Assessment of Control Algorithms for Mars Entry Vehicles with Flap-Based Trajectory Control Under Uncertainty

Daniel L. Engel* and Zachary R. Putnam†
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Soumyo Dutta‡
NASA Langley Research Center, Hampton, VA 23681, USA

Flap-based steering systems on blunt-body Mars entry vehicles may improve flight performance relative to existing bank-angle steering systems. Successful implementation of articulating aerodynamic flaps on a hypersonic entry vehicle requires an active control system to map angle of attack and sideslip angle commands to flap deflection commands. Here, a successive-linearization model predictive control algorithm, as well as a linear-quadratic regulator, are designed and assessed to address this multiple input multiple output control problem. These two control algorithms are assessed under uncertainty in Monte Carlo simulations for various flap configurations and command profiles. Results indicate that while both control algorithms provide successful command tracking in the presence of uncertainty, the model predictive controller provides tracking errors about half the value of those corresponding to the linear-quadratic regulator, indicating increased robustness. Comparison of various flap configurations showed the model predictive controller can be applied to various flap configurations successfully with only marginal performance differences between configurations.

I. Introduction

Guided entries of blunt-body entry vehicles to-date have used bank-angle steering for hypersonic trajectory control. Bank-angle steering has a long history of successful applications at Earth with the Apollo Command Module [1], Space Shuttle [2], and Orion crew vehicle [3]. The Mars Science Laboratory (MSL) [4] and Mars 2020 missions also used bank-angle steering, allowing for increasing ambitious Mars missions and a higher degree of landing accuracy. The Mars 2020 mission, in particular, achieved about 1 km of accuracy from the zero-divert point [5]. Guidance algorithms on bank-angle steering vehicles typically select the bank angle magnitude to hit a longitudinal target, and the sign of the bank angle is periodically varied to manage the lateral error through open loop bank reversals [4]. With a bank angle commanded to the vehicle, simple phase-plane controllers have been used to command reaction control system (RCS) thrusters to obtain the desired movement around the neutrally stable bank axis [6].

While bank-angle has been successful so far, this hypersonic steering scheme has several disadvantages. The open-loop bank reversals used as a result of the coupling of control between the longitudinal and lateral motion can result in decreased positional accuracy at terminal descent initiation (TDI) [7]. While retropropulsion may be used to fly out these errors, correcting positional errors due to bank reversals has the potential to add propellant mass to the vehicle. Bank-angle steering systems often require a center-of-gravity (CG) offset, in addition to spacecraft packaging, to achieve a non-zero trim angle of attack (α) and associated lift-to-drag ratio (L/D) for steering. Achieving the desired CG offset for MSL and Mars 2020 required tungsten masses, which were a significant fraction of the landed payload mass [8]. This may be problematic when scaling to a larger vehicle mass or higher lift-to-drag ratio that may add tons of mass to the vehicle in the form of ballast, which also must be jettisoned during entry [9]. Lastly, the RCS jets on the backshell of a bank-angle steering vehicle add additional hot gasses to the vehicle's hypersonic wake, complicating the aerothermal environment behind vehicle and interfering with the control authority of the thrusters [10–13].

Recent studies have investigated using the angle of attack and sideslip angle (β) for trajectory control, i.e α-β steering. Also referred to as direct force control (DFC), α-β steering has been investigated for Mars entry missions [9, 14, 15], as well as for aerocapture missions at Mars [16], Venus [17], Titan [18], and Neptune [19]. A vehicle using α-β steering avoids the need for open-loop reversals characteristic of bank-angle steering [7], although performing β reversals is

*PhD Candidate, Department of Aerospace Engineering, 104 S. Wright St., AIAA Student Member
†Adjunct Assistant Professor, Department of Aerospace Engineering, 104 S. Wright St., AIAA Senior Member
‡Aerospace Engineer, Atmospheric Flight and Entry Systems Branch, 1 N. Dryden St., M/S 489, AIAA Associate Fellow

optimal for certain objective functions [20]. Results have shown that α-β steering may provide a greater robustness to atmospheric dispersions [21], better landing mass fraction, and a lower propulsion system usage for powered descent, relative to bank-angle steering [9]. α-β steering is also advantageous in that these systems do not nominally need large ballast masses for CG offsets. Furthermore, a vehicle using α-β steering does not necessarily need a RCS, although a mechanism to regulate the bank angle or bank rate is likely necessary [22]. Several actuation concepts for α-β steering have been studied, including a morphing vehicle structure [7], CG movement systems [23], and aerodynamic flaps [24].

Controlling α and β on a blunt-body entry vehicle using several different effectors is a multiple input multiple output (MIMO) control problem and is more challenging than controlling the bank angle due to the vehicle's motion occurring over several statically stable axes. For an α-β steering system to be feasible and provide performance advantages over bank-angle steering, such a system must provide good control performance. Relatively simple proportional-derivative-integral (PID) controllers have been used in studies of a human-scale vehicle using flaps and morphing structures during Mars entries [7, 14], but control performance was not extensively studied. Both reference [23] and studies of the Pterodactyl concept [24, 25] investigated a linear-quadratic regulator (LQR) for a CG movement system in a Mars entry capsule and a deployable Earth entry vehicle with 8 flaps, respectively. Model predictive control (MPC), an advanced control technique, has been studied for thruster firings and control surface deflections on the X-33 spaceplane [26, 27] and was used to assess several flap configurations on a blunt body [28]. In this study a successive-linearization MPC algorithm is formulated and applied to a blunt-body entry vehicle with flaps, along with LQR. The developed controllers and simulation environment are capable of being applied to an arbitrary flap configuration and command profile. This study then goes beyond previous studies by assessing flap controller performance under uncertainty in the vehicle aerodynamics, mass properties, and initial conditions through Monte Carlo simulation.

II. Method

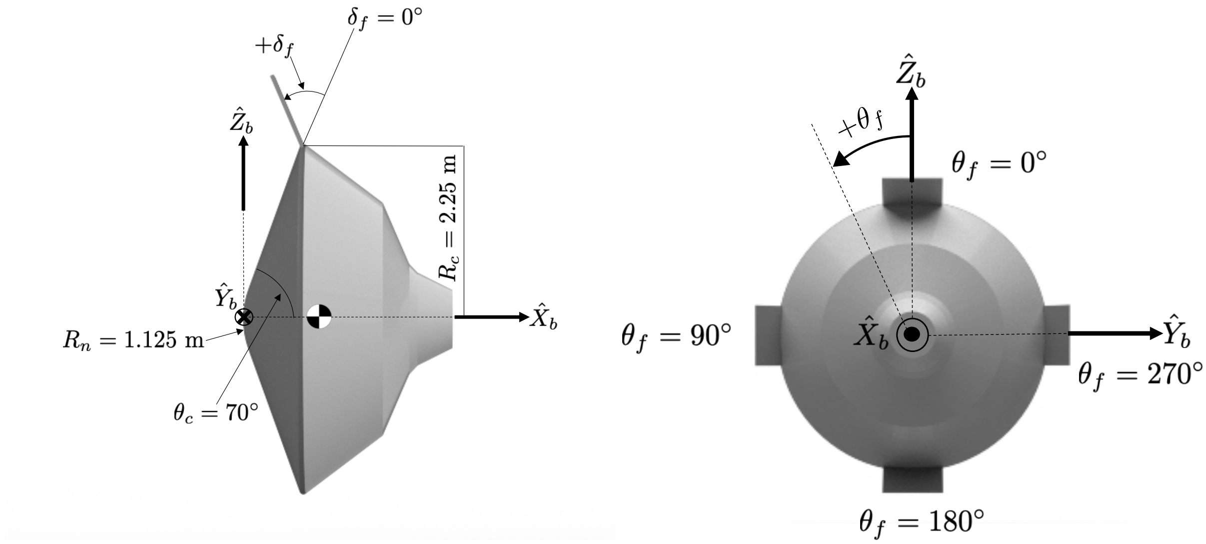
A. Vehicle Model and Coordinate System

This study considers an MSL-like axisymmetric blunt body entry vehicle. Flaps are placed at the shoulder of the forebody heatshield at flap location angle θ_f, and the flap deflection angle, δ_f, is the compliment of the angle between the flap and the cone (see Fig. 1). For example, a deflection angle of 20 deg corresponds to the flap perpendicular to the flow at $\alpha = 0$ deg, and a deflection angle of -70 deg corresponds to a flap parallel to the flow at $\alpha = 0$ deg. Flap areas are parameterized as a percentage of the aerodynamic reference area, and several different flap positions around the vehicle are possible, as shown in Fig. 2. The mass properties of the entry vehicle body are given in Table 1, and the moment of inertia values I_{XX}, I_{YY}, and I_{ZZ} are the diagonal components of the principal moment of inertia tensor, $\mathbf{I}$. The CG is located on the centerline and is located 17 cm forward of the MSL CG postion along the X-axis [29].

Table 1 Mass Properties of Entry Vehicle

Parameter	Value
I_{XX}	5688.61 kg-m^2
I_{YY}	3772.07 kg-m^2
I_{ZZ}	3772.07 kg-m^2
CG Location	$[1.18\ 0\ 0]^T$ m

Figure 3 shows the coordinate system and aerodynamic angles on the vehicle in the body frame (black) and wind frame (red), and the relationship between α and β, the freestream velocity vector $\mathbf{V}_\infty$, and the vehicle velocity components in the body frame (u, v, w) can be seen in equations (1), (2), and (3) [30]. A positive α and β were defined in this study as rotations about the positive $\hat{Y}$ and $\hat{Z}$ axes, with respect to the vehicle velocity vector. The transformation from the wind to body frame is equivalent to a $1-3-2$ sequential rotation of the vehicle by the bank angle (σ), β, and α, respectively. Additional details on the vehicle model and coordinate system can be found in Ref. [28].



(a) **(b)**

Fig. 1 Blunt body with flap-based steering system concept [22]: (a) vehicle dimensions and (b) flap positions.

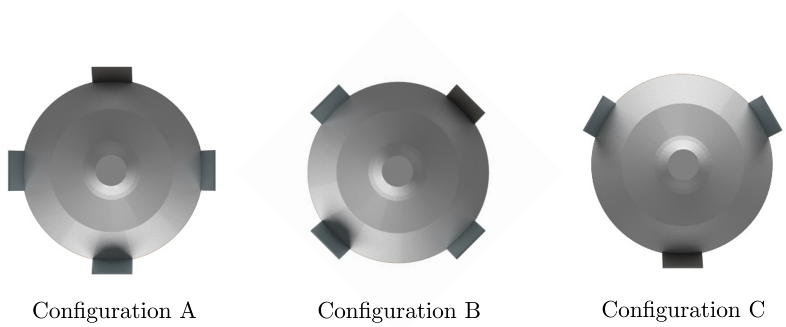


Fig. 2 Flap position options on entry vehicle.

$$u = |\mathbf{V}_\infty| \cos(\beta) \cos(\alpha) \tag{1}$$
$$v = |\mathbf{V}_\infty| \sin(\beta) \tag{2}$$
$$w = |\mathbf{V}_\infty| \cos(\beta) \sin(\alpha) \tag{3}$$

B. Equations of Motion

The controllers designed in this study use flap deflections to rotate the vehicle to achieve desired α and β commands. The performance of the controllers are assessed by simulating the three degree-of-freedom (3DOF) rotation of the flapped entry vehicle at a specified flight condition (i.e. dynamic pressure and Mach number), assuming the rotational motion is decoupled from the translational motion of the vehicle during entry. The angular kinematics associated with the vehicle are:

$$\dot{\sigma} = -\omega_x \cos\alpha \sec\beta - \omega_z \sin\alpha \sec\beta \tag{4}$$
$$\dot{\alpha} = \omega_x \cos\alpha \tan\beta + \omega_y + \omega_z \sin\alpha \tan\beta \tag{5}$$
$$\dot{\beta} = -\omega_x \sin\alpha + \omega_z \cos\alpha \tag{6}$$

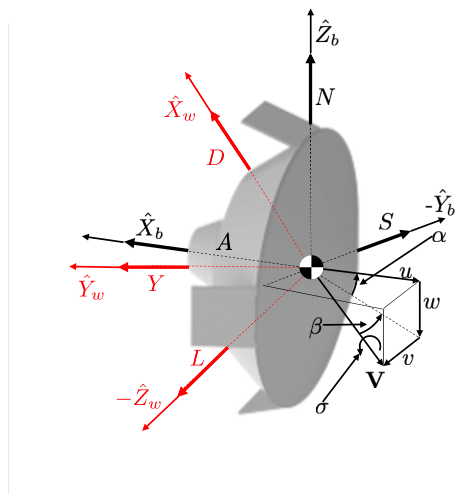


Fig. 3 Vehicle coordinate system and aerodynamic angles.

And the angular dynamics are given by:

$$\begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \mathbf{I}^{-1} \left[\mathbf{M}_{body} + \mathbf{M}_{dynamic} + \sum_{i=1}^{n} \left(\mathbf{M}_{flap,i} \right) - \omega_B \times \mathbf{I} \omega_B \right] \tag{7}$$

Here $\mathbf{M}_{body}$ and $\mathbf{M}_{dynamic}$ are the static and dynamic aerodynamic moment vectors of the vehicle body, respectively, and are given by:

$$\mathbf{M}_{body} = q_\infty s_{ref} c_{ref} \begin{bmatrix} C_l \\ C_m \\ C_n \end{bmatrix} \tag{8}$$

$$\mathbf{M}_{dynamic} = \frac{q_\infty s_{ref} c_{ref}^2}{2V_\infty} \begin{bmatrix} \omega_x C_{l_p} \\ \omega_y C_{m_q} \\ \omega_z C_{n_r} \end{bmatrix} \tag{9}$$

Where the aerodynamic reference area $s_{ref} = \pi {R_c}^2$, aerodynamic reference chord $c_{ref} = 2R_c$, angular velocity vector $\omega_B = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$, and $q_\infty = \frac{1}{2}\rho_\infty V_\infty^2$ is the freestream dynamic pressure. V_∞ and ρ_∞ are the freestream velocity magnitude and density, respectively. The static and dynamic aerodynamic moment coefficients C_l, C_m, C_n, C_{l_p}, C_{m_q}, and C_{n_r} are functions of α, β, the Mach number, and the CG position. Aerodynamic data from the Mars Science Laboratory [29] and Phoenix missions [31] were used to construct an aerodynamics database for these aerodynamic coefficients, and C_{l_p} is assumed to be zero. The static aerodynamic moment from the i^{th} flap, $\mathbf{M}_{flap,i}$, is a function of α, β, θ_f, and δ_f. The aerodynamic moments from each flap is calculated using Modified Newtonian aerodynamics [32], a first-order approximation for hypersonic blunt bodies [33]. This implementation considers the contributions of the front and sides of the flaps, and more details on the Modified Newtonian calculations used to determine the aerodynamic moments from the falps can be found in Ref. [28].

While this study is primarily concerned with the 3DOF rotational motion of the entry vehicle, it may be desired to assess how controller tracking impacts a vehicle's entry trajectory. To do this, the 3DOF translational motion of the entry vehicle over an oblate, rotating Mars can also be considered. These equations of motion are provided below for a vehicle using α-β steering. Here, r is the radial distance of the entry vehicle to the center of Mars, θ_r is the longitude, ϕ is the latitude, V_r is the Mars-relative velocity magnitude, γ_r is the Mars-relative flight-path angle, ψ is the Mars-relative azimuth angle, and Ω is the rotation rate of Mars.

$$\dot{r} = V_r \sin\gamma_r \tag{10}$$

$$\dot{\theta}_r = \frac{V_r \cos\gamma_r \sin\psi}{r\cos\phi} \tag{11}$$

$$\dot{\phi} = \frac{V_r \cos\gamma_r \cos\psi}{r} \tag{12}$$

$$\dot{V}_r = -\frac{D}{m} - g_r \sin\gamma_r - g_\phi \cos\gamma_r \cos\psi + \Omega^2 r \cos\phi(\sin\gamma_r \cos\phi - \cos\gamma_r \sin\phi\cos\psi) \tag{13}$$

$$\begin{aligned}\dot{\gamma}_r = \frac{1}{V_r}\bigg[&\frac{L}{m} + (V_r^2/r - g_r)\cos\gamma_r + g_\phi \sin\gamma_r \cos\psi + 2\Omega V_r \cos\phi\sin\psi \\ &+\Omega^2 r\cos\phi(\cos\gamma_r\cos\phi + \sin\gamma_r\cos\psi\sin\phi)\bigg]\end{aligned} \tag{14}$$

$$\begin{aligned}\dot{\psi} = \frac{1}{V_r}\bigg[&\frac{Y}{m\cos\gamma_r} + \frac{V_r^2}{r}\cos\gamma_r\sin\psi\tan\phi + g_\phi\frac{\sin\psi}{\cos\gamma_r} \\ &-2\Omega V_r(\tan\gamma_r\cos\psi\cos\phi - \sin\phi) + \frac{\Omega^2 r}{\cos\gamma_r}\sin\psi\sin\phi\cos\phi\bigg]\end{aligned} \tag{15}$$

g_r and g_ϕ are the components of the acceleration due to gravity, including J_2 effects [34]:

$$g_r = \frac{\mu}{r^2}\left[1 + J_2\left(\frac{R_m}{r}\right)^2(1.5 - 4.5\sin^2\phi)\right] \tag{16}$$

$$g_\phi = \frac{\mu}{r^2}\left[J_2\left(\frac{R_m}{r}\right)^2(3\sin\phi\cos\phi)\right] \tag{17}$$

And D, L, and Y are the aerodynamic drag, lift, and side forces given by:

$$D = q_\infty s_{ref} C_D \tag{18}$$

$$L = q_\infty s_{ref} C_L \tag{19}$$

$$Y = q_\infty s_{ref} C_Y \tag{20}$$

Where the aerodynamic coefficients C_D, C_L, and C_Y are also determined from the vehicle body aerodynamics database, and the density used to calculate q_∞ is determined using the Mars Global Reference Atmospheric Model [35].

C. Successive-Linearization Model Predictive Control Algorithm

MPC is a control method for MIMO systems in which an online optimization problem is solved to obtain the necessary control action. This optimization problem includes a performance index and constraints, potentially including inequality constraints on the states and inputs and equality constraints enforcing the plant model. This ability to handle constraints when calculating the control is an attractive feature of MPC, relative to other control algorithms such as PID or LQR. MPC is typically applied in a receding horizon, where the predictions within the optimization march further into the future every time the controller is called [36]. Some implementations of MPC utilize a linear time-invariant (LTI) system plant, but a single LTI model was found to be insufficient to capture the nonlinear attitude dynamics of the entry vehicle with flaps considered in this study. In a past study, a fully nonlinear MPC was utilized for this problem at the cost of a high computational burden [28]. Furthermore, fully nonlinear MPC is a non-convex problem with no guarantees of converging to to an optimal solution [36]. A successive-linearization MPC algorithm can potentially obtain similar performance to a fully nonlinear MPC algorithm by linearizing the nonlinear model every time the controller is called to obtain a locally-accurate LTI model to be used over short time horizons, and such an algorithm was implemented in Ref. [37] for robotics applications. With a successive linearization MPC algorithm, the optimization problem solved to determine the control action is convex, which guarantees an optimal solution can be quickly found. The successive-linearization MPC algorithm used for this application is described below in its general form and is inspired by the algorithms described in Refs. [37] and [38], as well as the MATLAB MPC Toolbox [39].

The MPC algorithm operates over a finite time horizon where the state is predicted into the future for a prediction horizon p, where p is a finite number of prediction steps. c control actions occurring at consistent times to the prediction steps are also considered, where c is the control horizon. If $c < p$, then the the control action over the remaining prediction steps is constant at its final value. This MPC algorithm uses a quadratic programming vector $\mathbf{z}$ which is a $(pn_x + cn_u) \times 1$ vector, where n_x and n_u are the number of states and inputs, respectively. Essentially, $\mathbf{z}$ contains p sets of the state x, followed by c steps of the input u. The goal of the algorithm is to choose $\mathbf{z}$ to achieve desired controller performance, while also satisfying constraints. A quadratic cost J_{MPC} is considered, in which it is desired to track some reference states $\mathbf{x_{ref}}$ while also using small control actions:

$$J_{MPC} = \mathbf{z}^T H \mathbf{z} + \mathbf{f}^T \mathbf{z} \tag{21}$$

Here H is a $(pn_x + cn_u) \times (pn_x + cn_u)$ matrix. H corresponds to the quadratic term in the cost and has the block diagonal form

$$H = \begin{bmatrix} Q_1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & Q_2 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q_p & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & R_0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & R_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & R_{c-1} \end{bmatrix} \tag{22}$$

Here Q and R are diagonal matrices, where each of the entries are weights on the states and inputs, respectively. Each of the Q and R matrices in H need not be equal, however for this application, the weights are chosen to be constant over the entire prediction horizon, i.e. $Q_1 = Q_2 = \cdots = Q_p$ and $R_0 = R_1 = \cdots = R_{c-1}$. $\mathbf{f}$ is a $pn_x + cn_u \times 1$ vector corresponding to the linear component of the cost and is given by:

$$f = -2 \begin{bmatrix} Q_1 \mathbf{x_{ref}} \\ Q_2 \mathbf{x_{ref}} \\ \vdots \\ Q_p \mathbf{x_{ref}} \\ \mathbf{0} \end{bmatrix} \tag{23}$$

To obtain control actions from the MPC optimization problem that make sense for the given vehicle, equality constraints describing the vehicle dynamics must be provided in the optimization problem. This is done by linearizing and discretizing the nonlinear plant model. Upon calling the MPC algorithm, the nonlinear state derivative model $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ is linearized about the current state $\mathbf{x_0}$ and current input $\mathbf{u_0}$, with $\dot{\mathbf{x}}_\mathbf{0} = \mathbf{f}(\mathbf{x_0}, \mathbf{u_0})$, into the form:

$$\dot{\mathbf{x}}_\mathbf{L} = A(\mathbf{x_L} - \mathbf{x_0}) + B(\mathbf{u_L} - \mathbf{u_0}) + \dot{\mathbf{x}}_\mathbf{0} \tag{24}$$

Here, $\mathbf{x_L}$ and $\mathbf{u_L}$ are the state and input vectors for the linearized continuous-time system. A and B are Jacobian matrices, given by:

$$A = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x_0}, \mathbf{u_0}} \qquad B = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\mathbf{x_0}, \mathbf{u_0}} \tag{25}$$

The MPC works by solving a finite-size optimization problem and thus requires a discrete-time model. The linearized continuous-time model in Eq. (24) can then be discretized in order to incorporate a plant model into the MPC optimization problem. Following a similar procedure to that of Ref. [37], the solution to the linearized equations of motion can be used to discretize the system. Over a single step between time $t = T_s k$ and $t = T_s(k+1)$, this solution of the linearized equations of motion is

$$\mathbf{x_L}[k+1] = e^{AT_s} \mathbf{x_L}[k] + \int_{kT_s}^{T_s(k+1)} e^{A(T_s(k+1)-\tau)} \left(B\mathbf{u_L}(\tau) + \dot{\mathbf{x}}_\mathbf{0} - A\mathbf{x_0} - B\mathbf{u_0}\right) d\tau \tag{26}$$

Here, T_s is the sampling time, in which the system is discretized, and the total length of time the MPC predicts into the future is equal to pT_s. A zero-order hold is then applied to the system, assuming control inputs are constant over each step. In the case that A is invertible, this yields

$$\mathbf{x_L}[k+1] = e^{AT_s}\mathbf{x_L}[k] + A^{-1}\left(e^{AT_s} - I\right)\left(B\mathbf{u_L}[k] + \dot{\mathbf{x}}_\mathbf{0} - A\mathbf{x_0} - B\mathbf{u_0}\right) \tag{27}$$

Expanding this expression yields

$$\mathbf{x_L}[k+1] = e^{AT_s}\mathbf{x_L}[k] + A^{-1}\left(e^{AT_s} - I\right)B\left(\mathbf{u_L}[k] - \mathbf{u_0}\right) + A^{-1}\left(e^{AT_s} - I\right)\dot{\mathbf{x}}_\mathbf{0} - A^{-1}\left(e^{AT_s} - I\right)A\mathbf{x_0} \tag{28}$$

Using the Taylor expansion of e^{AT_s}, it can be seen that

$$A^{-1}e^{AT_s}A = A^{-1}\left(I + AT_s + \frac{(AT_s)^2}{2!} + \frac{(AT_s)^3}{3!} + \cdots\right)A = e^{AT_s} \tag{29}$$

After making the following substitutions

$$A_D = e^{AT_s} \quad B_D = A^{-1}\left(e^{AT_s} - I\right)B \quad \bar{\mathbf{x}}_\mathbf{D} = A^{-1}\left(e^{AT_s} - I\right)\dot{\mathbf{x}}_\mathbf{0} + \mathbf{x_0} \tag{30}$$

and replacing $\mathbf{x_L}[k+1]$ and $\mathbf{x_L}[k]$ with $\mathbf{x_D^{k+1}}$ and $\mathbf{x_D^k}$, respectively, the following discrete-time model is obtained.

$$\mathbf{x_D^{k+1}} = A_D(\mathbf{x_D^k} - \mathbf{x_0}) + B_D(\mathbf{u_D^K} - \mathbf{u_0}) + \bar{\mathbf{x}}_\mathbf{D} \tag{31}$$

In implementation A may not be invertible, which means the following integrals will need to be computed.

$$B_D = \int_{kT_s}^{T_s(k+1)} e^{A(T_s(k+1)-\tau)}Bd\tau \quad \bar{\mathbf{x}}_\mathbf{D} = \int_{kT_s}^{T_s(k+1)} e^{A(T_s(k+1)-\tau)}\dot{\mathbf{x}}_\mathbf{0}d\tau + \mathbf{x_0} \tag{32}$$

From Ref. [40], the following expressions can be used to evaluate the integrals for B_D and $\bar{\mathbf{x}}_\mathbf{D}$.

$$\begin{bmatrix} A_D & B_D \\ 0 & I \end{bmatrix} = e^{\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}T_s} \tag{33}$$

$$\begin{bmatrix} A_D & \bar{\mathbf{x}}_\mathbf{D} - \mathbf{x_0} \\ 0 & I \end{bmatrix} = e^{\begin{bmatrix} A & \dot{\mathbf{x}}_\mathbf{0} \\ 0 & 0 \end{bmatrix}T_s} \tag{34}$$

The discrete time system in Eq. (31) can be enforced as an equality constraint across the prediction horizon by incorporating it into the following linear equation:

$$A_{eq}\mathbf{z} = \mathbf{b_{eq}} \tag{35}$$

In the above equation, A_{eq} is a $pn_x \times (pn_x + cn_u)$ block matrix of the form

$$A_{eq} = \begin{bmatrix} -I & 0 & 0 & \cdots & 0 & 0 & B_D & 0 & 0 & \cdots & 0 & 0 \\ A_D & -I & 0 & \cdots & 0 & 0 & 0 & B_D & 0 & \cdots & 0 & 0 \\ 0 & A_D & -I & \cdots & 0 & 0 & 0 & 0 & B_D & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -I & 0 & 0 & 0 & 0 & \cdots & B_D & 0 \\ 0 & 0 & 0 & \cdots & A_D & -I & 0 & 0 & 0 & \cdots & 0 & B_D \end{bmatrix} \tag{36}$$

And $\mathbf{b_{eq}}$ is a $pn_x \times 1$ vector of the form

$$\mathbf{b_{eq}} = \begin{bmatrix} B_D\mathbf{u_0} - \mathbf{\bar{x}_D} \\ A_D\mathbf{x_0} + B_D\mathbf{u_0} - \mathbf{\bar{x}_D} \\ A_D\mathbf{x_0} + B_D\mathbf{u_0} - \mathbf{\bar{x}_D} \\ \vdots \\ A_D\mathbf{x_0} + B_D\mathbf{u_0} - \mathbf{\bar{x}_D} \end{bmatrix} \tag{37}$$

The vector $\mathbf{z}$ contains the discrete time state $\mathbf{x_D^k}$ and input $\mathbf{u_D^K}$ as its entries and has the following form

$$\mathbf{z} = \begin{bmatrix} \mathbf{x_D^1}^T & \mathbf{x_D^2}^T & \cdots & \mathbf{x_D^p}^T & \mathbf{u_D^0}^T & \mathbf{u_D^1}^T & \cdots & \mathbf{u_D^{c-1}}^T \end{bmatrix}^T \tag{38}$$

Besides merely enforcing an approximation to the vehicle dynamics in the MPC optimization, it can be useful to enforce constraints on the input magnitudes and rates to assure that the commanded inputs are reasonable for the system. Constraints on the input magnitudes are of the form

$$\mathbf{u_D^K} \leq \mathbf{u_{max}} \quad -\mathbf{u_D^K} \leq -\mathbf{u_{min}} \tag{39}$$

Constraints on the input rates can be placed by assuming the maximum magnitude of a difference in subsequent input commands is equal to the maximum rate multiplied by T_s:

$$\mathbf{u_D^K} - \mathbf{u_D^{K-1}} \leq \mathbf{\dot{u}_{max}} T_s \quad -\left(\mathbf{u_D^K} - \mathbf{u_D^{K-1}}\right) \leq -\mathbf{\dot{u}_{min}} T_s \tag{40}$$

Both of these constraints can be incorporated in the following matrix form

$$P\mathbf{z} \leq h \tag{41}$$

Here P is a $4pn_u \times (pn_x + cn_u)$ matrix containing values of 1 and -1, and $\mathbf{h}$ is a $4pn_u$ vector containing the values of the constraints. With the cost, equality constraints, and inequality constraints formed, they can all be combined into the following MPC optimization problem:

$$\boxed{\begin{aligned} \min_{\mathbf{z}} \quad & J_{MPC} = \mathbf{z}^T H \mathbf{z} + \mathbf{f}^T \mathbf{z} \\ \text{s.t.} \quad & A_{eq}\mathbf{z} = \mathbf{b_{eq}} \\ & P\mathbf{z} \leq \mathbf{h} \end{aligned}} \tag{42}$$

The above quadratic programming problem is convex and can be quickly solved with a commercial solver. For this implementation, the MPC optimization problem is solved using the quadprog function in MATLAB with the active-set algorithm [41]. Solving this optimization problem yields a solution $\mathbf{z}^*$ which minimizes J_{MPC}. The component of $\mathbf{z}^*$ corresponding to $\mathbf{u_D^0}$ is commanded as the input to be utilized, i.e. only the first input of the solution to the MPC optimization problem is directly used. $\mathbf{z}^*$ is then passed back into the quadratic programming solver as the initial guess for the subsequent call to the controller. $\mathbf{z}_0 = \mathbf{0}$ is used as the initial guess for the first call to the controller.

D. Linear-Quadratic Regulator

LQR is also discussed here as a simpler MIMO control alternative to MPC. Similar to the MPC algorithm described above, LQR involves linearizing the state derivative about an equilibrium point $(\mathbf{x_e}, \mathbf{u_e})$ into the form $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$. A and B are found here similarly to Eq. (25), with the exception that the Jacobians are evaluated at $(\mathbf{x_e}, \mathbf{u_e})$. The control law for LQR is

$$\mathbf{u} = -K\mathbf{x_c} + \mathbf{u_e} \tag{43}$$

Where $\mathbf{x_c} = \mathbf{x} - \mathbf{x_e}$, and K is a static gain matrix that solves the infinite time-horizon optimization problem

$$J_{LQR} = \int_0^\infty \left(\mathbf{x_c}^T Q \mathbf{x_c} + \mathbf{u}^T R \mathbf{u}\right) dt \tag{44}$$

As with MPC, the matrices Q and R, respectively, in Eq. (44) allow the user to penalize the relative importance of state-tracking accuracy and control effort, for each of the different states and inputs. Although the controller is linearized around the equilibrium state $\mathbf{x_e}$, in implementation, $\mathbf{x_e}$ can be replaced with the commanded state when calculating $\mathbf{x_c}$ in Eq. (43) in order to track states other than the equilibrium state. It is possible to solve the optimization problem in (44) every time the controller is called to obtain a new K. In this implementation, however, K is only computed once offline at the beginning of the simulation, and each call to the controller solely requires the matrix operations in Eq. (43).

E. Simulation Environment and Control Application

Both the successive-linearization MPC algorithm and LQR were implemented into a controller simulation environment for a blunt-body Mars entry vehicle with flaps. The simulation environment numerically integrates equations of motion (4)-(7) as the truth dynamics with a fourth-order Runge-Kutta integration scheme and a constant time step of 0.01 s. The nonlinear state derivative equations which are linearized to be used with MPC and LQR are similar to those used in the truth dynamics, with the exception that a cubic fit to the moment coefficient data from the aerodynamics database is used to allow for analytical Jacobians to be calculated. The simulation environment allows for controllers to be automatically generated and simulated for a vehicle configuration with an arbitrary number of flaps, flap area, and flap positioning. The state vector to be controlled by the vehicle is $x^T = \begin{bmatrix} \alpha & \beta & \omega_y & \omega_z & e_\alpha & e_\beta \end{bmatrix}^T$. Here e_α and e_β are integral error states used in both controllers to reduce steady state error. The time derivatives of these integral error states are:

$$\dot{e}_\alpha = \alpha - \alpha_{des} \tag{45}$$

$$\dot{e}_\beta = \beta - \beta_{des} \tag{46}$$

Where α_{des} and β_{des} are the desired angle of attack and sideslip angle, respectively. Both controllers use the flap deflections to try to track commanded α and β values while also keeping ω_y and ω_z near zero. Every time a controller is called, the vehicle obtains a new set of commanded flap deflections. The simulation environment takes into account that these flaps cannot move instantaneously and cannot exceed certain flap deflection angles, subjecting the deflections to the rate, acceleration, and position limits given in Table 2. The MPC algorithm incorporates these flap deflection and deflection rate limits using (39) and (40), while LQR does not use any constraints in the calculation of flap deflection commands. Relevant parameters in the design of the MPC and LQR controllers are given in Tables 3 and 4, respectively. Discussions of how these parameters can impact vehicle performance are given in Ref. [28]. In this implementation, the equilibrium state for LQR is when all states are equal to zero. The subscripts in Q or R denote the the state or input in which the given weight is applied to. Note the lower controller frequency of MPC is a result of its higher computational requirements. All controller simulations were performed in MATLAB R2021b on a 2021 M1 MacBook Pro with 32 GB RAM.

Table 2 Flap Deflection Limits

Parameter	Value
Minimum flap deflection angle	−70 deg
Maximum flap deflection angle	20 deg
Minimum flap deflection rate	−18 deg/s
Maximum flap deflection rate	18 deg/s
Minimum flap deflection acceleration	−1000 deg/s^2
Maximum flap deflection acceleration	1000 deg/s^2

III. Results

A. Nominal Performance

Controller tracking performance was first tested for a nominal case without any uncertainty in the truth dynamics. MPC and LQR were both designed for Configuration A in Fig. 2, and several α and β profiles were considered for assessing different sets of steering commands and flight conditions. The commands corresponding to profile 1 are α and β commands created by scaling bank commands from the Apollo Final Phase guidance algorithm, the commands in profiles 2 and 3 come from solutions to optimal control problems. The dynamic pressure and Mach number profiles corresponding to the trajectories for each of these profiles are shown in Fig. 4 and are fed into the simulation environment. Results showing the α and β command tracking for this trajectory are shown in Figures 5, 6, and 7 for profiles 1, 2, and 3, respectively. These plots indicate that MPC is able to do a better job tracking the command profile than LQR.

Table 3 MPC Parameters

Parameter	Value
Q_α	4
Q_β	4
Q_{ω_y}	1.5625
Q_{ω_z}	1.5625
Q_{e_α}	5
Q_{e_β}	5
R_{δ_f}	0.05
T_s	0.2 s
p	6
c	4
Controller frequency	5 Hz

Table 4 LQR Parameters

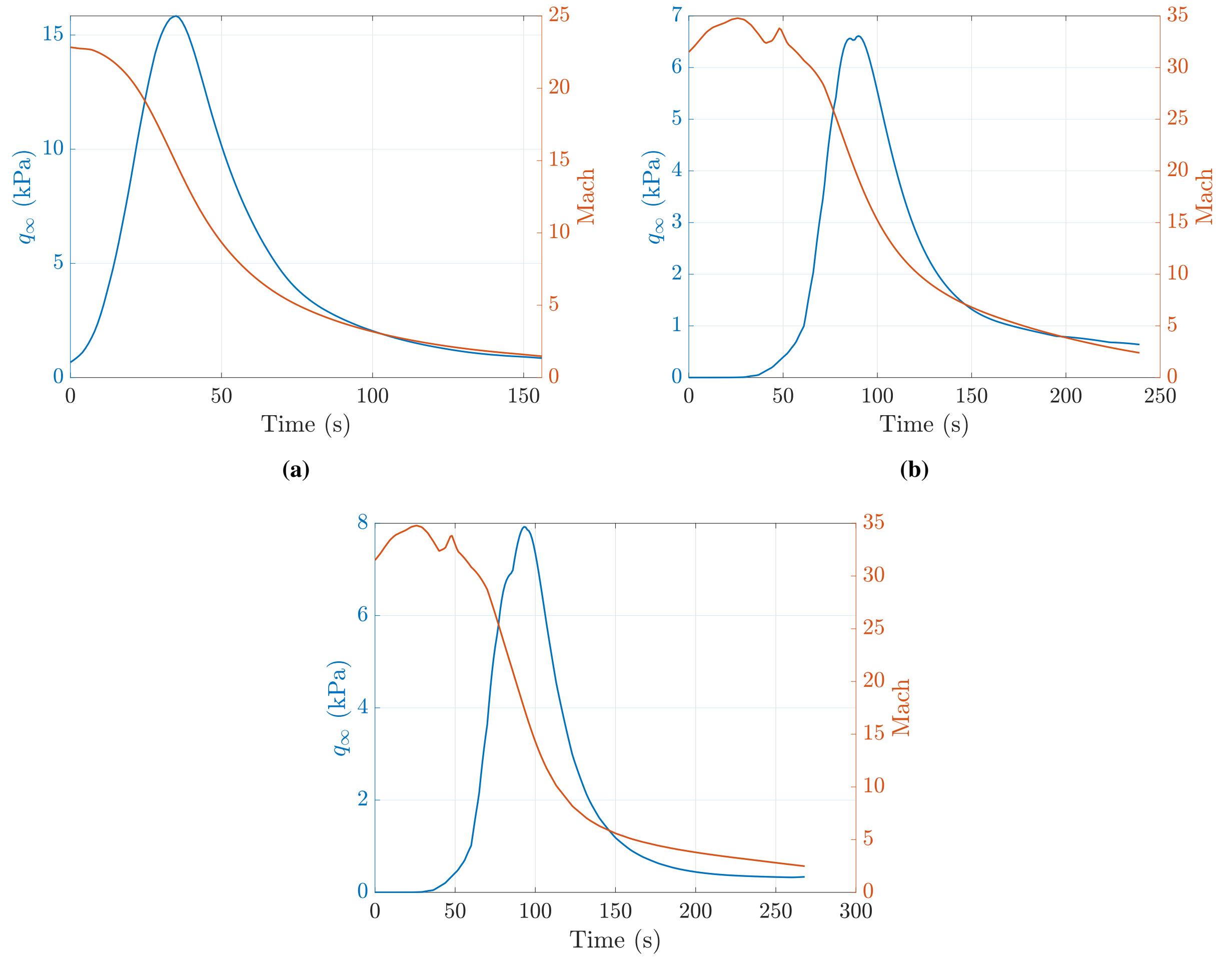
Parameter	Value
Q_α	80
Q_β	80
Q_{ω_y}	60
Q_{ω_z}	60
Q_{e_α}	65
Q_{e_β}	65
R_{δ_f}	20
u_e	-30 deg
Controller frequency	50 Hz

Qualitatively, the LQR results tend to lag behind the command profile, relative to MPC, and there tends to be a larger and longer-lasting oscillatory behavior for LQR near the start of the trajectories when the dynamic pressure is low.

This improved tracking performance of MPC can be verified quantitatively by considering the integral error in the tracking.

$$e_\alpha = \int_{t_0}^{t_f} |\alpha - \alpha_{des}| dt \qquad e_\beta = \int_{t_0}^{t_f} |\beta - \beta_{des}| dt \tag{47}$$

These numerical results are reported in Table 5, where the integral errors are larger for LQR than for MPC. Part of the improved performance of MPC here, relative to LQR, is that both controllers have been designed to provide good tracking and stability for dispersed cases in Monte Carlo simulations. In the process of tuning LQR, it was found that a nominal LQR case could obtain similar tracking performance to MPC if R_{δ_f} for LQR was decreased to obtain a more aggressive controller response. While this more aggressive LQR controller worked well for a single case, attempting to use it for different command profiles and in uncertainty analysis often led the vehicle to become unstable and tumble out of control. LQR was therefore made more conservative to accommodate a variety of command profiles and uncertainty by increasing R_{δ_f}; this comes at the cost of decreased tracking accuracy in the nominal case. The tumbling behavior of LQR when it is too aggressive were not observed for MPC and is likely due to LQR not enforcing constraints on the flap deflections or flap deflection rates. Without these constraints, the commanded flap deflections from LQR can end up being too aggressive for the rate- and acceleration-limited flaps to achieve, resulting in a deviation in the actual flap commands, compared to what LQR commands. This eventually results in a build up of flap oscillations and the vehicle

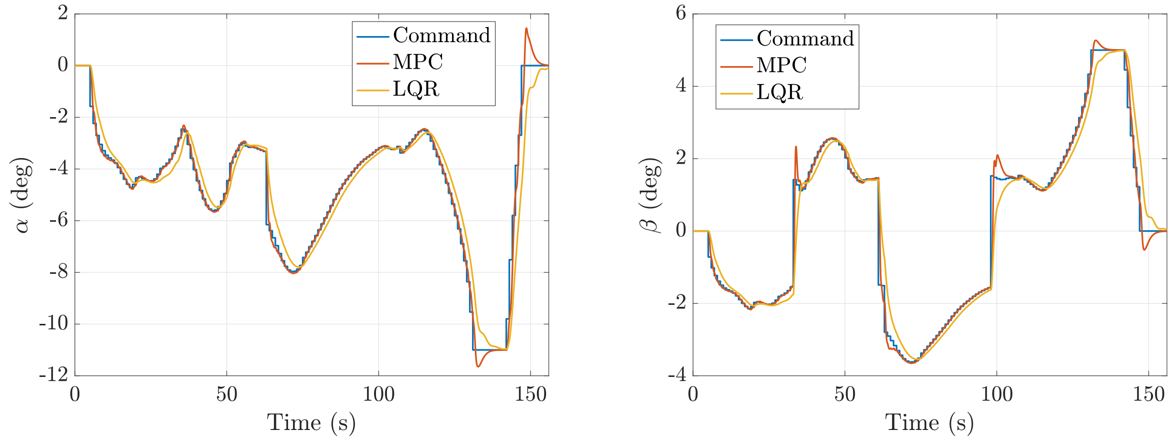


(c)

Fig. 4 Dynamic pressure and Mach number profile for controller simulation: (a) profile 1, (b) profile 2, and (c) profile 3.

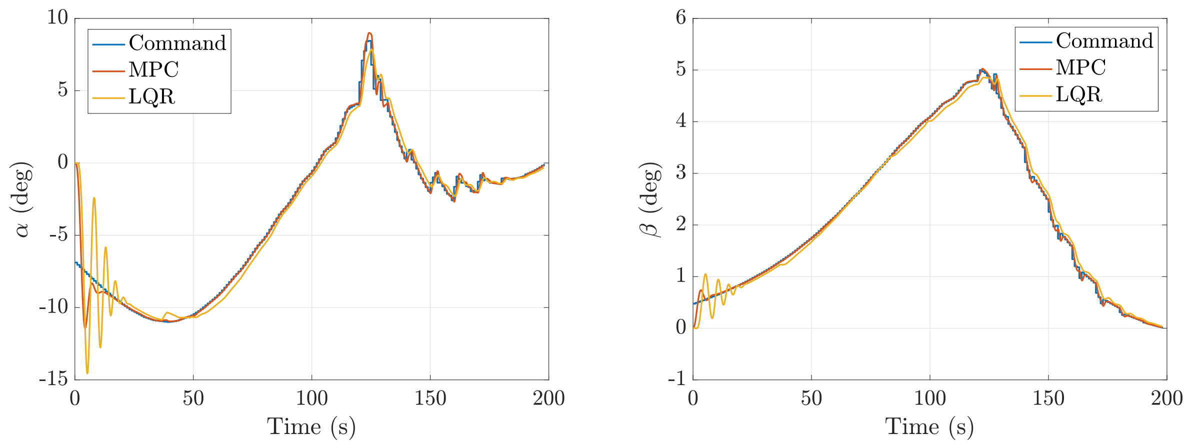
tumbling. A main difference between MPC and LQR is that MPC incorporates some of these flap deflection constraints into its calculation of the commanded flap deflections, resulting in flap commands that the flap actuators can closely achieve. Furthermore, the MPC has also has an advantage over LQR in that this implementation linearizes the plant model every time the controller is called, providing a more accurate model, in contrast with LQR which utilizes a static gain matrix computed offline. The flap deflections used to obtain the tracking in Figures 5-7 are shown in Figures 8-10, respectively. Both MPC and LQR have similar shaped flap deflection profiles, which are vertically shifted from each other and reflect the shape of the α and β profiles. The flap deflections from MPC have more a more aggressive "back and fourth" behavior, relative to LQR. This behavior likely allows for the more aggressive tracking that MPC provides and may also be due to the controller frequencies, of which MPC is run ten times lower, relative to LQR. Even though MPC here does a better job of tracking the commands, compared to LQR, these results show that there can potentially be many different flap deflection profiles that result in similar tracking performance. Hence, it is the relative deflection of the flaps which allows the vehicle to successfully track a command profile, rather than the absolute size of the flap deflections, themselves.

Although α and β tracking is the primary concern for this blunt-body vehicle with flaps, rolling moments from the sides of the flaps, as well as yawing motion at a non-zero α, results in the vehicle deviating from its nominal bank angle of zero deg. The resulting banking of the vehicle is shown in Fig. 11 for the different command profiles and control algorithms. Here, bank is a state not being controlled by the vehicle, although it may be possible to simultaneously



(a) **(b)**

Fig. 5 Nominal command profile tracking for profile 1 with MPC and LQR: (a) angle of attack and (b) sideslip angle.



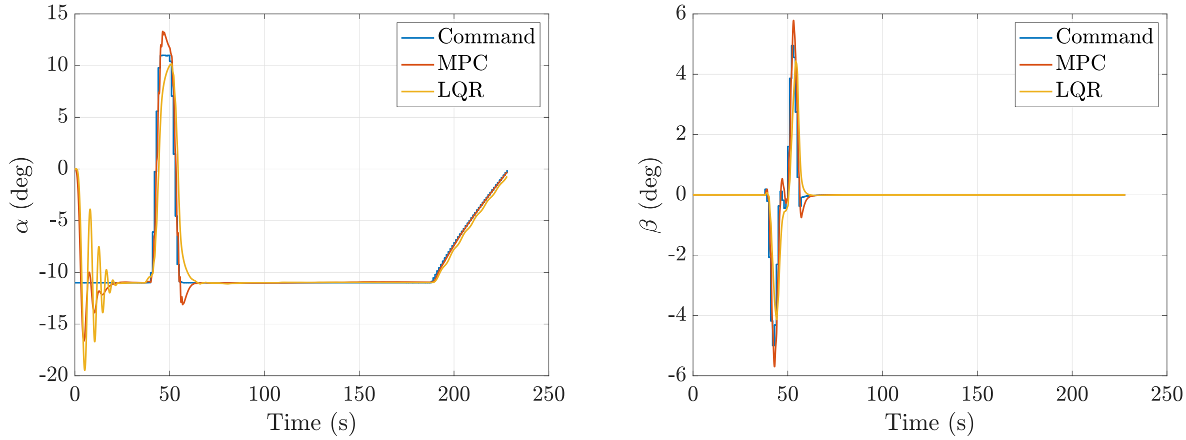
(a) **(b)**

Fig. 6 Nominal command profile tracking for profile 2 with MPC and LQR: (a) angle of attack and (b) sideslip angle.

Table 5 Integral Error for Nominal Case

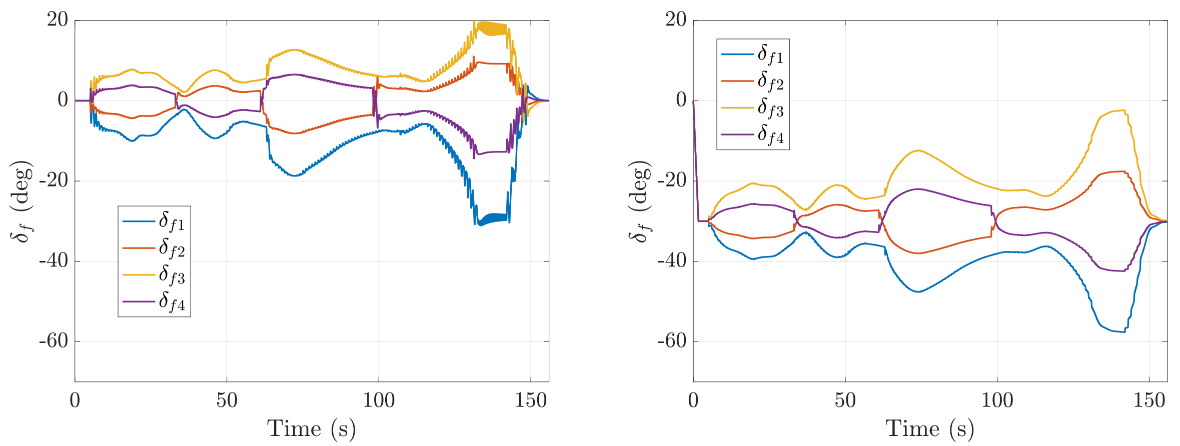
	e_α, deg-s	e_β, deg-s
Profile 1: MPC	9.826	8.886
Profile 1: LQR	30.756	24.184
Profile 2: MPC	19.971	2.866
Profile 2: LQR	51.419	10.463
Profile 3: MPC	38.958	6.970
Profile 3: LQR	74.351	14.208

control bank in addition to α and β by using the sides of the flaps. Even though the induced bank magnitude is small in



(a) **(b)**

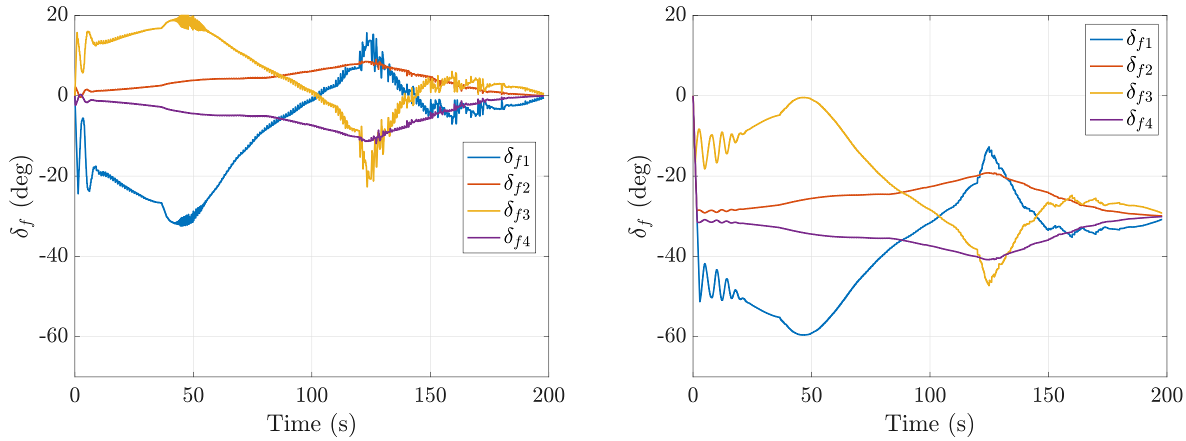
Fig. 7 Nominal command profile tracking for profile 3 with MPC and LQR: (a) angle of attack and (b) sideslip angle.



(a) **(b)**

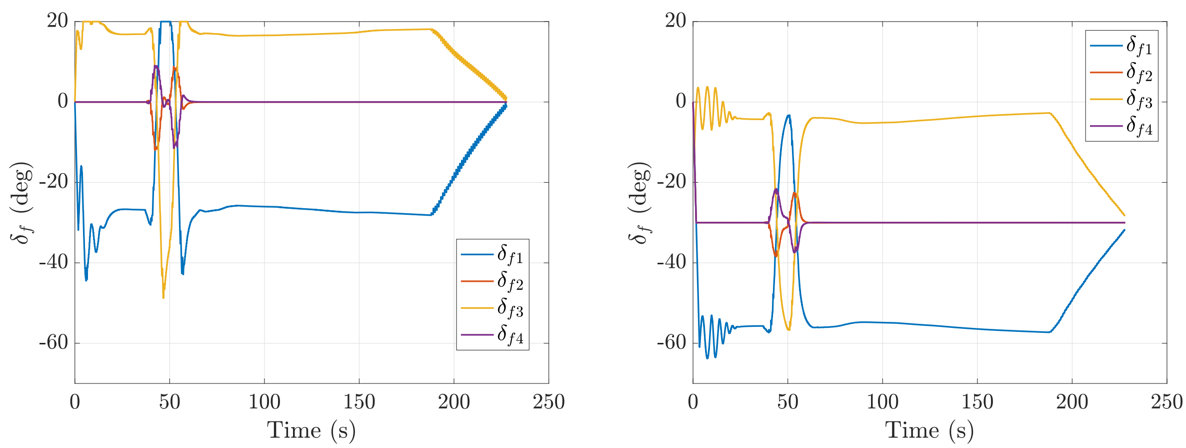
Fig. 8 Nominal flap deflections for command tracking of profile 1: (a) MPC and (b) LQR.

some cases, for profile 2, the induced bank magnitude exceeds 30 deg. The induced bank experienced by flap-based steering systems is a problematic for such systems and may result in an additional roll or bank mechanism, such as a rudder or RCS, needing to be included on a flight vehicle. Lastly, despite its increased tracking performance, the successive-linearization MPC requires a higher computational load than LQR (which only requires simple matrix addition and multiplication). Despite this, the mean time for this profile to obtain the commanded flap deflections once the MPC is well below the 0.2 s between subsequent calls to the controller. For example, for profile 1, the mean time to obtain commanded deflections was 0.0029 s.



(a) **(b)**

Fig. 9 Nominal flap deflections for command tracking of profile 2: (a) MPC and (b) LQR.



(a) **(b)**

Fig. 10 Nominal flap deflections for command tracking of profile 3: (a) MPC and (b) LQR.

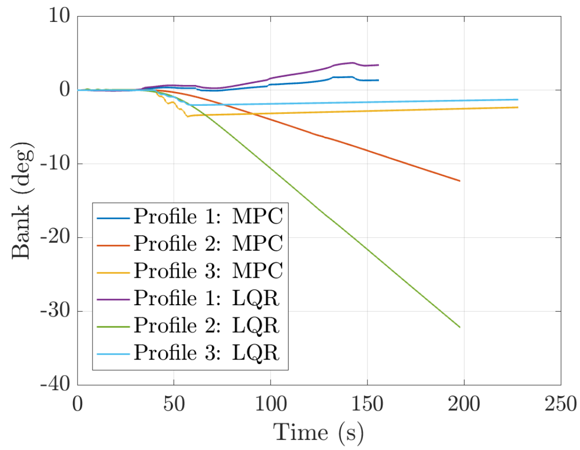


Fig. 11 Bank profile as a result of α and β tracking in nominal case.

B. Monte Carlo Performance: Effect of Control Algorithm

Uncertainty quantification through Monte Carlo simulation was used to assess the robustness of the flap-based steering system for both LQR and MPC. 1000 Monte Carlo Samples were used in each simulation, and dispersions were placed on the initial attitude and attitude rates of the vehicle, the moment coefficients, the CG position, and the moment of inertia tensor. A summary of the dispersions utilized are shown in Table 6. The dispersions are only included in the truth dynamics, and both controllers still use the same nominal linearization of the equations of motion. These Monte Carlo simulations were performed for each of the three command profiles shown previously for Configuration A with 3% area flaps. This uncertainty quantification is to assess controller performance, not entry guidance; in a given Monte Carlo simulation, all samples use the same command profile to test controller tracking. Results containing all 1000

Table 6 Monte Carlo Dispersions

Variable	Dispersion	Distribution
Axial CG position	±25 mm 3σ	Gaussian
Radial CG position	±0.55 mm 3σ	Gaussian
Radial CG angle	0:360 deg	Uniform
Moment of inertia matrix	±5% 3σ	Gaussian
Moment coefficient adder amplitude	±0.0046 3σ	Gaussian
Moment coefficient multiplier amplitude	±35% 3σ	Gaussian
Random fluctuations in moment coefficient adder and multiplier	-10%:10%	Uniform
Initial α and β	-2:2 deg	Uniform
Initial ω_y and ω_z	-5:5 deg/s	Uniform

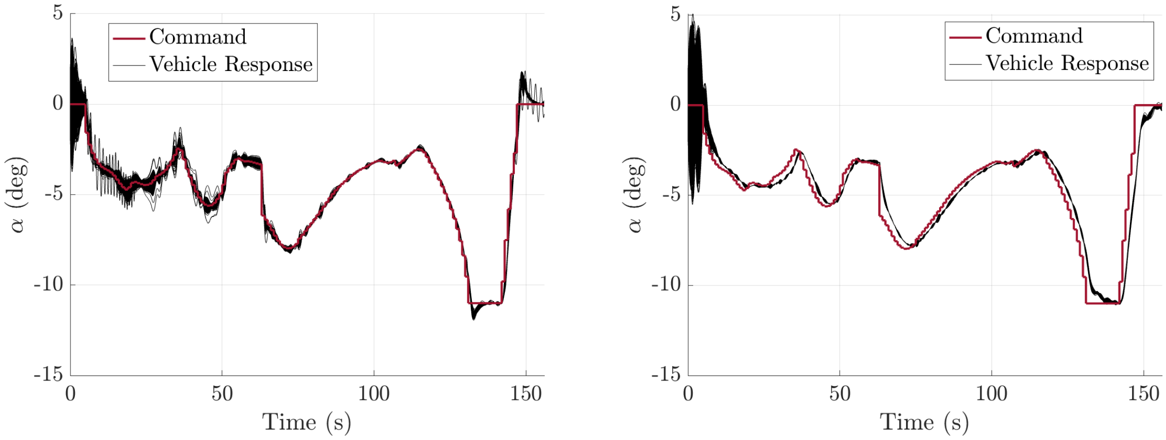
profiles using both controller options are shown in Figures 12 and 13, 14 and 15, and 16 and 17 for α and β, for each of the three command profiles. These results indicate both MPC and LQR are able to successfully track the command profiles under uncertainty, with a range of dispersions. Some dispersions, such as those in the moment coefficient multiplier, CG position, and moments of inertia, generally have little impact on controller performance and the resulting α-β tracking. Other dispersions, however, have larger impacts. Non-zero initial conditions result in an initial transient phase with large oscillations, as can be seen in most of the tracking results. These oscillations are further impacted by the low dynamic pressure available for the flaps at the start of each of these command profiles, which make it challenging to quickly achieve the desired commands and damp oscillations. The LQR results have a visibly larger oscillation magnitude, as well as a longer time to damp out the oscillations. Another significant dispersion is the moment coefficient adder, which can result in an offset in tracking with the MPC (see Figures 12a and 13a). This tracking offset decreases over the course of the trajectories due to corrections from integral error terms, which increase in their contribution over time. Lastly, although the moment coefficient multiplier typically does not significantly impact controller performance, profiles 1 and 3 for MPC have a case with noticeable oscillations as a result of a moment coefficient multiplier larger than the 3σ value. This multiplier is not an issue for LQR, which has state feedback to quickly null errors. Mean results for the mean and integral errors of tracking performance of the Monte Carlo simulations are provided in Table 7. Included for the integral errors are both the total integral error over the trajectory, as well as the steady integral error corresponding to the remaining results after 10 s, once the oscillations from the initial transient phase have been damped. The given results indicate that the MPC provides and improved controller performance under uncertainty, relative to LQR. The lower error for MPC is a result of both smaller oscillations during the transient phase and decreased tracking error during the steady phase. The higher steady integral error for LQR as well as the noticeable lag of LQR behind the command profile are evidence of the improved performance of MPC. Among all command profiles, MPC shows lower mean and integral errors, relative to LQR. With profile 1, for example, the MPC has mean integral errors 2.82 and 2.29 times lower than LQR for α and β, respectively. Generally, the standard deviation of the integral errors are also lower for MPC, relative to LQR, although LQR has the opposite trend, with LQR having a slightly lower standard deviation for integral errors. Standard deviation results are provided in Table 8. Despite the increased spread of results for MPC with profile 1, the highest integral error for MPC is still lower than the lowest integral error for LQR. As mentioned in the preceding section, LQR had to be made intentionally conservative to not have any trajectories that tumbled during any of the Monte Carlo samples for any of the considered command profiles. It is possible that larger and/or unmodeled dispersions, as well as more challenging command profiles could lead to further instabilities with LQR. Despite this, both controllers likely can provide suitable performance for a flight vehicle, especially if LQR is made conservative enough. Flight implementation of MPC may be more challenging, however, due to its higher computational cost on the limited processing power available from rad-hard flight computers.

Table 7 Mean Monte Carlo Results: Control Algorithm

	MPC: 1	LQR:1	MPC: 2	LQR: 2	MPC: 3	LQR: 3
Integral α error	29.898 deg-s	84.376 deg-s	55.731 deg-s	135.200 deg-s	104.753 deg-s	195.456 deg-s
Integral β error	23.617 deg-s	54.100 deg-s	17.077 deg-s	72.714 deg-s	27.932 deg-s	85.364 deg-s
Mean α error	0.191 deg	0.537 deg	0.279 deg	0.677 deg	0.456 deg	0.850 deg
Mean β error	0.150 deg	0.344 deg	0.086 deg	0.364 deg	0.123 deg	0.371 deg
Integral α error (steady)	25.923 deg-s	73.129 deg-s	28.270 deg-s	87.429 deg-s	60.852 deg-s	131.799 deg-s
Integral β error (steady)	20.459 deg-s	45.055 deg-s	6.631 deg-s	36.006 deg-s	16.960 deg-s	49.482 deg-s
Downrange difference	0.981 km	0.208 km	0.766 km	1.479 km	1.521 km	1.605 km
Crossrange difference	0.076 km	0.070 km	0.071 km	0.197 km	0.091 km	0.153 km
Terminal altitude	-	-	-	-	9.287 km	9.161 km

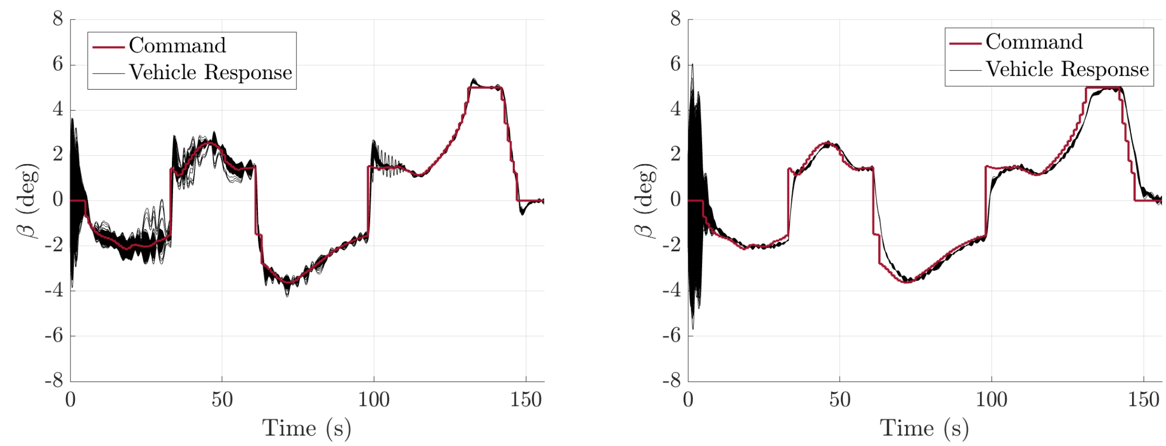
Table 8 Monte Carlo Standard Deviation Results: Control Algorithm

	MPC: 1	LQR:1	MPC: 2	LQR: 2	MPC: 3	LQR: 3
Integral α error	3.793 deg-s	3.256 deg-s	18.225 deg-s	35.543 deg-s	27.704 deg-s	44.210 deg-s
Integral β error	4.493 deg-s	3.243 deg-s	7.903 deg-s	27.405 deg-s	8.257 deg-s	27.642 deg-s



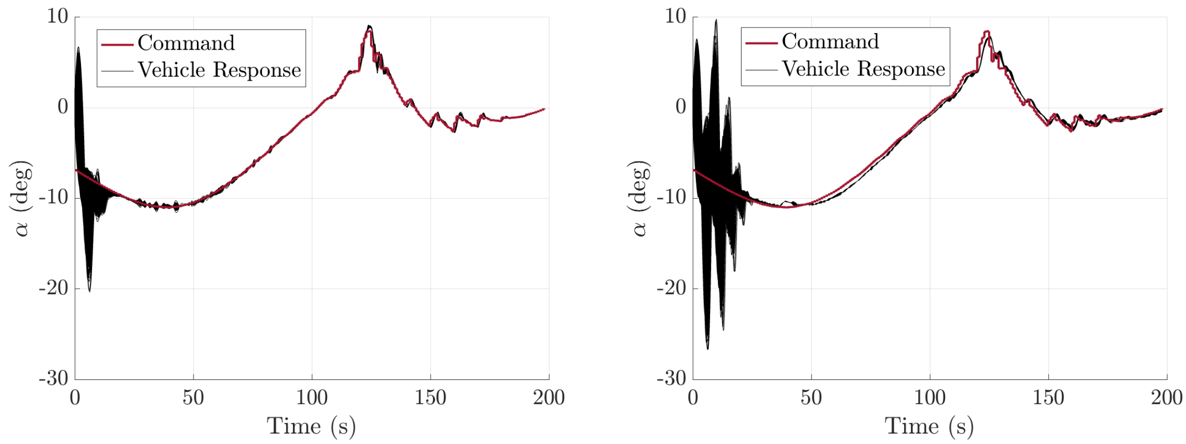
(a) **(b)**

Fig. 12 Monte Carlo results for angle of attack in profile 1: (a) MPC and (b) LQR.



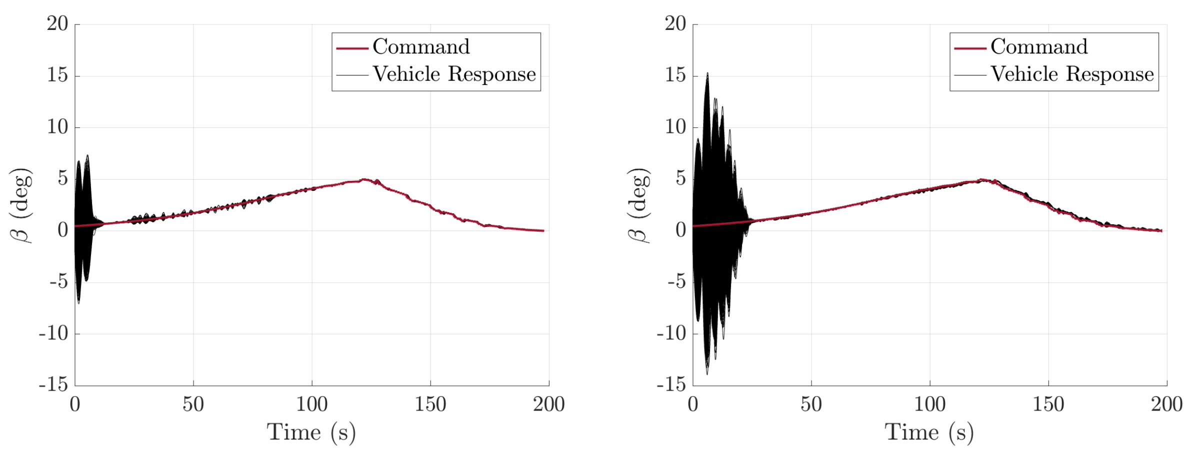
(a) **(b)**

Fig. 13 Monte Carlo results for sideslip angle in profile 1: (a) MPC and (b) LQR.



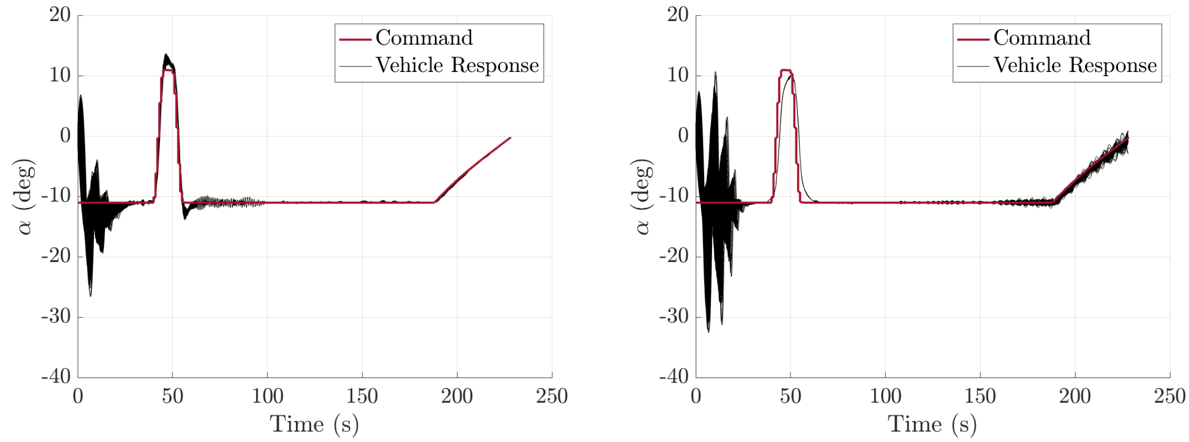
(a) **(b)**

Fig. 14 Monte Carlo results for angle of attack in profile 2: (a) MPC and (b) LQR.



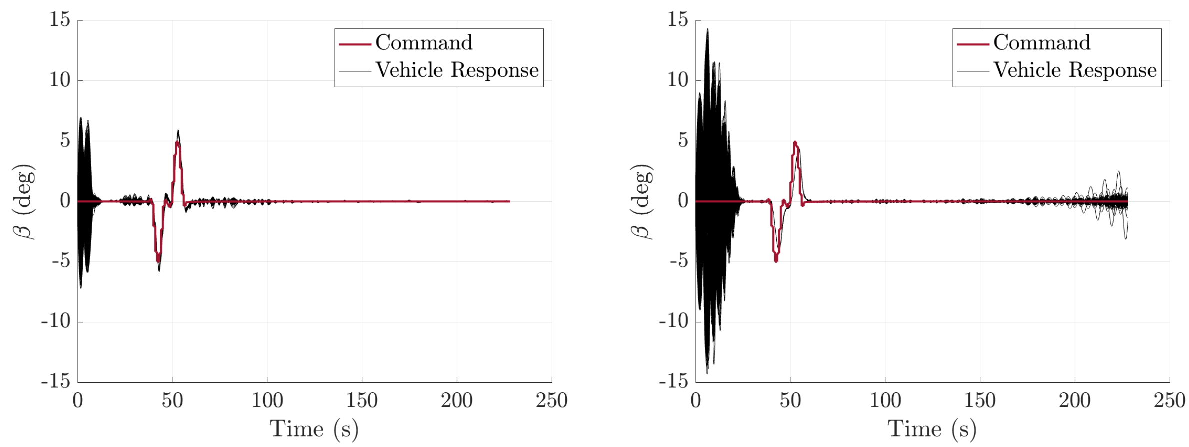
(a) **(b)**

Fig. 15 Monte Carlo results for sideslip angle in profile 2: (a) MPC and (b) LQR.



(a) (b)

Fig. 16 Monte Carlo results for angle of attack in profile 3: (a) MPC and (b) LQR.



(a) (b)

Fig. 17 Monte Carlo results for sideslip angle in profile 3: (a) MPC and (b) LQR.

Once complete, the α and β profiles from the Monte Carlo simulation can then be passed through Eq. (10)-(15) to simulate the translational motion during entry. Initial conditions with an initial altitude of 135 km, inertial velocity of 6.1 km/s, inertial flight-path angle of -15.5 deg, inertial azimuth angle of 90 deg, and latitude and longitude of zero deg are used and allow for an assessment of the effect of α and β tracking of the controllers on entry flight performance. Results are shown in Fig. 18 for the resulting entry trajectories corresponding to the MPC command tracking of profile 1. As with the controller tracking results, 1000 samples are considered, plotted here in an altitude versus velocity plot and a crossrange versus downrange plot. The results indicate that tracking accuracy of the MPC results in a tight clustering trajectories, despite the imperfect tracking of the MPC in the presence of dispersions. In particular, the maximum difference in terminal position for the trajectories are about 0.981 km and 0.076 km for downrange and crossrange, respectively. Note that these entry trajectories only to assess the effect of varied command tracking performance and do not consider dispersions in the translational motion, such as density or aerodynamic force coefficients. Additionally the trajectory calculations considered here assume the vehicle has RCS or other control system to hold the vehicle at zero bank; only the impacts α and β on the trajectory are considered, not any induced bank. Comparing the downrange and crossrange results in Table 7 for the different command profiles between MPC and LQR indicates both control algorithms result in similar performance. LQR often has larger differences in downrange and crossrange, relative to MPC, although for profile 1, LQR has a noticeably smaller downrange difference, relative to MPC. This trend for profile 1 is likely due to the larger standard deviation in tracking error for MPC, relative to LQR, as shown in Table 8. Additionally, profile 3, which seeks to maximize the final altitude during entry to increase margin in the descent timeline, has about a 0.13 km increase in altitude for MPC, relative to LQR. This indicates that the improved tracking performance of MPC has some small but present impacts on actual entry flight performance.

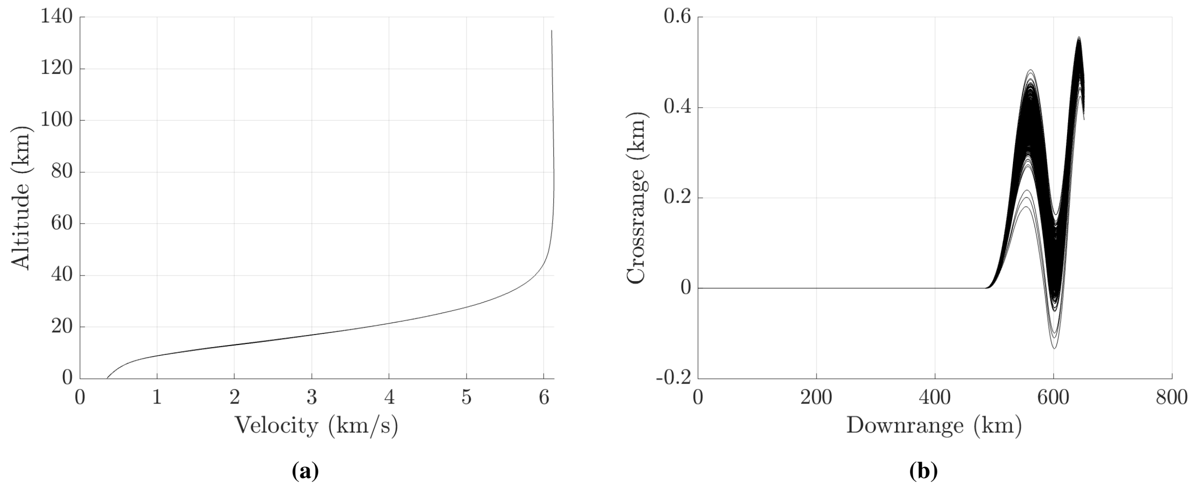


Fig. 18 Translational motion results from Monte Carlo α and β profiles: (a) altitude versus velocity and (b) crossrange versus downrange.

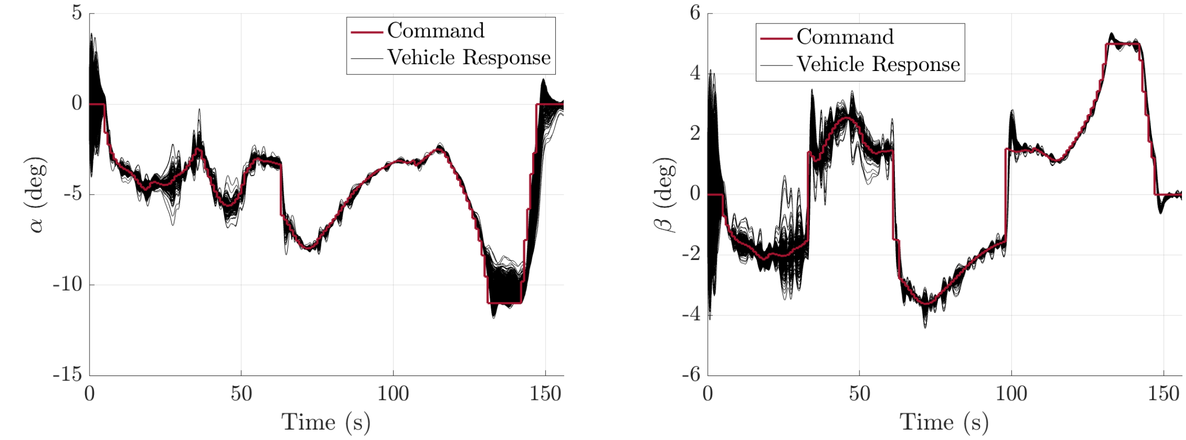
C. Monte Carlo Performance: Effect of Flap Configuration

Control system performance in the presence of uncertainty was also evaluated for various flap configurations. The different flap configurations include varied flap areas on Configuration A, as well as Configuration B, and Configuration C, and results were evaluated using the MPC algorithm with profile 1. Each case used the same tuning of the MPC, and the results in Table 9 indicate the tracking errors for α and β, as well as the difference in downrange and crossrange errors improve as flap area increases. In particular with 2% area flaps, the vehicle lacks sufficient control authority to trim to the desired commands, resulting in an increased error. This tracking performance is shown in Fig. 19. Provided the vehicle has flap areas which have the control authority available to trim to the desired commands, the vehicle can then successfully track the commands. This is evident for example by the 2% flaps having an integral error for α 1.48 higher than for the vehicle with 3% area flaps. There appear to be performance enhancements of 4% flaps over 3% flaps for α and β tracking due to the increased control authority available, however these improvements are marginal.

Table 9 Mean Monte Carlo Results: Flap Configuration

	A: 2%	A: 3%	A: 4%	B: 3%	C: 4%
Integral α error	44.213 deg-s	29.898 deg-s	28.718 deg-s	29.480 deg-s	29.737 deg-s
Integral β error	25.686 deg-s	23.617 deg-s	22.548 deg-s	25.283 deg-s	24.956 deg-s
Mean α error	0.282 deg	0.191 deg	0.183 deg	0.188 deg	0.189 deg
Mean β error	0.164 deg	0.150 deg	0.144 deg	0.161 deg	0.159 deg
Integral α error (steady)	39.562 deg-s	25.923 deg-s	25.102 deg-s	25.377 deg-s	25.722 deg-s
Integral β error (steady)	21.833 deg-s	20.459 deg-s	19.770 deg-s	21.838 deg-s	21.757 deg-s
Downrange difference	1.413 km	0.981 km	0.727 km	0.842 km	0.785 km
Crossrange difference	0.113 km	0.076 km	0.047 km	0.078 km	0.059 km

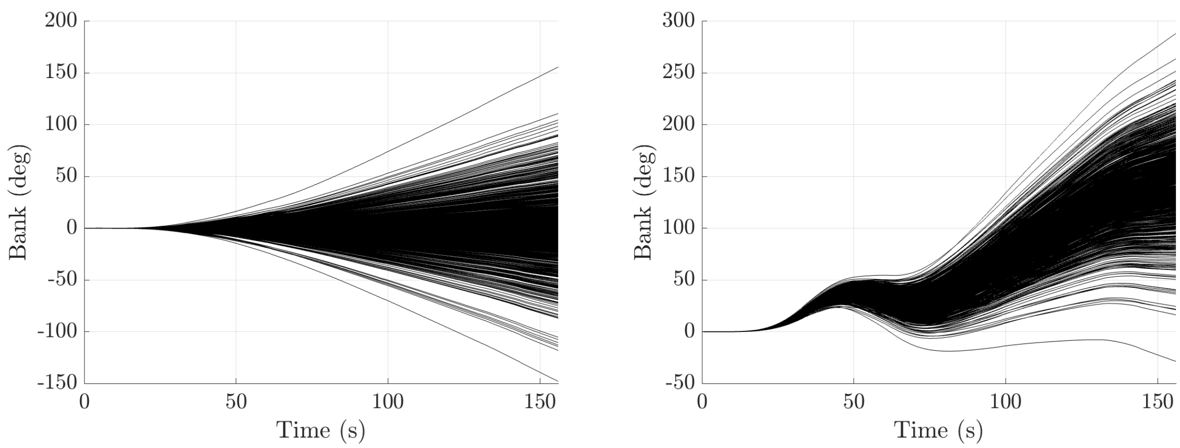
Comparing configurations A, B, and C to each other also indicates only marginal performance differences in α and



(a) **(b)**

Fig. 19 Monte Carlo results for 2% flaps: (a) angle of attack and (b) sideslip angle.

β tracking or the downrange and crossrange differences. This again indicates that this successive linearization MPC provides effective performance for various flap configurations. Provided that a given configuration can achieve the desired commands, there are many possible configurations that could be used. These results considering uncertainty are similar to the findings in Ref. [28], which considered nominal cases. One disadvantage of using Configuration C is the rolling moments from the sides of the flaps do not cancel out as much as for confiurations A and B. This results in the potential for a significant amount of bank during entry, even in nominal cases. Even though α and β command profiles are tracked successfully during Monte Carlo simulations, the bank angle remains an uncontrollable state. When the CG is located off the center line as a result of uncertainty, the normal and side aerodynamic forces on the vehicle induce an additional rolling moment. The results in Fig. 20 show that both configurations A and C have a significant amount of induced bank over the entry. This induced banking can be on the order of several hundred degrees and would be problematic on a flight vehicle unless mitigated against, although the mean terminal bank angle is -0.65 deg and -143.70 deg for configurations A and C, respectively. This again shows that a method for regulating the bank angle is necessary, even for vehicles using α-β steering and also indicates the induced bank problem may be more challenging on vehicles lacking a fully symmetric flap configuration.



(a) **(b)**

Fig. 20 Monte Carlo results for bank angle: (a) Configuration A and (b) Configuration B.

IV. Conclusions

In this study a successive linearization model predictive controller and linear-quadratic regulator for a Mars entry vehicle with flaps were designed and tested under uncertainty. Results indicated both controllers are able to track command profiles for the angle of attack and sideslip angle in nominal cases and when there is uncertainty such that the controller plant model differs from the truth dynamics. In the presence of uncertainty, the model predictive controller was found to have mean and integral errors on the order of two times lower than associated with the linear-quadratic regulator. The linear-quadratic regulator sometimes resulted in the entry vehicle tumbling, forcing a conservative tuning in order to provide successful tracking in all Monte Carlo samples. The lack of tumbling behavior for the model predictive controller is indicative of increased robustness relative to the linear-quadratic regulator, although using the model predictive controller requires increased computational resources. The model predictive controller is successful in being applied to several vehicle configurations in the presence of uncertainty without re-tuning. Provided a configuration has sufficient control authority to trim to desired angle of attack and sideslip angle commands, there is only a marginal differences in command tracking. Induced bank as a result of an off-axis center-of-gravity was found to be a potential drawback of systems modulating angle of attack and sideslip angle on blunt-body entry vehicles, as they may be forced to also carry a reaction control system for managing the bank angle. Induced bank on entry vehicles with an odd number of flaps may be further problematic, as the net aerodynamic moment from the sides of the flaps can result in an average of several hundred degrees of induced bank.

V. Acknowledgments

This work was supported by a NASA Space Technology Graduate Research Opportunity.

References

[1] Graves, C. A., and Harpold, J. C., "Apollo Experience Report: Mission Planning for Apollo Reentry," Tech. Rep. NASA-TN-D-6725, NASA Manned Spacecraft Center, 1972.

[2] Harpold, J. C., and Graves, C. A., "Shuttle Entry Guidance," Tech. Rep. NASA-TM-D-79949, NASA Johnson Space Center, 1979.

[3] Cichan, T., Norris, S. D., and Marshall, P., "Orion: EFT-1 Flight Test Results and EM-1/2 Status," *AIAA SPACE 2015 Conference and Exposition*, 2015. doi:10.2514/6.2015-4414.

[4] Mendeck, G. F., and McGrew, L. C., "Entry Guidance Design and Postflight Performance for 2011 Mars Science Laboratory Mission," *Journal of Spacecraft and Rockets*, Vol. 51, No. 4, 2014. doi:10.2514/1.A32737.

[5] Way, D. W., Dutta, S., Zumwalt, C., and Blette, D., "Assessment of the Mars 2020 Entry, Descent, and Landing Simulation," *AIAA SciTech Forum*, American Institute of Aeronautics and Astronautics Inc., 2022. doi:10.2514/6.2022-0421.

[6] Calhoun, P. C., and Queen, E. M., "Entry Vehicle Control System Design for the Mars Science Laboratory," *Journal of Spacecraft and Rockets*, Vol. 43, No. 2, 2006, pp. 324–329. doi:10.2514/1.19650.

[7] Cianciolo, A. D., Lugo, R. A., Korzun, A. M., Adam C. Slagle, A. C., Queen, E. M., Dillman, R. A., and Powell, R. W., "Low Lift-to-Drag Morphing Shape Design," *AIAA SciTech Forum*, American Institute of Aeronautics and Astronautics Inc., 2020. doi:10.2514/6.2020-1266.

[8] Korzun, A. M., Murphy, K. J., and Edquist, K. T., "Supersonic Aerodynamic Characteristics of Blunt Body Trim Tab Configurations," *Fluid Dynamics and Co-located Conferences*, American Institute of Aeronautics and Astronautics Inc., 2013. doi:10.2514/6.2013-2809.

[9] Cianciolo, A. D., and Powell, R. W., "Entry, Descent, and Landing Guidance and Control Approaches to Satisfy Mars Human Landing Mission Criteria," *Advances in the Astronautical Sciences*, Vol. 160, No. AAS 17-254, 2017.

[10] Steltzner, A. D., San Martin, M. A., Rivellini, T. P., Chen, A., and Kipp, D., "Mars Science Laboratory Entry, Descent, and Landing System Development Challenges," *Journal of Spacecraft and Rockets*, Vol. 51, No. 4, 2014. doi:10.2514/1.A32866.

[11] Dyakonov, A. A., Schoenenberger, M., Scallion, W. I., Van Norman, J. W., Novak, L. A., and Tang, C. Y., "Aerodynamic Interference Due to MSL Reaction Control System," *41st AIAA Thermophysics Conference*, American Institute of Aeronautics and Astronautics Inc., 2009. doi:10.2514/6.2009-3915.

[12] Schoenenberger, M., Van Norman, J., Rhode, M., and Paulson, J., "Characterization of Aerodynamic Interactions with the Mars Science Laboratory Reaction Control System Using Computation and Experiment," *51st AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics Inc., 2013. doi:10.2514/6.2013-971.

[13] Dyakonov, A. A., Glass, C. E., Desai, P. N., and Van Norman, J. W., "Analysis of Effectiveness of Phoenix Entry Reaction Control System," *Journal of Spacecraft and Rockets*, Vol. 48, No. 5, 2011. doi:10.2514/1.40965.

[14] Lugo, R. A., Powell, R. W., and Cianciolo, A. D., "Overview of a Generalized Numerical Predictor-Corrector Targeting Guidance with Application to Human-Scale Mars Entry, Descent, and Landing," *AIAA SciTech Forum*, American Institute of Aeronautics and Astronautics Inc., 2020. doi:10.2514/6.2020-0846.

[15] Lugo, R. A., Karlgaard, C. D., Powell, R. W., and Cianciolo, A. D., "Integrated Flush Air Data Sensing System Modeling for Planetary Entry Guidance with Direct Force Control," *AIAA SciTech Forum*, American Institute of Aeronautics and Astronautics Inc., 2019. doi:10.2514/6.2019-0663.

[16] Matz, D. A., and Cerimele, C. J., "Development of a Numeric Predictor-Corrector Aerocapture Guidance for Direct Force Control," *AIAA SciTech Forum*, American Institute of Aeronautics and Astronautics Inc., 2020. doi:10.2514/6.2020-0847.

[17] Lugo, R. A., Dutta, S., Matz, D., Johnson, B. J., Pensado, A. R., Roelke, E., Aguirre, J. T., and Powell, R., "Performance Analysis of SmallSat Aerocapture at Venus," American Institute of Aeronautics and Astronautics Inc., 2023. doi:10.2514/6.2023-0878.

[18] Engel, D. L., Dutta, S., and Putnam, Z. R., "Numerical Predictor-Corrector Based Guidance Scheme for Aero-Gravity Assist at Titan for Enceladus Missions," *AAS Spaceflight Mechanics Meeting*, American Astronautical Society, 2021. AAS 21-201.

[19] Deshmukh, R. G., Spencer, D. A., and Dutta, S., "Investigation of Direct Force Control for Aerocapture at Neptune," *Acta Astronautica*, Vol. 175, 2020, pp. 375–386. doi:10.1016/j.actaastro.2020.05.047.

[20] Engel, D. L., and Putnam, Z. R., "Optimal Range Capabilities for Low-Lift-to-Drag Ratio Entry Vehicles," *AAS Spaceflight Mechanics Meeting*, American Astronautical Society, 2023. AAS 23-181.

[21] Matz, D. A., Cerimele, C. J., and Sostaric, R. R., "Comparison of Aerocapture Performance Using Bank Control and Direct Force Control with Two Human-Scale Vehicles at Mars," *AIAA SciTech Forum*, American Institute of Aeronautics and Astronautics Inc., 2020. doi:10.2514/6.2020-1738.

[22] Engel, D. L., Skolnik, N. L., and Putnam, Z. R., "Comparison of Flap and Bank-Angle Steering for Mars Entry," *Journal of Spacecraft and Rockets*, Vol. 59, No. 5, 2022, pp. 1671–1685. doi:10.2514/1.A35253.

[23] Atkins, B. M., and Queen, E. M., "Internal Moving Mass Actuator Control for Mars Entry Guidance," *Journal of Spacecraft and Rockets*, Vol. 52, No. 5, 2015, pp. 1294–1310. doi:10.2514/1.A32970.

[24] Okolo, W. A., Margolis, B. W., D'Souza, S. N., and Barton, J. D., "Pterodactyl: Development and Comparison of Control Architectures for a Mechanically Deployed Entry Vehicle," *AIAA SciTech Forum*, American Institute of Aeronautics and Astronautics Inc., 2020. doi:10.2514/6.2020-1012.

[25] Margolis, B., Okolo, W., D'Souza, S. N., and Johnson, B. J., "Pterodactyl: Guidance and Control of a Symmetric Deployable Entry Vehicle using an Aerodynamic Control System," *AIAA Scitech 2021 Forum*, American Institute of Aeronautics and Astronautics Inc., 2021. doi:10.2514/6.2021-0764.

[26] Soest, W. R. V., Chu, Q. P., and Mulder, J. A., "Combined Feedback Linearization and Constrained Model Predictive Control for Entry Flight," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 2, 2006, pp. 427–434. doi:10.2514/1.14511.

[27] Recasens, J. J., Chu, Q. P., and Mulder, J. A., "Robust Model Predictive Control of a Feedback Linearized System for a Lifting-Body Re-Entry Vehicle," *AIAA Guidance, Navigation, and Control Conference*, Vol. 4, American Institute of Aeronautics and Astronautics Inc., 2005. doi:10.2514/6.2005-6147.

[28] Engel, D., and Putnam, Z. R., "Control of a Blunt Body Mars Entry Vehicle with Flaps Using Model Predictive Control," *AIAA SCITECH 2022 Forum*, American Institute of Aeronautics and Astronautics Inc., 2022. doi:10.2514/6.2022-1651.

[29] Dyakonov, A. A., Schoenenberger, M., and Van Norman, J. W., "Hypersonic and Supersonic Static Aerodynamics of Mars Science Laboratory Entry Vehicle," *43rd AIAA Thermophysics Conference*, American Institute of Aeronautics and Astronautics Inc., 2012. doi:10.2514/6.2012-2999.

[30] Etkin, B., *Dynamics of Atmospheric Flight*, 2^{nd} ed., John Wiley Sons, Inc., New York, 1972, Chap. 4.

[31] Edquist, K. T., Desai, P. N., and Schoenenberger, M., "Aerodynamics for Mars Phoenix Entry Capsule," *Journal of Spacecraft and Rockets*, Vol. 48, No. 5, 2011, pp. 713–726. doi:10.2514/1.46219.

[32] Lees, L., "Hypersonic Flow," *Journal of Spacecraft and Rockets*, Vol. 40, No. 5, 2004. doi:10.2514/2.6897.

[33] Anderson, J. D., *Hypersonic and High-Temperature gas Dynamics*, 2nd ed., American Institute of Aeronautics and Astronautics Inc., Reston, VA., 2006, Chap. 3.

[34] Lu, P., Cerimele, C. J., Tigges, M. A., and Matz, D. A., "Optimal Aerocapture Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 4, 2015, pp. 553–565. doi:10.2514/1.G000713.

[35] Justus, C., "A Mars Global Reference Atmospheric Model (MARS-GRAM) for Mission Planning and Analysis," *28th Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics Inc., 1990. doi:10.2514/6.1990-4.

[36] Camacho, E. F., and Bordons, C., *Model Predictive Control*, 2nd ed., Springer, 2007.

[37] Zhakatayev, A., Rakhim, B., Adiyatov, O., Baimyshev, A., and Varol, H. A., "Successive Linearization Based Model Predictive Control of Variable Stiffness Actuated Robots," *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 2017. doi:10.1109/AIM.2017.8014275.

[38] Wang, Y., and Boyd, S., "Fast Model Predictive Control Using Online Optimization," *IEEE Transactions on Control Systems Technology*, Vol. 18, No. 2, 2010, pp. 267–278. doi:10.1109/TCST.2009.2017934.

[39] MathWorks, "Model Predictive Control Toolbox," , 2021. Version 7.1.

[40] DeCarlo, R., *Linear Systems: A State Variable Approach with Numerical Implementation*, Prentice Hall, 1989, Vol. 2, pp. 215–216.

[41] Gill, P., Murray, W., and Wright, M., *Numerical Linear Algebra and Optimization*, Vol. 2, Addison Wesley, 1991.