A Framework for Approximating Perturbed Optimal Control Problems

Riley Link [†] and Ethan Ebbighausen [‡]

Project advisor: A.Alexanderian[§], J.Hart[¶]

Abstract. We consider trajectory optimal control problems in which parameter uncertainty limits the applicability of control trajectories computed prior to travel. Hence, efficient trajectory adjustment is needed to ensure successful travel. However, it is often prohibitive or impossible to recalculate the optimal control in-transit due to strict time constraints or limited onboard computing resources. Thus, we propose a framework for quick and accurate trajectory approximations by using post-optimality sensitivity information. This allows the reduction of uncertain parameter space and an instantaneous approximation of the new optimal controller while using sensitivity data computed and stored pretransit.

1. Introduction.

Optimal control problems pose physically difficult questions as optimization problems. Given a physical system, often constrained by differential equations, a control function is found such that an objective or cost function is minimized. This method is commonly used in aerospace applications such as the famous example of determining when to apply thrusters on the Apollo 11 mission [2].

In aerospace applications, the governing equations typically contain parameters that are subject to uncertainty. Examples include environmental parameters such as atmospheric density or physical parameters such as drag and lift coefficients. This parameter uncertainty restricts the usefulness of the precomputed solution to the optimal control problem. A small change in lift or drag forces due to, for example, heat stress causing deformation of a shuttle, can greatly influence how an aircraft maintains trajectory. For small changes, feedback controllers may maintain trajectory [11]. In more extreme cases, an optimal trajectory must be recomputed. We consider these in-transit computations.

Recalculation of the optimal path is computationally expensive, time-consuming, and lacks any guarantee of obtaining a favorable solution. If a new optimal path is required to avoid failure, a quick and efficient approximation of the optimal path is ideal [8, 3].

One approach to this computation is presented in [5]. Their adaptive flight control scheme for an F-18 aircraft focuses on in-flight failures that they simulate, and it helps dampen the affects of parameter uncertainties. We present a new method to approach this approximation.

Section 2 outlines our methods in generality, allowing for its application to a wide range of problems. Our first goal is to reduce the number of significant parameters using global sensitivity analysis. Then, rather than totally recomputing a solution, we focus on approximating a solution, generalizing a method presented in [1]. By storing and interpolating sensitivity information, which requires less memory due to parameter reduction, we may further reduce in-transit computation time without significant sacrifices of accuracy. These three steps, (i)

[†]Creighton University (rileylink03@gmail.com).
[‡]UNC-CH (etebbighausen@gmail.com).
[§]North Carolina State University (aalexan3@ncsu.edu).
[¶]Sandia National Laboratories (joshart@sandia.gov).

Copyright © SIAM
Unauthorized reproduction of this article is prohibited

parameter reduction, (ii) sensitivity information interpolation, and (iii) approximating using a general method from [1], individually constitute ways to improve computation time given uncertainty, and we consider the efficacy of each. Hence, our approach is to adapt the optimal trajectory dynamically based on sensitivity information. Typical feedback control which relies on reactive adjustments, works well for small deviations but may struggle with large deviations, and so this new approach can handle those larger perturbations given its adjustments made using precomputed sensitivity information.

Thus, we want to emphasize that our method is intended to complement, rather than replace, existing feedback control techniques, which remain well-suited for handling small deviations. Additionally, this approach represents a singular strategy within optimal control, and its applicability should be evaluated based on the specific problem at hand. Users should consider its usefulness in conjunction with other established methods to determine the most effective solution for their particular application.

Section 3 applies this solution method to a model Space Shuttle Trajectory problem from [4]. There we solve for the optimal control angle which maximizes the longitude traveled in a model space shuttle's reentry. This provides an in-depth example that demonstrates the strengths and limitations of each part of the procedure.

2. Generalized Framework for Approximation of the Optimal Control.

As noted above, our framework breaks down into 3 main steps.

(i) Conduct global sensitivity analysis to understand which parameters we may remove from consideration.
(ii) Interpolate post-optimality local sensitivities to store onboard.
(iii) Use the interpolated data to make quick and accurate controller approximations for perturbed parameters.

In particular, we perturb important parameters and solve this new optimal control problem by relying on a method of using the sensitivity information to construct a differential equation whose solution is the new optimal control.

2.1. Problem Formulation.

We consider a general trajectory control governed by a system of ordinary differential equations of the form,

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}, u; \boldsymbol{p}) & t \in (0, T), \\ \boldsymbol{x}(0) = \boldsymbol{x}_0, \end{cases} \tag{2.1}$$

where t is time, with final time $T > 0$, $\boldsymbol{x} : [0, T] \to \mathbb{R}^n$ is the state vector with initial condition $\boldsymbol{x}_0$, $u : [0, T] \to \mathbb{R}$ is the trajectory controller, and $\boldsymbol{p}$ is the parameter vector containing P parameters. To normalize parameter changes, we non-dimensionalize parameters to

$$p_i = (1 + \beta_0 \theta_i)\bar{p}_i, \quad \text{for } i = 1, 2, \ldots, P \tag{2.2}$$

where $\bar{p}_i$ is the nominal value of the parameter, and θ_i is the reparameterization. The factor β_0 determines the amount of change taken in these parameters. For example, if $\beta_0 = 0.1$ this simulates a maximum change of 10% from nominal parameter values.

To find the optimal control $u^*(t)$, we minimize a cost function,

$$\min_u J(\boldsymbol{x}(t), u(t), t; \boldsymbol{\theta}). \tag{2.3}$$

and impose inequality constraints and terminal state equality constraints.

Some optimal control solvers strictly enforce these constraints through spectral methods, but constraints may also be imposed through penalty terms in the cost function J. The latter method is less accurate, but is useful in situations with low computation power such as the motivating in-transit case.

To solve this problem computationally, we discretize the time interval $[0, T]$ as $t_0, t_1, ..., t_N$ for a chosen $N \in \mathbb{N}$. We also linearly discretize the controller by storing the function $\boldsymbol{u}_N(t)$ as a vector $[u_0, u_1, \ldots, u_N]$ where

$$\boldsymbol{u}_N(t) = \sum_{i=0}^{N} u_i \phi_i(t) \tag{2.4}$$

for basis functions $\phi_i(t)$ given by

$$\phi_i(t) = \begin{cases} \frac{t-t_{i-1}}{t_i-t_{i-1}} & t \in [t_{i-1}, t_i] \\ \frac{t_{i+1}-t}{t_{i+1}-t_i} & t \in [t_i, t_{i+1}] \\ 0 & \text{otherwise.} \end{cases} \tag{2.5}$$

2.2. Dimension Reduction. The various uncertain parameters may influence the problem at highly different degrees. Sensitivity information attempts to capture this information. In the present work, this sensitivity analysis helps us locate those parameters most important to the problem and thus we are locating parameters which have less influence on the optimal control. Fixing unimportant parameters at their nominal values greatly reduces the complexity of the problem when considering possible parameter changes and uncertainty estimation.

We consider the parameters $(\theta_1, ..., \theta_{N_p}) \in \Theta = \Theta_1 \times \Theta_2 \times \cdots \times \Theta_P \subset \mathbb{R}^P$, for a space of parameters Θ on which we define the probability law

$$\mu(d\boldsymbol{\theta}) = \prod_{j=1}^{N_p} \pi_j(\theta_j) d\theta_j \tag{2.6}$$

for the parameters. We wish to compute Sobol indices, which measure the amount of output variance that is caused by variance in input parameters [10]. It is important to note that Sobol indices come in three different forms, main, interaction, and total. The main Sobol index is simply the ratio of the variance of a single input, say the parameter θ_j, by the total variance of the output, in our case the optimal controller $\boldsymbol{u}$. Thus, this main index measures the direct effects of the input variable and does not account for any interaction effects with other variables. These interaction indices capture how the combined variations of input parameters like θ_i and θ_j affect the model output. The total Sobol index captures the contribution of an input variable to the output variance, accounting for both its main effects and all interaction effects involving the input variable.

These total Sobol indices provide a clear ranking of how important input parameters are. Total Sobol indices range from 0 to 1, yet will not sum to 1 given the overlapping of interaction terms. These indices would allow the most unimportant parameters to be reduced to their nominal values and ignored for perturbation analysis. However, computing them

can be expensive. Therefore, we bound Sobol indices using derivative-based global sensitivity measures (DGSMs) as described in [6].

Note that, if Sobol indices could be precisely computed, the main Sobol indices would provide a more valid reduction of parameter space. This is because, when setting a reduced parameter to its nominal value, we are then ignoring all interaction effects, leaving only the direct effects captured by the main Sobol indices. As such, the main Sobol indices would better describe how important each parameter is and whether it should be retained in the model.

However, since we are working with upper bounds on Sobol indices, the total index also provides a valid measure of whether or not we should retain or discard a parameter. The total index accounts for both direct and interaction effects, providing a comprehensive measure of a parameter's maximum possible contribution to output variance. For example, given a input parameter with a small upper bound on its total index, we can then safely say that even its maximum possible contribution (direct and interaction effects) to the output variance is negligible.

Nevertheless, the use of upper bounds comes with limitations. While we can confidently identify parameters with negligible total contributions, if given a large upper bound, i.e. an upper bound above 1, we cannot infer the relative importance of the parameter and its direct versus interaction effects. Thus, we leave these parameters with unclear significance to be further analyzed through the perturbation-based methods in our framework.

The structure of computing the upper bounds on total Sobol indices is as follows. Let $\boldsymbol{u} : \Theta \to \mathbb{R}^{N+1}$ be a vector-valued function. The derivative-based global sensitivity measure of $\boldsymbol{u}$ with respect to parameter j is defined as

$$\mathcal{N}_j(u) = \sum_{i=1}^{N} \int_{\Theta} \left(\frac{\partial u_i}{\partial \theta_j} \right)^2 \mu(d\boldsymbol{\theta}). \tag{2.7}$$

These DGSMs may be use to generate an upper bound on the Sobol indices. In the simple case of a uniform random variable, these connect to the Sobol indices via the following theorem. More complex and general versions may be found in [6].

Theorem 2.1. *Let $\boldsymbol{u}$ be a vector-valued random variable and let Γ be its covariance matrix. Assume the inputs θ_i are independent and identically distributed uniform random variables $\mathcal{U}(a, b)$. Then,*

$$\mathbb{S}_j^{tot}(\boldsymbol{u}) \leq c_j^p \frac{\mathcal{N}_j(\boldsymbol{u})}{Tr(\boldsymbol{\Gamma})} \tag{2.8}$$

where $c_j^p = \frac{(b-a)^2}{\pi^2}$.

Using this inequality to bound the Sobol indices from above, the DGSMs can show when parameters are unimportant and may be ignored for small perturbations or totally removed from perturbation analysis. This reduces the computational expense of solving updated problems.

Numerically, the DGSMs may be computed via a Monte Carlo method and the terms $\frac{\partial u_i}{\partial \theta_j}$ may computed via Hyper-Differential Sensitivity Analysis as seen in Section 2.2.1. In particular, a quasi-Monte Carlo (QMC) method employing low-discrepancy sequences to sample

points, as presented in [9], is shown to converge faster than a standard Monte Carlo method when computing DGSMs. It is also possible to vectorize the process by interchanging the QMC summation and DGSM summation. With this interchange we have,

$$\mathcal{N}_j(\boldsymbol{u}) \approx \frac{1}{M}\sum_{k=1}^{M}\sum_{i=1}^{N}\left(\frac{\partial u_i^*}{\partial \theta_{j,k}}\right)^2. \tag{2.9}$$

Using equation (2.9), upper bounds on Sobol indices for each parameter θ_i can be calculated. By ignoring the unimportant parameters the complexity of the optimal control problem is reduced and when coupled with Hyper-Differential Sensitivity Analysis, the computational cost for this reduction is feasible.

2.2.1. Hyper-Differential Sensitivity Analysis. Hyper-Differential Sensitivity Analysis (HDSA) is a local sensitivity method that considers the change in the solution of an optimal control problem with respect to perturbations of parameters appearing in the model [7]. It encapsulates this change by computing the derivatives $\frac{\partial \boldsymbol{u}^*}{\partial \theta_i}$, where $\boldsymbol{u}^*$ is the discretized optimal solution and θ_i are relevant parameters. It is possible to use numerical differentiation to compute these, but it diminishes the error to use implicit differentiation as in [7].

Considering $\nabla_u J(u^*(\theta), \theta) = 0$, implicit differentiation with respect to θ gives

$$\left(\frac{\partial^2 J}{\partial u_i \partial u_j}\right)\left(\frac{\partial u_j^*}{\partial \theta_k}\right) + \left(\frac{\partial^2 J}{\partial u_i \partial \theta_k}\right) = 0, \quad \text{for } i, j = 0, \ldots, N \text{ and } k = 1, \ldots, P \tag{2.10}$$

or

$$\boldsymbol{D} = -\boldsymbol{H}^{-1}\boldsymbol{B} \tag{2.11}$$

where $\boldsymbol{H} = (\frac{\partial^2 J}{\partial u_i \partial u_j})_{ij}$ is the Hessian of J with respect to u, which is assumed to be invertible at the optimal solution, $\boldsymbol{B} = (\frac{\partial^2 J}{\partial u_i \partial \theta_j})_{ij}$ is a matrix of mixed partials with respect to the controller $\boldsymbol{u}$ and parameters $\boldsymbol{\theta}$, and $\boldsymbol{D}$ is the matrix of sensitivities $\frac{\partial \boldsymbol{u}_i^*}{\partial \theta_k}$. Since $\boldsymbol{u}$ is discretized, the $\boldsymbol{D}$ matrix consists of local sensitivities of the discretized $\boldsymbol{u}$ with respect to each parameter $\boldsymbol{\theta}$. The entries of $\boldsymbol{D}$ are used for calculating DGSMs and for approximating $\boldsymbol{u}^*$ as in the next section and the Hessian and mixed partials matrix may themselves be computed by standard numerical differentiation techniques such as complex step differentiation or sensitivity equations.

2.2.2. Sensitivity Equations. The calculation of the gradient $\frac{\partial J}{\partial u_i}$ of the cost function $J(\boldsymbol{x}(t), \boldsymbol{u}(t), t; \boldsymbol{\theta})$, is important in both HDSA and the convergence and precision of the optimization process. Thus, employing an exact calculation rather than a numerical approximation is crucial. We choose to achieve this through the use of sensitivity equations.

Given a system of ODEs with state variables x_i and a discretized controller $\boldsymbol{u}$, sensitivity of the state variables with respect to the controller can be calculated via sensitivity equations. This is done by treating the discretized controller $\boldsymbol{u}$ as a parameter in the ODEs such that

$$\begin{aligned} \boldsymbol{x}'(t; \boldsymbol{u}) &= \boldsymbol{f}(t, \boldsymbol{x}(t; \boldsymbol{u}); \boldsymbol{u}) \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0(\boldsymbol{u}). \end{aligned} \tag{2.12}$$

By treating the controller this way it is possible to differentiate with respect to $\boldsymbol{u}$

$$\frac{\partial x_i'}{\partial u_j} = \frac{\partial \boldsymbol{f}}{\partial x_i}\frac{\partial x_i}{\partial u_j} + \frac{\partial \boldsymbol{f}}{\partial u_j}. \tag{2.13}$$

The derivatives $\frac{\partial x_i}{\partial u_j}$ can be used to calculate a more accurate gradient, $\frac{\partial J}{\partial u_j}$, when differentiating the cost function directly. Appendix C shows calculations of sensitivity equations for the example problem shown in section 3.

2.3. Approximating the Optimal Control Problem. In case of parameter change during flight (either an actual change occurs or parameters are remeasured at better accuracy), it is necessary to shift the precomputed trajectory to match the physical system, but the time pressure and computational restrictions of in-flight computation impedes the ability to recompute an optimal control, especially in the case of a highly non-convex objective function. Instead, we shift to approximation of the optimal control. In particular, we focus on a method relying on the sensitivity information described in the previous section.

2.3.1. Generalizing an Approximation Method. From subsection 2.2.1, we obtain the matrix $\boldsymbol{D} = (\frac{\partial u_i^*}{\partial \theta_j})_{ij}$. For small changes in $\boldsymbol{\theta}$ or changes with respect to less-important parameters, we may apply a first-order Taylor series approximation,

$$\boldsymbol{u}^*(\boldsymbol{\theta}_1) = \boldsymbol{u}^*(\boldsymbol{\theta}_0) + \boldsymbol{D}(\boldsymbol{u}^*(\boldsymbol{\theta}_0), \boldsymbol{\theta}_0)(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0) + \mathcal{O}(||\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0||_2), \tag{2.14}$$

for nominal parameters $\boldsymbol{\theta}_0$ and perturbed parameters $\boldsymbol{\theta}_1$. This approximation is already relatively accurate. Computing the derivative information $\boldsymbol{D}$ offline and storing it in vehicle systems allows for a rapid approximation. Higher order approximations may greatly improve accuracy if derivatives are highly accurate as well.

The inaccuracy of the linear approximation depends on the curvature of the optimal controller $\boldsymbol{u}^*$ with respect to $\boldsymbol{\theta}$. This problem may be addressed by stepping over $\boldsymbol{\theta}$ as presented in [1]. This process first parameterizes the step from $\boldsymbol{\theta}_0$ to $\boldsymbol{\theta}_1$ as $\boldsymbol{\theta}(t) = \boldsymbol{\theta}_0 + t(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0)$. Then,

$$\begin{aligned} \frac{\partial \boldsymbol{u}^*}{\partial t} &= \frac{\partial \boldsymbol{u}^*}{\partial \boldsymbol{\theta}}\frac{\partial \boldsymbol{\theta}}{\partial t} \\ &= \boldsymbol{D}(\boldsymbol{u}^*(\boldsymbol{\theta}(t)), \boldsymbol{\theta}(t))(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0). \end{aligned} \tag{2.15}$$

This creates a differential equation for $\boldsymbol{u}$ which may be evaluated to obtain $\boldsymbol{u}^*(\boldsymbol{\theta}_1) = \boldsymbol{u}^*(\boldsymbol{\theta}(1))$. In [1], this is presented via a time-stepping, or modified forward-Euler method. We pick some $M \in \mathbb{N}$, set $h = \frac{1}{M}$ and $t_m = mh$, and apply the recursion

$$\boldsymbol{u}_{m+1}^* = \boldsymbol{u}_m^* + h\boldsymbol{D}(\boldsymbol{u}_m^*, \boldsymbol{\theta}(t_m))(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0) \tag{2.16}$$

so $\boldsymbol{u}^*(\boldsymbol{\theta_1}) \approx \boldsymbol{u}_M^*$. Since this method does require re-computation of the derivative at various points, it trades accuracy for some computational speed. This is well-known to have a local truncation error proportional to the square of the step size, or $\mathcal{O}(||\boldsymbol{\theta_1} - \boldsymbol{\theta_0}||_2^2)$. In the case of in transit changes or large perturbations, this accuracy becomes extremely important. However, due to compounding error in calculating derivatives and steps, the number of step sizes that

improve the approximation is limited by the system involved. Any number of other numerical differential equation methods may be applied instead of a forward-Euler method, such as higher order Runge-Kutta methods which may help control when the ODE acts stiff at the cost of computation time.

2.3.2. Interpolation of the Key Jacobian. In n^{th}-order Runge Kutta, the number of calculations of the derivative is on the scale of $\mathcal{O}(nMNP)$ for M the number of steps, N the discretization number, and P the number of parameters. Thus, the computation of these derivatives throttles our increase of the order for accuracy. Depending on the curvature of the objective, sacrificing some accuracy of the derivatives to improve the order of the approximation may improve the approximation of the controller.

In particular, we focus on approximating the key Jacobian, $\boldsymbol{D} = (\frac{\partial u_i^*}{\partial \theta_j})_{ij}$. We precompute the sensitivity matrix, $\boldsymbol{D}$, on a mesh of values in the hypercube Θ, and store grid-call interpolation functions to approximate the derivative at given points. While this can be taxing in terms of computational power prior to an experiment, it drastically improves computation time in-situ. In some cases it improves accuracy due to the compounding error as the number of ODE steps increases. We will refer to the ODE method using these derivatives for inputs as the Interpolated Step method in the example shown in section 3.

Ironically, an increased number of steps tends to outweigh higher order methods in terms of accuracy in practice. While several other ODE methods, such as Runge Kutta 3, Adams-Bashforth multistep, Adams-Moulton multistep, and Gaussian Quadrature, were considered, we will focus on the forward Euler method moving forward and show the effectiveness of the interpolated step in section 3. Additionally, while it is not implemented in this paper, storing higher-order derivatives to approximate the Jacobian $\frac{\partial u_i}{\partial \theta_j}$ could similarly improve computational efficiency without sacrificing much accuracy.

3. Numerical Results. To demonstrate the effectiveness and efficacy of our method, we provide an example of a 2 degree-of-freedom space shuttle re-entry problem.

Less complex problems, such as the classical Zermelo's navigation problem, were considered but omitted as the small parameter space and ease of solution did not warrant parameter reduction or approximation, and did not display noticeable change with their application. However, with more complex systems, such as the 2 DOF space shuttle re-entry, in-transit parameter shifts can cause catastrophic results and thus require approximations to the optimal control to be made. Furthermore, these problems contain many physical parameters and thus demonstrate the effectiveness of our parameter reduction method and approximation technique.

To achieve our proposed method, we implement this numerically in MATLAB. Algorithm 3.1 provides a summary of the proposed approach.

Algorithm 3.1 Computation of Mid-Trajectory Approximations

1: Solve the nominal trajectory problem ($\boldsymbol{\theta}_0 = \mathbf{0}$) for states $\boldsymbol{x}^*$ and optimal controller $\boldsymbol{u}^*$
2: Subdivide the hypercube Θ into m equally distributed samples. At each sample compute the optimal trajectory and take local hyper-differential sensitivities, $\boldsymbol{D}$ (2.11), about the new trajectory. These sensitivities form the basis used for interpolation
3: Compute DGSMs (2.7) and reduce the parameter space using the Sobol Index upper bounds (2.8)
4: Organize the sensitivities computed across all of Θ into a key Jacobian of sensitivities for only the important parameters.
5: Assume a random parameter change, $\boldsymbol{\theta}_1$, at t_0 such as $\theta_j \in \mathcal{U}([-1, 1])$
6: Using (2.16), step across $\boldsymbol{\theta}_0 \to \boldsymbol{\theta}_1$ and interpolate the key Jacobian to calculate the sensitivities at each step in order to return the approximate trajectory $\boldsymbol{u}_{IS}$

3.1. Space Shuttle Problem. We consider a 2 degree-of-freedom space shuttle re-entry problem posed in [4]. With 7 parameters to consider, this gives a sufficient amount of complexity to examine the method described above. In particular, we note that the parameter reductions and Jacobian interpolation do not substantially affect the accuracy of the approximated optimal control (measured through norm difference of the controls and difference of the associated costs). Further, we display the ability of this approximation method to mimic the new optimal solution while noting some instability in this approximation method.

3.1.1. Problem Formulation. The motion of the space shuttle is defined by the following set of equations:

$$\dot{\boldsymbol{x}}(t) = \begin{cases} \dot{h}(t) &= v(t)\sin(\gamma(t)) \\ \dot{\phi}(t) &= \frac{v(t)}{r}\cos(\gamma(t)) \\ \dot{v}(t) &= -\frac{D}{m} - g\sin(\gamma(t)) \\ \dot{\gamma}(t) &= \frac{L}{mv(t)} + \cos(\gamma(t))\left(\frac{v(t)}{r} - \frac{g}{v(t)}\right) \end{cases} \tag{3.1}$$

h	altitude (ft)
ϕ	longitude (deg)
v	velocity (ft/sec)
γ	flight path angle (rad)

The initial conditions, terminal conditions, and constraints of the system are as follows:

$$\boldsymbol{x}(0) = \begin{cases} h(0) &= 260000 \\ v(0) &= 25600 \\ \gamma(0) &= -\frac{\pi}{180} \end{cases} \tag{3.2}$$

$$\boldsymbol{x}(T) = \begin{cases} h(T) &= 80000 \\ v(T) &= 2500 \\ \gamma(T) &= -5\frac{\pi}{180} \end{cases} \tag{3.3}$$

$$\boldsymbol{\eta}(t, \boldsymbol{x}, u; \boldsymbol{\theta}) = \begin{cases} h(t) &\geq 0 \\ v(t) &\geq 1 \\ |\gamma(t)| &\leq 89\frac{\pi}{180} \\ |u(t)| &\leq \frac{\pi}{2} \end{cases} \tag{3.4}$$

Appendix A contains the specific aerodynamic and atmospheric forces on the shuttle.

The goal of the space shuttle re-entry is to maximize longitudinal distance, $\phi(t) := x_2(t)$, across $t \in (0, T)$. This is achieved by finding the optimal angle of attack $u : [0, T] \to \mathbb{R}$ such that the cost function $J(\boldsymbol{x}(t), u(t), t; \boldsymbol{\theta})$ is minimized. Thus let the optimal control problem be formulated as,

$$\min_u J(\boldsymbol{x}(t), u(t), t; \boldsymbol{\theta}) = \min_u \left[-x_2(T) + S(\boldsymbol{x}(t))\right] \tag{3.5}$$

where $S(\boldsymbol{x}(t))$ is a function of penalty terms enforcing terminal conditions (3.3) and state constraints (3.4). The full cost function is shown in appendix B. In particular, we take a weighted penalty for each state and exponentiate to ensure the cost is twice-differentiable.

We analyze seven parameters of interest that are subject to change during flight or to uncertainty in measurement:

$$\boldsymbol{p} = [m, \rho_0, a_0, a_1, b_0, b_1, b_2] \tag{3.6}$$

where m is the mass of the space shuttle, ρ_0 the initial atmospheric density, and a_i and b_i are dimensionless parameters found in the coefficients of lift and drag. As shown in subsection 2.1, these parameters are nondimensionalized to $\boldsymbol{\theta} \in \Theta$ according to equation (2.2).

To solve the problem computationally, we linearly discretize the controller according to equation (2.4). Thus the discretized controller is $\boldsymbol{u} : [0, T] \to \mathbb{R}^{N+1}$. A discretization value of $N = 10$ or $N = 20$ proved sufficient for computations with appropriate interpolation of the controller when necessary.

Sensitivity equations were used for efficient computation of the gradient, $\frac{\partial J}{\partial u_i}$, and the full computations can be seen in appendix C. As mentioned previously, the calculation of the gradient through this method also increases the accuracy of the local sensitivities, $\frac{\partial u_i^*}{\partial \theta_j}$, which are found through HDSA as outlined in subsection 2.2.1.

3.1.2. Parameter Space Reduction. We first assume that all nondimensionalied parameters, θ_j, follow a uniform distribution, $\mathcal{U}([-1, 1])$, which allows for a maximum 10% change in the parameters. This assumption is made due to the absence of further information on the atmospheric space of interest or information on the non-dimensional drag and lift coefficients.

We computed DGSM's and upper bounds on Sobol Indices according to equations (2.9) and (2.8) respectively. For a sample of $M = 800$ the results are shown in Table 1.

Table 1

DGSM's and upper bounds of Sobol Indices for parameters one through seven.

Parameter #	DGSM	Upper Bound
m	3.22×10^{-3}	2.13×10^{-2}
ρ_0	3.56×10^{-3}	2.35×10^{-2}
a_0	9.91×10^{-3}	6.54×10^{-2}
a_1	2.66×10^{0}	1.76×10^{1}
b_0	6.56×10^{-1}	4.33×10^{0}
b_1	5.39×10^{-3}	3.56×10^{-2}
b_2	7.50×10^{-1}	4.95×10^{0}

Table 1 shows that the mass m, atmospheric density ρ_0, a_0, and b_1 parameters are relatively unimportant. Consequently, we focus on perturbations to parameters a_1, b_0, and b_2. Thus we have reduced the total number of parameters to $N_p = 3$ and so $\Theta \subset \mathbb{R}^3$, allowing for quicker computations. This process is shown to not diminish accuracy through results in subsection 3.1.3.

3.1.3. Space Shuttle Results. At nominal parameters $\boldsymbol{\theta}_0 = \mathbf{0}$, the optimal solution is denoted u^*. This is the expected path the space shuttle would follow given no perturbations in parameters. We simulate a parameter change $\boldsymbol{\theta}_0 \to \boldsymbol{\theta}_1$ at $t = 2000$ in which only the important parameters found in subsection 3.1.2 are perturbed. The optimal solution for the parameter change is denoted u_{opt}. Figure 1 shows an approximation using our interpolated step method for a random parameter perturbation $\boldsymbol{\theta}_0 \to \boldsymbol{\theta}_1$ at $t = 2000$. This simulates a change halfway through the trajectory that requires a new path. In this example the approximation is shown to closely follow the new optimal solution in this parameter shift. However, note that in Figure 1, even the optimal solution violates some final state constraints. This is due to the fact that we impose constraints as penalties, allowing for some violations at the expense of an increase in the cost function.

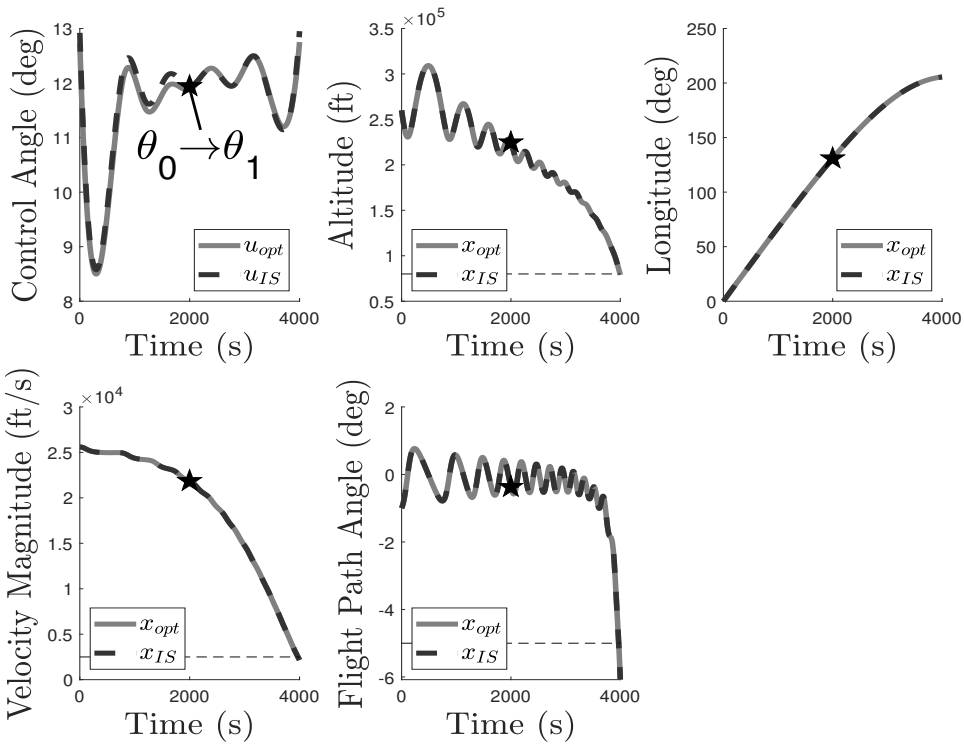

Figure 1. *Controller, $\boldsymbol{u}$, and state variables, $\boldsymbol{x}$, of the optimal solution and approximated solution for parameter shift $\boldsymbol{\theta}_0 \to \boldsymbol{\theta}_1$ at $t = 2000$. The terminal constraints are represented by the thin dashed lines.*

After 500 samples of random parameter perturbations, in which only the important parameters were perturbed, Figure 2 gives histograms of results to display the relative density of approximations.

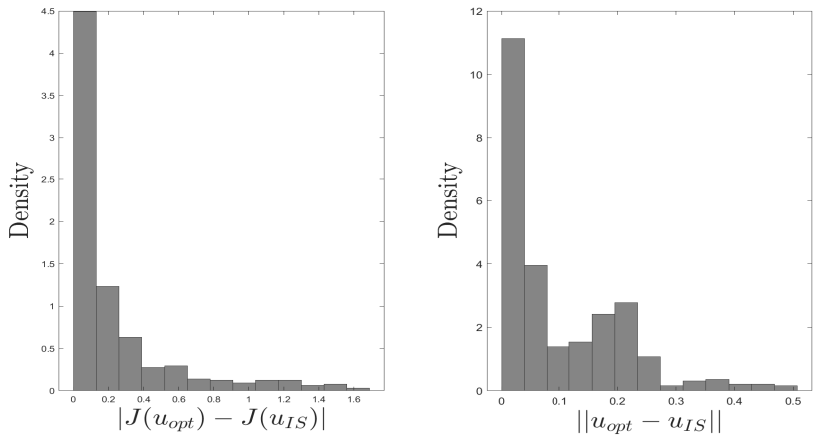

Figure 2. *Cost difference (left) and error norm (right) between optimal and approximated solutions for 500 parameter perturbations. Note that the cost is on the scale of* 2 *and* $||u||$ *is on the scale of* 0.68.

In Figure 2 we see that both the cost and the norm error skew right, thus implying most interpolated step approximations efficiently minimize cost and accurately approximate the optimal solution for the parameter change. We chose these two error metrics because we noticed that an approximation that minimizes error does not necessarily minimize cost and vice versa. Since we only have control over the costs here, this implies there are multiple paths that lead to similar results, and computationally this implies that this is a highly non-convex problem with many local minima. By using both metrics, a more complete view of the effectiveness of our approximation is shown.

Based on the scale of the cost and norm we notice that there are outliers for both cost and error measurements. Since we take random parameter perturbations, there are situations when they are beyond the ability of our optimizer (without extra care to find a better initial guess), and the approximation fails in a similar way to the re-optimized result.

Finally, Figure 2 displays a bimodal distribution with a main mode around 0 and a second mode around 0.2. We notice that this secondary mode seems to be a feature of the problem. As more samples are taken, the tail extends and the mode persists. However, we can calculate,

$$\mathbb{P}\left(\|u_{opt} - u_{IS}\| < 0.2\right) = 0.84.$$

Thus, there is an 84% chance that our approximation is below this secondary mode and thus is an excellent error approximation. Additionally, given that we are looking at a considerable 10% perturbation range, we would expect this probability to only increase with a more local approximation perturbation range of 5% or less.

Next, to verify the usefulness of parameter reduction, we use the 500 parameter changes and compare the new optimal solution using all 7 parameter perturbations, u^7_{opt}, with the case of only considering the 3 parameter perturbations identified in subsection 3.1.2, u^3_{opt}. The following table summarizes the results:

Table 2
Optimal Solutions and Parameter Reduction for 500 Perturbations

	$J\left(u_{opt}^{7}\right)$	$J\left(u_{opt}^{3}\right)$	$\|u_{opt-7p} - u_{opt-3p}\|$
Mean	1.37×10^{0}	1.11×10^{0}	6.21×10^{-2}
Median	-1.63×10^{0}	-1.60×10^{0}	2.11×10^{-2}

where $||u||$ is on the scale of 0.68. The median difference, 0.0211, represents about a 3% change in the controller norm and an even smaller change in the optimal solution's cost. Thus, optimizing with the reduced parameters returns solutions with nearly the exact same cost and almost no difference in path.

Note that in this study, the constraints are encoded as penalties in the objective, so instability represents itself as the discrepancy between the median and mean in the approximation costs, whereby the small norm difference highlights the nearness to the optimizer output.

Table 3 considers the linear (or forward-Euler) approximation over the same sample of perturbed parameters sub-scripted as u_{lin}, both over 7 parameters and 3 parameters with notation as above. Table 4 compares the norm differences to the reoptimized solution.

Table 3
500 Perturbed Parameters Linear Approximation Costs without Reduction v.s. with Reduction

	$J\left(u_{opt}^{7}\right)$	$J\left(u_{lin}^{7}\right)$	$J\left(u_{lin}^{3}\right)$
Mean	1.37×10^{0}	2.54×10^{3}	6.83×10^{2}
Median	-1.63×10^{0}	-9.68×10^{-1}	-9.42×10^{-1}

Table 4
500 Perturbed Parameters Linear Approximation Errors without Reduction v.s. with Reduction

	$\|u_{lin}^{7} - u_{lin}^{3}\|$	$\|u_{opt}^{7} - u_{lin}^{7}\|$	$\|u_{opt}^{7} - u_{lin}^{3}\|$	$\|u_{opt}^{3} - u_{lin}^{7}\|$	$\|u_{opt}^{3} - u_{lin}^{3}\|$
Mean	4.27×10^{-2}	1.48×10^{-1}	1.45×10^{-1}	1.44×10^{-1}	1.37×10^{-1}
Median	4.37×10^{-2}	1.21×10^{-1}	1.19×10^{-1}	1.26×10^{-1}	1.17×10^{-1}

Table 3 provides two main results. First, the difference in median values of $J\left(u_{\text{lin}}^{7}\right)$ and $J(u_{\text{lin}}^{3})$, i.e. -9.68×10^{-1} and -9.42×10^{-1} respectively, is minuscule, around a 3% difference. This suggests that parameter reduction is an effective method for reducing computation cost while maintaining accuracy. Secondly, we see that on average a linear approximation does not do a good job of minimizing cost and has many outliers that greatly reduce its accuracy. This is shown by the high mean costs of 2.54×10^{3} and 6.83×10^{2}.

Table 4 provides similar results. We see that the norm difference between approximations using the full parameter space versus the reduced space is extremely small, 4.37×10^{-2}. This once again suggests that the parameter space reduction does not reduce accuracy of approximations. However, we see that error between any optimal solution and linear approximation is large, such as $||u_{opt}^{7} - u_{lin}^{7}||$ median of 1.21×10^{-1} which is an 82% increase from the nominal normed controller. This once again shows that a linear approximation is not effective at approximating an optimal solution.

Tables 5 and 6 provide similar data but for our new interpolated step approximation method denoted as u_{IS}. The approximations were computed for the same 500 parameter perturbations and for both the full 7 dimensional parameter space and the reduced 3 dimensional parameter space.

Table 5

500 Perturbed Parameters Interpolated Step Approximation Costs without Reduction v.s. with Reduction

	$J\left(u_{opt}^{7}\right)$	$J\left(u_{IS}^{7}\right)$	$J\left(u_{IS}^{3}\right)$
Mean	1.37×10^{0}	1.82×10^{0}	1.34×10^{0}
Median	-1.63×10^{0}	-1.46×10^{0}	-1.39×10^{0}

Table 6

500 Perturbed Parameters Interpolated Step Approximation Errors without Reduction v.s. with Reduction

	$\left\|u_{IS}^{7}-u_{IS}^{3}\right\|$	$\left\|u_{opt}^{7}-u_{IS}^{7}\right\|$	$\left\|u_{opt}^{7}-u_{IS}^{3}\right\|$	$\left\|u_{opt}^{3}-u_{IS}^{7}\right\|$	$\left\|u_{opt}^{3}-u_{IS}^{3}\right\|$
Mean	7.56×10^{-2}	1.10×10^{-1}	1.15×10^{-1}	1.20×10^{-1}	1.04×10^{-1}
Median	4.63×10^{-2}	5.90×10^{-2}	5.42×10^{-2}	7.00×10^{-2}	4.70×10^{-2}

Table 5 shows a similar narrative to the case without interpolation. First, we see that the mean and median are distant because as mentioned earlier, imposing constraints as penalties represents itself as discrepancy. However, this has greatly improved the stability of the result compared to the linear case. The mean and median costs now closely resemble that of the optimized result, showing that our approximation method does a great job of minimizing cost, just as seen and discussed for figure 2. Finally, we once again see that the minimal difference between the 7- and 3-parameter cases, median costs of -1.46 and -1.39 respectively, justifies the use of parameter reduction as it also greatly reduces the computational load of using the interpolation.

Table 6 also shows similar norm error results to the linear case. The norm difference between the 7- and 3-parameter approximations, $\left\|u_{IS}^{7}-u_{IS}^{3}\right\|$ having a median of 4.63×10^{-2}, is small enough to again imply that parameter reduction is an effective method for reducing computation cost while maintaining accuracy. Additionally, we see much lower errors showing that the interpolated step approximation method accurately replicates the optimal solution while only using precomputed data.

3.2. Space Shuttle Discussion. Using the general methods of HDSA, parameter reduction, and our interpolated step method, the shuttle trajectory problem (3.5) was adjusted accurately for the parameter perturbations without sacrificing robustness. Figure 2 shows right skew that implies the interpolated step approximation minimizes cost and accurately approximates the optimal solution. Table 2 verifies that parameter reduction does not affect the optimized solutions accuracy. Tables 3, 4, 5, and 6 all show that reducing the parameter space of statistically unimportant parameters does not affect the accuracy of approximated solutions. Additionally, the difference in approximation accuracy from tables 3 and 4 to tables 5 and 6, verify that our instantaneous interpolated step approximation method does an excellent job at approximating a new optimal controller.

We observe a significant reduction in computation time, as shown in Table 7. As mentioned earlier, recalculating the optimal solution in-transit is often prohibitive given the lack of any guarantee of a favorable solution in any short period of time.

Table 7
500 Perturbed Parameters Approximations Mean Computation Times

	time(u_{opt})	time(u_{IS})
Mean (s)	3.19×10^1	1.12×10^{-1}

Table 7 highlights the considerable decrease in computation time between obtaining the optimal solution and our interpolated step approximation. Note that the primary factor that is limiting the approximation from achieving even faster computation times is the speed of the interpolation function in MATLAB. Additionally, keep in mind that the data in Table 7 was acquired on a local workstation as opposed to the specialized chips and hardware that would be onboard this aircraft problem.

Concisely, our interpolation improves in-situation computation time and approximation stability for both cost and norm error metrics. When coupled with the parameter reduction, the load of precomputation for this method is reduced without a significant sacrifice of accuracy.

4. Conclusion. Optimal control under parameter uncertainty restricts the usefulness of the precomputed optimal solution due to in-transit parameter changes. Recalculation of the optimal path is seen to be computationally expensive, time-consuming, and lacks any guarantee of obtaining a favorable solution. The proposed framework of coupling parameter space reduction, an interpolated step approximation, and the use of hyper-differential sensitivity analysis (HDSA), focuses on a quick and accurate in-transit approximation of the optimal controller. For problems in aerospace, by precomputing sensitivity data and solving a simpler problem due to parameter space reduction, instantaneous approximations are able to save the vehicle from inevitable catastrophe.

As stated in the introduction, we want to emphasize that our method is intended to be used in conjunction with established methods for handling small parameter perturbations such as feedback control. The space shuttle example of Betts verifies the parameter reduction and interpolated-step approximation methods. While this is a limited application, the generality of these methods allows their accuracy to be controlled by the user and allows for more complex application. Stricter methods of applying constraints would also likely limit the instability possible in the approximated solution. It is important to note that this approach is one of many in the toolbox for solving perturbation issues in optimal control problems and thus users should consider other established methods alongside this new approach to determine the most effective and practical solution for the application under consideration. Yet, the proposed approach has applications in aerospace trajectory control, parameter uncertainty calibration, and broad applicability to optimal control problems in which parameter space reduction could improve computation time with little to no reduction in accuracy.

Acknowledgments. As a part of the North Carolina State University's Directed Research for Undergraduates in Mathematics and Statistics (DRUMS) program, our research was sup-

ported in part by NSF/DMS-2051010, as well as the National Security Agency Grant No. H98230-23-1-0009.

We also thank Alen Alexanderian, Joseph Hart, and Paul Spears for their useful insights and oversight of this project.

Appendix A. Parameters and Variables of the Space Shuttle Problem. Given below are the parameters and variables for the 2 DOF space shuttle problem described in section 3.

Table 8
Static Parameters

Parameter Name	Symbol	Value
Vehicle Mass	m	$\frac{20300}{32.173}$ $(slug)$
Aerodynamic Reference Area	S	2690 (ft^2)
Nominal Atmospheric Density	ρ_0	0.002378 (lb/ft^3)
Scale Height Factor	h_r	23800 (ft)
Radius of the Earth	Re	20902900 (ft)
GM for Earth	μ	$0.14076539 \times 10^{17}$ (ft^3/s^2)

Table 9
Dimensionless Drag and Lift Coefficients

Parameter Name	Symbol	Value
Constant Lift	a_0	-0.20704
Linear Lift	a_1	0.029244
Constant Drag	b_0	0.07854
Linear Drag	b_1	-0.61592×10^{-2}
Quadratic Drag	b_2	0.621408×10^{-3}

Table 10
Functional Variables

Function Name	Variable	Function
Atmospheric Density	$\rho(h)$	$\rho_0 \exp\left(\frac{h}{h_r}\right)$ (lb/ft^3)
Distance from Earth's Center	$r(h)$	$Re + h$ (ft)
Force of Gravity	$g(h)$	$\frac{\mu}{r^2}$ (ft/s^2)
Angle of Attack	$\hat{u}$	$u\frac{180}{\pi}$ (deg)
Coefficient of Lift	$c_L(\hat{u})$	$a_0 + a_1\hat{u}$ (-)
Coefficient of Drag	$c_D(\hat{u})$	$b_0 + b_1\hat{u} + b_2\hat{u}^2$ (-)
Drag Force	$D(c_D, \rho, v)$	$0.5c_D S\rho v^2$ $(lbft/s^2)$
Lift Force	$L(c_L, \rho, v)$	$0.5c_L S\rho v^2$ $(lbft/s^2)$

Appendix B. Space Shuttle Cost Function. Given below is the complete cost function for the 2 DOF space shuttle problem. The first term aims to maximize longitudinal distance. The next three penalties enforce terminal conditions and the final four penalties enforce state constraints. The β_i are tuning parameters. These tuning parameter values were chosen experimentally and are going to be application dependent.

$$
\begin{aligned}
J(u) &= \frac{-x_2(t_f, u)}{\overline{x_2}} \\
&+ \beta_1 \left(\frac{x_1(t_f, u) - h_f}{h_f}\right)^2 + \beta_2 \left(\frac{x_3(t_f, u) - v_f}{v_f}\right)^2 + \beta_3 \left(\frac{x_4(t_f, u) - g_f}{g_f}\right)^2 \\
&+ \beta_4 \left(\frac{(\sum_{(x_1(t_i)<0} x_1(t_i)^4)}{\overline{x_1}^4}\right) + \beta_5 \left(\frac{(\sum_{(x_3(t_i)<1}(x_3(t_i) - 1)^4))}{\overline{x_3}^4}\right) \\
&+ \beta_6 \left(\frac{(\sum_{(x_4(t_i)<\frac{-89\pi}{180}}(x_4(t_i) + \frac{89\pi}{180})^4))}{\overline{x_4}^4}\right) + \beta_7 \left(\frac{(\sum_{x_4(t_i)>\frac{89\pi}{180}}(x_4(t_i) - \frac{89\pi}{180})^4)}{\overline{x_4}^4}\right)
\end{aligned}
$$

Appendix C. Sensitivity Equations. The sensitivity equations compute $\frac{dJ}{du_i}$ using a manipulation of chain rule that allows the exploitation of ODE solvers to make these derivatives more accurate than finite difference or complex step when calculated numerically. From the

defining dynamical systems, we have $\frac{dx}{dt} = F$. Differentiating this gives

$$\frac{\partial^2 x}{\partial u_i \partial t} = \frac{\partial F}{\partial x}\frac{\partial x}{\partial u_i} + \frac{\partial F}{\partial u}\frac{\partial u}{\partial u_i}$$

Since the mixed partials are continuous, we may apply Clairaut's theorem that $\frac{\partial^2 x}{\partial u_i \partial t} = \frac{\partial^2 x}{\partial t \partial u_i}$, establishing an ODE against time from which we solve for $\frac{\partial x}{\partial u_i}$. In the context of the space shuttle, we can set up this ODE as

$$\frac{dF}{dx} = \begin{bmatrix} 0 & 0 & \sin(x_4) & x_3\cos(x_4) \\ \frac{-x_3\cos(x_4)}{r^2} & 0 & \frac{\cos(x_4)}{r} & -\frac{x_3}{r}\sin(x_4) \\ \frac{-D}{h_r} + \frac{2g\sin(x_4)}{r} & 0 & \frac{2D}{x_3} & -g\cos(x_4) \\ \frac{-L}{x_3 m h_r} + cos(x_4)(\frac{-x_3}{r^2} + \frac{2g}{x_3 r}) & 0 & \frac{L}{m x_3^2} + \cos(x_4)(\frac{1}{r} + \frac{g}{x_3^2}) & \sin(x_4)(\frac{x_3}{r} - \frac{g}{x_3}) \end{bmatrix}$$

and

$$\frac{dF}{du} = \begin{bmatrix} 0 \\ 0 \\ \frac{-1}{2m}(S\rho_0 e^{-x_1/h_r} x_3^2)(b_1 + 2b_2\hat{u})(\frac{180}{\pi}) \\ \frac{1}{m}(S\rho_0 e^{-x_1/h_r} x_3^2)(a_1)(\frac{180}{\pi}) \end{bmatrix}$$

Through applications of the Lebesgue differentiation theorem and Leibniz integration rule, as well as considering the piecewise portions of the cost, we arrive at

$$\begin{aligned}
\frac{\partial J}{\partial u_i} &= \frac{\overline{x_2}(-\frac{dx_2}{du_i}(t_f,u)) + x_2(t_f,u)\overline{\frac{\partial dx_2}{\partial u_i}})}{(\overline{x_2})^2} \\
&+ 2\beta_1 \frac{(x_1(t_f,u) - h_f)}{h_f^2}\frac{\partial x_1}{\partial u_i}(t_f,u) + 2\beta_2 \frac{(x_3(t_f,u) - v_f)}{v_f^2}\frac{\partial x_3}{\partial u_i}(t_f,u) + 2\beta_3 \frac{(x_4(t_f,u) - g_f)}{g_f^2}\frac{\partial x_4}{\partial u_i}(t_f,u) \\
&+ 4\beta_4 \left(\frac{(\overline{x_1}^4(\sum_{x_1(t_i)<0} 4x_1(t_i)^3 \frac{\partial x_1}{\partial u_i}(t_i,u)) - (\sum_{x_1(t_i)<0} x_1(t_i)^4)(4\overline{x_1}^3\overline{\frac{\partial x_1}{\partial u_i}})}{\overline{x_1}^8} \right) \\
&+ 4\beta_5 \left(\frac{(\overline{x_3}^4(\sum_{x_3(t_i)<1} 4(x_3(t_i)-1)^3 \frac{\partial x_3}{\partial u_i}(t_i,u)) - (\sum_{x_3(t_i)<1} (x_3(t_i)-1)^4)(4\overline{x_3}^3\overline{\frac{\partial x_3}{\partial u_i}})}{\overline{x_3}^8} \right) \\
&+ 4\beta_6 \left(\frac{(\overline{x_4}^4(\sum_{x_4(t_i)<\frac{-89\pi}{180}} 4(x_4(t_i)+\frac{89\pi}{180})^3 \frac{\partial x_4}{\partial u_i}(t_i,u)) - (\sum_{x_4(t_i)<\frac{-89\pi}{180}} (x_4(t_i)+\frac{89\pi}{180})^4)(4\overline{x_4}^3\overline{\frac{\partial x_4}{\partial u_i}})}{\overline{x_4}^8} \right) \\
&+ 4\beta_7 \left(\frac{(\overline{x_4}^4(\sum_{x_4(t_i)>\frac{89\pi}{180}} 4(x_4(t_i)-\frac{89\pi}{180})^3 \frac{\partial x_4}{\partial u_i}(t_i,u)) - (\sum_{x_4(t_i)>\frac{89\pi}{180}} (x_4(t_i)-\frac{89\pi}{180})^4)(4\overline{x_4}^3\overline{\frac{\partial x_4}{\partial u_i}})}{\overline{x_4}^8} \right)
\end{aligned}$$

REFERENCES

[1] A. Alexanderian, J. Hart, and M. Stevens, *A new perspective on parameter study of optimization problems*, Applied Mathematics Letters, 140 (2023), p. 108548.

[2] M. Athans and P. Falb, *Optimal Control: An Introduction to the Theory and Its Applications*, McGraw-Hill, New York, NY, 1966.

[3] F. Bernelli-Zazzera, M. Lavagna, R. Armellin, P. D. Lizia, and A. Morselli, *Trajectory optimization under uncertainties*, tech. report, Department of Aerospace Engineering, Politecnico di Milano, 2012.

[4] J. T. Betts, *Practical Methods for Optimal Control Using Nonlinear Programming, Third Edition*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2020, https://doi.org/10.1137/1.9781611976199, https://epubs.siam.org/doi/abs/10.1137/1.9781611976199, https://arxiv.org/abs/https://epubs.siam.org/doi/pdf/10.1137/1.9781611976199.

[5] J. Burken, N. Nguyen, and B. Griffin, *Adaptive Flight Control Design with Optimal Control Modification for F-18 Aircraft Model*, American Institue of Aeronautics and Astronautics, 2010, https://doi.org/10.2514/6.2010-3364, https://arc.aiaa.org/doi/abs/10.2514/6.2010-3364, https://arxiv.org/abs/https://arc.aiaa.org/doi/pdf/10.2514/6.2010-3364.

[6] H. L. Cleaves, A. Alexanderian, H. Guy, R. C. Smith, and M. Yu, *Derivative-based global sensitivity analysis for models with high-dimensional inputs and functional outputs*, SIAM Journal on Scientific Computing, 41 (2019), pp. A3524–A3551, https://doi.org/10.1137/19M1243518, https://doi.org/10.1137/19M1243518, https://arxiv.org/abs/https://doi.org/10.1137/19M1243518.

[7] J. Hart, B. van Bloemen Waanders, L. Hood, and J. Parish, *Sensitivity-driven experimental design to facilitate control of dynamical systems*, Journal of Optimization Theory and Applications, 196 (2023), pp. 855–881.

[8] M. Huang and Z. Liang, *Approach guidance for rendezvous of non-cooperative targets based on onboard trajectory planning*, in Advances in Guidance, Navigation and Control, L. Yan, H. Duan, and Y. Deng, eds., Singapore, 2023, Springer Nature Singapore, pp. 6206–6216.

[9] S. Kucherenko, M. Rodriguez-Fernandez, C. Pantelides, and N. Shah, *Monte carlo evaluation of derivative-based global sensitivity measures*, Reliability Engineering & System Safety, 94 (2009), pp. 1135–1148, https://doi.org/https://doi.org/10.1016/j.ress.2008.05.006, https://www.sciencedirect.com/science/article/pii/S095183200800152X. Special Issue on Sensitivity Analysis.

[10] I. Sobol, *Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates*, Mathematics and Computers in Simulation, 55 (2001), pp. 271–280, https://doi.org/https://doi.org/10.1016/S0378-4754(00)00270-6, https://www.sciencedirect.com/science/article/pii/S0378475400002706. The Second IMACS Seminar on Monte Carlo Methods.

[11] S. Zhang, F. Liao, Y.-Q. Cheng, and K.-F. He, *Aircraft trajectory control with feedback linearization for general nonlinear systems*, in 2020 Chinese Control And Decision Conference (CCDC), 2020, pp. 3937–3944, https://doi.org/10.1109/CCDC49329.2020.9164469.