Performance Optimization and Guidance of a Low-Altitude Skid-To-Turn Vehicle. Part II: Optimal Guidance and Control

Miriam E. Dennis*
Anil V. Rao†

University of Florida
Gainesville, FL 32611

The problem of guidance and control of an air-to-surface low-altitude skid-to-turn vehicle is considered. The objective is to steer the vehicle to a ground target from an initial state such that the vehicle remains at a constant low altitude for as long as possible and performs a bunt maneuver (negative sensed-acceleration load) rapidly at the end of the trajectory in order to attain terminal target conditions. The vehicle is modeled as a point mass in motion over a flat Earth, and the vehicle is controlled using thrust magnitude, angle of attack, and sideslip angle. The guidance and control problem is posed as a decreasing-horizon optimal control problem and is re-solved numerically at constant guidance cycles. The work described in this paper is the second part of a two-part sequence on trajectory optimization and guidance of a skid-to-turn vehicle. In both cases, the objective is to minimize the time taken by the vehicle to complete a bunt maneuver subject to the following constraints: dynamic, boundary, state, path, and interior-point event constraints. In the second part of this two-part study, a numerical guidance law is employed to re-solve the optimal control problem on a shrinking horizon. An assessment is made as to the time required to re-solve the optimal control problem both in the absence and presence of a time delay, where the time delay is the amount of time required to solve the numerical approximation of the optimal control problem at the start of each guidance cycle. The results of this study identify that in the absence and presence of a time delay, the size of the mesh on subsequent guidance cycles decreases, the time required to solve the reduced-horizon optimal control problem is small compared to the guidance cycle duration time, and the terminal conditions are met with sufficient accuracy. Thus, the results show that the guidance law presented in this paper has the potential to solve optimal control problems in real-time.

Nomenclature

$\mathbf{a}$	=	Vector Field That Defines Reference Dynamics
$\tilde{\mathbf{a}}$	=	Vector Field That Defines Perturbed Dynamics
C	=	Computation Time
$E^{(s)}$	=	Number of Mesh Intervals in Expired Horizon on Guidance Cycle s
G	=	Guidance Cycle Duration
g_0	=	Standard Acceleration Due to Gravity, m·s^{-2}
h	=	Altitude Over Flat Earth
$K^{(s)}$	=	Total Number of Mesh Intervals on Guidance Cycle s
M	=	Total Number of Mesh Refinement Iterations
N	=	Total Number of Legendre-Gauss-Radau Quadrature Collocation Points on a Mesh

*Ph.D. Student, Department of Mechanical and Aerospace Engineering. E-mail: dennism09@ufl.edu.
†Associate Professor, Department of Mechanical and Aerospace Engineering. Erich Farber Faculty Fellow and University Term Professor. E-mail: anilvrao@ufl.edu. Corresponding Author.

s	=	Guidance Cycle
S	=	Total Number of Guidance Cycles
t	=	Time on $[t_0, t_f]$
$t_e^{(s)}$	=	Time on $[t_0^{(s)}, t_f^{(s)}]$ Corresponding to Terminal Time of Expired Horizon on Guidance Cycle s
t_0	=	Initial Time
t_b	=	Phase Transition Time
t_f	=	Terminal Time
T_d	=	Computation Time Delay
$T_e^{(s)}$	=	Time on $[-1, +1]$ Corresponding to Terminal Time of Expired Horizon on Guidance Cycle s
$\mathbf{u}$	=	Control Input to Dynamical System
$\mathbf{u}^*$	=	Optimal Control
v	=	Speed
x	=	Downtrack Displacement
$\mathbf{y}$	=	State of Dynamical System
$\mathbf{y}^*$	=	Optimal State of Dynamical System
$\tilde{\mathbf{y}}$	=	State of Dynamical System Using Perturbed Dynamics
$\Delta\mathbf{y}(t)$	=	$\tilde{\mathbf{y}}(t) - \mathbf{y}^*(t)$
y	=	Crosstrack Displacement
z	=	Down Displacement
α	=	Pitch Angle
β	=	Sideslip Angle
γ	=	Flight Path Angle
ψ	=	Azimuth Angle
τ	=	Time on $[-1, +1]$

I. Introduction

Optimal control problems arise frequently in many engineering applications due to the need to optimize performance of a controlled dynamical system. In general, optimal control problems do not have analytic solutions and, thus, must be solved numerically. Numerical methods for optimal control fall into two broad categories: indirect methods and direct methods. In an indirect method, the first-order variational optimality conditions are derived, and the optimal control problem is converted to a Hamiltonian boundary-value problem (HBVP). The HBVP is then solved numerically using a differential-algebraic equation solver. In a direct method, the state and control are approximated, and the optimal control problem is transcribed into a finite-dimensional nonlinear programming problem[1] (NLP). The NLP is then solved numerically using well developed software for solving NLPs.[2–4] Even in cases where an accurate approximation to the solution of an optimal control problem can be computed, disturbances in the actual system along with measurement errors lead to suboptimal performance and constraint violations in the actual system. If the difference between the motion of the reference system and that of the actual system is large, it may be necessary to re-solve the optimal control problem (that is, perform an optimal midcourse correction) in real-time based on the current state of the actual system. When re-solving the optimal control problem is necessary, it is desirable that the re-optimized solution be obtained sufficiently quickly so that this new solution can be implemented in the actual system. Because most optimal control problems must be solved numerically, in order to realize real-time optimal control in a constrained nonlinear dynamical system, it is necessary to develop computational methods.

A problem of current interest within the optimal control community is the ability to solve a constrained nonlinear optimal control problem sufficiently quickly that it could potentially be used for outer-loop control (guidance). Such an approach, generically termed *optimal guidance*, provides the ability to maintain high performance in real time. A particular problem that falls within the aforemetioned application realm are air-to-surface missiles where the goal is to minimize the time required to reach a target from an initial state. This research focuses on the application of a computational framework for the real-time solution

of optimal control problems for a skid-to-turn low-altitude air-to-surface missile. The problem of interest is one where the vehicle starts in a level flight configuration and is guided to terminal constraints on the position and velocity of the vehicle while being subject to a maximum altitude and crosstrack constraints during flight. The approach used in this paper is to employ the recently developed class of direct *Gaussian quadrature orthogonal collocation methods*[5–19] as an outer loop control (guidance law). In particular, the hp–adaptive Gaussian quadrature collocation approach is employed where it is possible to design a mesh that has the potential to provide equivalent or greater accuracy using a significantly smaller mesh than would be required using a traditional fixed-order collocation method. As a result, the computational efficiency of an hp method is potentially much greater when compared to that of a fixed-order method.

Previous work on trajectory optimization of air-to-surface missiles includes Refs. 20–22. Specifically, Ref. 20 studied the problem of trajectory optimization of a skid-to-turn air-to-surface missile with seeker angle constraints using a direct shooting method. Ref. 21 studied a problem similar to that of Ref. 20 using a bank-to-turn vehicle. Finally, Ref. 22 studied the problem of trajectory regulation using a linear quadratic follower approach to calculate a closed-loop in-flight control. In addition, Ref. 22 employed a preview term in order to predict the future trajectory from the current state. The approaches developed in Refs. 20–22 employed a shooting method for trajectory optimization and tracked the computed reference trajectory using a neighboring optimal control approach.[23]

Different from the research described in Refs. 20–22, in this paper, the problem of closed-loop guidance for a skid-to-turn air-to-surface missile is investigated using a Legendre-Gauss-Radau collocation method[6–14] for both the trajectory generation and the closed-loop control. The work described in this paper is the second part of a two-part sequence on trajectory optimization and guidance of a skid-to-turn vehicle. First, a reference time-optimal trajectory and control are obtained for a vehicle that starts at level flight at a low altitude and terminates at a prescribed location and nearly vertically downward orientation. Closed-loop optimal guidance is then implemented by simulating the flight of the vehicle over fixed guidance cycles using a perturbed model that is different from the reference model. The terminal state obtained from the simulation of the flight at the end of a guidance cycle using the perturbed dynamic model is used as the initial state for the solution of the optimal control problem on the remaining horizon. The process of simulating the vehicle flight using the non-reference model and re-solving the optimal control problem is repeated at the start of each guidance cycle. In order to efficiently solve the optimal control problems that arise as the horizon shrinks with each guidance cycle, the shrinking-horizon mesh-generation method of Ref. 24 reduces the mesh. In particular, at the start of each guidance cycle, a reduced mesh compared with the mesh used on the previous guidance cycle is generated by retaining only the unexpired portion of the horizon since the last guidance update. As was shown in Ref. 24, the unexpired portion of the mesh was well placed when the optimal control problem was solved at the previous guidance update. As a result, the remaining mesh was shown in Ref. 24 to be well placed for use as a starting mesh for the current guidance update. As a result of using the unexpired portion of the mesh as a starting point, little mesh refinement is required in order to obtain the solution on the unexpired horizon. This approach makes it possible to rapidly generate a control for use on the current guidance cycle.

This approach is applied to the aforementioned air-to-surface missile problem using two different modeling assumptions. First, it is assumed that the optimal control problem can be solved instanteously, that is, that the solution to the optimal control problem on the unexpired horizon can be obtained in the absence of a computation time delay. Second, it is assumed that the solution to the optimal control problem on the remaining horizon, for use on the next guidance cycle, is available only at the time when the solution has been obtained. Using this second modeling assumption, a time delay is introduced that is equal to the time required to solve the optimal control problem. The performance of the system using these two modeling assumptions is then assessed and compared.

This paper is organized as follows. Section II provides an overview of the computational guidance and control law used in this paper. Section III provides the description of the optimal control problem example and the offline solution. Section IV shows the results and discussion of solving the optimal control problem listed in Section III in the absence and presence of a computation time delay. Finally, Section V provides conclusions on this work.

II. Computational Guidance and Control Law

The guidance and control law implemented in this paper is based on Ref. 24. An overview of this method is described below. Section A provides an overview of the guidance law and the implementation of the optimal control problem within the guidance law. Section B describes the mesh-generation method used in the guidance law to generate the reduced-horizon optimal control problem that is solved on each guidance cycle. Section C describes the absence or presence of a computation time delay within the guidance law. Finally, Section D gives an overview of the procedures to use the computational guidance and control method.

A. Solution to Optimal Control Problem

Suppose the solution to the optimal control problem is given on the horizon $t \in [t_0^{(s)}, t_f^{(s)}]$ where $t_0^{(s)}$ is the start of guidance cycle $s \in [1, \ldots, S]$, S is the number of guidance cycles (that is, S is the number of times the control is updated during the motion), $t_0^{(s)} = t_0 + sG$ is the initial time of the guidance cycle, and G is the guidance cycle duration.

Consider the following scenario where the optimal control is implemented on the current guidance cycle, $[t_0^{(s)}, t_e^{(s)}]$ where $t_e^{(s)} = t_0 + (s+1)G$ is the end time of the guidance cycle. Because the perturbed model dynamics are not the same as the reference dynamics, the motion of the system will evolve differently. As the actual motion differs from the reference solution, the dynamics at the end of a guidance cycle will not match the reference solution, $\tilde{\mathbf{y}}(t_e^{(s)}) \neq \mathbf{y}(t_e^{(s)})$, at the guidance cycle $s \in [0, 1, \ldots, S]$. Also, it is important to note that over the first time interval $t \in [t_0, t_0 + G]$ (that is, when $s = 0$), the motion of the actual system is obtained using the optimal control from the solution of the reference optimal control problem. Thus, the optimal control must be updated to generate a solution that compensates for these deviations. To correct these deviations, the optimal control problem is re-solved on the unexpired horizon, $t \in [t_e^{(s)}, t_f^{(s)}]$. This shorter-horizon optimal control problem is referred to as the reduced-horizon optimal control problem, and the start time of this reduced-horizon optimal control problem, denoted a guidance update time, is $t_0^{(s)} = t_e^{(s-1)}$.

The process of re-solving the reduced-horizon optimal control problem on the unexpired horizon, $t \in [t_0^{(s)}, t_f^{(s)}]$, combined with the motion of the system evolving over a guidance cycle, $[t_0^{(s)}, t_e^{(s)}]$, is repeated for each guidance cycle, $s \in [1, 2, \ldots, S]$. The schematic of Fig. 1 shows how the process is implemented as a guidance law on guidance cycle s where the guidance cycle is over the time $t \in [t_0^{(s)}, t_e^{(s)}]$. The boundary conditions of the optimal control problem are updated, the optimal control problem is solved on the unexpired horizon, and the updated control is used as the system dynamics evolve over the guidance cycle with the possibility of a computation time delay, where the time delay is equal to the computation time required to solve the nonlinear programming problem (NLP) associated with the optimal control problem solved on the unexpired horizon. (See Section C for more details associated with the time delay.)

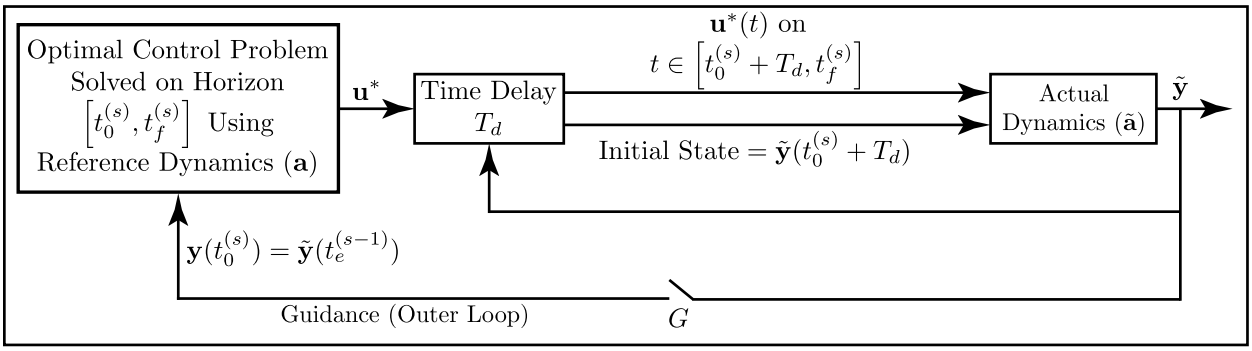
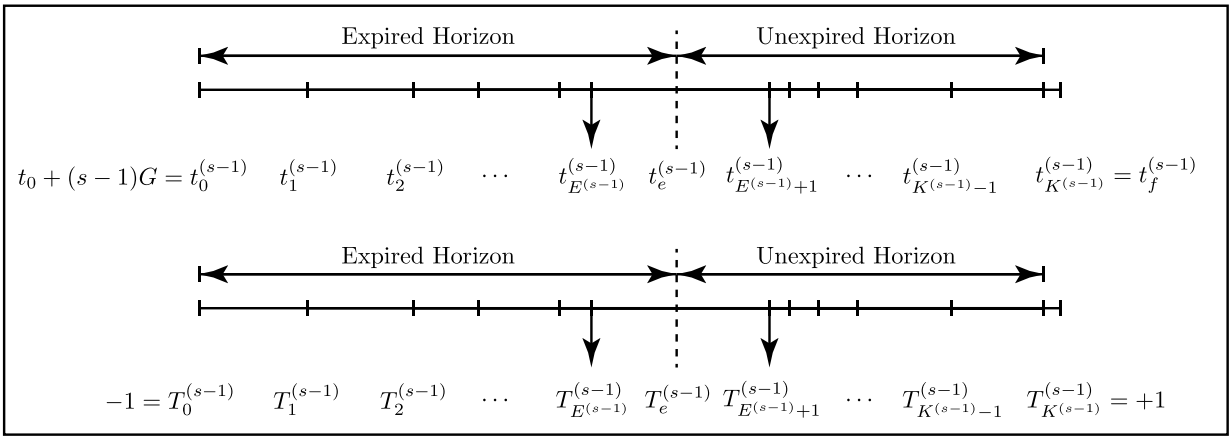


Figure 1: Schematic of a reduced-horizon optimal control problem used as a guidance law.

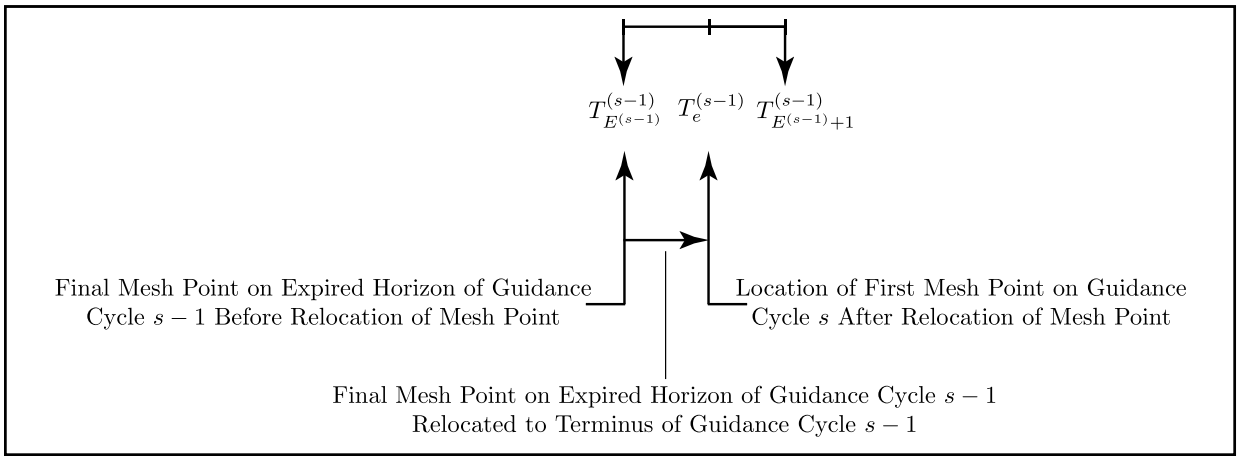
B. Mesh-Generation Method

An important component of this computation guidance law is the ability to solve the reduced-horizon optimal control problem in a quick manner, such that not much time is spent re-solving for the control to be used on the current guidance cycle. This requires that the mesh on the unexpired horizon be well-placed such that little time is required to solve the associated NLP and that the NLP must only be solved once (thus, no mesh refinement is necessary). The overview of this mesh-generation method for use within the guidance law is as follows.

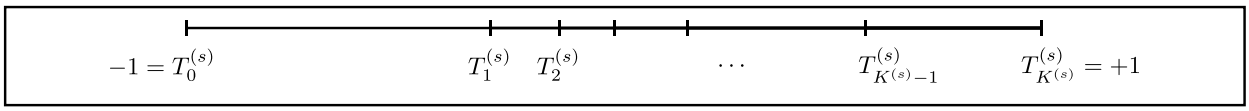
Suppose that the solution has been solved on the guidance cycle $s-1$. The solution carries with it a mesh, mesh intervals, and collocation points. Knowing the previous solution, the starting time of the current guidance cycle s (the start of the unexpired horizon) $t_0^{(s)} = t_e^{(s-1)}$, and the state at the end of the previous cycle $\mathbf{y}(t_0^{(s)}) = \tilde{\mathbf{y}}(t_e^{(s-1)})$, a new mesh can be generated on the current guidance cycle s that is well placed to be solved rapidly. Figure 2 shows the steps taken to transform the mesh. First, the mesh intervals are classified as exisiting in the expired or unexpired horizon. Second, the last mesh point in the expired horizon is relocated to the end of the guidance cycle $s-1$ (the beginning of the current guidance cycle s). The mesh points in the unexpired horizon remain at the same time. Finally, the mesh points on the unexpired horizon (combined with the relocation of the final mesh point on the expired horizon) are transformed to the domain $\tau \in [-1, +1]$. This transformed mesh is now ready to be used for the current guidance cycle s.



(a) Expired and unexpired horizons on guidance cycle $s-1$.



(b) Final mesh point on the expired horizon of guidance cycle $s-1$ as shown in Fig. 2a relocated to the terminus of guidance cycle $s-1$.



(c) Mesh on guidance cycle s, where $T_0^{(s)} = -1$ and $T_{K^{(s)}}^{(s)} = +1$ correspond, respectively, to $T_e^{(s-1)}$ and $T_{K^{(s-1)}}^{(s-1)}$.

Figure 2: Three-part schematic for mesh-generation as part of the method for guidance and control using hp LGR collocation and sparse nonlinear programming.

C. Inclusion of Computation Time Delay

The guidance law described in Section A can be implemented in the absence or presence of a computation time delay. In the absence of a time delay, it is assumed that the sparse nonlinear programming problem (NLP) associated with the hp Legendre-Gauss-Radau (LGR) collocation method can be solved instantaneously, and thus, the control is available to be used over the entire current guidance cycle. In the presence of a time delay, it is assumed that the sparse NLP cannot be solved instantaneously. Let C be the computation time required to solve the sparse NLP associated with the hp LGR collocation method on the horizon $t \in [t_0^{(s)}, t_f^{(s)}]$, and set $T_d = C$ as the time delay. Because the updated control is not available at the start of the current guidance cycle, the control solved for on the previous guidance cycle on the horizon $t \in [t_0^{(s-1)}, t_f^{(s-1)}]$ is used until the new control is available. Thus, the control from the previous guidance cycle is used to simulate the actual dynamics from $[t_0^{(s)}, t_0^{(s)} + T_d]$. After the new control is available, it is used to simultate the dynamics over the rest of the current guidance cycle from $[t_0^{(s)} + T_d, t_e^{(s)}]$ where $t_e^{(s)} = t_0 + (s+1)G$ and is the end of the guidance cycle s. In the absence of a time delay, it is assumed that $T_d = 0$, and the new control is available for use at $t = t_0^{(s)}$ onwards.

D. Computational Guidance and Control Method

Using the procedures mentioned in Sections A–C, the computational guidance and control method of this paper is as follows. First, the solution on guidance cycle $s-1$ is interpolated to the mesh for use on guidance cycle s. This interpolated mesh is the initial guess for solving the sparse NLP associated with the hp LGR collocation method on guidance cycle s. Also, the NLP constraints are updated with the initial time and state at the start of the current guidance cycle s. As the mesh on guidance cycle s is smaller than the mesh on guidance cycle $s-1$, the sparse NLP that is solved on the current guidance cycle s is smaller than the sparse NLP solved on the previous guidance cycle $s-1$. In the absence of a time delay ($T_d = 0$), the control solved on the horizon $[t_0^{(s)}, t_f^{(s)}]$ is used over the entire current guidance cycle s from $[t_0^{(s)}, t_e^{(s)}]$. In the presence of a time delay ($T_d \neq 0$), the control generated on the previous horizon $[t_0^{(s-1)}, t_f^{(s-1)}]$ is used in the simulation of the dynamics from $[t_0^{(s)}, t_0^{(s)} + T_d]$ until the control on the current guidance cycle s is available for use. Then, the control is used in the simulation of the dynamics for the rest of the guidance cycle from $[t_0^{(s)} + T_d, t_0^{(s)} + G]$. This method is repeated for each guidance cycle $s \in [1, 2, \ldots, S]$.

III. Optimal Control Problem

The optimal control problem under consideration is stated as follows. Determine the final time (t_d, t_f) in each phase, state $(x(t), y(t), z(t), v(t), \gamma(t), \psi(t))$, and control $(\alpha(t), \beta(t), T(t))$ that minimizes the cost function

$$J = t_f \left(t_f - t_b\right)^2 + 0.5 \int_{t_b}^{t_f} |h - h_{\max}^{(1)}| dt \tag{1}$$

subject to the dynamic constraints in each phase $p \in [1, 2]$,

$$\begin{aligned}
\dot{x} &= v \cos\gamma \cos\psi, \\
\dot{y} &= v \cos\gamma \sin\psi, \\
\dot{z} &= -v \sin\gamma, \\
m\dot{v} &= (T - A)\cos\alpha\cos\beta - Y\sin\beta - N\sin\alpha\cos\beta - mg\sin\gamma, \\
mv\dot{\gamma} &= (T - A)\sin\alpha + N\cos\alpha - mg\cos\gamma, \\
mv\dot{\psi}\cos\gamma &= (T - A)\cos\alpha\sin\beta + Y\cos\beta - N\sin\alpha\sin\beta,
\end{aligned} \tag{2}$$

the state inequality path constraint in the first phase,

$$h_{\min}^{(1)} \leq h^{(1)} \leq h_{\max}^{(1)}, \tag{3}$$

the linkage constraints on the time and state between phases 1 and 2,

$$\begin{aligned}
t_f^{(p)} - t_0^{(p+1)} &= 0, \\
x(t_f^{(p)}) - x(t_0^{(p+1)}) &= 0, \\
y(t_f^{(p)}) - y(t_0^{(p+1)}) &= 0, \\
z(t_f^{(p)}) - z(t_0^{(p+1)}) &= 0, \quad p \in [1], \\
v(t_f^{(p)}) - v(t_0^{(p+1)}) &= 0, \\
\gamma(t_f^{(p)}) - \gamma(t_o^{(p+1)}) &= 0, \\
\psi(t_f^{(p)}) - \psi(t_0^{(p+1)}) &= 0,
\end{aligned} \tag{4}$$

and the boundary conditions

$$\begin{aligned}
&t_0 = t_0^{(s)}, \quad (x(t_0), x(t_f)) = (x_0^{(s)}, x_f)\ \text{m}, \quad (y(t_0), y(t_f)) = (y_0^{(s)}, y_f)\ \text{m}, \\
&(z(t_0), z(t_f)) = (z_0^{(s)}, z_f)\ \text{m}, \quad (v(t_0), v(t_f)) = (v_0^{(s)}, v_f)\ \text{m}\cdot\text{sec}^{-1}, \quad (\gamma(t_0), \gamma(t_f)) = (\gamma_0^{(s)}, \gamma_f)\ \text{rad}, \\
&\psi(t_0) = \psi_0^{(s)}\ \text{rad},
\end{aligned} \tag{5}$$

where x, y. and z are the displacements in the downtrack, crosstrack, and vertical down directions, $h = -z$ is the altitude, v is the speed, γ is the flight path angle, ψ is the azimuth angle (measured clockwise from the downtrack direction), β is the sideslip angle, α is the angle of attack, and T is the thrust. β, α, and T are the controls for this problem. Furthermore, A, Y, and N are the aerodynamic axial, side, and normal forces, respectively. In this example, the initial conditions $(x_0^{(s)}, y_0^{(s)}, z_0^{(s)}, v_0^{(s)}, \gamma_0^{(s)}, \psi_0^{(s)})$ change depending upon the initial time $t_0^{(s)}$ where $s \in [0, 1, 2, \ldots, S]$, and S is the number of guidance cycles. Finally, the initial conditions corresponding to the full-horizon (reference) optimal solution are given as

$$\begin{aligned}
&t_0^{(0)} = 0\ \text{s}, \quad x(t_0^{(0)}) = 0\ \text{m}, \quad y(t_0^{(0)}) = 0\ \text{m}, \\
&z(t_0^{(0)}) = -31\ \text{m}, \quad v(t_0^{(0)}) = 915\ \text{m}\cdot\text{sec}^{-1}, \quad \gamma(t_0^{(0)}) = 0\ \text{deg}, \\
&\psi(t_0^{(0)}) = 0\ \text{deg},
\end{aligned} \tag{6}$$

while the terminal conditions are given as

$$\begin{aligned}
&x(t_f) = 30500\ \text{m}, \quad y(t_f) = 6100\ \text{m}, \quad z(t_f) = 0\ \text{m}, \\
&v(t_f) = 915\ \text{m}\cdot\text{sec}^{-1}, \quad \gamma(t_f) = -85\ \text{deg}, \quad \psi(t_f) = \text{Free},
\end{aligned} \tag{7}$$

and the limits on the altitude in the first phase are

$$h_{\min}^{(1)} = 10\ \text{m}, \quad h_{\max}^{(1)} = 170\ \text{m}, \tag{8}$$

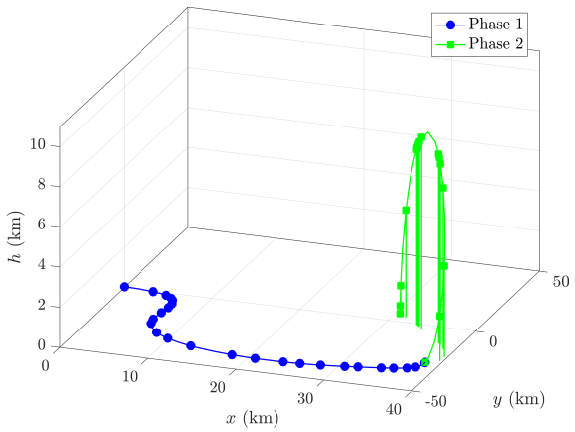
where t_f corresponds to the minimum time obtained on the reference optimal solution on the entire horizon $t \in [t_0^{(0)}, t_f^{(0)}]$. The objective of this example is to demonstrate the capability of the computational guidance and control method of Section II in re-solving the optimal control problem for use as a guidance law every G time units. In the analysis that follows, the model used to describe the actual motion of the vehicle is obtained via a random perturbation of the aerodynamic forces, that is, the aerodynamic forces of the actual vehicle is given as

$$\tilde{A} = q_1 A \quad , \quad \tilde{N} = q_2 N \quad , \quad \tilde{Y} = q_3 Y, \tag{9}$$

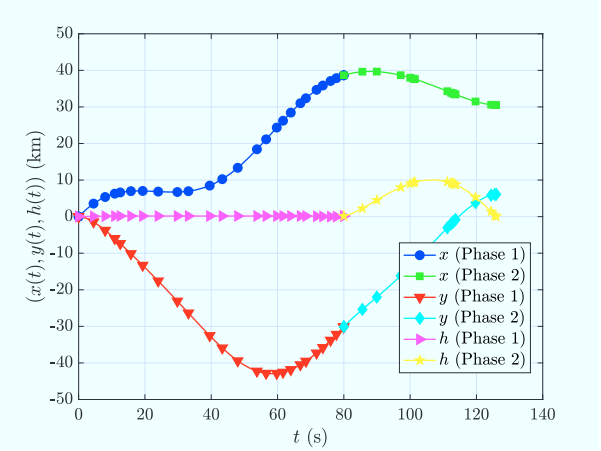
where (q_1, q_2, q_3) are random variables drawn on each guidance cycle from a uniform distribution on the interval $[a, b]$. Thus, the dynamics, used to integrate the motion over each guidance cycle, are of the form

$$\begin{aligned}
\dot{x} &= v\cos\gamma\cos\psi, \\
\dot{y} &= v\cos\gamma\sin\psi, \\
\dot{z} &= -v\sin\gamma, \\
m\dot{v} &= (T - \tilde{A})\cos\alpha\cos\beta - \tilde{Y}\sin\beta - \tilde{N}\sin\alpha\cos\beta - mg\sin\gamma, \\
mv\dot{\gamma} &= (T - \tilde{A})\sin\alpha + \tilde{N}\cos\alpha - mg\cos\gamma, \\
mv\dot{\psi}\cos\gamma &= (T - \tilde{A})\cos\alpha\sin\beta + \tilde{Y}\cos\beta - \tilde{N}\sin\alpha\sin\beta,
\end{aligned} \tag{10}$$

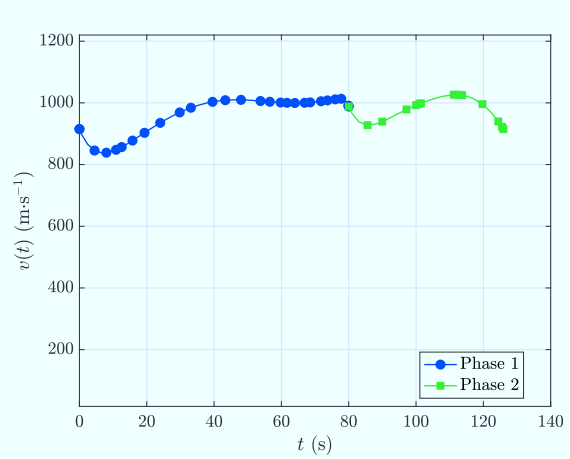
It is seen that the actual dynamics given in Eq. (10) differ from those of the reference dynamics given in Eq. (2). The reference optimal solution on the entire time horizon $t \in [t_0^{(0)}, t_f^{(0)}]$ is shown in Figs. 3–5. It can be seen that the solution consists of an initial ascent to a near constant altitude in phase one about the altitude constraint in Eq. (3). In addition, the first phase terminates with a slight decrease in altitude followed by an increase in altitude as the vehicle prepares to execute a bunt maneuver, as seen in Fig. 4. In the second phase, the bunt maneuver starts as the vehicle climbs to its maximum trajectory altitude and then descends in order to meet terminal conditions. It is seen from the solution that the vehicle never attains the maximum allowable speed $v_{\max} = 1220\ \mathrm{m} \cdot \mathrm{s}^{-1}$; however, the thrust is at maximum $T_{\max} = 8900\ N$ for the entire solution as seen in Fig. 5c.



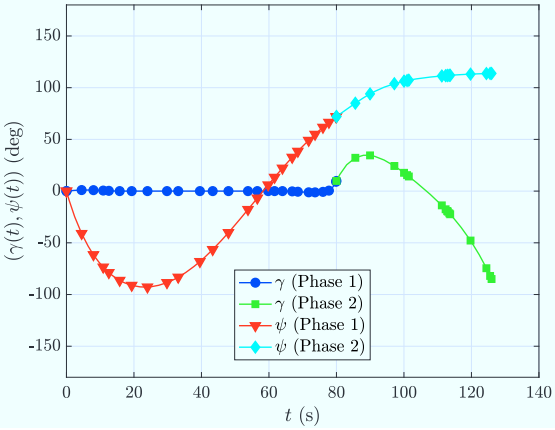
(a) Three-dimensional view of solution.



(b) Downtrack, $x(t)$, crosstrack, $y(t)$, and altitude, $h(t)$, vs. t.



(c) Speed, $v(t)$, vs. t.



(d) Flight path angle, $\gamma(t)$, and azimuth angle, $\psi(t)$, vs. t.

Figure 3: Reference solution of state, $(x(t), y(t), h(t), v(t), \gamma(t), \psi(t))$, vs. t.

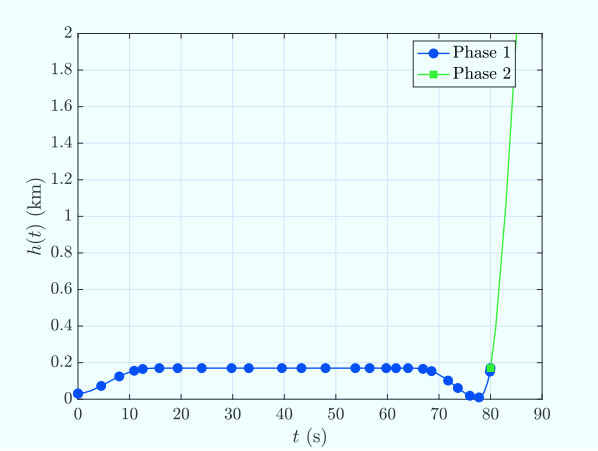
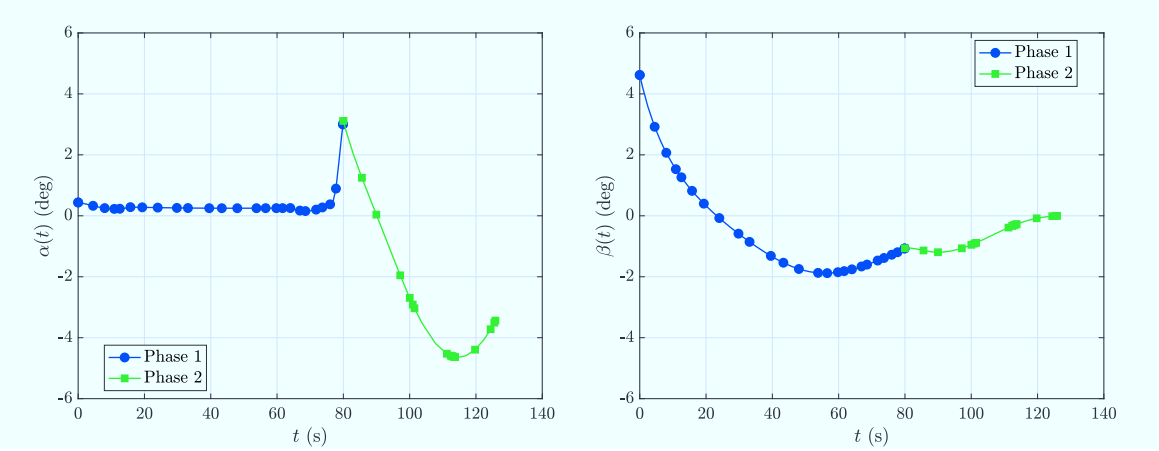
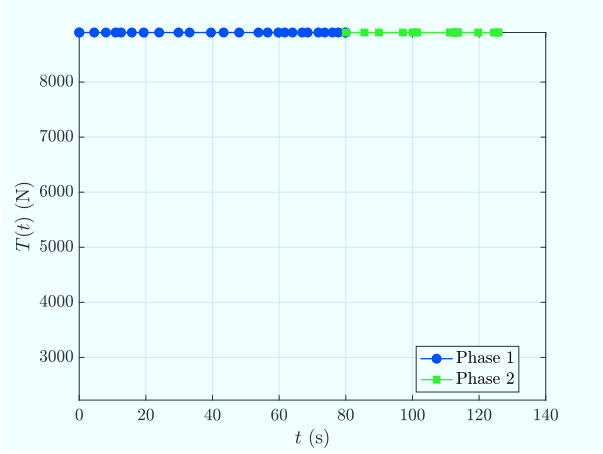


Figure 4: Reference solution of altitude in phase one, $h^{(1)}(t)$, vs. t .



(a) Pitch angle, $\alpha(t)$, vs. t.

(b) Slip angle, $\beta(t)$, vs. t.



(c) Thrust magnitude, $T(t)$, vs. t.

Figure 5: Reference solution of control, $(\alpha(t), \beta(t), T(t))$, vs. t .

IV. Results of Guidance Law and Discussion

In this section, an example is considered where the guidance law given in Section II is employed. The optimal control problem described in Section III is re-solved at constant duration guidance cycles, where the initial conditions for the start of each guidance are obtained from the simulation of the actual dynamics over the previous guidance cycle. In this example, the guidance cycle duration is $G = 3$ s. The example is solved excluding and including a computation time delay as described in Section II.C. In the absence of a time delay, the perturbed dynamics of Eq. (10) are simulated from $[t_0^{(s)}, t_e^{(s)}]$ (the current guidance cycle) using a cubic spline interpolation of the time series of the hp LGR collocation control that has computed on the current unexpired horizon $[t_0^{(s)}, t_f^{(s)}]$ with the initial conditions obtained from the simulation of the perturbed dynamics at the end of the previous guidance cycle $s-1$. In the presence of a time delay, the perturbed dynamics of Eq. (10) are simulated from $[t_0^{(s)}, t_0^{(s)} + T_d]$ using a cubic spline interpolation of the time series of the hp LGR collocation control that was computed on the previous unexpired horizon $[t_0^{(s-1)}, t_f^{(s-1)}]$ and $[t_0^{(s)}+T_d, t_e^{(s)}]$ using a cubic spline interpolation of the time series of the hp LGR collocation control that was computed on the current unexpired horizon $[t_0^{(s)}, t_f^{(s)}]$.

All results shown in this section were obtained using the MATLAB® optimal control software $\mathbb{GPOPS}-\mathbb{II}$[25] with the nonlinear programming problem (NLP) solver *IPOPT*,[4] where *IPOPT* was employed in full Newton (second derivative) using the default NLP solver tolerance of $\epsilon_{\text{NLP}} = 10^{-7}$. Furthermore, in addition to the mesh-generation strategy described in Section II.B, any necessary mesh refinement was performed using the mesh refinement method described in Ref. 14 using a minimum of three and maximum of 14 allowable collocation points per interval and a mesh refinement relative-error accuracy tolerance of $\epsilon_{\text{mesh}} = 10^{-4}$. The perturbed dynamics were simulated using the MATLAB® differential equation solver `ode113` using an integration error tolerance of 10^{-5}. In order to employ *IPOPT* in full Newton mode, all required first and second derivatives were obtained using the algorithmic differentiation software *ADiGator*.[26] Finally, all computations shown in this section were on a 3.1 GHz Intel Core i7 MacBook Pro running Mac OS-X version 10.12.6 (Sierra) with 16GB 1867MHz DDR3 RAM and MATLAB® Version R2016a (build 9.0.0.341360) and all computation (CPU) times, denoted C, are in reference to this aforementioned machine.

A. Performance Excluding Time Delay

Suppose now the the optimal guidance and control method operates in the absence of a time delay ($T_d = 0$). Then, starting with the reference initial conditions of Eq. (6), the optimal control problem given in Eqs. (1)–(5) is re-solved at the start of each guidance cycle $s \in [1, 2, 3, \ldots, S]$ on the decreasing horizon $[t_0^{(s)}, t_f^{(s)}]$ (where $t_0^{(s)} = t_0 + sG$). Furthermore, the simulation of the actual dynamics is performed with the random variables (q_1, q_2, q_3) in Eq. (9) drawn from uniform distribution $[a, b] = [0.9, 1.1]$. The state obtained by simulating the actual dynamics of Eq (10) on the intervals $t \in [t_0^{(s)}, t_0^{(s)} + G]$ (where $s \in [0, 1, 2, \ldots, S]$) using the control computed on $[t_0^{(s)}, t_f^{(s)}]$ via the hp LGR collocation method is shown in Fig. 7 alongside the reference state as shown in Fig. 3. In addition, Fig. 8 shows the control used in the simulation of the actual dynamics alongside the reference control as shown in Fig. 5. The thrust is not shown in Fig. 8, because the thrust found in the guidance solution is indistinguishable from the reference solution and remains at its maximum througout the problem solution. It is seen that the simulated solution over each of the guidance cycles is indistinguishable from the reference solution, even though the model used in the simulation [Eq. (10)] is different from the reference model [Eq. (2)].

Now let $\Delta\mathbf{y}(t) = \tilde{\mathbf{y}}(t) - \mathbf{y}^*(t)$ be the difference between the state, $\tilde{\mathbf{y}}(t)$, obtained using the actual perturbed dynamics of Eq. (10) and the state, $\mathbf{y}^*(t)$, obtained from the reference solution of the optimal control problem using the hp LGR collocation method. Furthermore, let t_f^* be the terminal time obtained from the numerical solution of the optimal control problem using the hp LGR collocation method over the last guidance cycle. Then, the differences $(\Delta x(t_f^*), \Delta y(t_f^*), \Delta z(t_f^*), \Delta v(t_f^*), \Delta\gamma(t_f^*))$ are given, respectively, as $(\Delta x(t_f^*), \Delta y(t_f^*), \Delta z(t_f^*), \Delta v(t_f^*), \Delta\gamma(t_f^*)) = (1.2180 \times 10^{-1}$ m, -2.7017×10^{-1} m, -1.3662×10^{-1} m, -1.5282×10^{-1} m $\cdot$ sec^{-1}, and -2.4306×10^{-4} rad). It is seen that these are, respectively, $\mathcal{O}(10^{-1})$ m, $\mathcal{O}(10^{-1})$ m, $\mathcal{O}(10^{-1})$ m, $\mathcal{O}(10^{-1})$ m $\cdot$ sec^{-1}, and $\mathcal{O}(10^{-4})$ rad.

Next, Fig. 9 shows the number of mesh refinements, M, the number of collocation points, N, the number of mesh intervals, K, and the computation time, C, required to re-solve the optimal control problem

every G time units for each guidance cycle $s \in [1, \ldots, S]$. It is seen that for most of the guidance cycles the optimal control problem is solved in less that 0.9 sec using the hp LGR collocation method. Moreover, it is seen that a minimal amount of mesh refinement is required through all of the guidance cycles in that all of the solutions are obtained on the first mesh without requiring any mesh refinement. Finally, it is seen that the size of the mesh progressively decreases as the guidance cycles evolve such that in the last guidance cycle the mesh consists of 11 collocation points (that is, $N = 11$) and two mesh intervals (that is, $K = 2$). Thus, for the vast majority of guidance cycles the optimal control problem is not only solved efficiently, but the mesh size decreases making it possible to re-solve the optimal control problem using a progressively smaller finite-dimensional approximation of the optimal control problem.

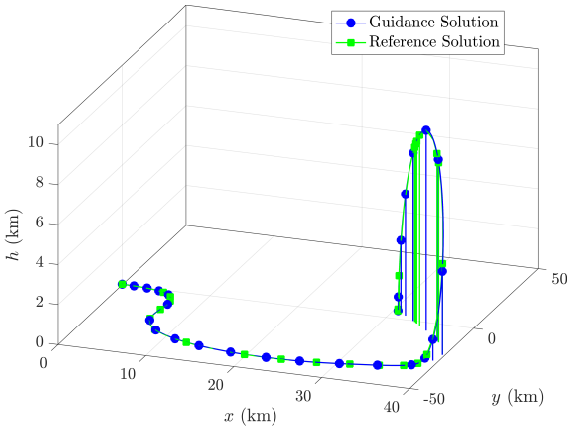
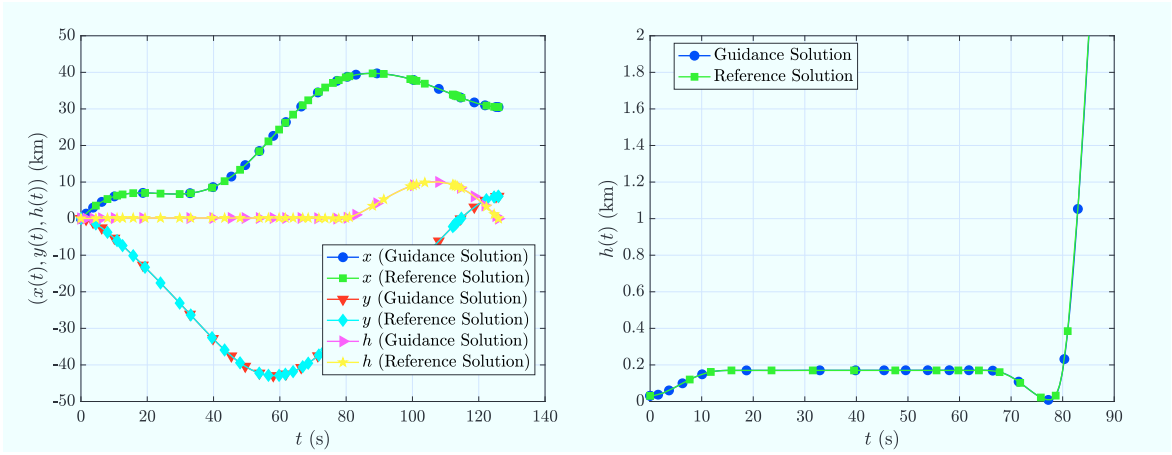
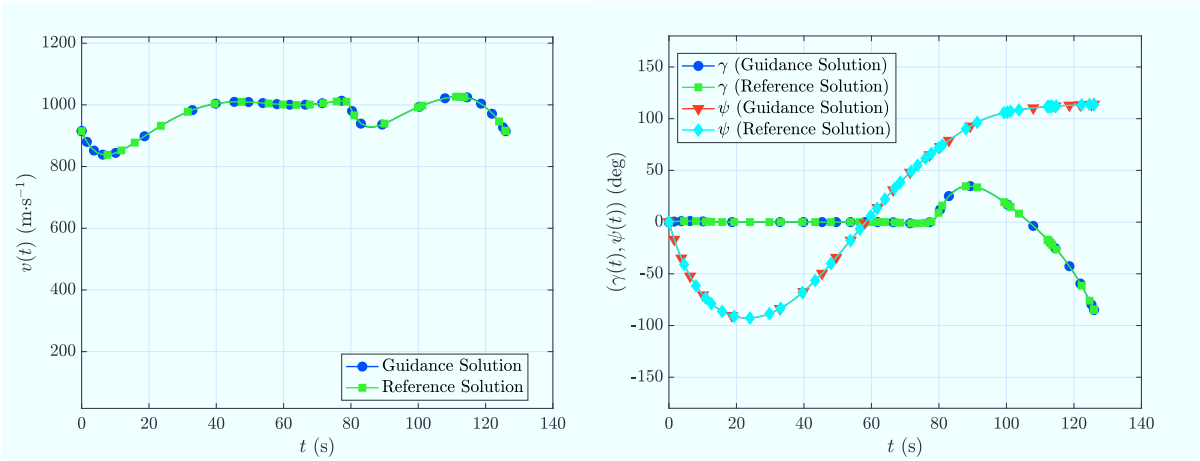


Figure 6: Three-Dimensional Trajectory Obtained via Numerical Integration of the Actual Dynamics as Given in Eq. (10) Over Each Guidance Cycle Using the Control Generated by Solving the Optimal Control Problem of Eqs. (1)–(5) in the Absence of the Time Delay.



(a) Downtrack, $x(t)$, crosstrack, $y(t)$, and altitude, $h(t)$, vs. t.

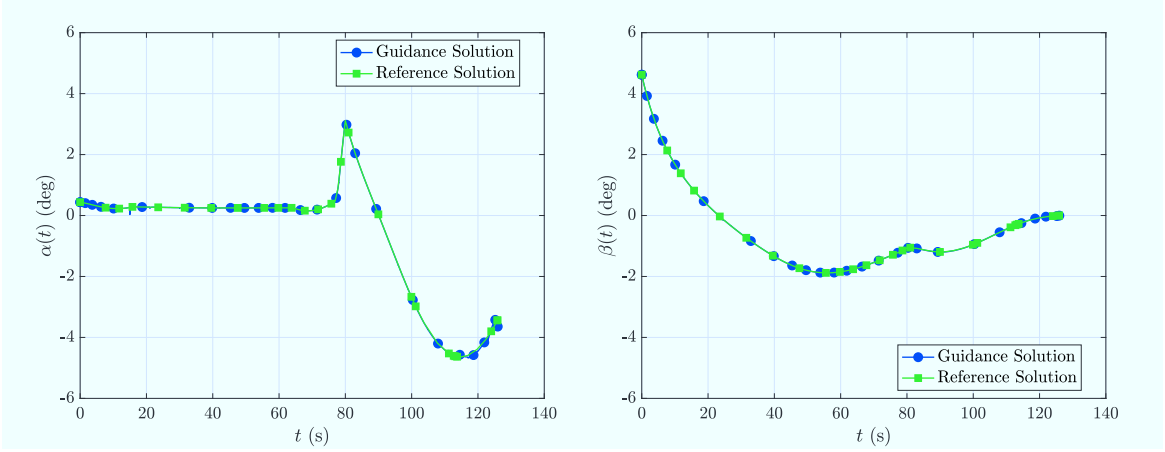
(b) Altitude, $h^{(1)}(t)$, vs. t.



(c) Speed, $v(t)$, vs. t.

(d) Flight path angle, $\gamma(t)$, and azimuth angle, $\psi(t)$, vs. t.

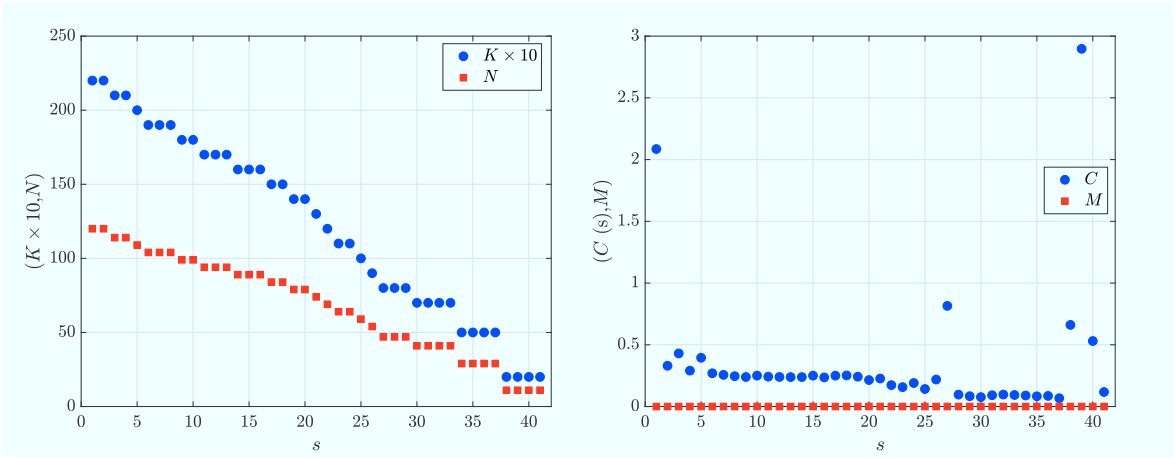
Figure 7: State Solution Obtained via Numerical Integration of the Actual Dynamics as Given in Eq. (10) Over Each Guidance Cycle Using the Control Generated by Solving the Optimal Control Problem of Eqs. (1)–(5) in the Absence of the Time Delay.



(a) Pitch angle, $\alpha(t)$, vs. t.

(b) Slip angle, $\beta(t)$, vs. t.

Figure 8: Control Solution Obtained via Numerical Integration of the Actual Dynamics as Given in Eq. (10) Over Each Guidance Cycle Using the Control Generated by Solving the Optimal Control Problem of Eqs. (1)–(5) in the Absence of the Time Delay.



(a) K, the number of mesh intervals, N, the number of collocation points, vs. s, guidance cycle number.

(b) C, the computational time, M, the number of mesh refinements, vs. s, guidance cycle number.

Figure 9: Mesh Characteristics of the Optimal Control Solution Over Each Guidance Cycle in the Absence of the Time Delay.

B. Performance Including Time Delay

Suppose now that the optimal guidance and control method of this paper operates in the presence of a time delay ($T_d \neq 0$). Then, starting with the reference initial conditions of Eq. (6), the optimal control problem given in Eqs. (1)–(5) is re-solved at the start of each guidance cycle $s \in [1, 2, 3, \ldots, S]$ on the decreasing horizon $[t_0^{(s)}, t_f^{(s)}]$ (where $t_0^{(s)} = t_0 + sG$). Furthermore, the simulation of the actual dynamics is performed with the random variables (q_1, q_2, q_3) in Eq. (9) drawn from uniform distribution $[a, b] = [0.9, 1.1]$. The state obtained by simulating the actual dynamics of Eq. (10) on the intervals $t \in [t_0^{(s)}, t_0^{(s)} + G]$ (where $s \in [0, 1, 2, \ldots, S]$) via the hp LGR collocation method is shown in Fig. 11 alongside the reference state as shown in Fig. 3. In addition, Fig. 12 shows the control used in the simulation of the actual dynamics alongside the reference control as shown in Fig. 5. The thrust is not shown in Fig. 12, because the thrust found in the guidance solution is indistinguishable from the reference solution and remains at its maximum througout the problem solution. It is seen that the simulated solution over each of the guidance cycles is indistinguishable from the reference solution, even though the model used in the simulation [Eq. (10)] is different from the reference model [Eq. (2)].

Similar to the approach used to analyze the results in the absence of a time delay, let $\Delta\mathbf{y}(t) = \tilde{\mathbf{y}}(t) - \mathbf{y}^*(t)$ be the difference between the state, $\tilde{\mathbf{y}}(t)$, obtained using the actual perturbed dynamics of Eq. (10) and the state, $\mathbf{y}^*(t)$, obtained from the reference solution of the optimal control problem using the hp LGR collocation method. Furthermore, let t_f^* be ther terminal time obtained from the numerical solution of the optimal control problem using the hp LGR collocation method over the last guidance cycle. Then, the differences $(\Delta x(t_f^*), \Delta y(t_f^*), \Delta z(t_f^*), \Delta v(t_f^*), \Delta\gamma(t_f^*))$ are given, respectively, as $(\Delta x(t_f^*), \Delta y(t_f^*), \Delta z(t_f^*), \Delta v(t_f^*), \Delta\gamma(t_f^*))$ = $(7.7279 \times 10^{-2}$ m, -1.4750×10^{-1} m, -3.9003×10^{-2} m, 5.9419×10^{-2} m$\cdot$sec^{-1}, and -4.7030×10^{-4} rad). It is seen from these are, respectively, $\mathcal{O}(10^{-2})$ m, $\mathcal{O}(10^{-1})$ m, $\mathcal{O}(10^{-2})$ m, $\mathcal{O}(10^{-2})$ m$\cdot$sec^{-1}, and $\mathcal{O}(10^{-4})$ rad.

Next, Fig. 13 shows the number of mesh refinements, M, the number of collocation points, N, the number of mesh intervals, K, and the computation time, C, required to re-solve the optimal control problem every G time units for each guidance cycle $s \in [1, \ldots, S]$. It is seen that for most of the guidance cycles the optimal control problem is solved in less that 1 sec using the hp LGR collocation method. Moreover, it is seen that a minimal amount of mesh refinement is required through all of the guidance cycles in that all of the solutions are obtained on the first mesh with no mesh refinement except for the last guidance cycle. Finally, it is seen that the size of the mesh progressively decreases as the guidance cycles evolve such that in the last guidance cycle the mesh consists of 12 collocation points (that is, $N = 12$) and two mesh intervals (that is, $K = 2$). Thus, for the vast majority of guidance cycles the optimal control problem is not only solved efficiently, but the mesh size decreases making it possible to re-solve the optimal control problem using a progressively smaller finite-dimensional approximation of the optimal control problem.

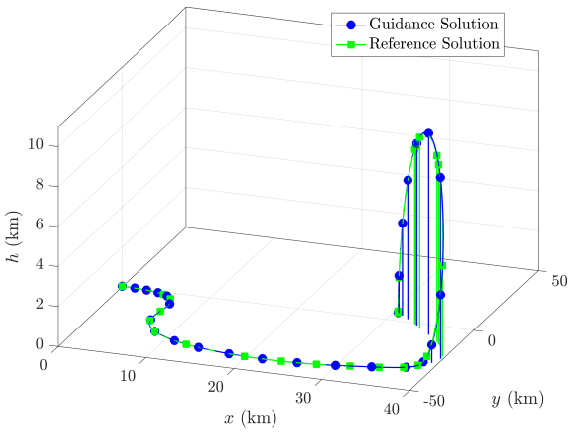
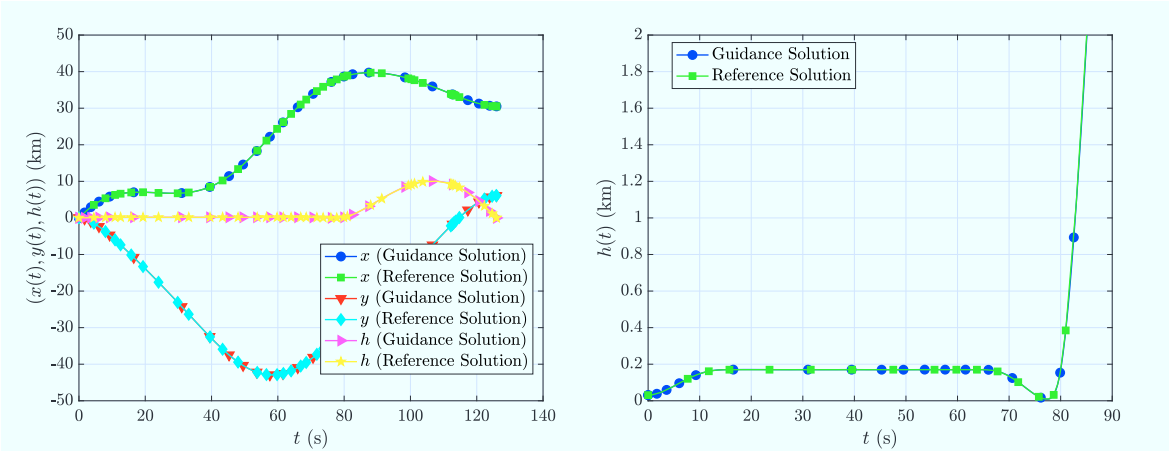
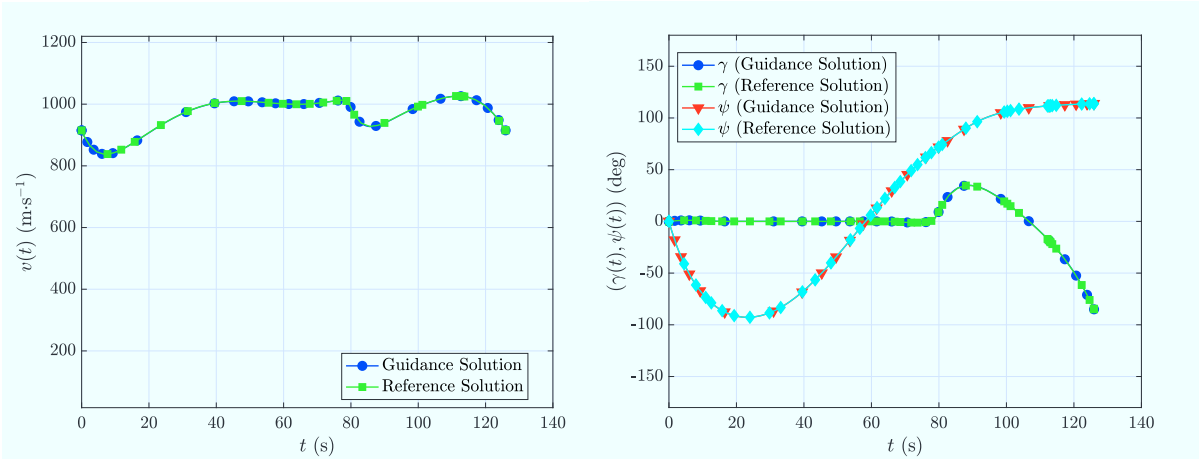


Figure 10: Three-Dimensional Trajectory Obtained via Numerical Integration of the Actual Dynamics as Given in Eq. (10) Over Each Guidance Cycle Using the Control Generated by Solving the Optimal Control Problem of Eqs. (1)–(5) in the Presence of the Time Delay.



(a) Downtrack, $x(t)$, crosstrack, $y(t)$, and altitude, $h(t)$, vs. t.

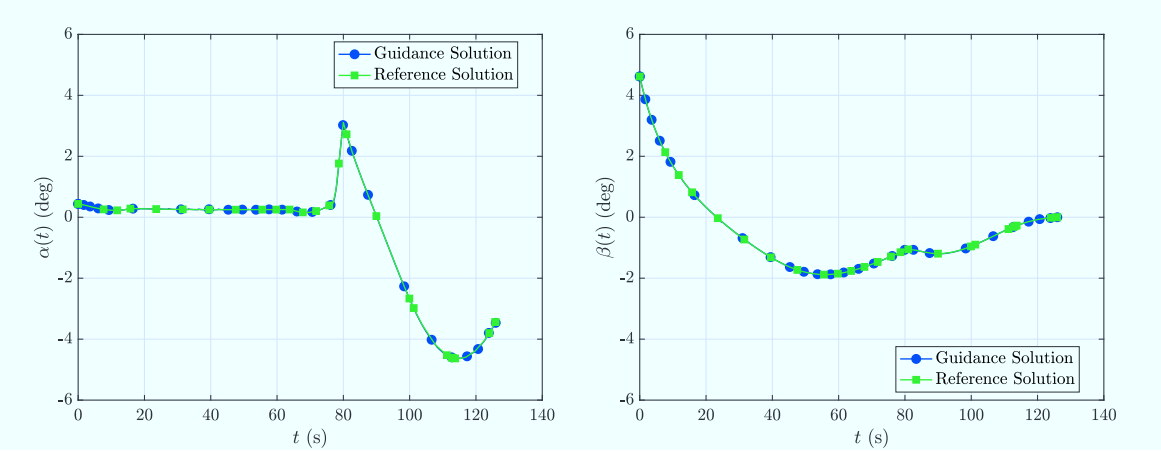
(b) Altitude, $h^{(1)}(t)$, vs. t.



(c) Speed, $v(t)$, vs. t.

(d) Flight path angle, $\gamma(t)$, and azimuth angle, $\psi(t)$, vs. t.

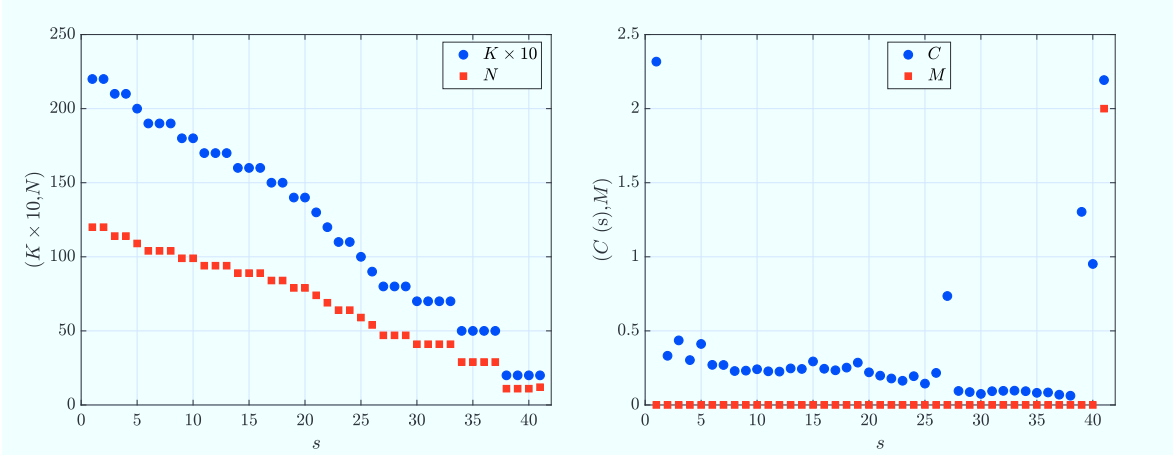
Figure 11: State Solution Obtained via Numerical Integration of the Actual Dynamics as Given in Eq. (10) Over Each Guidance Cycle Using the Control Generated by Solving the Optimal Control Problem of Eqs. (1)–(5) in the Presence of the Time Delay.



(a) Pitch angle, $\alpha(t)$, vs. t.

(b) Slip angle, $\beta(t)$, vs. t.

Figure 12: Control Solution Obtained via Numerical Integration of the Actual Dynamics as Given in Eq. (10) Over Each Guidance Cycle Using the Control Generated by Solving the Optimal Control Problem of Eqs. (1)–(5) in the Presence of the Time Delay.



(a) K, the number of mesh intervals, N, the number of collocation points, vs. s, guidance cycle number.

(b) C, the computational time, M, the number of mesh refinements, vs. s, guidance cycle number.

Figure 13: Mesh Characteristics of the Optimal Control Solution Over Each Guidance Cycle in the Presence of the Time Delay.

C. Discussion

This example illustrates key features of the computational optimal guidance and control method. The first key feature pertains to the size of the mesh. As the guidance cycle number increases, and the problem's horizon decreases, the size of the mesh decreases both in number of intervals and number of collocation points. As a result, the mesh generated on the current guidance cycle is smaller that the mesh on the previous guidance cycle as seen in Figs. 9a and 13a. Second, the problem requires no mesh refinement over all of the guidance cycles in the absence of a time delay and requires mesh refinement on only one guidance cycle in the presence of a time delay. Specifically, this can be seen in Figs. 9b and 13b. These results show that the mesh-generation strategy described in Section II provides a mesh at the start of the guidance cycle that is close to a mesh that satisfies the mesh refinement accuracy tolerance.

The second key feature pertains to the computational efficiency with which a solution is returned by the NLP solver that satisfies the mesh refinement accuracy tolerance. For the bunt maneuver problem, the guidance cycle was $G = 3$ s. It is seen from Figs. 9b and 13b that in all cases the time required to solve the sparse NLP is shorter than the guidance cycle. Specifically, the maximum times required to re-solve the sparse NLP in the absence and presence of the time delay are approximately 2.7 s and 2.3 s. However, for the majority of the guidance cycles, the average time to solve the sparse NLP is approximately 0.25 s (in the absence of a time delay) and approximately 0.3 s (in the presence of a time delay).

The third feature pertains to the accuracy with which the terminal conditions are attained via simulation of the actual (perturbed) dynamics. It can be seen in both solution sets that the state and control guidance solutions closely resemble the reference solution. The attained terminal conditions differ from the specified terminal conditions both in the absence and presence of a time delay by slightly less than one meter in downrange, crossrange, and altitude, slightly less than one meter per second in speed, and much less than one degree in flight path angle. These results indicate that the method applied is capable of compensating for the significant perturbations in the dynamic model relative to the reference model.

V. Conclusions

In the second part of this two-part study, the performance of this computational guidance and control law applied to a skid-to-turn vehicle model is assessed. The optimal control problem is posed as a decreasing-horizon two-phase bunt maneuver problem. The application of the method described is used to solve the decreasing-horizon problem. The mesh from the previous guidance cycle is used to solve the optimal control problem on the current guidance cycle; the mesh characteristics on each subsequent guidance cycle is reduced. The results shown in the absence and presence of a computation time delay for this method show that the method has the potential to solve optimal control problems in real-time using adaptive Gaussian quadrature collocation.

Acknowledgments

The authors acknowledge support for this research from the U.S. Office of Naval Research under grant N00014-15-1-2048, from the U.S. National Science Foundation under grants CBET-1404767, DMS-1522629, and CMMI-1563225, and from the Naval Air Warfare Center Weapons Division, China Lake, California under Contract No N68936-17-C-0011.

References

[1]Betts, J. T., *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, SIAM Press, Philadelphia, 2nd ed., 2009.

[2]Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, Vol. 47, No. 1, January 2002, pp. 99–131.

[3]Gill, P. E., Murray, W., and Saunders, M. A., *User's Guide for SNOPT Version 7: Software for Large Scale Nonlinear Programming*, February 2006.

[4]Biegler, L. T. and Zavala, V. M., "Large-Scale Nonlinear Programming Using IPOPT: An Integrating Framework for Enterprise-Wide Optimization," *Computers and Chemical Engineering*, Vol. 33, No. 3, March 2008, pp. 575–582.

[5]Benson, D. A., Huntington, G. T., Thorvaldsen, T. P., and Rao, A. V., "Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, November-December 2006, pp. 1435–

1440.

[6]Garg, D., Patterson, M. A., Darby, C. L., Francolin, C., Huntington, G. T., Hager, W. W., and Rao, A. V., "Direct Trajectory Optimization and Costate Estimation of Finite-Horizon and Infinite-Horizon Optimal Control Problems via a Radau Pseudospectral Method," *Computational Optimization and Applications*, Vol. 49, No. 2, June 2011, pp. 335–358. DOI: 10.1007/s10589–00–09291–0.

[7]Garg, D., Patterson, M. A., Hager, W. W., Rao, A. V., Benson, D. A., and Huntington, G. T., "A Unified Framework for the Numerical Solution of Optimal Control Problems Using Pseudospectral Methods," *Automatica*, Vol. 46, No. 11, November 2010, pp. 1843–1851. DOI: 10.1016/j.automatica.2010.06.048.

[8]Garg, D., Hager, W. W., and Rao, A. V., "Pseudospectral Methods for Solving Infinite-Horizon Optimal Control Problems," *Automatica*, Vol. 47, No. 4, April 2011, pp. 829–837. DOI: 10.1016/j.automatica.2011.01.085.

[9]Francolin, C. C., Hager, W. W., and Rao, A. V., "Costate Approximation in Optimal Control Using Integral Gaussian Quadrature Collocation Methods," *Optimal Control Applications and Methods*, Vol. 36, No. 4, July–August 2015, pp. 381–397.

[10]Darby, C. L., Hager, W. W., and Rao, A. V., "An hp-Adaptive Pseudospectral Method for Solving Optimal Control Problems," *Optimal Control Applications and Methods*, Vol. 32, No. 4, July–August 2011, pp. 476–502.

[11]Darby, C. L., Hager, W. W., and Rao, A. V., "Direct Trajectory Optimization Using a Variable Low-Order Adaptive Pseudospectral Method," *Journal of Spacecraft and Rockets*, Vol. 48, No. 3, May–June 2011, pp. 433–445.

[12]Patterson, M. A., Hager, W. W., and Rao, A. V., "A ph Mesh Refinement Method for Optimal Control," *Optimal Control Applications and Methods*, Vol. 36, No. 4, July–August 2015, pp. 398–421.

[13]Liu, F., Hager, W. W., and Rao, A. V., "Adaptive Mesh Refinement for Optimal Control Using Nonsmoothness Detection and Mesh Size Reduction," *Journal of the Franklin Institute*, Vol. 352, No. 10, October 2015, pp. 4081–4106.

[14]Liu, F., Hager, W. W., and Rao, A. V., "Adaptive Mesh Refinement for Optimal Control Using Using Decay Rates of Legendre Polynomial Coefficients," *IEEE Transactions on Control System Technology*, 2017, pp. 10.1109/TCST.2017.2702122.

[15]Hager, W. W., Hou, H., and Rao, A. V., "Lebesgue Constants Arising in a Class of Collocation Methods," *IMA Journal of Numerical Analysis*, Published Online: 24 December 2016. DOI: 10.1093/imanum/drw060.

[16]Hager, W. W., Hou, H., and Rao, A. V., "Convergence Rate for a Gauss Collocation Method Applied to Unconstrained Optimal Control," *Journal of Optimization Theory and Applications*, Vol. 169, 2016, pp. 801–824.

[17]Hager, W. W., Hou, H., and Rao, A. V., "Convergence Rate for a Radau Collocation Method Applied to Unconstrained Optimal Control," 2015, arXiv.org/abs/1508.03783.

[18]Hager, W. W., Hou, H., Mohapatra, S., and Rao, A. V., "Convergence Rate for an hp-Collocation Method Applied to Unconstrained Optimal Control," 2016, arXiv.org/abs/1605.02121.

[19]Hager, W. W., Mohapatra, S., and Rao, A. V., "Convergence Rate for a Gauss Collocation Method applied to Constrained Optimal Control," 2016, arXiv.org/abs/1607.02798.

[20]Farooq, A. and Limebeer, D. J. N., "Trajectory Optimization for Air-to-Surface Missiles with Imaging Radars," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 5, September–October 2002, pp. 876–887.

[21]Farooq, A. and Limebeer, D. J. N., "Optimal Trajectory Regulation for Radar Imaging Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 6, November–December 2005, pp. 1157–1170.

[22]Farooq, A. and Limebeer, D. J. N., "Optimal Trajectory Regulation for Radar Imaging Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 4, July–August 2008, pp. 1076–1092.

[23]Stengel, R. F., *Optimal Control and Estimation*, Dover Publications, Mineola, New York, 1994.

[24]Rao, A. V. and Hager, W. W., "Mesh-Generation Method for Real-Time Optimal Control Using Adaptive Gaussian Quadrature Collocation," *2018 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2018-0848, Kissimmee, Florida, 8 – 12 January 2018, https://arc.aiaa.org/doi/abs/10.2514/6.2018-0848.

[25]Patterson, M. A. and Rao, A. V., "$\mathbb{GPOPS}-\mathbb{II}$, A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming," *ACM Transactions on Mathematical Software*, Vol. 41, No. 1, October 2014, pp. 1:1–1:37.

[26]Weinstein, M. J. and Rao, A. V., "Algorithm: ADiGator, a Toolbox for the Algorithmic Differentiation of Mathematical Functions in MATLAB," *ACM Transactions on Mathematical Software*, Accepted for Publication, May 2017.