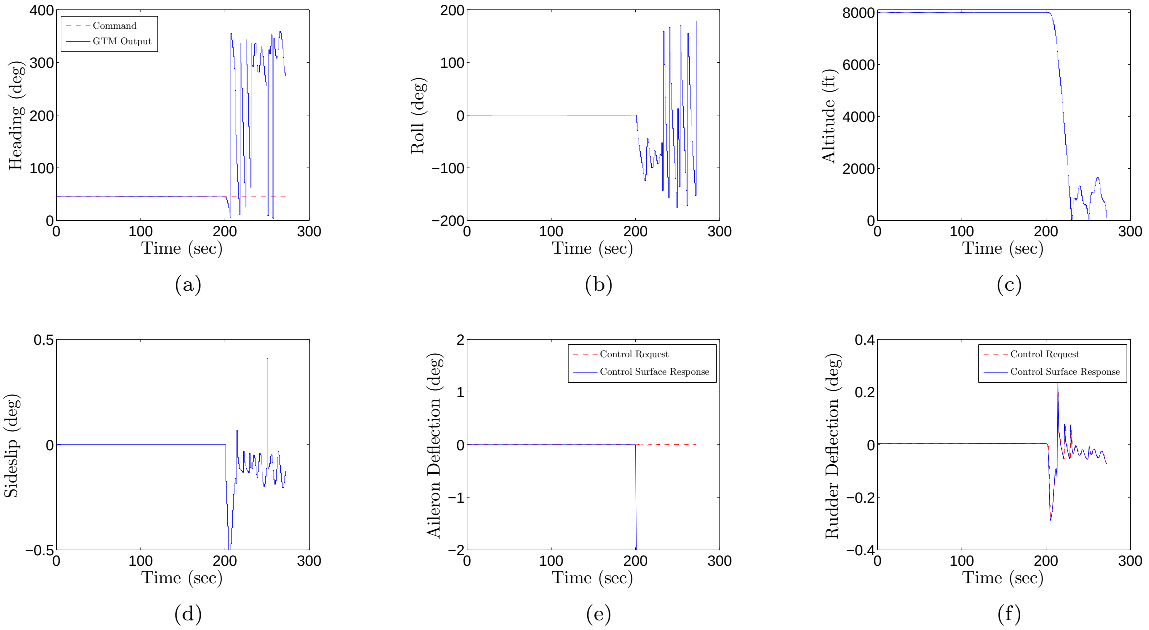
Table 4. Summary of controller performance for jammed actuators. (C) Centralized RCAC, (Co) constrained RCAC, (A) aileron jammed, and (R) rudder jammed. The notation used is, Y: the controller was able to track the command, and N: the controller was not able to track the command. The centralized controller performs as well or better when an actuator is jammed, though neither controller is robust to a jammed rudder.



(a) (b) (c)

(d) (e) (f)

Figure 9. Heading command using the centralized architecture, with a -2 deg aileron jam at 200 sec. (a) Heading, (b) roll, (c) altitude, (d) sideslip, (e) aileron deflection, and (f) rudder deflection. At the onset of the fault, the centralized controller immediately changes the rudder deflection to compensate for the jammed aileron.



(a) (b) (c)

(d) (e) (f)

Figure 10. Heading command using the constrained architecture, with a -2 deg aileron jam at 200 sec. (a) Heading, (b) roll, (c) altitude, (d) sideslip, (e) aileron deflection, and (f) rudder deflection. The three constrained controllers are not able to compensate for the fault.

that i^* is not known, and $\tilde{\mathcal{H}}$ can be given by an accurate estimate of H_i for any i. The second case assumes that i^* is not known, and $\tilde{\mathcal{H}}$ is given solely by H_j, where j is a constant and therefore does not necessarily equal i. For the scope of this paper, we assume that $j = i^*$.

We focus on testing the robustness of RCAC for the jammed actuator case. For all test cases, the aircraft is given a zero-deg roll command instead of a change-of-heading command in order to directly address the rolling and pitching moments caused by aileron and rudder jam. The controller gains are first allowed to converge during the inital level straight-line flight, and then the actuator failure occurs at 200 sec.

IV.A. Case 1: Erroneous Markov Parameter Choice

To determine how different Markov parameters can impact the performance of the controller, we choose $i = 1,2,3,4,5$ and obtain H_i from the linearized system. We then test the robustness of RCAC under each of the five Markov parameters in various angles of jammed actuator faults. For each Markov parameter, an optimal tuning parameter is obtained to provide a baseline indication of the best achievable performance of RCAC in each case. The optimal tunings for the first five Markov parameters are given by

$$\tilde{\mathcal{H}} = H_1, i = 1, n_c = 16, P_0 = 0.01, \eta_0 = 0.0003, \tag{39}$$

$$\tilde{\mathcal{H}} = H_2, i = 2, n_c = 16, P_0 = 0.003, \eta_0 = 0.0007, \tag{40}$$

$$\tilde{\mathcal{H}} = H_3, i = 3, n_c = 16, P_0 = 0.0005, \eta_0 = 0.0005, \tag{41}$$

$$\tilde{\mathcal{H}} = H_4, i = 4, n_c = 16, P_0 = 0.0003, \eta_0 = 0.0005, \tag{42}$$

$$\tilde{\mathcal{H}} = H_5, i = 5, n_c = 16, P_0 = 0.001, \eta_0 = 0.0007, \tag{43}$$

We compare the performance of RCAC under each choice of Markov parameters, each tested at its optimal tuning parameters. Figure 11 shows the performance of the controller using H_2 under a -8-deg jammed rudder deflection angle. Figure 12 shows the performance of the controller using H_4 under a $-6-deg$ jammed aileron deflection angle.

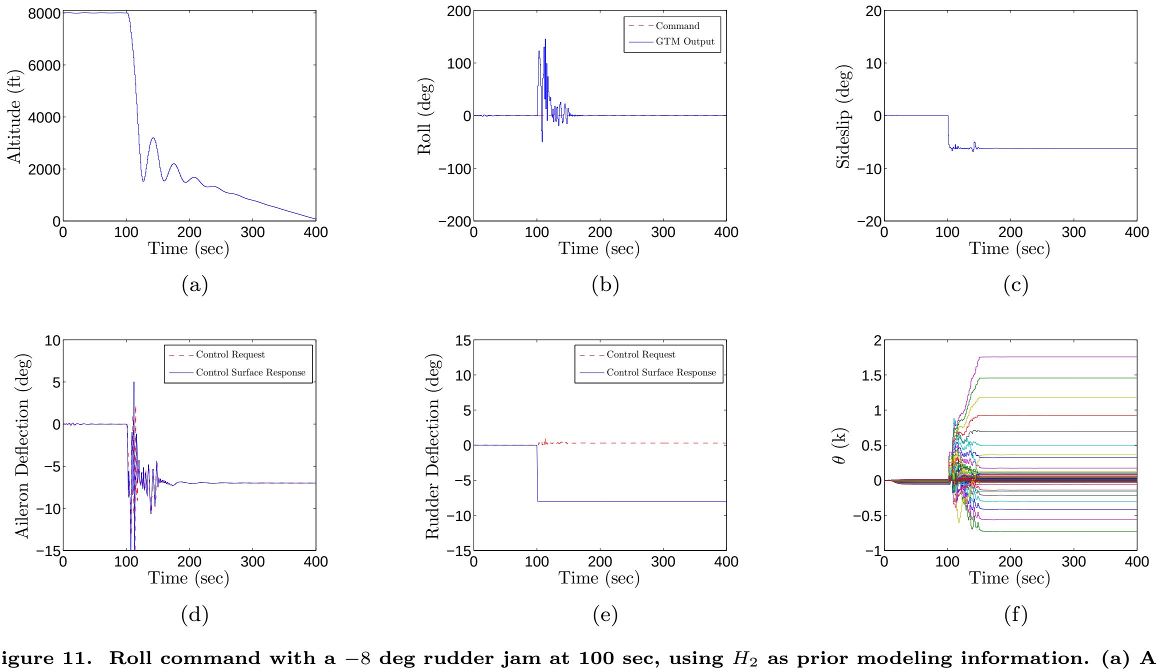


Figure 11. Roll command with a -8 deg rudder jam at 100 sec, using H_2 as prior modeling information. (a) Altitude, (b) roll, (c) sideslip, (d) aileron deflection, (e) rudder deflection, and (f) adaptive gains. The adaptive gains of the centralized controller change in response to the fault and converge within approximately 50 sec.

Table 5 summarizes the performance of RCAC under the first 5 Markov parameters at their optimal tunings.

Jam Angle (deg)	H1,A	H1,R	H2,A	H2,R	H3,A	H3,R	H4,A	H4,R	H5,A	H5,R
-10	N	N	N	N	N	N	N	N	N	N
-9	N	Y	N	N	N	N	N	N	N	N
-8	N	Y	N	Y	N	Y	N	N	N	N
-7	N	Y	N	Y	N	N	N	N	Y	N
-6	N	Y	N	Y	N	Y	Y	Y	N	N
-5	N	N	N	Y	Y	Y	Y	Y	Y	Y
-4	N	N	N	Y	N	Y	Y	Y	Y	Y
-3	N	N	N	Y	Y	Y	Y	Y	Y	Y
-2	N	N	N	Y	Y	Y	N	Y	Y	Y
-1	N	Y	N	Y	Y	Y	Y	Y	Y	Y
0	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
1	N	Y	N	Y	Y	Y	Y	Y	Y	Y
2	N	N	N	Y	Y	Y	Y	Y	Y	Y
3	N	N	N	Y	Y	Y	Y	Y	Y	Y
4	N	N	N	Y	N	Y	Y	Y	N	Y
5	N	N	N	Y	Y	Y	Y	Y	Y	Y
6	N	N	N	Y	N	Y	Y	N	N	N
7	N	Y	N	Y	N	Y	N	N	Y	Y
8	N	Y	N	Y	N	Y	N	N	N	N
9	N	N	N	Y	N	N	N	N	N	N
10	N	N	N	N	N	N	N	N	N	N

Table 5. Summary of controller performance for jammed actuators. (H1) First Markov parameter, (H2) second Markov parameter, (H3) third Markov parameter, (H4) fourth Markov parameter, (H5) fifth Markov parameter, (A) aileron jammed, and (R) rudder jammed. The notation used is, Y: the controller was able to track the command, and N: the controller was not able to track the command. H_3, H_4, H_5 produce the best results. Both H_1 and H_2 exhibit low robustness with respect to jammed aileron faults and high robustness with respect to jammed rudder faults. H_4 and H_5 exhibit slightly lower robustness than H_3 for jammed aileron and rudder faults.

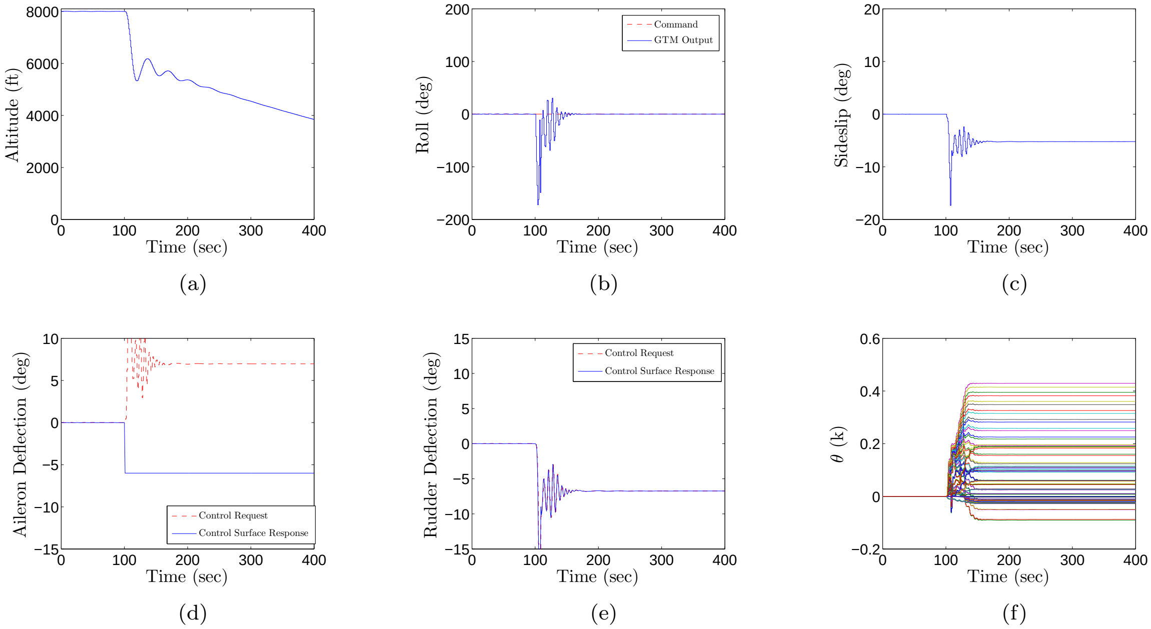


Figure 12. Roll command with a -6 deg aileron jam at 100 sec, using H_4 as prior modeling information. (a) Altitude, (b) roll, (c) sideslip, (d) aileron deflection, (e) rudder deflection, and (f) adaptive gains. The adaptive gains of the centralized controller change in response to the fault and converge within approximately 50 sec.

IV.B. Case 2: Discrepancy between Markov Parameter Choice and $\tilde{\mathcal{H}}$

To determine how a mismatch between i and H_{i^*} can impact the performance of the controller, we choose $i = 1,2,3,4,5$ and obtain $H_{i^*} = H_3$ from the linearized system. Keeping $\tilde{\mathcal{H}} = H_3$ the same for each simulation while varying i, we test the robustness of RCAC to jammed actuator faults. Note that Markov parameter choice is used to determine the number of past time steps used to construct the retrospectively optimized controls $\hat{U}(k-1)$, therefore RCAC may produce different outputs despite using the same $\tilde{\mathcal{H}}$.

For each Markov parameter, an optimal tuning parameter is obtained to provide a baseline indication of the best achievable performance of RCAC in each case. The optimal tunings for the first five Markov parameters are given by

$$\tilde{\mathcal{H}} = H_3, i = 1, n_c = 16, P_0 = 0.001, \eta_0 = 0.0007 \tag{44}$$

$$\tilde{\mathcal{H}} = H_3, i = 2, n_c = 16, P_0 = 0.003, \eta_0 = 0.0007 \tag{45}$$

$$\tilde{\mathcal{H}} = H_3, i = 3, n_c = 16, P_0 = 0.0005, \eta_0 = 0.0005, \tag{46}$$

$$\tilde{\mathcal{H}} = H_3, i = 4, n_c = 16, P_0 = 0.001, \eta_0 = 0.001 \tag{47}$$

$$\tilde{\mathcal{H}} = H_3, i = 5, n_c = 16, P_0 = 0.001, \eta_0 = 0.0009 \tag{48}$$

We compare the performance of each choice of i while using the same $\tilde{\mathcal{H}}$. Figure 13 shows the performance of the controller using $i = 5$ under a -4-deg jammed aileron deflection angle. Figure 14 shows the performance of the controller using $i = 1$ under an 8-deg jammed rudder deflection angle.

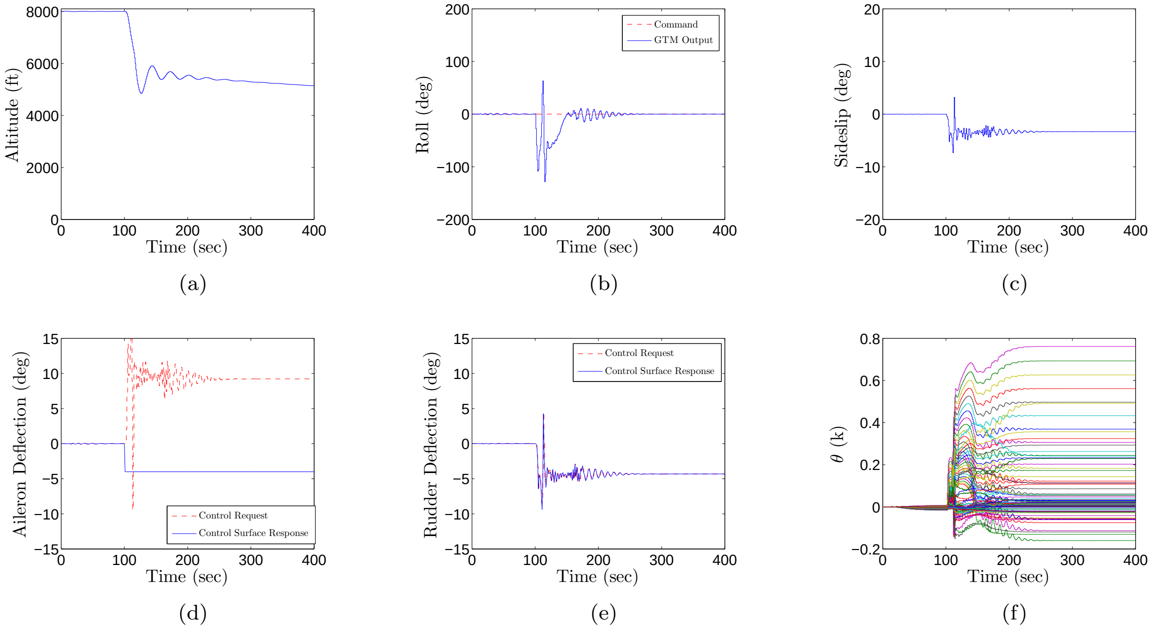


Figure 13. Roll command with a -4 deg aileron jam at 100 sec, using H_5 as prior modeling information. (a) Altitude, (b) roll, (c) sideslip, (d) aileron deflection, (e) rudder deflection, and (f) adaptive gains. The adaptive gains of the centralized controller change in response to the fault and converge within approximately 150 sec.

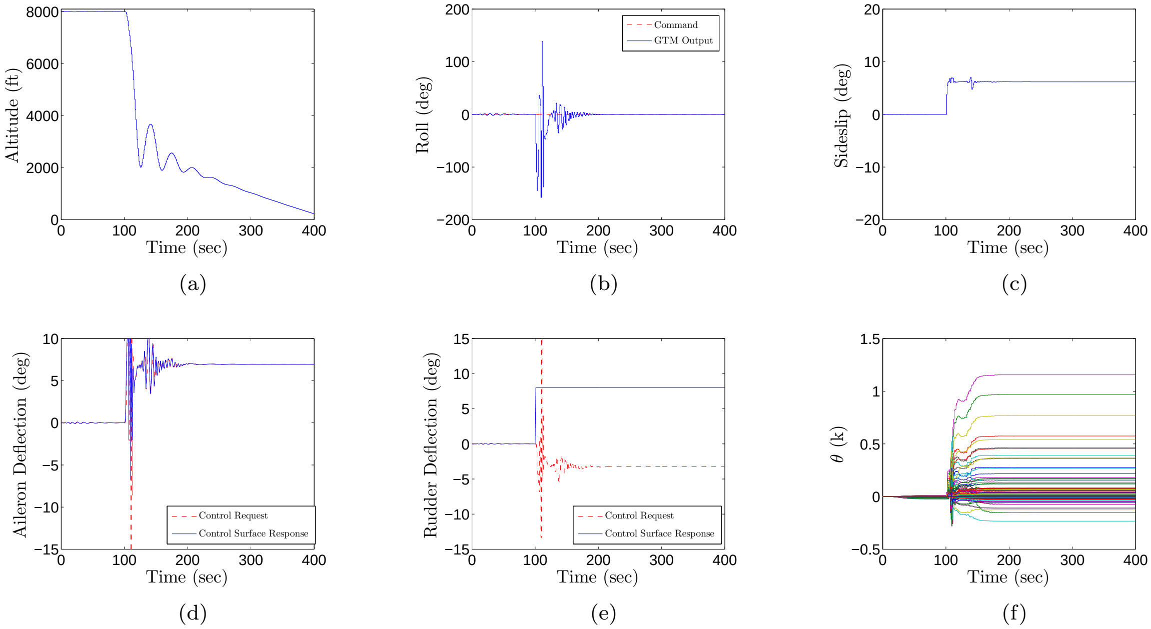


Figure 14. Roll command a 8 deg rudder jam at 100 sec, using H_1 as prior modeling information. (a) Altitude, (b) roll, (c) sideslip, (d) aileron deflection, (e) rudder deflection, and (f) adaptive gains. The adaptive gains of the centralized controller change in response to the fault and converge within approximately 100 sec.

Table 6 summarizes the adaptive controller's performance under the first 5 Markov parameters and the same matrix $\tilde{\mathcal{H}}$.

Jam Angle (deg)	H1,A	H1,R	H2,A	H2,R	H3,A	H3,R	H4,A	H4,R	H5,A	H5,R
-10	N	N	N	N	N	N	N	N	N	N
-9	N	N	N	N	N	N	N	N	N	N
-8	N	Y	N	Y	N	Y	N	N	N	N
-7	N	Y	N	Y	N	N	N	Y	N	Y
-6	N	Y	N	N	N	Y	N	Y	N	Y
-5	Y	Y	Y	N	Y	Y	N	Y	N	Y
-4	Y	Y	Y	Y	N	Y	Y	Y	Y	Y
-3	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
-2	Y	Y	N	Y	Y	Y	Y	Y	Y	Y
-1	N	N	N	Y	Y	Y	Y	Y	Y	Y
0	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
1	N	N	Y	Y	Y	Y	Y	Y	Y	Y
2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
3	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
4	Y	Y	Y	Y	N	Y	Y	Y	Y	Y
5	Y	Y	N	Y	Y	Y	N	Y	N	Y
6	N	Y	N	Y	N	Y	N	Y	N	Y
7	N	Y	N	Y	N	Y	N	Y	N	N
8	N	Y	N	N	N	Y	N	N	N	N
9	N	Y	N	N	N	N	N	N	N	N
10	N	N	N	N	N	N	N	N	N	N

Table 6. Summary of controller performance for jammed actuators. (H1) First Markov parameter, (H2) second Markov parameter, (H3) third Markov parameter, (H4) fourth Markov parameter, (H5) fifth Markov parameter, (A) aileron jammed, and (R) rudder jammed. The notation used is, Y: the controller was able to track the command, and N: the controller was not able to track the command. Setting $i = 3$ produces the best results. $i = 2$ produced matching robustness with respect to jammed rudder faults, while $i = 1, 2, 4, 5$ exhibit similar robustness with respect to jammed aileron faults.

V. Conclusions

We compared two implementations of RCAC for adaptive flight control of the NASA GTM model under various unknown control-surface faults,. To compare the two controller architectures, we considered the case in which heading is the performance variable and the elevator, ailerons, and rudder are subjected to sudden unknown faults. The centralized RCAC controller is a 3x1 SIMO controller, whose gains are updated by RCAC. In the constrained configuration, the gains of each controller are adapted independently by three separate RCAC blocks. It was found that the centralized architecture exhibits increased robustness to control-surface faults.

Both controllers were tested with four types of unknown actuator faults, namely, magnitude and rate saturation, deadzone, and jams. In most cases, both controller architectures was able to follow a 90-deg heading-change command over a wide range of unknown faults. The most challenging case was found to be a jammed rudder, in which case neither the centralized nor the constrained controller was entirely successful. In all but one case, centralized RCAC at least matches the performance of the constrained architecture, and in a few cases exceeds it.

We also characterized the robustness of the centralized architecture with respect to prior modeling information. We considered the case in which the roll angle is the performance variable and the aileron and rudder are subjected to sudden unknown faults. The controller is first tested with respect to the first five Markov parameters and their corresponding $\tilde{\mathcal{H}}$. RCAC is then tested by using the same $\tilde{\mathcal{H}}$ for the first five Markov parameters.

The robustness of the controller was tested with respect to the jammed aileron and rudder fault. In most cases, the controller was able to maintain a zero-deg roll command over a wide range of Markov parameter choices and their corresponding $\tilde{\mathcal{H}}$. When the matrix $\tilde{\mathcal{H}}$ can be correctly identified from a given choice of Markov parameter, we found that the controller performed best for Markov parameter choices higher than the third Markov parameter. When the matrix $\tilde{\mathcal{H}}$ is identified, but the corresponding Markov parameter is uncertain, we found that the controllers produce similar performances. For the first case, picking the incorrect Markov parameter can lead to a steep decline in the robustness of an actuator.

Future research will focus on further characterizing the robustness of RCAC, with the goal of obtaining a rubric for maximizing the controller's performance in the events of both uncertain actuator faults and prior modeling information.

References

[1] R. Bailey and T. Jordan. NASA Langley's AirSTAR Testbed: A subscale flight test capability for flight dynamics and control system experiments. In *Proc. AIAA Guid. Nav. Contr. Conf.*, Honolulu, HI, August 2008. AIAA-2008-6660.

[2] A. M. D'Amato, E. D. Sumer, and D. S. Bernstein. Frequency-domain stability analysis of retrospective-cost adaptive control for systems with unknown nonminimum-phase zeros. In *Proc. IEEE Conf. Dec. Contr.*, pages 1098–1103, Orlando, FL, December 2011.

[3] A. M. D'Amato, E. D. Sumer, and D. S. Bernstein. Retrospective cost adaptive control for systems with unknown nonminimum-phase zeros. In *Proc. AIAA Guid. Nav. Contr. Conf.*, Portland, OR, August 2011. AIAA-2011-6203.

[4] A. M. D'Amato, E. D. Sumer, K. S. Mitchell, A. V. Morozov, J. B. Hoagg, and D. S. Bernstein. Adaptive output feedback control of the NASA GTM model with unknown nonminimum-phase zeros. In *Proc. AIAA Guid. Nav. Contr. Conf.*, Portland, OR, August 2011. AIAA-2011-6204.

[5] M. S. Fledderjohn, M. S. Holzel, H. Palanthandalam-Madapusi, R. J. Fuentes, and D. S. Bernstein. A comparison of least squares algorithms for estimating markov parameters. In *Proc. Amer. Contr. Conf.*, pages 3735–3740, Baltimore, MD, June 2010.

[6] J. B. Hoagg and D. S. Bernstein. Retrospective cost adaptive control for nonminimum-phase discrete-time systems part 1: The ideal controller and error system, part 2: The adaptive controller and stability analysis. In *Proc. Conf. Dec. Contr.*, pages 893–904, Atlanta, GA, December 2010.

[7] J. B. Hoagg and D. S. Bernstein. Retrospective cost model reference adaptive control for nonminimum-phase discrete-time systems, part 1: The ideal controller and error system; part 2: The adaptive controller and stability analysis. In *Proc. Amer. Contr. Conf.*, pages 2927–2938, San Francisco, CA, June 2011.

[8] J. B. Hoagg, M. A. Santillo, and D. S. Bernstein. Discrete-time adaptive command following and disturbance rejection for minimum phase systems with unknown exogenous dynamics. *IEEE Trans. Autom. Contr.*, 53:912–928, 2008.

[9] N. Hovakimyan, C. Cao, E. Kharisov, E. Xargay, and I. Gregory. Adaptive controller for safety critical systems: Guaranteed robustness with fast adaptation. *IEEE Control Systems Magazine*, pages 54–106, October 2011.

[10] I. Hwang, S. Kim, Y. Kim, and C. Seah. A survey of fault detection, isolation, and reconfiguration methods. *IEEE Trans. Contr. Sys. Tech.*, 18:636–653, 2010.

[11] A. Murch. A flight control system architecture for the NASA AirSTAR flight test infrastructure. Honolulu, HI, August 2008. AIAA-2008-6990.

[12] M. A. Santillo and D. S. Bernstein. Adaptive control based on retrospective cost optimization. *AIAA J. Guid. Contr. Dyn.*, 33:289–304, 2010.

[13] E.D. Sumer and D.S. Bernstein. Adaptive decentralized noise and vibration control with conflicting performance objectives. In *Proc. DSCC*, pages 1–10, Fort Lauderdale, FL, October 2012.