AIAA Guidance, Navigation, and Control Conference
13 - 16 August 2012, Minneapolis, Minnesota

AIAA 2012-4624

Retrospective Cost Adaptive Control of Spacecraft Attitude

Gerardo Cruz,[*] Anthony M. D'Amato,[†] and Dennis S. Bernstein[‡]

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

We apply retrospective cost adaptive control (RCAC) to spacecraft attitude control. First, we develop results for angular rate control. These results are then extended to attitude control. We examine two problems for each of the controllers. For both problems, the spacecraft has an arbitrary initial angular rate, and in the case of attitude control, an arbitrary initial attitude. The objective for the first problem is to bring the body to rest and to a specified attitude in the attitude control case. The second problem seeks to bring the spacecraft to spin about a specified body axis, which, in the case of attitude control, is inertially pointed. We first test the algorithm using an estimate of the spacecraft's Markov parameters obtained from discretization of the linearized Euler's and Poisson's equations. Then, we limit the dependence on knowledge of the mass properties by removing inertia information from the Markov parameter. Finally, we test for robustness by scaling the Markov parameter and rotating the actuator matrix.

[*] Graduate Student, Department of Aerospace Engineering

[†] Graduate Student, Department of Aerospace Engineering

[‡] Professor, Department of Aerospace Engineering

Copyright © 2012 by Gerardo Cruz. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

I. Introduction

Control of rigid body spacecraft is a widely studied problem [1]. Controller tuning requires knowledge of the mass properties of the spacecraft as well as information about actuator placement and orientation. The effects of sensor and actuator misalignments, measurement errors, time delays, and other errors are minimized through control analysis and design. Robustness is achieved through large gain and phase margins, controller performance is evaluated through extensive Monte Carlo simulations. These techniques, though effective, require significant time and effort. Spacecraft projects with limited resources can benefit from control algorithms that are insensitive to modeling errors.

Adaptive techniques for spacecraft attitude control are developed in [2–8]. These techniques allow the controller to tune itself to the actual dynamics of the spacecraft. Thus, they are useful when a sufficiently accurate model of the spacecraft is not available for fixed controller synthesis.

Retrospective cost adaptive control (RCAC) is a multi-input, multi-output direct adaptive controller that requires limited modeling information. This method has been applied to systems with unknown nonminimum-phase zeros [13–15] for stabilization, command following, and disturbance rejection. RCAC has also been tested on nonlinear plants such as multiple linkages [16, 17].

In this paper we apply RCAC to spacecraft attitude control. First, the algorithm is implemented as a regulator for the angular rate of a rigid body governed by Euler's equation. This type of controller can be used to detumble a spacecraft after launch-vehicle separation; this is the rate-only motion-to-rest (M2R-R) problem. Then, we consider commanded spins about an arbitrary body axis; this constitutes the rate-only motion-to-spin (M2S-R) problem. Initially, we use knowledge of the inertia and actuator alignment to compute the parameters required by RCAC. The algorithm is then tested for robustness to scaling of the inertia and actuator matrices.

The angular rate controllers are then extended to attitude control. Attitude kinematics are included in the spacecraft model, and attitude-dependent motion-to-rest (M2R-A) and motion-to-spin (M2S-A) maneuvers are tested. For M2R-A, the spacecraft has an initial attitude and angular rate, and the objective is to bring the spacecraft to rest at a specified attitude. For M2S-A, the spacecraft has an arbitrary initial attitude and angular rate, and the objective is to bring the spacecraft to spin about a specified body axis that is pointed inertially. The body spin axis need not be a principal axis. As in the angular rate control case, both problems are first examined using complete knowledge of the inertia and actuator alignment. Robustness is then examined through scaling and misalignments.

II. Spacecraft Model

For the spacecraft model, we consider a single rigid body controlled by force or torque actuators, such as thrusters or magnetic torque devices. We consider only the rotational motion of the spacecraft while ignoring the translational motion of the spacecraft's center of mass; therefore we consider only the torque applied by the force actuators. We define a body-fixed frame for the spacecraft, whose origin is chosen to be the center of mass, and specify an inertial frame to determine the attitude of the spacecraft. The spacecraft equations of motion are given by Euler's and Poisson's equations,

$$J\dot{\omega} = (J\omega) \times \omega + B_{SC}u + z_d, \tag{1}$$

$$\dot{R} = R\omega^{\times}, \tag{2}$$

respectively, where $\omega \in \mathbb{R}^3$ is the angular rate of the spacecraft frame with respect to the inertial frame resolved in the spacecraft frame and $J \in \mathbb{R}^{3\times 3}$ is the constant, positive-definite inertia matrix of the spacecraft, that is, the inertia dyadic of the spacecraft relative to the spacecraft center of mass resolved in the spacecraft frame. Furthermore, $R \in \mathbb{R}^{3\times 3}$ is the proper orthogonal matrix (that is, the rotation matrix) that transforms the components of a vector resolved in the spacecraft frame into the components of the same vector resolved in the inertial frame, and $\omega^{\times}$ is the skew-symmetric cross-product matrix of ω.

The components of the vector $u \in \mathbb{R}^{l_u}$ represent independent control inputs, while the matrix $B_{SC} \in \mathbb{R}^{3\times l_u}$ determines the applied torque about each axis of the spacecraft frame due to u as given by the product $B_{SC}u$. The vector z_d represents disturbance torques, that is, all internal and external torques applied to the spacecraft aside from control torques. These disturbances may be due to onboard components, gravity gradients, solar pressure, atmospheric drag, or the ambient magnetic field. For convenience in (1), (2) we omit the argument t, recognizing that ω, R, u, and z_d are time-varying quantities.

We assume that both rate (inertial) and attitude (noninertial) measurements are available. Gyro measurements $y_{\rm rate} \in \mathbb{R}^3$ provide measurements of the angular rate resolved in the spacecraft frame, that is,

$$y_{\rm rate} = \omega. \tag{3}$$

For simplicity, we assume that rate measurements are available without noise and bias. In practice, rate bias can be corrected by using attitude measurements and filtering techniques.

Attitude is measured indirectly using sensors such as star trackers. The attitude measurement is determined to be

$$y_{\rm attitude} = R. \tag{4}$$

The objective of the attitude control problem is to determine control inputs such that the spacecraft attitude given by R follows a commanded attitude trajectory given by a possibly time-varying C^1 rotation matrix $R_{\rm d}(t)$. For $t \geq 0$, $R_{\rm d}(t)$ is given by

$$\dot{R}_{\rm d}(t) = R_{\rm d}(t)\omega_{\rm d}(t)^\times, \tag{5}$$

$$R_{\rm d}(0) = R_{\rm d0}, \tag{6}$$

where $\omega_{\rm d}$ is the desired, possibly time-varying angular rate. The attitude error, that is, the rotation between $R(t)$ and $R_{\rm d}(t)$, is given by

$$\tilde{R} \overset{\triangle}{=} R_{\rm d}^{\rm T} R, \tag{7}$$

which satisfies Poisson's equation

$$\dot{\tilde{R}} = \tilde{R}\tilde{\omega}^\times, \tag{8}$$

where the angular rate error $\tilde{\omega}$ is defined by

$$\tilde{\omega} \overset{\triangle}{=} \omega - \tilde{R}^{\rm T}\omega_{\rm d}. \tag{9}$$

We then rewrite (1) in terms of the angular rate error (9) as

$$J\dot{\tilde{\omega}} = [J(\tilde{\omega} + \tilde{R}^{\rm T}\omega_{\rm d})] \times (\tilde{\omega} + \tilde{R}^{\rm T}\omega_{\rm d}) + J(\tilde{\omega} \times \tilde{R}^{\rm T}\omega_{\rm d} - \tilde{R}^{\rm T}\dot{\omega}_{\rm d}) + Bu + z_d. \tag{10}$$

III. The RCAC Algorithm

RCAC is a discrete-time output-feedback controller that minimizes the command-following error corresponding to the performance variable z. The algorithm does not require detailed plant information, instead, RCAC uses knowledge of Markov parameters.

Consider the MIMO discrete-time system

$$x(k+1) = Ax(k) + Bu(k), \tag{11}$$

$$y_0(k) = E_1 x(k), \tag{12}$$

$$z(k) = y_0(k) - E_0 r(k), \tag{13}$$

where $x(k) \in \mathbb{R}^{l_x}$, $y_0(k) \in \mathbb{R}^{l_y}$, $z(k) \in \mathbb{R}^{l_z}$, $u(k) \in \mathbb{R}^{l_u}$, $r(k) \in \mathbb{R}^{l_w}$, and $k \geq 0$.

A. Retrospective Cost

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

$$H_i \overset{\triangle}{=} E_1 A^{i-1} B. \tag{14}$$

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

$$x(k) = A^n x(k-n) + \sum_{i=1}^{n} A^{i-1} Bu(k-i) \tag{15}$$

thus

$$z(k) = E_1 A^n x(k-n) + \bar{H}\bar{U}(k-1) - E_0 r(k), \tag{16}$$

where

$$\bar{H} \overset{\triangle}{=} \begin{bmatrix} H_1 & \cdots & H_n \end{bmatrix} \in \mathbb{R}^{l_z \times n l_u}$$

and

$$\bar{U}(k-1) \overset{\triangle}{=} \begin{bmatrix} u(k-1) \\ \vdots \\ u(k-n) \end{bmatrix} \in \mathbb{R}^{n l_u}.$$

Next, assume we know $l_{\mathcal{H}}$ Markov parameters, rearrange the columns of $\bar{H}$ and the entries of $\bar{U}(k-1)$ then partition the resulting matrix and vector so that

$$\bar{H}\bar{U}(k-1) = \mathcal{H}'U'(k-1) + \mathcal{H}U(k-1). \tag{17}$$

where $U \in \mathbb{R}^{l_{\mathcal{H}} l_u}$ and $U' \in \mathbb{R}^{(n-l_h)l_u}$. Furthermore, $\mathcal{H} \in \mathbb{R}^{l_z \times l_{\mathcal{H}} l_u}$ and $\mathcal{H}' \in \mathbb{R}^{l_z \times l_u(n-l_{\mathcal{H}})}$ are the known and unknown Markov parameters respectively. For example, if $\bar{H} = \begin{bmatrix} H_1 & H_2 & H_3 \end{bmatrix}$, we can divide it into $\mathcal{H}' = \begin{bmatrix} H_1 & H_3 \end{bmatrix}$ with $U'(k-1) = \begin{bmatrix} u(k-1) \\ u(k-3) \end{bmatrix}$ then $\mathcal{H} = H_2$ with the corresponding $U = u(k-2)$. We can rewrite (16) as

$$z(k) = \mathcal{S}(k) + \mathcal{H}U(k-1), \tag{18}$$

where

$$\mathcal{S}(k) \overset{\triangle}{=} E_1 A^n x(k-n) - E_0 r(k) + \mathcal{H}'U'(k-1) \tag{19}$$

collects all the unknown parameters of the system.

Let s be a positive integer, then for $j = 1, \ldots, s$, we add a delay k_j in (18) so that $0 \leq k_1 \leq k_2 \leq \cdots \leq k_s$. The delayed performance is

$$z(k-k_j) = \mathcal{S}_j(k-k_j) + \mathcal{H}_j U_j(k-k_j-1). \tag{20}$$

where

$$\mathcal{S}_j(k-k_j) \overset{\triangle}{=} E_1 A^n x(k-k_j-n) - E_0 r(k-k_j) + \mathcal{H}'_j U'_j(k-k_j-1), \tag{21}$$

and (17) becomes

$$\bar{H}\bar{U}(k-k_j-1) = \mathcal{H}'_j U'_j(k-k_j-1) + \mathcal{H}_j U_j(k-k_j-1). \tag{22}$$

We stack $z(k-k_1),\ldots,z(k-k_s)$, and define the *extended performance*

$$Z(k) \triangleq \begin{bmatrix} z(k-k_1) \\ \vdots \\ z(k-k_s) \end{bmatrix} \in \mathbb{R}^{sl_z}. \tag{23}$$

Therefore,

$$Z(k) \triangleq \tilde{\mathcal{S}}(k) + \tilde{\mathcal{H}}\tilde{U}(k-1), \tag{24}$$

where

$$\tilde{\mathcal{S}}(k) \triangleq \begin{bmatrix} \mathcal{S}_1(k-k_1) \\ \vdots \\ \mathcal{S}_s(k-k_s) \end{bmatrix} \in \mathbb{R}^{sl_z}, \tag{25}$$

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

$$\tilde{U}(k-1) \triangleq \begin{bmatrix} u(k-q_1) \\ \vdots \\ u(k-q_g) \end{bmatrix} \in \mathbb{R}^{gl_u}, \tag{26}$$

where $k_1 \le q_1 < q_2 < \cdots < q_g \le k_s + n$. 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, and $\tilde{\mathcal{H}} \in \mathbb{R}^{sl_z \times gl_u}$ is constructed according to the structure of $\tilde{U}(k-1)$.

We also define the *retrospective performance*,

$$\hat{z}(k-k_j) \triangleq \mathcal{S}_j(k-k_j) + \mathcal{H}_j\hat{U}_j(k-k_j-1), \tag{27}$$

where the past controls $U_j(k-k_j-1)$ in (20) are replaced by the retrospective controls $\hat{U}_j(k-k_j-1)$, which are computed in (36) below. As in (23), we define the *extended retrospective performance*

$$\hat{Z}(k) \triangleq \begin{bmatrix} \hat{z}(k-k_1) \\ \vdots \\ \hat{z}(k-k_s) \end{bmatrix} \in \mathbb{R}^{sl_z}, \tag{28}$$

thus

$$\hat{Z}(k) = \tilde{\mathcal{S}}(k) + \tilde{\mathcal{H}}\hat{\tilde{U}}(k-1), \tag{29}$$

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 the extended performance in (24) from the extended retrospective performance in (29) yields

$$\hat{Z}(k) = Z(k) - \tilde{\mathcal{H}}\tilde{U}(k-1) + \tilde{\mathcal{H}}\hat{\tilde{U}}(k-1). \tag{30}$$

Thus, we define the *retrospective cost function*

$$J(\hat{\tilde{U}}(k-1),k) \triangleq \hat{Z}^{\mathrm{T}}(k)R(k)\hat{Z}(k), \tag{31}$$

where $R(k) \in \mathbb{R}^{l_z s \times l_z s}$ is a positive-definite performance weighting.

The goal is to determine the retrospective controls $\hat{\tilde{U}}(k-1)$ that minimize the retrospective performance $\hat{Z}(k)$. These retrospectively optimized control values $\hat{\tilde{U}}(k-1)$ are then used to update the controller. To this end we solve the optimization problem in (31).

Expanding (31) with (30) yields

$$J(\hat{\tilde{U}}(k-1),k) = \hat{\tilde{U}}(k-1)^{\rm T}\mathcal{A}(k)\hat{\tilde{U}}(k-1) + \hat{\tilde{U}}^{\rm T}(k-1)\mathcal{B}^{\rm T}(k) + \mathcal{C}(k), \tag{32}$$

where

$$\mathcal{A}(k) \triangleq \tilde{\mathcal{H}}^{\rm T}R(k)\tilde{\mathcal{H}}, \tag{33}$$

$$\mathcal{B}(k) \triangleq 2\tilde{\mathcal{H}}^{\rm T}R(k)[Z(k) - \tilde{\mathcal{H}}\tilde{U}(k-1)], \tag{34}$$

$$\mathcal{C}(k) \triangleq Z^{\rm T}(k)R(k)Z(k) - 2Z^{\rm T}(k)R(k)\tilde{\mathcal{H}}\tilde{U}(k-1) + \tilde{U}^{\rm T}(k-1)\tilde{\mathcal{H}}^{\rm T}R(k)\tilde{\mathcal{H}}\tilde{U}(k-1). \tag{35}$$

Given a full column rank $\tilde{\mathcal{H}}$, $\mathcal{A}(k)$ is positive definite and $J(\hat{\tilde{U}}(k-1),k)$ has a unique global minimizer which is the optimized retrospective control

$$\hat{\tilde{U}}(k-1) = -\frac{1}{2}\mathcal{A}^{-1}(k)\mathcal{B}(k). \tag{36}$$

B. Controller Construction

We design a strictly proper time-series controller of order $n_{\rm c}$ given by

$$u(k) = \sum_{i=1}^{n_{\rm c}} M_i(k)u(k-i) + \sum_{i=1}^{n_{\rm c}} N_i(k)z(k-i), \tag{37}$$

where, for all $i = 1,\ldots,n_{\rm c}$, $M_i(k) \in \mathbb{R}^{l_u\times l_u}$ and $N_i(k) \in \mathbb{R}^{l_u\times l_z}$. The control (37) can be expressed as

$$u(k) = \theta(k)\phi(k-1), \tag{38}$$

where

$$\theta(k) \triangleq [M_1(k) \ \cdots \ M_{n_{\rm c}}(k) \ N_1(k) \ \cdots \ N_{n_{\rm c}}(k)] \in \mathbb{R}^{l_u\times n_{\rm c}(l_u+l_z)} \tag{39}$$

and

$$\phi(k-1) \triangleq \begin{bmatrix} u(k-1) \\ \vdots \\ u(k-n_{\rm c}) \\ z(k-1) \\ \vdots \\ z(k-n_{\rm c}) \end{bmatrix} \in \mathbb{R}^{n_{\rm c}(l_u+l_z)}. \tag{40}$$

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

Define the cumulative cost function

$$J_{\rm R}(\theta(k)) \triangleq \sum_{i=q_g+1}^{k} \lambda^{k-i}\|\phi^{\rm T}(i-q_g-1)\theta^{\rm T}(k) - \hat{u}^{\rm T}(i-q_g)\|^2 + \lambda^k(\theta(k)-\theta(0))P^{-1}(0)(\theta(k)-\theta(0))^{\rm T}, \tag{41}$$

where $\|\cdot\|$ is the Euclidean norm and, for some $\varepsilon \in (0,1)$, $\lambda(k) \in (\varepsilon, 1]$ is the forgetting factor, and $P(0) \in \mathbb{R}^{n_{\rm c}(l_u+l_z)\times n_{\rm c}(l_u+l_z)}$ is the initial covariance matrix. Minimizing (41) yields

$$\begin{aligned}\theta^{\rm T}(k) \triangleq \theta^{\rm T}(k-1) + \beta(k)P(k-1)\phi(k-q_g-1)[\phi^{\rm T}(k-q_g-1)P(k-1)\phi(k-q_g-1) + \lambda(k)]^{-1} \\ \cdot [\theta(k-1)\phi(k-q_g-1) - \hat{u}(k-q_g)]^{\rm T},\end{aligned} \tag{42}$$

where $\beta(k)$ is a switch on the control such that

$$\beta(k) = \begin{cases} 0 & k < k_{\text{on}} \\ 1 & k \geq k_{\text{on}} \end{cases} \tag{43}$$

and k_{on} is the time step at which we wish the controller to start operating.

The error covariance is updated by

$$\begin{aligned} P(k) \overset{\triangle}{=} & [1-\beta(k)]P(k-1) + \beta(k)\lambda^{-1}(k)P(k-1) - \beta(k)\lambda^{-1}(k)P(k-1)\phi(k-q_g-1) \\ & \cdot [\phi^{\mathrm{T}}(k-q_g-1)P(k-1)\phi(k-q_g-1) + \lambda(k)]^{-1}\phi^{\mathrm{T}}(k-q_g-1)P(k-1). \end{aligned} \tag{44}$$

We initialize the error covariance matrix as $P(0) = \gamma I$, where $\gamma > 0$. Furthermore, the updates (42) and (44) are based on the g^{th} component of $\hat{\tilde{U}}(k-1)$. However any or all of the components of $\hat{\tilde{U}}(k-1)$ may be used in the update of $\theta(k)$ and $P(k)$.

IV. Performance Variable for Angular Rate Control

For a discrete-time plant the Markov parameters are obtained by evaluating (14). However, the angular rate and attitude control problems are governed by nonlinear, continuous-time equations. Thus, for angular rate control we linearize and discretize (10) to obtain the required Markov parameters to use in RCAC.

In the angular rate control problem the attitude error is ignored. Let $\tilde{R} = I_3$, which corresponds to zero attitude error, then the performance variable for RCAC becomes

$$z = \omega - \omega_d. \tag{45}$$

Next, we linearize Euler's equation in (10) about a desired equilibrium $\tilde{\omega}_e$, which yields the Jacobian

$$A_c(\tilde{\omega}_e, \omega_d) = \frac{\partial \dot{\tilde{\omega}}}{\partial \tilde{\omega}} = J^{-1}\left[-(\tilde{\omega}_e + \omega_d)^{\times} J + [J(\tilde{\omega}_e + \omega_d)]^{\times}\right] - \omega_d^{\times}, \tag{46}$$

Similarly, the input matrix for the linearized system is given by

$$B_c = \frac{\partial \dot{\omega}}{\partial u} = J^{-1}B_{SC}. \tag{47}$$

For the angular rate control problem the output matrices are given by $C = I$ and $D = 0$.

We obtain the discrete-time dynamics matrix from

$$A = e^{A_c h}, \tag{48}$$

where A_c is given by (46) and h is the controller time step. The discrete-time input matrix is given by

$$B = \int_0^h e^{A\tau}B_c d\tau. \tag{49}$$

The discrete-time output matrices are $E_1 = C$ and $D_d = D$. These discrete-time matrices are used in (14) to compute the Markov parameter for RCAC. For this paper we use the first Markov parameter

$$H_1 = E_1 B. \tag{50}$$

V. Numerical Examples for Angular Rate Control

Consider a rigid spacecraft with the inertia matrix

$$J = \begin{bmatrix} 5 & -0.1 & -0.5 \\ -0.1 & 2 & 1 \\ -0.5 & 1 & 3.5 \end{bmatrix} \text{ kg-m}^2 \tag{51}$$

Parameter	Value
n_c	3
P_0	$100I$
R	I
s	1
k_s	1
θ_0	0
$\lambda(k)$	1
h	0.1
k_{on}	54

Table 1: RCAC Parameters.

and let the spacecraft be fully actuated such that $B_{SC} = I_3$. Furthermore, let all disturbances be zero. The RCAC parameters used are shown in Table 1. The parameter k_{on}, which controls the switch $\beta(k)$ in (43) and denotes the number of time steps before the controller turns on, is given by

$$k_{\text{on}} = (l_u + l_y)n_c l_u. \tag{52}$$

The Markov parameter H_1 is computed using the linearization in Section IV evaluated at $\tilde{\omega}_e = 0$ and ω_{d} which depends on the type of maneuver.

We consider angular rate control for both detumbling (M2R-R) and spin (M2S-R) maneuvers. For each maneuver we examine robustness to unknown changes in the inertia by using modified Markov parameters.

A. M2R-R Maneuvers

Let the initial motion of the spacecraft be described by

$$\omega(0) = \frac{0.1}{\sqrt{3}} \left[\begin{array}{ccc} 1 & -1 & 1 \end{array} \right]^{\text{T}} \text{rad/sec.} \tag{53}$$

The goal of the controller is to bring the spacecraft to rest, that is,

$$\omega_d = \left[\begin{array}{ccc} 0 & 0 & 0 \end{array} \right]^{\text{T}} \text{rad/sec.} \tag{54}$$

Given the output matrix $E_1 = I$, the discrete-time dynamics matrix in (48) evaluated at $\tilde{\omega}_e = 0$ and $\omega_d = 0$ so that $A_c = 0$, and the input matrix in (49), the Markov parameter in (50) is

$$H_1 = B = hJ^{-1}B_{SC}. \tag{55}$$

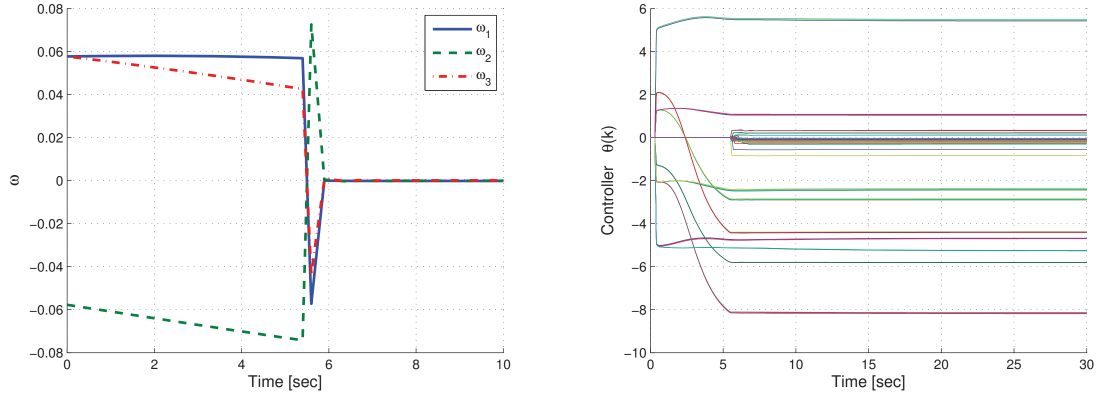
Figure 1 shows the closed-loop performance for this maneuver. The angular rate about each axis is shown in Figure 1a and the controller coefficients $\theta(k)$ are shown in Figure 1b.

B. Effect of the Markov Parameters on Convergence

The Markov parameter in (55) shows that, for M2R-R, we utilize three pieces of information: the mass distribution, the actuator alignments, and the controller time step. We wish to limit the information used by constructing an arbitrary Markov parameter. We define a matrix $\hat{H}_1$ to use in place of the Markov parameter H_1. Then, since that the time step h is a scaling parameter, we can replace it with a positive scalar α and tune the controller. Thus, we begin with

$$\hat{H}_1 = \alpha J^{-1}B_{SC}, \tag{56}$$

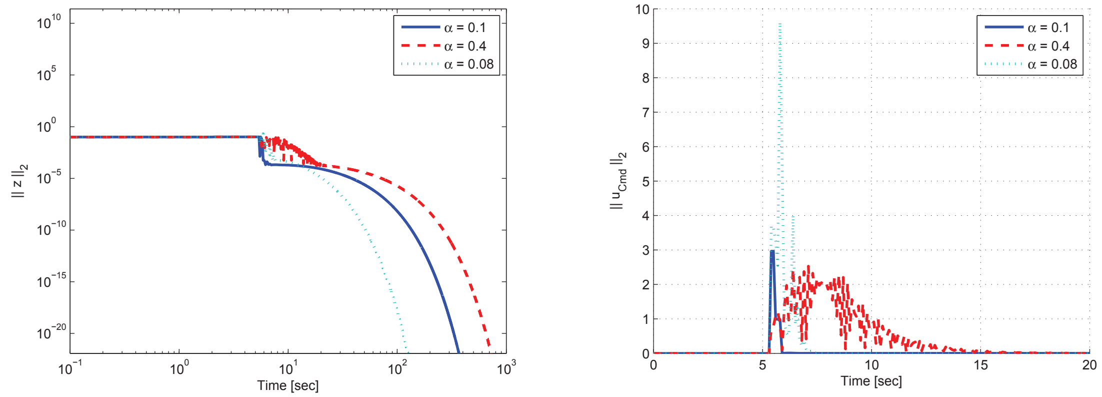
where α is a positive number. We test this choice of $\hat{H}_1$ by varying the parameter α.



(a) Angular Rate $[\mathrm{rad/sec}]$

(b) Controller Coefficients

Figure 1: RCAC closed-loop performance for M2R-R using the Markov parameter H_1 in (55).



(a) Log Plot of Euclidean Norm of Performance

(b) Euclidean Norm of Control Input

Figure 2: Comparison of performance variable and commanded control input for RCAC for the M2R-R maneuver using various values of α in the Markov parameter $\hat{H}_1$ in (56).

Figure 2 shows the results of this analysis. Note that, as α increases, the controller applies less torque to the spacecraft, whereas as α decreases, more torque is applied. However, as we move away from the nominal value of $\alpha = h$ the controller diverges. Figure 3 shows how values of α that are both larger and smaller than the nominal value cause the controller coefficients θ to diverge. Thus, the control inputs also diverge. Note that the smaller value of α in Figures 3b and 3d is close to the working value of $\alpha = 0.08$ in Figure 2 which suggests that RCAC is very sensitive to scaling of H_1.

To overcome this problem we introduce a proportional saturation in the control input by scaling the control vector as

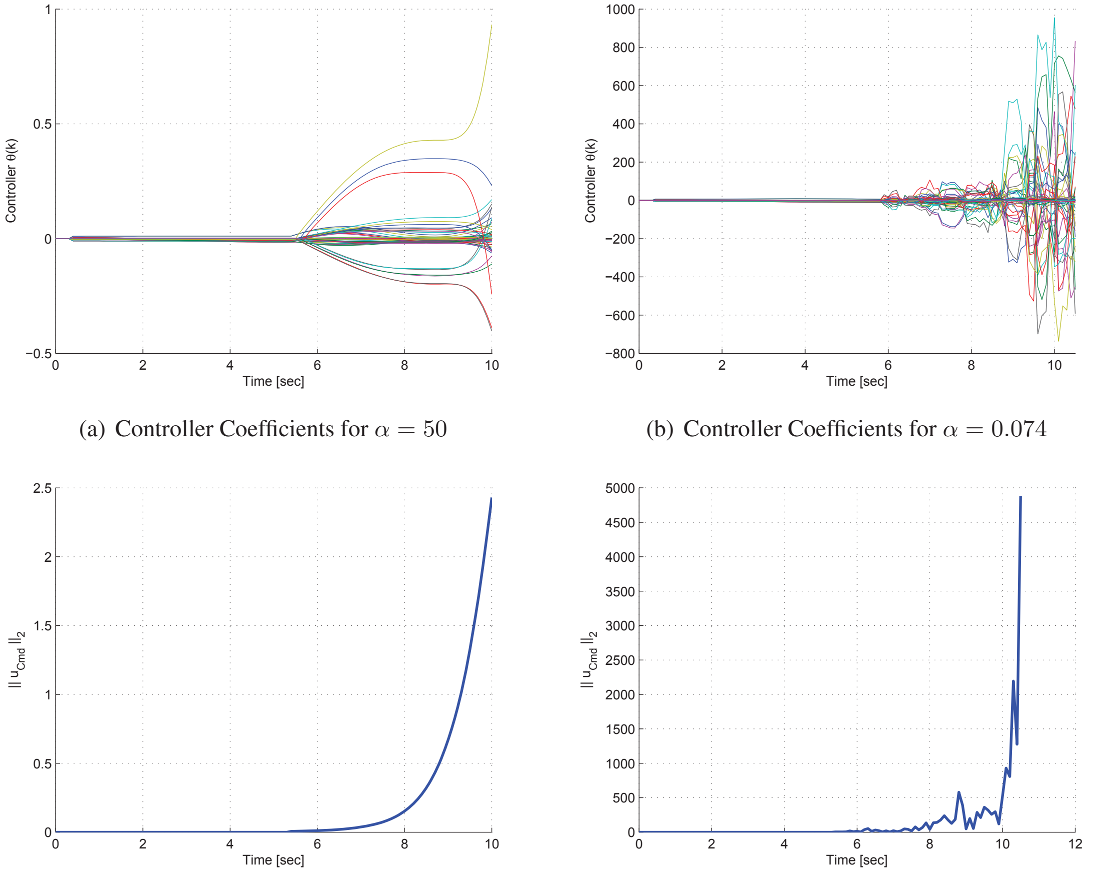
$$u_{\mathrm{sat}} = \begin{cases} u, & u \in \mathcal{B}, \\ \\ \eta u, & u \notin \mathcal{B}, \end{cases} \tag{57}$$

where $\mathcal{B}$ is a boundary defined by the saturation limits and η is the maximum scaling possible such that $u_{sat} \in \mathcal{B}$, that is,

$$\eta = \max_{\eta \in (0,1]} \{\eta : \eta u \in \mathcal{B}\}. \tag{58}$$

Figure 4 shows that this method enables RCAC to bring the system to rest using significantly off-nominal Markov parameters.

Note that varying α corresponds to changing the scale of the inertia, the actuator matrix, or both. Define a scaled inertia $J' = \beta_J J$ and a scaled actuator matrix $B'_{SC} = \beta_B B_{SC}$, where β_J and β_B are positive



(a) Controller Coefficients for $\alpha = 50$

(b) Controller Coefficients for $\alpha = 0.074$

(c) Euclidean Norm of Control Input for $\alpha = 50$

(d) Euclidean Norm of Control Input for $\alpha = 0.074$

Figure 3: Controller divergence for off-nominal values of $\alpha = 50$ and $\alpha = 0.074$ for M2R-R using Markov parameter $\hat{H}_1$ in (56).

numbers. Then, we can set $\alpha = \frac{\beta_B}{\beta_J}$ and obtain

$$\hat{H}_1 = J'^{-1}B'_{SC} = \frac{\beta_B}{\beta_J}J^{-1}B_{SC} = \alpha J^{-1}B_{SC}.$$

Thus, if RCAC is robust to changes in α, it is robust to scaling of the inertia and the actuator matrices. Therefore, saturation has made RCAC robust to scaling errors in the inertia J and the actuator matrix B_{SC} for M2R-R maneuvers. Furthermore, for a given saturation level, α can also be used as a tuning parameter for control authority.

With the saturation method added to the controller, we eliminate the inertia information used in $\hat{H}_1$ by removing J from (56) and obtain

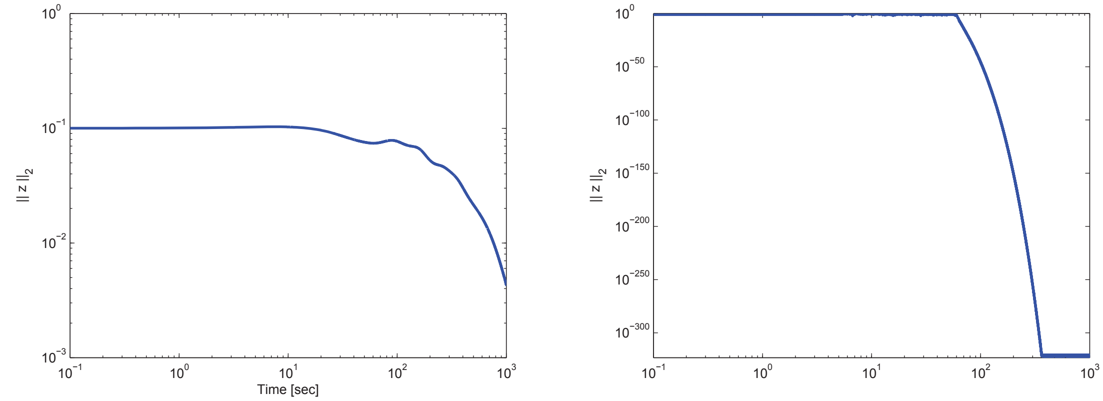
$$\hat{H}_1 = \alpha B_{SC}. \tag{59}$$

Removing the inertia from $\hat{H}_1$ hides information about the spacecraft axes coupling from RCAC. Figure 5 shows that RCAC completes the M2R-R maneuver considered in Figure 1 for various values of α. Notice that, as expected, decreasing α increases the control input.

C. M2S-R Maneuvers

To expand on the M2R-R example, we command the spacecraft to spin about an arbitrary body axis and examine the M2S-R problem. The initial angular rate is described by (53), and the controller goal is to make the spacecraft spin at the desired angular rate of

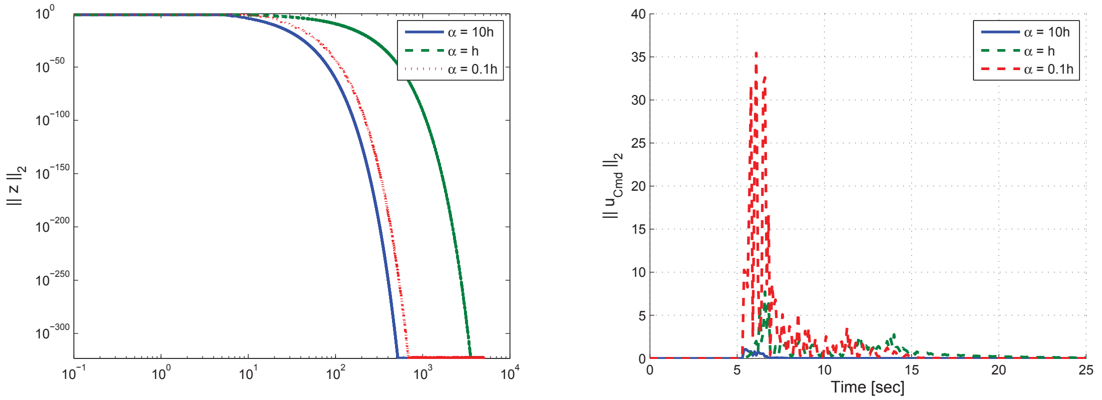
$$\omega_d = \frac{0.1}{\sqrt{3}} \begin{bmatrix} -1 & 1 & 1 \end{bmatrix}^{\mathrm{T}} \text{rad/sec}. \tag{60}$$



(a) $\alpha = 1000$, Saturation 0.001 N-m

(b) $\alpha = 0.001$, Saturation 10 N-m

Figure 4: Log plots of the Euclidean norm of performance, $\|z\|_2$, for M2R-R for off-nominal values of $\alpha = 1000$ and $\alpha = 0.001$ in the Markov parameter $\hat{H}_1$ in (56) with a saturation level of 1 N-m.



(a) Euclidean Norm of Performance

(b) Euclidean Norm of Unsaturated Control Input

Figure 5: Inertia-free RCAC performance for various values of α for M2R-R. Euclidean norm of performance, $\|z\|_2$ and unsaturated controller input, $\|u_{\mathrm{cmd}}\|$ using the Markov parameter $\hat{H}_1$ in (59). The saturation level is set to 1 N-m and $h = 0.1$ sec.

The spacecraft has the inertia matrix in (51) with the initial angular rate in (53). The RCAC parameters are chosen as in Table 1. Figure 6 compares the performance for the maneuver using the Markov parameters in (55) and (59). For both Markov parameter choices, RCAC completes the maneuver. However, the lack of information increases the settling time for the case using the inertia-free Markov parameter.

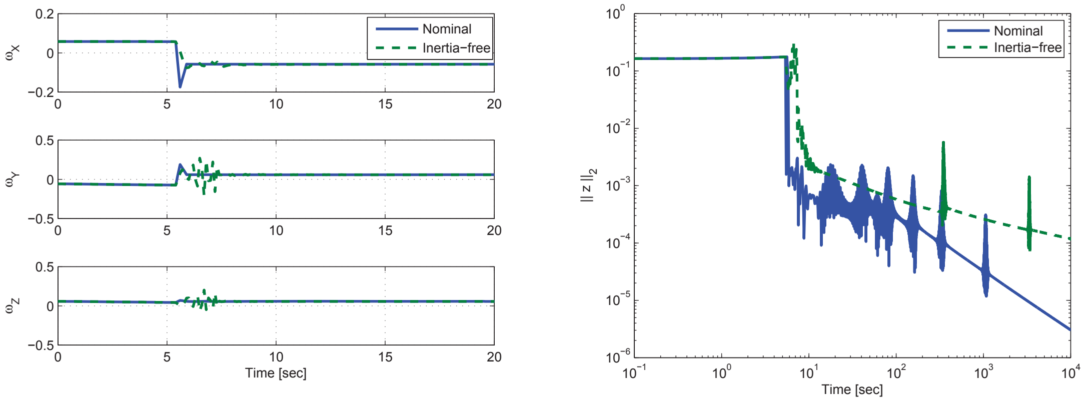
VI. RCAC Parameters for Attitude Control

We extend the results from Section V and include the attitude kinematics. Since RCAC requires a vector performance, the rotation matrix governed by Poisson's equation (2), cannot be used directly.

Thus, we formulate the attitude error dynamics by using the vector function of the attitude error matrix presented in [2]. For $i = 1, 2, 3$, let e_i denote the ith column of the 3×3 identity matrix and let $A_{\mathrm{att}} = \mathrm{diag}(a_1, a_2, a_3)$ be a diagonal positive-definite matrix, then

$$z_a \triangleq \sum_{1}^{3} a_i \left(\tilde{R}^{\mathrm{T}} e_i\right) \times e_i, \tag{61}$$

is a 3×1 vector measure of attitude error. Note that $z_a = 0$ when $\tilde{R} = I_3$. Thus, we use z_a as the attitude performance variable. The attitude error affects the rate error given as shown in (9). We redefine the angular



(a) Angular Rate $[\mathrm{rad/sec}]$

(b) Log Plot of Euclidean Norm of Performance

Figure 6: Angular rate, ω, and Euclidean norm of performance, $\|z\|_2$ for M2S-R. Plots compare convergence for RCAC using Markov parameters derived from the linearized dynamics in (55) with RCAC using inertia-free Markov parameters in (59) with a saturation level of 10 N-m and $\alpha = 0.1$.

rate performance as

$$z_\omega \overset{\triangle}{=} \omega - \tilde{R}^{\mathrm{T}}\omega_d. \tag{62}$$

The combined performance variable for the attitude control problem is given by

$$z = \begin{bmatrix} z_\omega \\ z_a \end{bmatrix}. \tag{63}$$

A. Markov Parameters

The Markov parameters required by RCAC serve as a mapping between the control input and the performance variable. To obtain the Markov parameters, we represent the SO(3) attitude kinematics as vector equations. Thus, we parameterize the SO(3) attitude as a vector composed by the rows of the rotation matrix. For the attitude error matrix $\tilde{R}$, this parameterization is

$$\tilde{R} = \begin{bmatrix} \tilde{r}_1 \\ \tilde{r}_2 \\ \tilde{r}_3 \end{bmatrix}, \tag{64}$$

where, for $i = 1, 2, 3$, $\tilde{r}_i \in \mathbb{R}^{1\times 3}$ is a row of the rotation matrix. We define the new attitude parameter

$$\tilde{r} = \begin{bmatrix} \tilde{r}_1 & \tilde{r}_2 & \tilde{r}_3 \end{bmatrix}^{\mathrm{T}}, \tag{65}$$

we express the performance variable, the dynamics in (10), and the kinematics in (8) in terms of $\tilde{r}$.

We rewrite the angular rate performance in (62) as

$$z_\omega = \omega - \mathcal{D}(\omega_d)\tilde{r}. \tag{66}$$

Where the operator $\mathcal{D}(x)$ for $x \in \mathbb{R}^3$ is

$$\mathcal{D}(x) \overset{\triangle}{=} \begin{bmatrix} x_1 I_3 & x_2 I_3 & x_3 I_3 \end{bmatrix}. \tag{67}$$

where $x_i,\ i = 1, 2, 3$, is the ith component of x. Then, we express the attitude performance z_a in terms of the new attitude parameter as

$$z_a = -M_a\tilde{r}, \tag{68}$$

where I_3 is the 3×3 identity matrix and

$$M_a \triangleq \begin{bmatrix} e_1^\times & e_2^\times & e_3^\times \end{bmatrix} \begin{bmatrix} a_1 I_3 & & \\ & a_2 I_3 & \\ & & a_3 I_3 \end{bmatrix}. \tag{69}$$

The attitude error dynamics in (7) become

$$\dot{\tilde{r}} = \begin{bmatrix} -\tilde{\omega}^\times & & \\ & -\tilde{\omega}^\times & \\ & & -\tilde{\omega}^\times \end{bmatrix} \tilde{r} \tag{70}$$

and Euler's equation in (10) is expressed as

$$\dot{\tilde{\omega}} = J^{-1} \left[\left[J \left(\tilde{\omega} + \mathcal{D}(\omega_d) r \right) \right] \times \left(\tilde{\omega} + \mathcal{D}(\omega_d) r \right) \right] + \tilde{\omega} \times \left[\mathcal{D}(\omega_d) r \right] - \mathcal{D}(\dot{\omega}_d) r + J^{-1} \left(Bu + z_d \right). \tag{71}$$

To obtain the Jacobian we stack the angular rate error $\tilde{\omega}$ and the attitude-error parameter $\tilde{r}$ to form a state vector. Next, we differentiate (70) and (71) with respect to this new state and evaluate it at a given equilibrium $\begin{bmatrix} \tilde{\omega}_e \\ \tilde{r}_e \end{bmatrix}$. For the rotational dynamics in (71) the derivative with respect to $\tilde{\omega}$ is

$$\frac{\partial \dot{\tilde{\omega}}}{\partial \tilde{\omega}} = J^{-1} \left[- \left(\tilde{\omega}_e + \mathcal{D}(\omega_d) \tilde{r}_e \right)^\times J + \left[J \left(\tilde{\omega}_e + \mathcal{D}(\omega_d) \tilde{r}_e \right) \right]^\times \right] - \left(\mathcal{D}(\omega_d) \tilde{r}_e \right)^\times. \tag{72}$$

Similarly, the derivative with respect to $\tilde{r}$ is

$$\frac{\partial \dot{\tilde{\omega}}}{\partial \tilde{r}} = J^{-1} \left[- \left(\tilde{\omega}_e + \mathcal{D}(\omega_d) \tilde{r}_e \right)^\times J \mathcal{D}(\omega_d) + \left[J \left(\tilde{\omega}_e + \mathcal{D}(\omega_d) \tilde{r}_e \right) \right]^\times \mathcal{D}(\omega_d) \right] + \tilde{\omega}_e^\times \mathcal{D}(\omega_d) - \mathcal{D}(\dot{\omega}_d). \tag{73}$$

Thus, we construct the Jacobian for the angular rate dynamics

$$A_\omega = \begin{bmatrix} \frac{\partial \dot{\tilde{\omega}}}{\partial \tilde{\omega}} & \frac{\partial \dot{\tilde{\omega}}}{\partial \tilde{r}} \end{bmatrix}. \tag{74}$$

For the attitude kinematics, we partition (70) for each row $\tilde{r}_i$ of $\tilde{R}$ as

$$\dot{\tilde{r}}_i = \left(\tilde{r}_i \times \tilde{\omega} \right)^{\mathrm{T}}. \tag{75}$$

Then, the derivative with respect to $\tilde{\omega}$ is

$$\frac{\partial \dot{\tilde{r}}_i}{\partial \tilde{\omega}} = -\tilde{r}_{e,i}^\times, \text{ for } i = 1, 2, 3. \tag{76}$$

Differentiating (75) with respect to $\tilde{r}_j$ yields

$$\frac{\partial \dot{\tilde{r}}_i}{\partial \tilde{r}_j} = \left[-\tilde{\omega}^\times \frac{\partial \tilde{r}_i}{\partial \tilde{r}_j} + \tilde{r}_i^\times \frac{\partial \tilde{\omega}}{\partial \tilde{r}_j} \right]^{\mathrm{T}}, \tag{77}$$

where

$$\frac{\partial \tilde{r}_i}{\partial \tilde{r}_j} = \begin{cases} I_3, & i = j, \\ 0_{3\times3}, & i \neq j, \end{cases} \tag{78}$$

and

$$\frac{\partial \tilde{\omega}}{\partial \tilde{r}_j} = \begin{cases} \mathcal{D}(\omega_d) \begin{bmatrix} I_3 & 0 & 0 \end{bmatrix}^{\mathrm{T}}, & j = 1, \\ \mathcal{D}(\omega_d) \begin{bmatrix} 0 & I_3 & 0 \end{bmatrix}^{\mathrm{T}}, & j = 2, \\ \mathcal{D}(\omega_d) \begin{bmatrix} 0 & 0 & I_3 \end{bmatrix}^{\mathrm{T}}, & j = 3. \end{cases} \tag{79}$$

Finally, we obtain the Jacobian for the attitude kinematics

$$A_a = \begin{bmatrix} \frac{\partial \dot{\tilde{r}}_1}{\partial \tilde{\omega}} & \frac{\partial \dot{\tilde{r}}_2}{\partial \tilde{\omega}} & \frac{\partial \dot{\tilde{r}}_3}{\partial \tilde{\omega}} \\ \frac{\partial \dot{\tilde{r}}_1}{\partial \tilde{r}_1} & \frac{\partial \dot{\tilde{r}}_2}{\partial \tilde{r}_1} & \frac{\partial \dot{\tilde{r}}_3}{\partial \tilde{r}_1} \\ \frac{\partial \dot{\tilde{r}}_1}{\partial \tilde{r}_2} & \frac{\partial \dot{\tilde{r}}_2}{\partial \tilde{r}_2} & \frac{\partial \dot{\tilde{r}}_3}{\partial \tilde{r}_2} \\ \frac{\partial \dot{\tilde{r}}_1}{\partial \tilde{r}_3} & \frac{\partial \dot{\tilde{r}}_2}{\partial \tilde{r}_3} & \frac{\partial \dot{\tilde{r}}_3}{\partial \tilde{r}_3} \end{bmatrix}^{\mathrm{T}}. \tag{80}$$

Thus, the dynamics matrix for the linearized continuous-time system is

$$A_{\mathrm{c}}(\tilde{\omega}_e, \tilde{r}_e, \omega_d) = \begin{bmatrix} A_\omega \\ A_a \end{bmatrix}, \tag{81}$$

with the input matrix

$$B_{\mathrm{c}} = \begin{bmatrix} J^{-1} B_{SC} \\ 0_{9\times 3} \end{bmatrix}, \tag{82}$$

and the output matrices

$$C = \begin{bmatrix} I_3 & 0_{3\times 9} \\ 0_{3\times 3} & M_a \end{bmatrix} \tag{83}$$

and

$$D = 0_{6\times 3}. \tag{84}$$

This continuous-time system is discretized as in (48) and (49) with $E_1 = C$ and $D_d = D$. We obtain the Markov parameters using (14).

VII. Numerical Examples for Attitude Control

Let the spacecraft inertia matrix be defined as in (51), $B_{SC} = I_3$, and $A_{\mathrm{att}} = I$. We use the RCAC parameters in Table 1 and let the parameter $k_{\mathrm{on}} = 81$ as given by (52). The Markov parameter is given by the linearized system in Section VI evaluated at $\tilde{R}_e = I_3$ and $\tilde{\omega}_e = 0$, ω_d depends on the maneuver.

We examine slew (M2R-A) and spin (M2S-A) maneuvers. For each maneuver we examine robustness to changes in the inertia using arbitrary Markov parameters.

A. M2R-A Maneuvers

Let the initial motion of the spacecraft be described by (53) and the initial attitude be given by

$$R(0) = I_3. \tag{85}$$

The goal of the controller is to bring the spacecraft to rest, that is $\omega_d = 0$, with the inertial attitude

$$R_d = \begin{bmatrix} 0.5000 & 0.5000 & 0.7071 \\ 0.5000 & 0.5000 & -0.7071 \\ -0.7071 & 0.7071 & 0.0000 \end{bmatrix}. \tag{86}$$

For the M2R-A problem, the Markov parameter given by (14) becomes

$$H_1 = \begin{bmatrix} I_3 & 0_{3\times 9} \\ 0_{3\times 3} & M_a \end{bmatrix} \int_0^h e^{A_c \tau} \begin{bmatrix} J^{-1} B_{SC} \\ 0_{9\times 3} \end{bmatrix} d\tau, \tag{87}$$

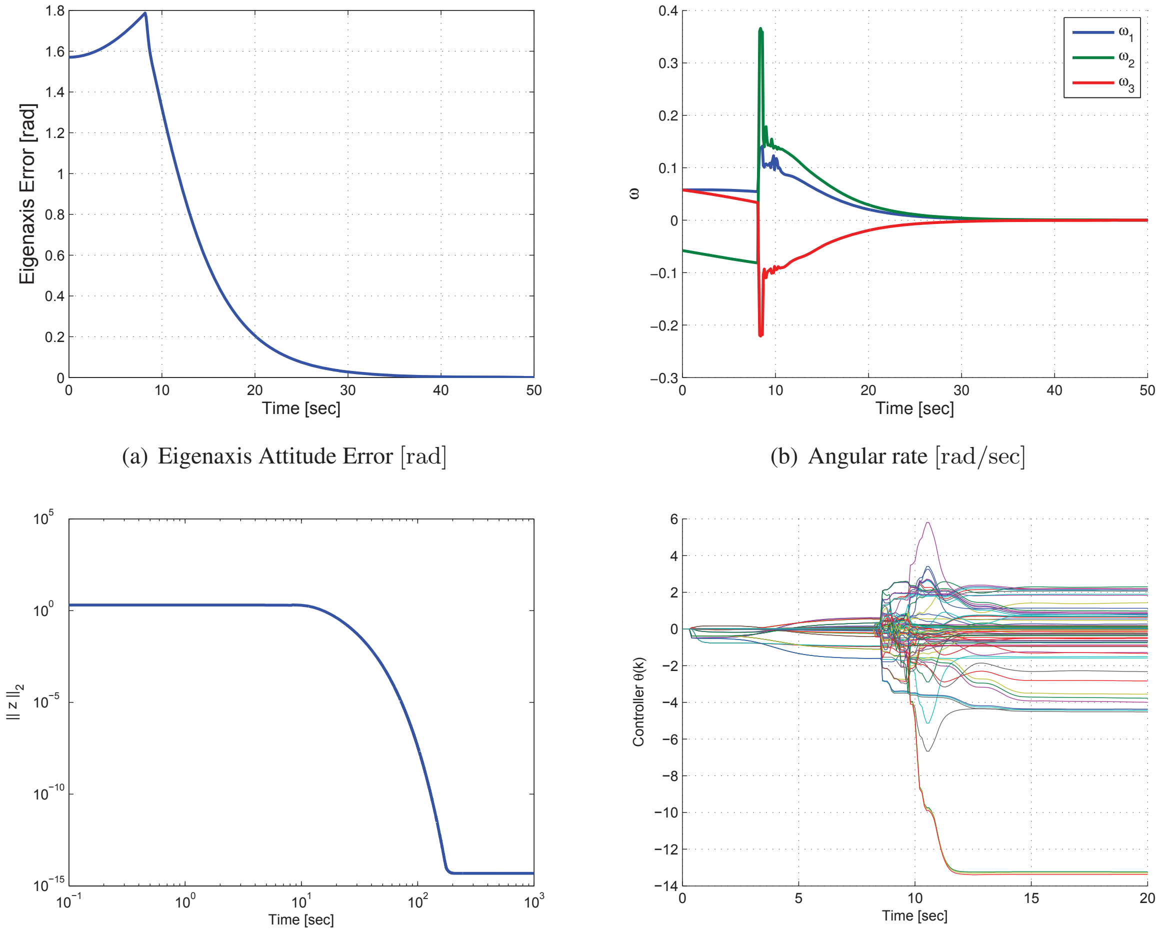
where the matrix A_c is given by (81) evaluated at $\tilde{\omega} = \omega_d = 0$ and $\tilde{R} = I_3$. We compute the matrix exponential

$$e^{A_c\tau} = \begin{bmatrix} I_3 & 0_{3\times 9} \\ \tau e_1^\times & \\ \tau e_2^\times & I_9 \\ \tau e_3^\times & \end{bmatrix} \tag{88}$$

and evaluate the integral term in (87). Thus, the M2R-A Markov parameter is

$$H_1 = \begin{bmatrix} hJ^{-1}B_{SC} \\ h^2 J^{-1}B_{SC} \end{bmatrix}. \tag{89}$$

Figure 7 shows the closed-loop performance for the M2R-A maneuver using the Markov parameter H_1 in (89). Note that the linear controller coefficients converge smoothly and quickly.



(a) Eigenaxis Attitude Error $[\mathrm{rad}]$

(b) Angular rate $[\mathrm{rad/sec}]$

(c) Euclidean Norm of Performance

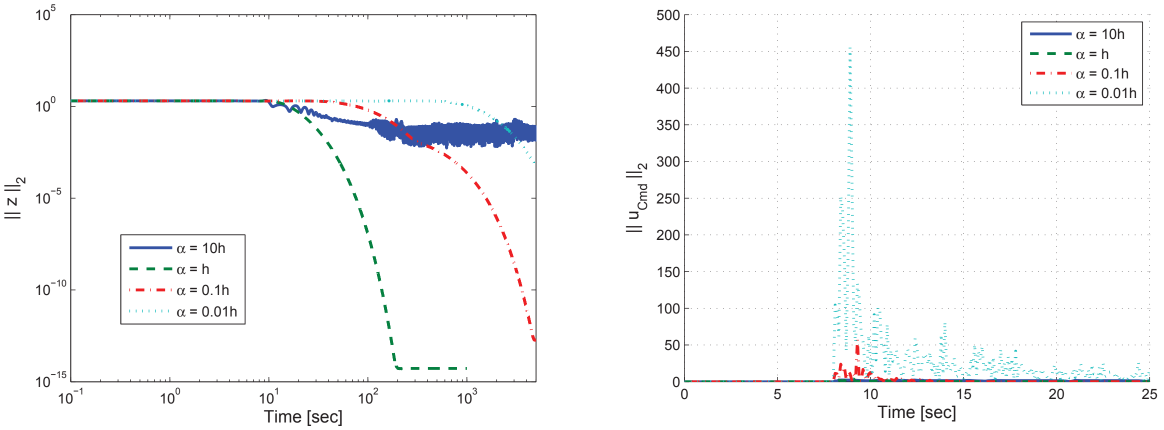
(d) Controller Coefficients

Figure 7: RCAC performance for M2R-A using Markov parameter H_1 in (89).

B. Effect of the Markov Parameters on Convergence

We use H_1 in (89) to define the arbitrary Markov parameter

$$\hat{H}_1 = \begin{bmatrix} \alpha B_{SC} \\ \alpha^2 B_{SC} \end{bmatrix}, \tag{90}$$



(a) Euclidean Norm of Performance

(b) Euclidean Norm of Unsaturated Control Input

Figure 8: Inertia-free RCAC performance for M2R-A using Markov parameter $\hat{H}_1$ in (90) with $h = 0.1$ sec and a saturation level of 1 N-m.

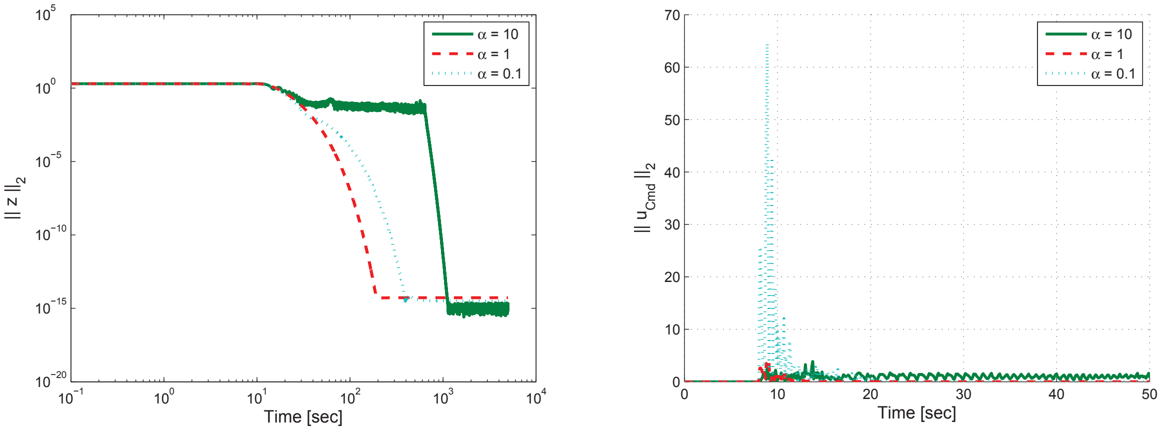
where we have removed the inertia matrix and replaced the time step h with a positive scaling parameter α. As in the M2R-R case, off-nominal values of α cause the controller to diverge. Thus, we implement the saturation method in (57) to mitigate this problem.

Figure 8 shows the controller's performance using various values of α. Note that, unlike the M2R-R case, saturation does not provide robustness to changes of α outside of the nominal value $\alpha = h$, especially for values greater than 1. This is due to the nonlinear term, α^2 in the lower portion of the matrix $\hat{H}_1$ in (90). When $\alpha \neq h$ the ratios between the entries in the top and bottom halves of the matrix $\hat{H}_1$ change from their nominal values, thus affecting the internal structure of the Markov parameter. Thus, we redefine the Markov parameter as

$$\hat{H}_1' = \alpha \left[\begin{array}{c} hB_{SC} \\ h^2 B_{SC} \end{array} \right]. \tag{91}$$

This scaling maintains the matrix element ratios at their nominal values.

The M2R-A maneuver in Figure 9 examines the effect of the Markov parameter $\hat{H}_1'$ given by (91). The trajectory for $\alpha = 1$ corresponds to the unscaled inertia-free Markov parameter. As shown by the performance variable plot in Figure 9a, the algorithm is now robust to changes in the inertia and to scaling of the input matrix B_{SC}. As expected, decreasing α increases the commanded control input.



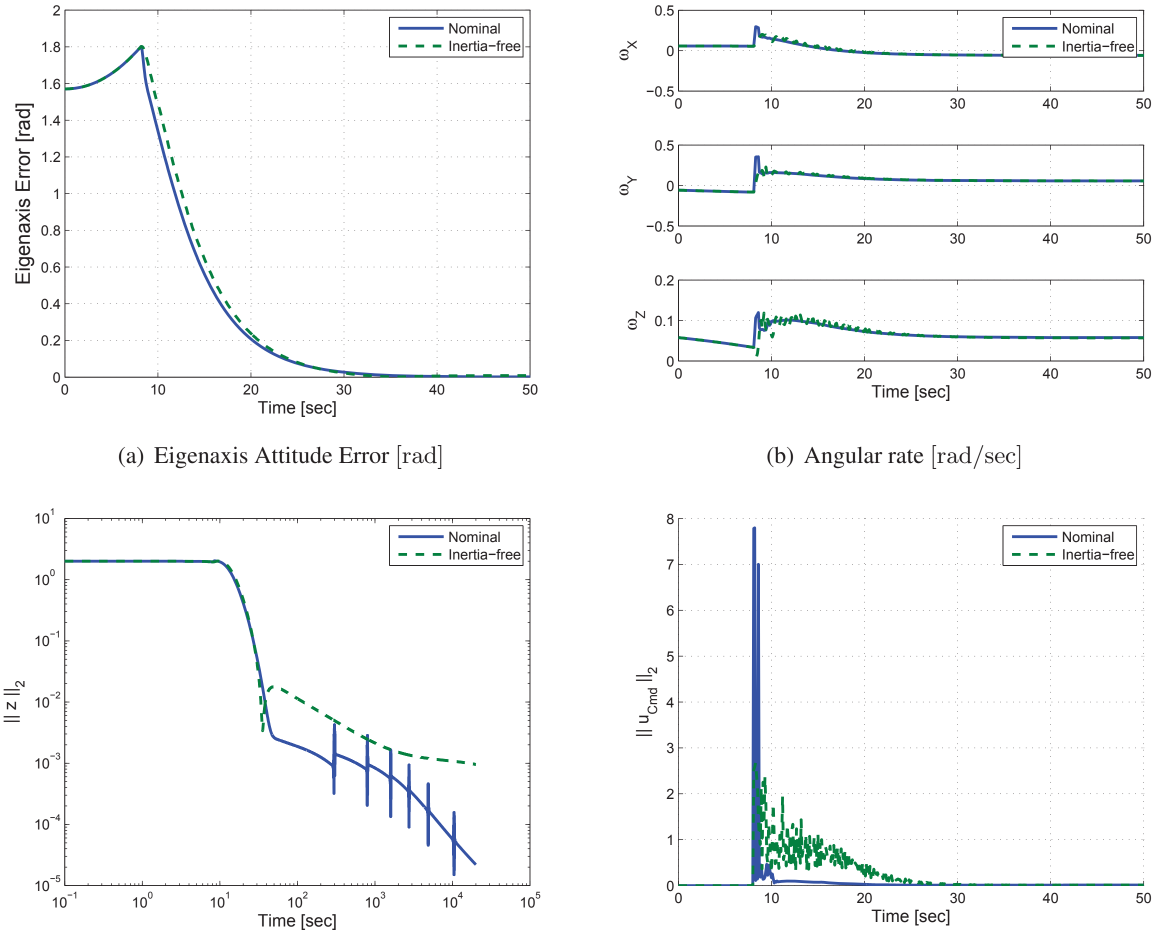
(a) Euclidean Norm of Performance

(b) Euclidean Norm of Unsaturated Control Input

Figure 9: Inertia-free RCAC performance for M2R-A using the inertia-free Markov parameter $\hat{H}_1'$ in (91) with $h = 0.1$ sec. The controller is robust to scaling of H_1.

C. M2S-A Maneuvers

To expand the M2R-A example we command the spacecraft to spin about a specified body axis aligned with a specific inertial attitude. Let the initial state of the spacecraft be as in the examples in the previous section, set the controller parameters as in Table 1, and let the desired angular rate ω_d be as described in (60). Since the commanded angular rate is non-zero the desired attitude evolves over time according to (6). Figure 10 compares the performance of RCAC using the nominal Markov parameter in (87) with RCAC using the inertia-free Markov parameter (91) with saturation for the M2S-A maneuver.



(a) Eigenaxis Attitude Error [rad]

(b) Angular rate [rad/sec]

(c) Euclidean Norm of Performance

(d) Euclidean Norm of Unsaturated Control Input

Figure 10: Comparison of RCAC performance for M2S-A. Plots compare RCAC using Markov parameter H_1 in (87) without saturation versus RCAC using the inertia-free Markov parameter $\hat{H}_1'$ in (91) with $\alpha = 1$, $h = 0.1$ sec, and a saturation limit of 1 N-m.

VIII. Robustness to Actuator Misalignment for M2R-A

We examine the robustness of RCAC for the M2R-A maneuver to actuator misalignment. We misalign each actuator in different directions by varying angles. Thus, for a spacecraft with three actuators, the actuator matrix is given by

$$B_{SC} = \left[\begin{array}{ccc} R_1 e_1 & R_2 e_2 & R_3 e_3 \end{array} \right]. \tag{92}$$

Each $R_i \in \mathbb{R}^{3\times 3}$ for $i = 1, 2, 3$ in (92) is a rotation matrix given by Rodrigues' equation

$$\mathcal{R}(\theta_B, \hat{n}_B) = \cos(\theta_B) I_3 + (1 - \cos(\theta_B)) n_B * n_B^{\mathrm{T}} + \sin(\theta_B) n_B^{\times}, \tag{93}$$

where θ_B is the misalignment angle and $\hat{n}_B \in \mathbb{R}^3$ is the misalignment axis.

A. Numerical Examples

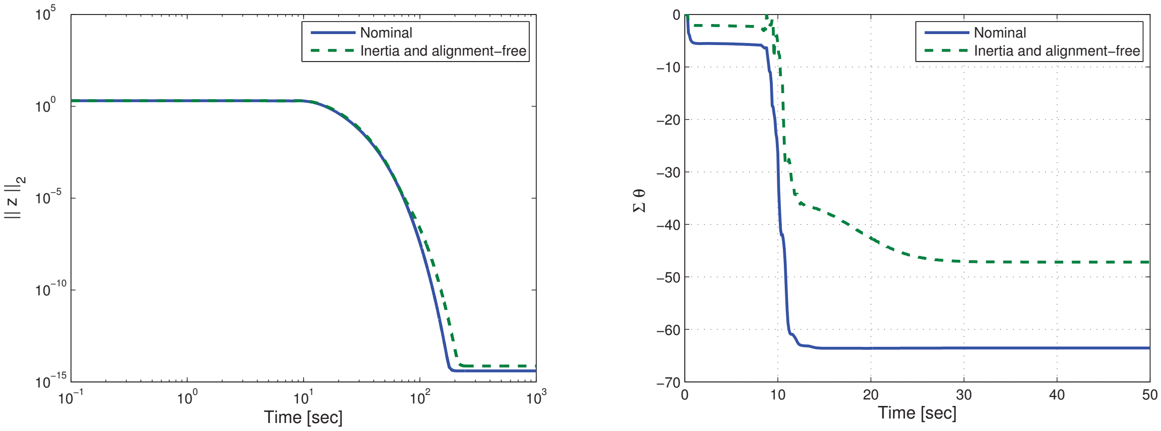
The initial conditions and RCAC parameters used are given in Section VII. First, we test a misalignment of $\theta_B = 30^\circ$ on each actuator with different misalignment axes so that the rotation matrices in (92) are

$$R_1 = \mathcal{R}(\theta_B, e_3), \tag{94}$$
$$R_2 = \mathcal{R}(\theta_B, e_1), \tag{95}$$
$$R_3 = \mathcal{R}(\theta_B, e_2). \tag{96}$$

where e_i for $i = 1, 2, 3$ is the ith column of I_3. Figure 11 compares the closed-loop performance using RCAC with full inertia and actuator knowledge given by the Markov parameter H_1 in (87) with RCAC using the inertia and misalignment-free Markov parameter

$$\hat{H}_1 = \alpha \left[\begin{array}{c} h\hat{B}_{SC} \\ h^2\hat{B}_{SC} \end{array} \right], \tag{97}$$

where $\hat{B}_{SC} = I_3$ is the ideal actuator matrix.



(a) Euclidean Norm of Performance

(b) Sum of Controller Coefficients

Figure 11: Comparison of RCAC performance for M2R-A with $\theta_B = 30^\circ$. Plots compare RCAC using Markov parameter H_1 in (87) without saturation versus RCAC using the inertia and alignment-free Markov parameter $\hat{H}_1$ in (97) with $\alpha = 1$, $h = 0.1$ sec, and a saturation limit of 1 N-m.

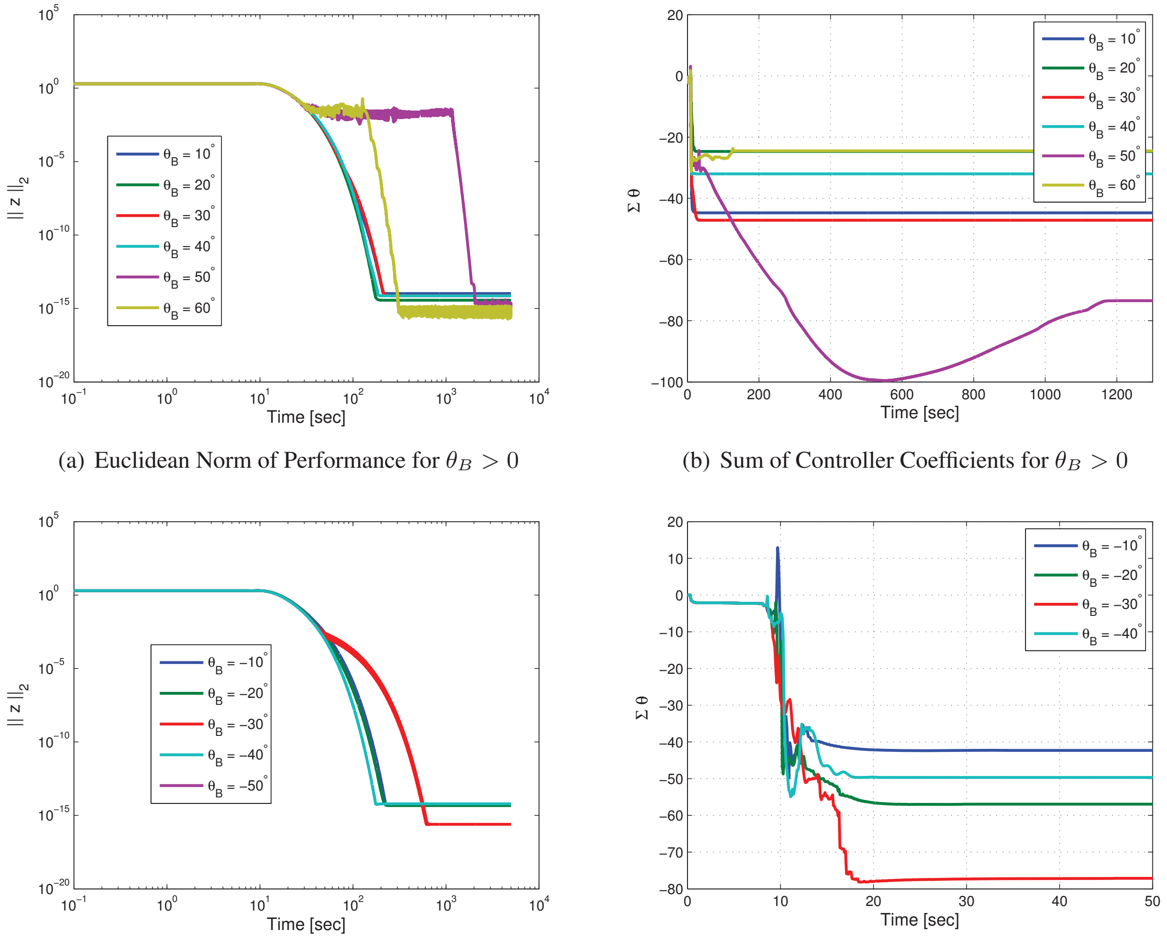
RCAC can complete the M2R-A maneuver without using knowledge of the misalignments in the actuator matrix B_{SC}. As shown in Figure 11b the lack of alignment information results in a longer settling time. To evaluate robustness, we test RCAC with larger misalignment angles.

Figure 12 shows the closed-loop performance when using the Markov parameter $\hat{H}_1$ in (97) for different misalignment angles. We increase the misalignment angle until the controller fails to complete the M2R-A maneuver. As shown in Figure 13, for angles greater than 60° or less than -40°, the controller coefficients converge but the system settles into a limit cycle.

IX. Conclusions and Future Research

The RCAC algorithm was used to control spacecraft angular rate and attitude using Markov parameters derived from the linearized Euler and Poisson equations. Numerical simulations indicate that the inertia information can be removed from the Markov parameter to obtain inertia-free attitude control. We also showed robustness to inertia and actuator scaling and achieve the shortest settling time when the controller time step h is known. Thus, RCAC can be used as an inertia-free angular rate and attitude controller for detumbling (M2R-R), slews (M2R-A), and spinning maneuvers (M2S-R and M2S-A).

Robustness to actuator misalignments was also examined. Numerical simulations of the M2R-A maneuver indicate that RCAC can handle misalignments of various angles about different axes. The simulation



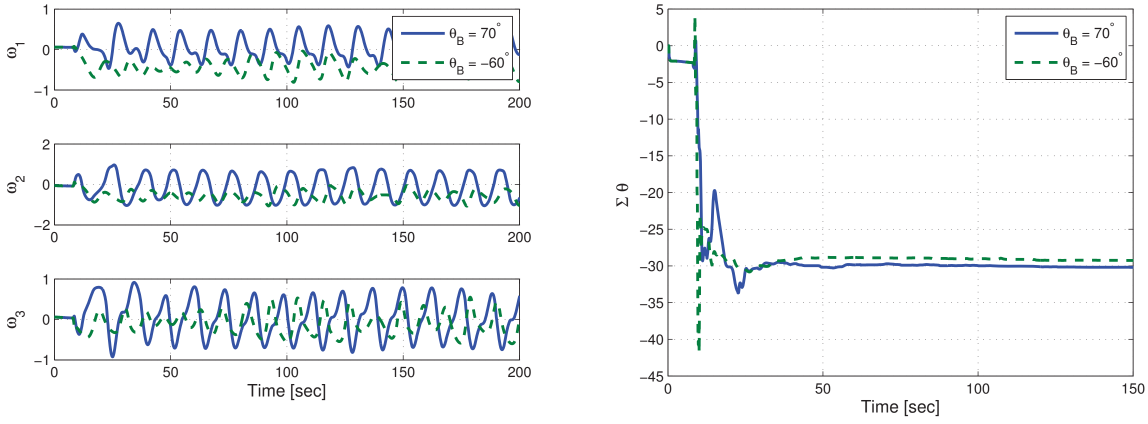
(a) Euclidean Norm of Performance for $\theta_B > 0$

(b) Sum of Controller Coefficients for $\theta_B > 0$

(c) Euclidean Norm of Performance for $\theta_B < 0$

(d) Sum of Controller Coefficients for $\theta_B < 0$

Figure 12: Comparison of RCAC performance for M2R-A for various misalignment angles θ_B. Plots compare the closed-loop performance using the inertia and alignment-free Markov parameter $\hat{H}_1$ in (97) with $\alpha = 1$, $h = 0.1$ sec, and a saturation limit of 1 N-m.



(a) Angula-rate

(b) Sum of Controller Coefficients

Figure 13: RCAC performance for M2R-A for misalignment angles $\theta_B = 70^\circ$ and $\theta_B = -60^\circ$ using the inertia and alignment-free Markov parameter $\hat{H}_1$ in (97) with $\alpha = 1$ and a saturation limit of 1 N-m. Although the controller coefficients converge, the system enters a limit cycle as shown by the angular rate plot.

results indicate that RCAC is robust to misalignments up to 50° in all directions. Larger misalignment errors result in RCAC entering a limit cycle due to the incorrect input-output information provided by the Markov parameter.

To fully understand the RCAC spacecraft system, the effect of the individual entries of the Markov parameter on controller behavior must be determined. Thus, further work will examine the relation between maneuver type, Markov parameter, and controller performance. Also, different spacecraft inertias and actuator placements, specifically underactuated spacecraft, will be considered.

Future applications will extend RCAC to spacecraft systems with changing inertia properties such as actuated solar arrays. Also of interest are spacecraft with flexible bodies such as gravity gradient booms and deployable antennas. Alternative actuators, such as reaction wheels, magnetic torquers, control moment gyros, and combinations of these, will be explored. RCAC can also address disturbance rejection and sensor noise problems.

References

[1] Hughes, P. C., *Spacecraft Attitude Dynamics*, Wiley, 1986; reprinted by Dover, 2008.

[2] Sanyal, A., Fosbury, A., Chaturvedi, N., and Bernstein, D. S., "Inertia-Free Spacecraft Attitude Tracking with Disturbance Rejection and Almost Global Stabilization," *AIAA J. Guid. Contr. Dyn.*, Vol. 32, pp. 1167-1178, 2009.

[3] Ahmed, J., Coppola, V. T. , and Bernstein, D. S., "Asymptotic Tracking of Spacecraft Attitude Motion with Inertia Matrix Identification," *AIAA J. Guid. Contr. Dyn.*, Vol. 21, pp. 684-691, 1998.

[4] Chaturvedi, N. A., Bernstein, D. S., Ahmed, J. ,Bacconi, F., and McClamroch, N. H., "Globally Convergent Adaptive Tracking of Angular Velocity and Inertia Identification of a 3-DOF Rigid Body," *IEEE Transactions on Control Systems Technology*, Vol. 14, No. 5, 2006, pp. 841-853.

[5] Egeland, O. and Godhavn, J.M., "Passivity-based Adaptive Attitude Control of a Rigid Spacecraft," *IEEE Transactions on Automatic Control*, Vol. 39, No. 4, 1994, pp. 842-846.

[6] Luo, W., Chu, Y., and Ling, K., "Inverse Optimal Adaptive Control for Attitude Tracking of Spacecraft ," *IEEE Transactions on Automatic Control*, Vol. 50, No. 11, 2005, pp. 1639-1654.

[7] Slotine, J.E. and Di Benedetto, M.D., "Hamiltonian Adaptive Control of Spacecraft," *IEEE Transactions on Automatic Control*, Vol. 35, No. 7, 1990, pp. 848-852.

[8] Singh, S.N. and Yim, W., "Nonlinear Adaptive Backstepping Design for Spacecraft Attitude Control Using Solar Radiation Pressure," *Proc. 41st IEEE Conf. on Decision and Control, 2002* , Vol. 2, pp. 1239-1244, December 2002.

[9] Venugopal, R. and Bernstein, D. S. , "Adaptive Disturbance Rejection Using ARMARKOV System Representations," *IEEE Trans. Contr. Sys. Tech.*, Vol. 8, pp. 257-269, 2000.

[10] Hoagg, J. B., Santillo, M. A., and Bernstein, D. S., "Discrete-Time Adaptive Command Following and Disturbance Rejection with Unknown Exogenous Dynamics," *IEEE Trans. Autom. Contr.*, Vol. 53, pp. 912-928, 2008.

[11] Santillo, M. A. and Bernstein, D. S., "Adaptive Control Based on Retrospective Cost Optimization," *AIAA J. Guid. Contr. Dyn.*, Vol. 33, pp. 289–304, 2010.

[12] Sumer, E. D., D'Amato, A. M., Morozov, A. V., Hoagg, J. B. , and Bernstein, D. S., "Robustness of Retrospective Cost Adaptive Control to Markov-Parameter Uncertainty," *Proc. Conf. Dec. Contr.*, pp. 6085–6090, Orlando, FL, December 2011.

[13] Hoagg, J. B. and Bernstein, D. S., "Retrospective Cost Model Reference Adaptive Control for Nonminimum-Phase Discrete-Time Systems, Part 1: The Adaptive Controller; Part 2: Stability Analysis," *Proc. Amer. Contr. Conf.*, pp. 2927–2938, San Francisco, CA, June 2011.

[14] D'Amato, A. M., Sumer, E. D., and Bernstein, D. S., "Retrospective Cost Adaptive Control for Systems with Unknown Nonminimum-Phase Zeros," AIAA Guid. Nav. Contr. Conf., Portland, OR, August 2011, AIAA-2011-6203.

[15] D'Amato, A. M., Sumer, E. D., and Bernstein, D. S.,"Frequency-Domain Stability Analysis of Retrospective-Cost Adaptive Control for Systems with Unknown Nonminimum-Phase Zeros," *Proc. Conf. Dec. Contr.*, pp. 1098–1103, Orlando, FL, December 2011.

[16] Morozov, A.V., Hoagg, J. B., and Bernstein, D. S., "Retrospective Adaptive Control of a Planar Multilink Arm with Nonminimum-Phase Zeros," *Proc. Conf. Dec. Contr.*, pp. 3706–3711, Atlanta, GA, December 2010.

[17] Isaacs, M. W., Hoagg, J. B., Morozov, A., and Bernstein, D. S. , "A Numerical Study on Controlling a Nonlinear Multilink Ar Using a Retrospective Cost Model Reference Adaptive Controller," *Proc. Conf. Dec. Contr.*, pp. 8008–8013, Orlando, FL, December 2011.