Engineering Notes

Analytic Costate Initialization from Rough State-Trajectory Estimates

Emmanuel E. Skamangas* and John A. Lawton†
Naval Surface Warfare Center, Dahlgren, Virginia 22448-5156
and
Jonathan T. Black‡
Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061

https://doi.org/10.2514/1.G005224

I. Introduction

IN THE development of optimal control solutions, there are two general approaches: the direct method and the indirect method. In the direct method, the control is directly manipulated (hence the name) in an iterative manner until an optimal result is achieved. In the indirect method, the iteration is applied to the costate or adjoint variables, on which the controls are dependent (through the necessary conditions of the maximum principle [1]), in an effort to achieve an optimal control. Both methods have strengths and weaknesses, and a significant body of knowledge exists for both methods. The direct method is generally more straightforward and can provide quick results provided that the initial guess for the optimal solution is sufficiently close to allow convergence. However, direct method approaches usually result in approximate optimal solutions. A significant body of work by Betts [2] and others [3–5] focuses on variations on the direct approach involving collocation. Indirect methods, on the other hand, provide very accurate solutions. Benson et al. [6] indicate that "The primary advantages of indirect methods are their high accuracy and the assurance that the solution satisfies first-order optimality conditions."

With regard to the indirect method, Betts [7], quoting Bryson and Ho [8], states, "The main difficulty with these methods is getting started ..." One of the difficulties to be addressed in using the indirect method for the solution of an optimal control problem is the necessity of obtaining estimates of the initial costates such that convergence to a solution is feasible and efficient. This primary difficulty lies in the fact that the arrival at an optimal solution is dependent on the selection of the initial costates, which unfortunately is not usually intuitive. Bushong and Lawton [9] develop estimates of the costates for an optimal trajectory by using a near-optimal parameterized solution. Jiang et al. [10] use a nominal trajectory to estimate the adjoint variables for a low-thrust application. Seywald and Kumar [11] discuss a method to estimate the costates for an optimal control problem using the indirect method using results obtained from a related discretized, direct-method solution. Lee and Bang [12] propose an initial guess structure for the costates based on known costate characteristics for the design of spiral orbit transfers. Martell and Lawton [13] provide a methodology that allows for estimation of the costates using finite differences obtained from a known trajectory and implemented through the use of an auxiliary optimization problem.

This Note revisits the approach of Martell and Lawton, replacing the auxiliary optimization problem with another auxiliary problem that has an analytic solution. Therefore, no numerical iterations are required for this present approach. The costate solutions from this new approach will then be used in a multiple-shooting method implementation to determine the optimal control for the same problem. It will be demonstrated that even with only rough estimates in the neighborhood of the optimal trajectory, this procedure converges, making it one more possible tool available to those solving optimal trajectory problems.

II. Optimal-Control Problem Statement

The focus of this Note is on finding estimates of costates, also known as adjoint variables, of the indirect method of solution, for a certain class of optimal control problems. The optimal control problem is, given the differential equations of the state, find the control-vector function of time $\boldsymbol{u}(t)$ that minimizes the cost function J.

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t) \tag{1}$$

(where $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector, $\boldsymbol{u} \in \mathbb{R}^m$ is the control vector, and t is time), the cost to be minimized

$$J(\boldsymbol{u}) = \phi(\boldsymbol{x}(t_f), t_f) + \int_{t_0}^{t_f} L(\boldsymbol{x}, \boldsymbol{u}, t)\,\mathrm{d}t \tag{2}$$

and the boundary conditions $\boldsymbol{\chi}(\boldsymbol{x}(t_0), t_0) = \boldsymbol{0}$ and $\boldsymbol{\psi}(\boldsymbol{x}(t_f), t_f) = \boldsymbol{0}$, find the control-vector function of time $\boldsymbol{u}(t)$ that minimizes the cost function J. The classic necessary condition for finding the optimal control, $\boldsymbol{u}$,

$$\boldsymbol{u}^* = \arg\max_{\boldsymbol{u}} J \tag{3}$$

is the maximum principle [1]. Specifically, the Hamiltonian is formed:

$$\mathcal{H} = \boldsymbol{\lambda}^T \boldsymbol{f} - L \tag{4}$$

where $\boldsymbol{\lambda}(t) \in \mathbb{R}^n$ is the costate vector. The costate governing differential equation is

$$\dot{\boldsymbol{\lambda}} = -\left(\frac{\partial \mathcal{H}}{\partial \boldsymbol{x}}\right)^T = -\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right)^T \boldsymbol{\lambda} + \left(\frac{\partial L}{\partial \boldsymbol{x}}\right)^T \tag{5}$$

The boundary conditions on $\boldsymbol{\lambda}$ are

$$\boldsymbol{\lambda}(t_0) = \left[\frac{\partial \boldsymbol{\chi}^T(\boldsymbol{x})}{\partial \boldsymbol{x}}\boldsymbol{\mu}\right]\Bigg|_{t=t_0} \tag{6}$$

$$\boldsymbol{\lambda}(t_f) = \left[-\frac{\partial \phi}{\partial \boldsymbol{x}} + \frac{\partial \boldsymbol{\psi}^T}{\partial \boldsymbol{x}}\boldsymbol{\nu}\right]\Bigg|_{t=t_f} \tag{7}$$

Presented as Paper 2020-0864 at the AIAA SciTech Forum 2020, Orlando, FL, January 6–10, 2020; received 2 March 2020; revision received 22 July 2021; accepted for publication 23 July 2021; published online 7 September 2021.

*Principal Engineer, Retired, Advanced Hypersonic Weapons and Guided Munitions Division, Dahlgren Division, E20. Member AIAA.

†Principal Engineer, Aerosciences Modeling and Analysis Branch, Dahlgren Division, E21. Senior Member AIAA.

‡Professor, Aerospace and Ocean Engineering (0203). Associate Fellow AIAA.

Given these definitions, the maximum principle states that the optimal $\boldsymbol{u}$ will satisfy the necessary condition

$$\boldsymbol{u}^* = \arg\max_{\boldsymbol{u}} \mathcal{H}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{u}, t) \tag{8}$$

where $\boldsymbol{x}^*(t)$ is the value of the state on the optimal trajectory at time t. In other words, at each time, on the optimal trajectory, the optimal control is the one that maximizes the Hamiltonian.

When the Hamiltonian is not linear in the control, $\boldsymbol{u}$, and there are no bounds on $\boldsymbol{u}$, then the maximum principle implies that

$$\frac{\partial \mathcal{H}}{\partial \boldsymbol{u}} = 0 \tag{9}$$

Equation (9) is solved for $\boldsymbol{u}$ to obtain the control on the optimal trajectory. As such, $\boldsymbol{u}$ is a function of the states and the costates on the optimal trajectory.

The class of problems to be solved by the technique of this Note is such that $(\partial \mathcal{H}/\partial \boldsymbol{u}) = 0$ is necessary throughout the whole trajectory. This is a large class of problems, to which many aerospace control problems belong.

The solution process requires the appropriate number of tranversality and state boundary conditions given by

$$\boldsymbol{\lambda}(t_0) = \left[\frac{\partial \boldsymbol{\chi}^T(\boldsymbol{x})}{\partial \boldsymbol{x}}\boldsymbol{\mu}\right]\Bigg|_{t=t_0} \tag{10}$$

$$\boldsymbol{\lambda}(t_f) = \left[-\frac{\partial \phi}{\partial \boldsymbol{x}} + \frac{\partial \boldsymbol{\psi}^T}{\partial \boldsymbol{x}}\boldsymbol{\nu}\right]\Bigg|_{t=t_f} \tag{11}$$

$$\boldsymbol{\chi}[\boldsymbol{x}(t_0), t_0] = \boldsymbol{0} \tag{12}$$

$$\boldsymbol{\psi}[\boldsymbol{x}(t_f), t_f] = \boldsymbol{0} \tag{13}$$

Additionally, if the final time is free, then the optimal t_f will be such that

$$\mathcal{H}(t_f) = \left[-\frac{\partial \phi}{\partial t} + \frac{\partial \boldsymbol{\psi}^T}{\partial t}\boldsymbol{\nu}\right]\Bigg|_{t=t_f} \tag{14}$$

The indirect method, in this case, has reduced the optimal control problem to the task of finding the parameters $\boldsymbol{\lambda}(t_0), \boldsymbol{\nu}, \boldsymbol{\mu}$, and t_f such that the conditions (10–13) are satisfied. The multiple-shooting method is one of the more successful methods for solving boundary-value problems of this type.

With an initial estimate for the costates, $\boldsymbol{\lambda}_0$, a guess for $\boldsymbol{\mu}$, the initial conditions, and the control from Eq. (9), one could numerically integrate the dynamic equations for the state and costates from time $t = 0$ forward to the estimate for t_f. This method, known as the simple-shooting method, suffers from the shortfall that small errors in the initial estimates will lead to uncontrollable growth of the costates as the integration progresses in time. An alternative to this approach, which specifically addresses this issue, is the multiple-shooting method. In this method, the trajectory is divided up into a number of nodes, and estimates for the costates are generated at each of the nodes. The integration proceeds as before, but now upon arriving at each node, the costates are "reset" to the estimates previously calculated for that node. The solution is found when the differences in the costates at all the nodes (i.e., between the forward-integrated values and the estimate values) are zero within a specified tolerance. In this manner, the number of nodes may be selected such that the costates do not diverge uncontrollably and the method converges to a solution.

III. Method for Estimating the Costates from In-the-Neighborhood Trajectories

We start with the finite difference method used in [13]. The adjoint variable differential equations are written as

$$\dot{\boldsymbol{\lambda}} = [\boldsymbol{A}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{u}}, t)]\boldsymbol{\lambda} + \frac{\partial L}{\partial x}\bigg|_{\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{u}}, t} \tag{15}$$

where

$$\boldsymbol{A} = -\left(\frac{\partial f}{\partial \boldsymbol{x}}\right)^T \tag{16}$$

and the tildes ($\sim$) denote values on the estimated trajectory.

Equation (15) can then be approximated by

$$[\boldsymbol{I}]\boldsymbol{\lambda}_{i+1} - [\boldsymbol{I} + \boldsymbol{A}_i(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{u}}_i, t)\Delta t_i]\tilde{\boldsymbol{\lambda}}_i = \Delta t_i \frac{\partial L}{\partial \boldsymbol{x}}\bigg|_{\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{u}}_i, t_i}; \quad i = 1, \ldots, w-1 \tag{17}$$

The time between nodes, which may vary from panel to panel, is given by

$$\Delta t_i = t_{i+1} - t_i \tag{18}$$

In matrix notation, Eq. (17) becomes

$$\begin{bmatrix} \boldsymbol{\alpha}_1 & \boldsymbol{I} & \boldsymbol{0} & \ldots & \ldots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\alpha}_2 & \boldsymbol{I} & \ldots & \ldots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \ldots & \boldsymbol{\alpha}_{w-2} & \boldsymbol{I} & \vdots \\ \vdots & \vdots & \ldots & \ldots & \boldsymbol{\alpha}_{w-1} & \boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} & \ldots & \ldots & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \\ \vdots \\ \boldsymbol{\lambda}_{w-1} \\ \boldsymbol{\lambda}_w \end{Bmatrix} = \begin{Bmatrix} \Delta t_1 \dfrac{\partial L}{\partial \boldsymbol{x}}\Big|_1 \\ \Delta t_2 \dfrac{\partial L}{\partial \boldsymbol{x}}\Big|_2 \\ \vdots \\ \Delta t_{w-1} \dfrac{\partial L}{\partial \boldsymbol{x}}\Big|_{w-1} \\ \boldsymbol{\lambda}(t_f) \end{Bmatrix} \tag{19}$$

where each $n \times n$ submatrix $\boldsymbol{\alpha}_i$ is given by

$$\alpha_i = -(\boldsymbol{I} + \boldsymbol{A}_i \Delta t_i) \tag{20}$$

(The reader should note that these matrix quantities, $\boldsymbol{\alpha}_i$, are not the angle of attack.)

Inverting the matrix on the left side of Eq. (19) and multiplying to both sides of the same equation yields

$$\begin{Bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \\ \vdots \\ \vdots \\ \boldsymbol{\lambda}_{w-1} \\ \boldsymbol{\lambda}_w \end{Bmatrix} = \begin{bmatrix} \boldsymbol{\alpha}_1^{-1} & -\boldsymbol{\alpha}_1^{-1}\boldsymbol{\alpha}_2^{-1} & \boldsymbol{\alpha}_1^{-1}\boldsymbol{\alpha}_2^{-1}\boldsymbol{\alpha}_3^{-1} & -\boldsymbol{\alpha}_1^{-1}\boldsymbol{\alpha}_2^{-1}\boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_4^{-1} & \ldots & \pm\boldsymbol{\alpha}_1^{-1}\boldsymbol{\alpha}_2^{-1}\ldots\boldsymbol{\alpha}_{w-1}^{-1} \\ \boldsymbol{0} & \boldsymbol{\alpha}_2^{-1} & -\boldsymbol{\alpha}_2^{-1}\boldsymbol{\alpha}_3^{-1} & \boldsymbol{\alpha}_2^{-1}\boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_4^{-1} & \ldots & \mp\boldsymbol{\alpha}_2^{-1}\boldsymbol{\alpha}_3^{-1}\ldots\boldsymbol{\alpha}_{w-1}^{-1} \\ 0 & 0 & \alpha_3^{-1} & -\alpha_3^{-1}\alpha_4^{-1} & \ldots & \pm\alpha_3^{-1}\alpha_4^{-1}\ldots\alpha_{w-1}^{-1} \\ \vdots & \vdots & \ldots & \ldots & \ldots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \ldots & \ldots & \alpha_{w-1}^{-1} & -\alpha_{w-1}^{-1} \\ \boldsymbol{0} & \boldsymbol{0} & \ldots & \ldots & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{Bmatrix} \Delta t_1 \dfrac{\partial L}{\partial \boldsymbol{x}}\Big|_1 \\ \Delta t_2 \dfrac{\partial L}{\partial \boldsymbol{x}}\Big|_2 \\ \vdots \\ \Delta t_{w-1} \dfrac{\partial L}{\partial \boldsymbol{x}}\Big|_{w-1} \\ \boldsymbol{\lambda}(t_f) \end{Bmatrix} \tag{21}$$

A brief note with regard to the signs of the terms in the last column of the matrix on the right-hand side of Eq. (21): The $w-1$ term in that column always has a negative sign associated with it. From there, the terms alternate in sign upward (i.e., the $w-2$ term is positive, the $w-3$ term is negative, and so on).

Multiplying the right side of Eq. (21) results in

$$\begin{Bmatrix} \lambda_1 \\ \lambda_2 \\ : \\ : \\ \lambda_{w-1} \\ \lambda_w \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{\alpha}_1^{-1}\Delta t_1 \dfrac{\partial L}{\partial \boldsymbol{x}}|_1 - \boldsymbol{\alpha}_1^{-1}\boldsymbol{\alpha}_2^{-1}\Delta t_2 \dfrac{\partial L}{\partial \boldsymbol{x}}\Big|_2 + \ldots \pm \boldsymbol{\alpha}_1^{-1}\boldsymbol{\alpha}_2^{-1}\ldots\boldsymbol{\alpha}_{w-1}^{-1}\boldsymbol{\lambda}(t_f) \\ \boldsymbol{\alpha}_2^{-1}\Delta t_2 \dfrac{\partial L}{\partial \boldsymbol{x}}\Big|_2 - \ldots \mp \boldsymbol{\alpha}_2^{-1}\boldsymbol{\alpha}_3^{-1}\ldots\boldsymbol{\alpha}_{w-1}^{-1}\boldsymbol{\lambda}(t_f) \\ : \\ : \\ \boldsymbol{\alpha}_{w-1}-1\Delta t_{w-1}\dfrac{\partial L}{\partial \boldsymbol{x}}\Big|_{w-1} - \boldsymbol{\alpha}_{w-1}^{-1}\boldsymbol{\lambda}_(t_f) \\ \boldsymbol{\lambda}(t_f) \end{Bmatrix} \tag{22}$$

Given that

$$\boldsymbol{\lambda}(t_f) = \left(\frac{\partial \boldsymbol{\phi}^T}{\partial \boldsymbol{x}} + \frac{\partial \boldsymbol{\psi}^T}{\partial \boldsymbol{x}}\boldsymbol{\nu}\right) \tag{23}$$

the costates are given by

$$\begin{Bmatrix} \lambda_1 \\ \lambda_2 \\ : \\ : \\ \lambda_{w-1} \\ \lambda_w \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{\alpha}_1^{-1}\Delta t_1 \dfrac{\partial L}{\partial \boldsymbol{x}}\Big|_1 - \ldots \mp \boldsymbol{\alpha}_1^{-1}\boldsymbol{\alpha}_2^{-1}\ldots\boldsymbol{\alpha}_{w-2}^{-1}\Delta t_{w-1}\dfrac{\partial L}{\partial \boldsymbol{x}}\Big|_{w-1} \\ -\boldsymbol{\alpha}_2^{-1}\Delta t_2 \dfrac{\partial L}{\partial \boldsymbol{x}}\Big|_2 + \ldots \pm \boldsymbol{\alpha}_2^{-1}\boldsymbol{\alpha}_3^{-1}\ldots\boldsymbol{\alpha}_{w-2}^{-1}\Delta t_{w-1}\dfrac{\partial L}{\partial \boldsymbol{x}}\Big|_{w-1} \\ : \\ : \\ \boldsymbol{\alpha}_{w-1}^{-1}\Delta t_{w-1}\dfrac{\partial L}{\partial \boldsymbol{x}}\Big|_{w-1} \\ \boldsymbol{0} \end{Bmatrix} + \begin{Bmatrix} \pm\boldsymbol{\alpha}_1^{-1}\boldsymbol{\alpha}_2^{-1}\ldots\boldsymbol{\alpha}_{w-1}^{-1} \\ \mp\boldsymbol{\alpha}_2^{-1}\boldsymbol{\alpha}_3^{-1}\ldots\boldsymbol{\alpha}_{w-1}^{-1} \\ : \\ : \\ -\boldsymbol{\alpha}_{w-1}^{-1} \\ \boldsymbol{I} \end{Bmatrix} \left(\frac{\partial \boldsymbol{\phi}^T}{\partial \boldsymbol{x}} + \frac{\partial \boldsymbol{\psi}^T}{\partial \boldsymbol{x}}\boldsymbol{\nu}\right)\Big|_{t=t_f} \tag{24}$$

which can be rewritten as

$$\begin{Bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \\ : \\ : \\ \boldsymbol{\lambda}_{w-1} \\ \boldsymbol{\lambda}_w \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \\ : \\ : \\ \boldsymbol{d}_{w-1} \\ \boldsymbol{0} \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{S}_1 \\ \boldsymbol{S}_2 \\ : \\ : \\ \boldsymbol{S}_{w-1} \\ \boldsymbol{I} \end{Bmatrix} \left(\frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}_f}^T + \frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}_f}^T \boldsymbol{\nu}\right) \tag{25}$$

$$= \begin{Bmatrix} \boldsymbol{d}_1 + \boldsymbol{S}_1\dfrac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}_f}^T + \boldsymbol{S}_1\dfrac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}_f}^T\boldsymbol{\nu} \\ \boldsymbol{d}_2 + \boldsymbol{S}_2\dfrac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}_f}^T + \boldsymbol{S}_2\dfrac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}_f}^T\boldsymbol{\nu} \\ : \\ : \\ \boldsymbol{d}_{w-1} + \boldsymbol{S}_{w-1}\dfrac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}_f}^T + \boldsymbol{S}_{w-1}\dfrac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}_f}^T\boldsymbol{\nu} \\ \dfrac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}_f}^T + \dfrac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}_f}^T\boldsymbol{\nu} \end{Bmatrix} \tag{26}$$

Now, if we add the initial condition function $\boldsymbol{\chi}$ at $t = 0$, transversality gives

$$\boldsymbol{\lambda}_1 = \frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{x}}\Big|^T_{\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{u}}_1, t_1} \boldsymbol{\mu} \tag{27}$$

Therefore,

$$\frac{\partial \boldsymbol{\chi}^T}{\partial \boldsymbol{x}}\boldsymbol{\mu} = \boldsymbol{d}_1 + \boldsymbol{S}_1\frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}_f}^T + \boldsymbol{S}_1\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}_f}^T\boldsymbol{\nu} \tag{28}$$

where $\boldsymbol{S}_1 = \pm\boldsymbol{\alpha}_1^{-1}\boldsymbol{\alpha}_2^{-1}\ldots\boldsymbol{\alpha}_{w-1}^{-1}$. We now need to find a solution for $\boldsymbol{\nu}$. Our motivation is the maximum principle, where the optimal control (for the class of problems that are the focus of this Note) will be the one that results in $(\partial\mathcal{H}/\partial\boldsymbol{u}) = 0$. By using $\boldsymbol{x}$ and $\boldsymbol{u}$ along nodes, we seek to find costates that leave $\partial\mathcal{H}/\partial\boldsymbol{u}$ as close to 0 as possible consistent with these data. With this in mind, we introduce the following auxiliary function for which a minimum will be found:

$$Q(\boldsymbol{\nu}) = \sum_{i=1}^{w}\frac{1}{2}\frac{\partial\mathcal{H}}{\partial\boldsymbol{u}}\frac{\partial\mathcal{H}}{\partial\boldsymbol{u}}^T \tag{29}$$

By $\partial\mathcal{H}/\partial\boldsymbol{u}$ in this expression we mean this partial derivative as evaluated at each of the nodes (i.e., evaluated at t_i), $(\partial\mathcal{H}/\partial\boldsymbol{u})|_{t_i}$. This is the main difference between the approach taken in [13] and this approach; this auxiliary function replaces the one in [13] and allows for an analytical solution.

Evaluating this partial derivative at the nodes gives

$$\frac{\partial\mathcal{H}}{\partial\boldsymbol{u}}\Big|_{t_i} = \boldsymbol{\lambda}^T\frac{\partial\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial\boldsymbol{u}}\Big|_{t_i} \tag{30}$$

$$= \hat{\boldsymbol{\lambda}}_i(\boldsymbol{\nu})^T\frac{\partial\boldsymbol{f}(\tilde{\boldsymbol{x}}_i,\tilde{\boldsymbol{u}}_i)}{\partial\boldsymbol{u}} \tag{31}$$

$$= \left[\boldsymbol{d}_i^T + \frac{\partial\boldsymbol{\phi}}{\partial\boldsymbol{x}_f}\boldsymbol{S}_i^T + \boldsymbol{\nu}^T\frac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{x}_f}\boldsymbol{S}_i^T\right]\boldsymbol{B}_i \tag{32}$$

where

$$\boldsymbol{B}_i = \frac{\partial\boldsymbol{f}(\tilde{\boldsymbol{x}}_i,\tilde{\boldsymbol{u}}_i)}{\partial\boldsymbol{u}} \tag{33}$$

Substituting Eqs. (30–33) into Eq. (29) results in

$$Q(\boldsymbol{\nu}) = \frac{1}{2}\sum_{i=1}^{w}\left[\boldsymbol{d}_i^T + \left(\frac{\partial\boldsymbol{\phi}}{\partial\boldsymbol{x}_f} + \boldsymbol{\nu}^T\frac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{x}_f}\right)\boldsymbol{S}_i^T\right] \boldsymbol{B}_i\boldsymbol{B}_i^T\left[\boldsymbol{d}_i + \boldsymbol{S}_i\left(\frac{\partial\boldsymbol{\phi}}{\partial\boldsymbol{x}_f}^T + \frac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{x}_f}^T\boldsymbol{\nu}\right)\right] \tag{34}$$

Taking the partial of J with respect to z and setting equal to 0 yields

$$\frac{\partial Q}{\partial \boldsymbol{\nu}} = \sum_{i=1}^{w}\left[\boldsymbol{d}_i^T + \left(\frac{\partial \phi}{\partial \boldsymbol{x}_f} + \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}_f}\right)\boldsymbol{S}_i^T\right]\boldsymbol{B}_i\boldsymbol{B}_i^T\boldsymbol{S}_i\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}_f}^T = \boldsymbol{0} \quad (35)$$

which has the solution

$$\boldsymbol{\nu} = -\left[\frac{\partial \boldsymbol{\psi}}{\partial x_f}\left(\sum_{i=1}^{w}\boldsymbol{S}_i^T\boldsymbol{B}_i\boldsymbol{B}_i^T\boldsymbol{S}_i\right)\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}_f}^T\right]^{-1} \cdot\left[\sum_{i=1}^{w}\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}_f}\boldsymbol{S}_i^T\boldsymbol{B}_i\boldsymbol{B}_i^T\boldsymbol{d}_i + \frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}_f}\left(\sum_{i=1}^{w}\boldsymbol{S}_i^T\boldsymbol{B}_i\boldsymbol{B}_i^T\boldsymbol{S}_i\right)\frac{\partial \phi}{\partial \boldsymbol{x}_f}^T\right] \quad (36)$$

Note that the summation $\sum_{i=1}^{w}\boldsymbol{S}_i^T\boldsymbol{B}_i\boldsymbol{B}_i^T\boldsymbol{S}_i$ is the finite-difference version of the controllability grammian. So, although each term of the summation is of rank $m < n$, the sum will be of rank n if the problem is controllable. This helps to ensure that the first term in Eq. (36) is invertible for problems that make sense (i.e., which are completely controllable). Equation (36) replaces the auxiliary optimization procedure of [13].

IV. Multiple Shooting Method

With estimates of the initial costates in hand, and along with the state and costate dynamics (and the initial conditions on the state), it is a simple matter to integrate the states and costates forward in time and compare the integrated values of the states and costates with the final conditions. This method, known as the simple shooting method, then allows the initial costates to be adjusted such that the differences between the integrated states and costates and the final conditions are negligible. However, the costates are often highly sensitive to the initial estimates; this sensitivity results in the costates being unstable. As such, minor changes in their initial values can lead to quickly diverging results.

To address this issue, we applied the multiple shooting method. In this method, the trajectory is divided into w nodes, and the values of the costates are calculated at each of the nodes using Eq. (26); these values are stored and will be used later to "reset" the costates to prevent unstable behavior. Figure 1a shows a notional depiction of the application of the multiple shooting method for the initial integration of λ_x. At t_1, λ_x is integrated forward until t_2 using Eq. (5). At this point, the value of λ_x is reset to the value of λ_x at t_2 computed and stored earlier. The integration continues to t_3, where the process is repeated until reaching the final node. Using a nonlinear system-of-equations solver, the costate estimates at each node are adjusted and stored, and the integration is executed again. The process continues until the differences $\Delta\lambda_x$, also known as defects, between the integrated and computed (stored) values, are driven to zero. Figure 1b shows the final result of this process for λ_x by using the nonlinear system-of-equations solver and iteratively integrating to determine the optimal trajectory.

To employ the solver, we established a vector of inputs, $\boldsymbol{p}$, and an associated vector of dependent values, $\boldsymbol{F}(\boldsymbol{p})$. The solver functions by manipulating the vector of inputs to drive each element of $\boldsymbol{F}(\boldsymbol{p})$ to zero (within a user-defined tolerance). Upon convergence of the solver, the final elements of $\boldsymbol{p}$ will be the solution to the system of linear equations.

The specific contents of $\boldsymbol{p}$ and $\boldsymbol{F}(\boldsymbol{p})$ will be discussed in Sec. VI; $\boldsymbol{p}$ chiefly contains the estimated costates at the nodes, plus other problem-specific parameters, whereas $\boldsymbol{F}(\boldsymbol{p})$ chiefly contains the defects at the nodes, plus additional conditions such as the enforcement of initial-time boundary conditions on the state and costate.

V. Illustrative Problem

The method discussed above is now applied to the problem of a projectile with a single control, α, the angle of attack, which is defined as the angle between the projectile body axis and the velocity vector (see Fig. 2). The projectile is fired from the surface of the Earth with a launch angle θ_0, which is measured from the vertical, at an initial velocity v_0. We will consider planar motion in a nonrotating, Earth-centered coordinate system. The projectile is fired at a target with a fixed location on the Earth.

We define the dynamics as follows:

$$\boldsymbol{x} = \begin{Bmatrix}\boldsymbol{r}\\ \boldsymbol{v}\end{Bmatrix} \quad (37)$$

where $\boldsymbol{r}$ is the two-dimensional position vector from the center of the Earth, and

$$\dot{\boldsymbol{x}} = \begin{Bmatrix}\boldsymbol{v}\\ \boldsymbol{a}\end{Bmatrix} \quad (38)$$

where $\boldsymbol{a}$, the acceleration, is given by

$$\boldsymbol{a} = (\boldsymbol{A} + \boldsymbol{N})/m + \boldsymbol{g} \quad (39)$$

where $\boldsymbol{A}$ is the axial force, $\boldsymbol{N}$ is the normal force, m is the projectile mass, and $\boldsymbol{g}$ is the acceleration due to gravity. $\boldsymbol{A}$ is aligned with the projectile body and its magnitude is given by

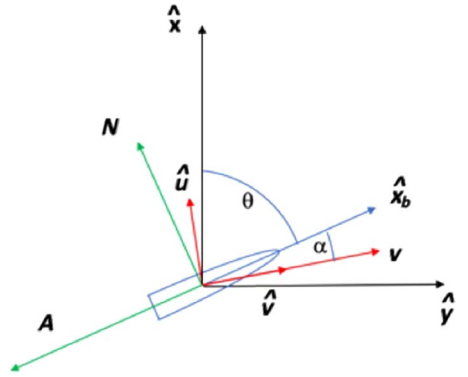
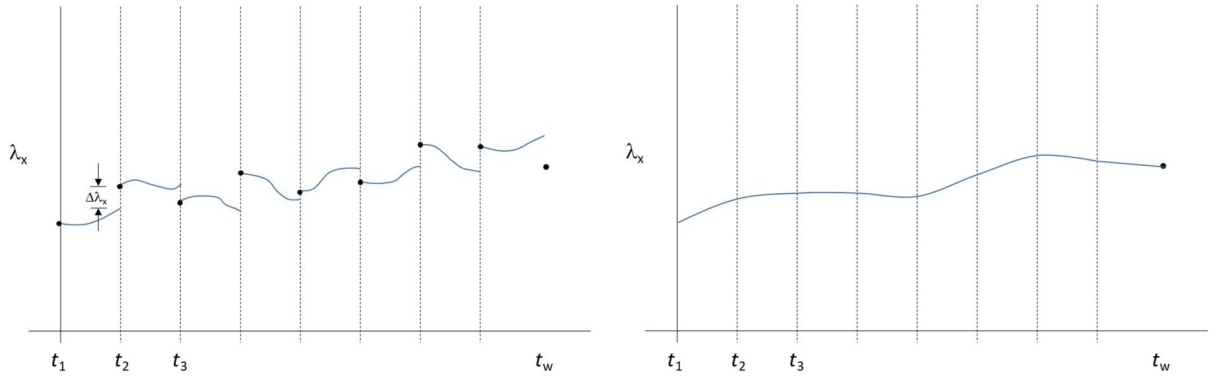


Fig. 2 Coordinate system showing projectile body orientation.



a) Nonconverged solution **b) Converged solution**

Fig. 1 λ_x for initial and final values of $\boldsymbol{p}$.

$$\|\boldsymbol{A}\| = \frac{1}{2}\rho v^2 \mathcal{S} C_A \tag{40}$$

where ρ is the density, v is the magnitude of the velocity, $\mathcal{S}$ is the reference area of the projectile, and C_A is the constant coefficient of the axial force. $\boldsymbol{N}$ is aligned perpendicular to the projectile body and its magnitude is given by

$$\|\boldsymbol{N}\| = \frac{1}{2}\rho v^2 \mathcal{S} C_{N_\alpha}\alpha \tag{41}$$

where C_{N_α} is the gradient of the coefficient of the normal force with respect to α. Both C_A and C_{N_α} are aerodynamic characteristics of the projectile. The initial conditions are given by

$$\boldsymbol{\chi} = \begin{Bmatrix} \boldsymbol{r}_0 - \boldsymbol{r}_L \\ \boldsymbol{v}_0 - \|\boldsymbol{v}_L\| \begin{bmatrix} \cos(\theta_0) \\ \sin(\theta_0) \end{bmatrix} \end{Bmatrix} \tag{42}$$

where the subscript 0 denotes the initial position and velocity, and L denotes the launch position and velocity. The final conditions are given by

$$\boldsymbol{\psi} = \{\boldsymbol{r}(t_f) - \boldsymbol{r}_T\} \tag{43}$$

where the subscript f denotes the final time and T denotes the target position.

We define the cost function as

$$J = -v^2(t_f) \tag{44}$$

and we construct the Hamiltonian to yield

$$\mathcal{H} = \boldsymbol{\lambda}_{\boldsymbol{r}}^T \boldsymbol{v} + \boldsymbol{\lambda}_{\boldsymbol{v}}^T \boldsymbol{a} \tag{45}$$

where $\boldsymbol{\lambda}_{\boldsymbol{r}}$ and $\boldsymbol{\lambda}_{\boldsymbol{v}}$ are the costates of the position and velocity, respectively. Taking the partial derivative of $\mathcal{H}$ with respect to the control α and setting equal to zero gives

$$\frac{\partial \mathcal{H}}{\partial \alpha} = \boldsymbol{\lambda}_{\boldsymbol{v}}^T \frac{\partial \boldsymbol{a}}{\partial \alpha} = 0 \tag{46}$$

To find $\partial \boldsymbol{a}/\partial\alpha$, we write $\hat{\boldsymbol{N}}$ and $\hat{\boldsymbol{A}}$ as

$$\hat{\boldsymbol{N}} = \hat{\boldsymbol{u}}\cos\alpha - \hat{\boldsymbol{v}}\sin\alpha \tag{47}$$

and

$$\hat{\boldsymbol{A}} = -\hat{\boldsymbol{u}}\sin\alpha - \hat{\boldsymbol{v}}\cos\alpha \tag{48}$$

where $\hat{\boldsymbol{v}}$ is the unit vector along the velocity vector and $\hat{\boldsymbol{u}}$ is perpendicular to $\hat{\boldsymbol{v}}$ in the "up" direction.

Defining a_N and a_A to be $\|\boldsymbol{N}\|/m$ and $\|\boldsymbol{A}\|/m$, respectively, we obtain

$$\boldsymbol{a}_{\text{aero}} = -a_N \sin\alpha\hat{\boldsymbol{v}} + a_N \cos\alpha\hat{\boldsymbol{u}} - a_A \cos\alpha\hat{\boldsymbol{v}} + a_A \sin\alpha\hat{\boldsymbol{u}} \tag{49}$$

Recognizing that $(\partial \boldsymbol{g}/\partial\alpha) = \boldsymbol{0}$ and therefore $(\partial \boldsymbol{a}/\partial\alpha) = \partial \boldsymbol{a}_{\text{aero}}/\partial\alpha$,

$$\begin{aligned}\boldsymbol{\lambda}_{\boldsymbol{v}}^T \frac{\partial \boldsymbol{a}}{\partial \alpha} = &\left[\left(-\frac{\partial a_N}{\partial \alpha} + a_A\right)\sin\alpha - \left(a_N + \frac{\partial a_A}{\partial \alpha}\right)\cos\alpha\right]\boldsymbol{\lambda}_{\boldsymbol{v}}^T\hat{\boldsymbol{v}} \\ &+ \left[\left(\frac{\partial a_N}{\partial \alpha} - a_A\right)\cos\alpha - \left(a_N + \frac{\partial a_A}{\partial \alpha}\right)\sin\alpha\right]\boldsymbol{\lambda}_{\boldsymbol{v}}^T\hat{\boldsymbol{u}} = 0\end{aligned} \tag{50}$$

Rearranging yields

$$\begin{aligned}&\left[\left(-\frac{\partial a_N}{\partial \alpha} + a_A\right)\boldsymbol{\lambda}_{\boldsymbol{v}}^T\hat{\boldsymbol{v}} - \left(a_N + \frac{\partial a_A}{\partial \alpha}\right)\boldsymbol{\lambda}_{\boldsymbol{v}}^T\hat{\boldsymbol{u}}\right]\sin\alpha \\ &\quad = \left[\left(a_N + \frac{\partial a_A}{\partial \alpha}\right)\boldsymbol{\lambda}_{\boldsymbol{v}}^T\hat{\boldsymbol{v}} + \left(-\frac{\partial a_N}{\partial \alpha} + a_A\right)\boldsymbol{\lambda}_{\boldsymbol{v}}^T\hat{\boldsymbol{u}}\right]\cos\alpha\end{aligned} \tag{51}$$

which can be solved for $\tan\alpha$:

$$\tan\alpha = \frac{\left[\left(a_N + \frac{\partial a_A}{\partial \alpha}\right)\boldsymbol{\lambda}_{\boldsymbol{v}}^T\hat{\boldsymbol{v}} + \left(-\frac{\partial a_N}{\partial \alpha} + a_A\right)\boldsymbol{\lambda}_{\boldsymbol{v}}^T\hat{\boldsymbol{u}}\right]}{\left[\left(-\frac{\partial a_N}{\partial \alpha} + a_A\right)\boldsymbol{\lambda}_{\boldsymbol{v}}^T\hat{\boldsymbol{v}} - \left(a_N + \frac{\partial a_A}{\partial \alpha}\right)\boldsymbol{\lambda}_{\boldsymbol{v}}^T\hat{\boldsymbol{u}}\right]} \tag{52}$$

We thus have an expression for the optimal control α, which is determined from the costates, which we can obtain using the method outlined in the previous section.

To solve for α from the adjoint variables, we express a_N and a_A as

$$a_N = \frac{q_\infty S}{m} C_{N_\alpha} \cdot \alpha, \; a_A = \frac{q_\infty S}{m} C_A \tag{53}$$

Taking partials with respect to the control α gives

$$\frac{\partial a_N}{\partial \alpha} = \frac{q_\infty S}{m} C_{N_\alpha}, \frac{\partial a_A}{\partial \alpha} = 0 \tag{54}$$

Combining these with Eq. (52) yields

$$\tan\alpha = \frac{C_{N_\alpha}\alpha\boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{v}} + (C_A - C_{N_\alpha})\boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{u}}}{(C_A - C_{N_\alpha})\boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{v}} - C_{N_\alpha}\alpha\boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{u}}} \tag{55}$$

We will now use Newton's method to solve for α. Rearranging Eq. (55) provides us with $F(\alpha)$

$$\begin{aligned}F(\alpha) = &\boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{v}}(C_A - C_{N_\alpha})\sin\alpha - \boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{u}} C_{N_\alpha} \cdot \alpha\sin\alpha \\ &- \boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{v}} C_{N_\alpha} \cdot \alpha\cos\alpha + \boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{u}}(C_{N_\alpha} - C_A)\cos\alpha = 0\end{aligned} \tag{56}$$

while the derivative with of $F(\alpha)$ with respect to α is

$$\begin{aligned}F'(\alpha) = &\boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{v}}(C_A - C_{N_\alpha})\cos\alpha - \boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{u}} C_{N_\alpha} \cdot \sin\alpha - \boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{u}} C_{N_\alpha} \cdot \alpha\cos\alpha \\ &- \boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{v}} C_{N_\alpha} \cdot \cos\alpha + \boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{v}} C_{N_\alpha} \cdot \alpha\sin\alpha - \boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{u}}(C_{N_\alpha} - C_A)\sin\alpha\end{aligned} \tag{57}$$

Each new estimate of α is obtained from

$$\alpha_{k+1} = \alpha_k - \frac{F(\alpha_k)}{F'(\alpha_k)} \tag{58}$$

To determine an initial guess for $\alpha_{k=0}$, we apply the small angle approximation to Eq. (56), which reduces to

$$\begin{aligned}&-\boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{u}} C_{N_\alpha} \cdot \alpha_{k=0}^2 + \boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{v}}(C_A - 2C_{N_\alpha})\alpha_{k=0} \\ &\quad + \boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{u}}(C_{N_\alpha} - C_A) = 0\end{aligned} \tag{59}$$

which results in an initial guess of

$$\begin{aligned}&\alpha_{k=0} \\ &= \frac{-\boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{v}}(C_A - 2C_{N_\alpha}) \pm \sqrt{(\boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{v}})^2(C_A - 2C_{N_\alpha})^2 - 4(\boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{u}})^2 C_{N_\alpha}(C_{N_\alpha} - C_A)}}{2\boldsymbol{\lambda}_{\boldsymbol{v}} \cdot \hat{\boldsymbol{u}} C_{N_\alpha}}\end{aligned} \tag{60}$$

VI. Application of the Methodology

To demonstrate the utility of this methodology, two cases are presented where the reference trajectories were generated using two different methods. In the first case, an approximate optimal trajectory was obtained from previously developed software employing a direct method. In the second case, a purely ballistic (i.e., $\alpha = 0$) depressed trajectory flown to the same range was used. The point of using the ballistic trajectory is that the reference trajectories do not need to be nearly optimal; they just need to be of the right shape to produce estimates for the costates that are close enough to permit convergence. In each case, the reference trajectory was then divided into $w - 1$ panels (defined by w nodes); 20 nodes were used in this problem. Using Eq. (36), a value for $\boldsymbol{\nu}$ was calculated and subsequently Eq. (26) was used to determine the $\boldsymbol{\lambda}$s at each of the nodes. With the $\boldsymbol{\lambda}$s in hand, Eqs. (56) and (59) were then used to calculate the control α.

In the first case, previously developed, direct-method software was executed to obtain an approximate optimal trajectory. The trajectory was divided into the same w nodes and the software executed to produce a control history (at the nodes) that would result in a near-optimal trajectory maximizing impact velocity. The output control history, at the nodes, was then used in a simulation based on the indirect method discussed previously. The simulation begins by recreating the approximate optimal trajectory and then produces the estimates for the costates. With these in hand, the dependent vector $\boldsymbol{F}(\boldsymbol{p})$ is iterated on by adjusting the input vector $\boldsymbol{p}$ until the values in $\boldsymbol{F}(\boldsymbol{p})$ are reduced to the tolerance of convergence.

For this specific application, the independent vector $\boldsymbol{p} \in \mathbb{R}^{n_p}$, where $n_p = 4w$, is defined by

$$\boldsymbol{p} = \begin{Bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \\ \vdots \\ \boldsymbol{\lambda}_{w-1} \\ \boldsymbol{\nu} \\ \mu \\ \theta_0 \end{Bmatrix} \tag{61}$$

where each $\boldsymbol{\lambda}_i$ is the four-dimensional costate vector (composed of λ_x, λ_y, λ_{v_x}, and λ_{v_y}) at the ith node.

A unique aspect of this application results from the fact that only the costates, and not the states, along with some transversality conditions, are included in $\boldsymbol{p}$. This has the advantage of keeping $\boldsymbol{p}$ smaller and was possible because the state variables were not nearly as unstable as the costates.

The corresponding dependent vector $\boldsymbol{F}(\boldsymbol{p})$: $\mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_p}$,

$$\boldsymbol{F}(\boldsymbol{p}) = \begin{Bmatrix} \Delta\boldsymbol{\lambda}_2 \\ \Delta\boldsymbol{\lambda}_3 \\ \vdots \\ \Delta\boldsymbol{\lambda}_{w-1} \\ \Delta\boldsymbol{\lambda}_w \\ \Delta\boldsymbol{r} \\ \boldsymbol{\lambda}_{v_x,0} - \mu * \cos\theta_0 \\ \boldsymbol{\lambda}_{v_y,0} - \mu * \sin\theta_0 \end{Bmatrix} \tag{62}$$

where the $\Delta\boldsymbol{\lambda}_i \in \mathbb{R}^4$ are the differences between the costate vector integrated forward from the previous node, $i - 1$, to the current node, i, and the previously stored $\boldsymbol{\lambda}_i$. For example,

$$\Delta\boldsymbol{\lambda}_2 = \boldsymbol{\lambda}_{2_{\text{integrated}}} - \begin{pmatrix} \boldsymbol{p}_5 \\ \boldsymbol{p}_6 \\ \boldsymbol{p}_7 \\ \boldsymbol{p}_8 \end{pmatrix}$$

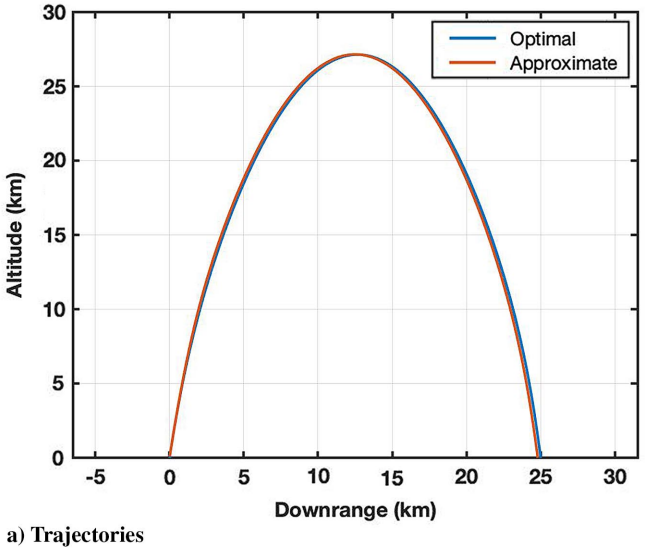
where $\boldsymbol{\lambda}_{2_{\text{integrated}}}$ comes from integrating

$$\boldsymbol{\lambda}_1 = \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \boldsymbol{p}_3 \\ \boldsymbol{p}_4 \end{pmatrix}$$

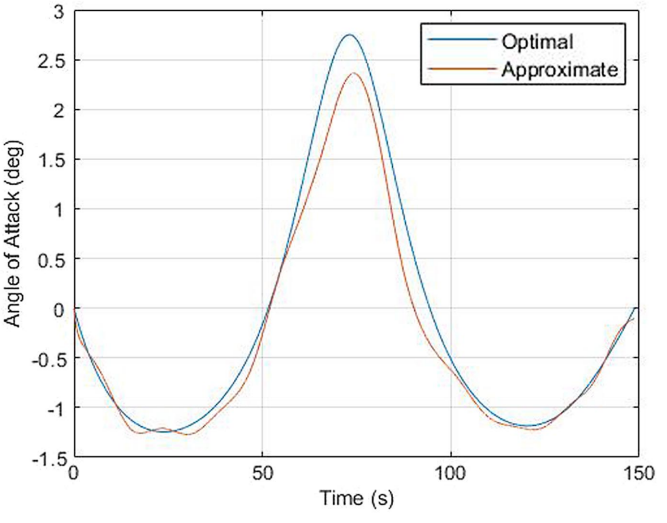
forward from t_1 ($t = 0$) to time t_2, using Eq. (5).

As to the last four terms of Eq. (62), $\Delta\boldsymbol{\lambda}_w$ (the wth node corresponds with t_f) is the difference between the integrated value of $\boldsymbol{\lambda}_w$ and $\boldsymbol{\lambda}_f$ from Eq. (7), $\Delta\boldsymbol{r}$ is the difference between the projectile position and the position of the target at time t_f, and the last two terms are from the initial transversality conditions. The numerical nonlinear-system-of-equations solver iterates on $\boldsymbol{p}$ until $\boldsymbol{F}(\boldsymbol{p}) \approx 0$. Figure 1b shows how the solver drives the $\Delta\lambda_x$ values to zero, as representative of what happens to all four values of the costate vector.

This approach successfully converged to an optimal solution with the results shown in the following figures. In each case shown below, "optimal" refers to the solution resulting from the multiple-shooting method approach. Figure 3a shows a comparison of the approximate



a) Trajectories



b) Angle-of-attack profiles

Fig. 3 Comparison of approximate and optimal trajectories.

optimal and optimal trajectories, indicating the degree to which the optimal trajectory matches the original approximate optimal one (because the velocity profiles of the two trajectories are similarly close, this plot is not included). Figure 3b shows a comparison of the approximate optimal angle-of-attack history and the one from the optimal trajectory. Notice that the approximate optimal history is not smooth as is the case for the optimal trajectory. This exemplifies one of the differences between the direct and indirect methods: whereas direct methods provide a good approximation of the optimal control, indirect methods typically result in a more precise optimal solution.

Next, the methodology was executed using control inputs from a purely ballistic case. A separate simulation was developed to generate the control history associated with a ballistic ($\alpha = 0$) case and that control history was used in an identical manner in the indirect method simulation (a depressed trajectory was chosen for this case because the lofted trajectory at this range is already close to the optimal solution). The methodology again converged to an optimal solution; a comparison of the ballistic trajectory and the resulting optimally controlled trajectory is shown in Fig. 4a. The optimal trajectory achieves the same range and time of flight as the ballistic trajectory, despite the fact that it flies significantly higher. On the optimal trajectory, the projectile is fired at a smaller launch angle ($\theta_0 = 62^\circ$) compared with the 74° launch angle for the ballistic trajectory. This flight through thinner air allows the projectile to arrive with a higher final velocity than the purely ballistic case (436 m/s vs 413 m/s) as specified by the cost function. A comparison of the velocity profiles is shown in Fig. 4b.

Figure 5 shows a history of the angle of attack for the optimal trajectory (the angle of attack for the ballistic trajectory is uniformly 0). The optimally controlled projectile gradually pitches over from the initial velocity vector for the first part of the trajectory, then pulls slight positive α's on the descent portion of the flight. This control history results in the significant increase in final velocity mentioned previously.

Figures 6 and 7 show comparisons of the initial estimates for the costates with the converged costate histories for the approximate optimal and ballistic cases. Figures 6a and 6b show the position costate estimates, whereas Figs. 7a and 7b show the estimates of the velocity costates. For the approximate optimal case, the plots of the initial estimates are fairly close to the final histories; this is expected because the original approximate trajectory is very close to the final optimal trajectory. As can be seen, the estimates derived from the approximate optimal trajectory are much closer to the histories produced during the generation of the optimal trajectory than those produce from the ballistic trajectory. However, even the coarser estimates from the ballistic trajectory are sufficient to allow the method to converge to a solution. It is worth mentioning that

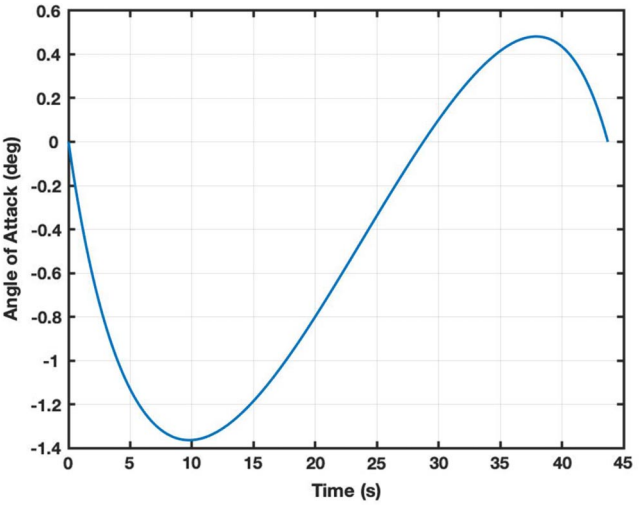
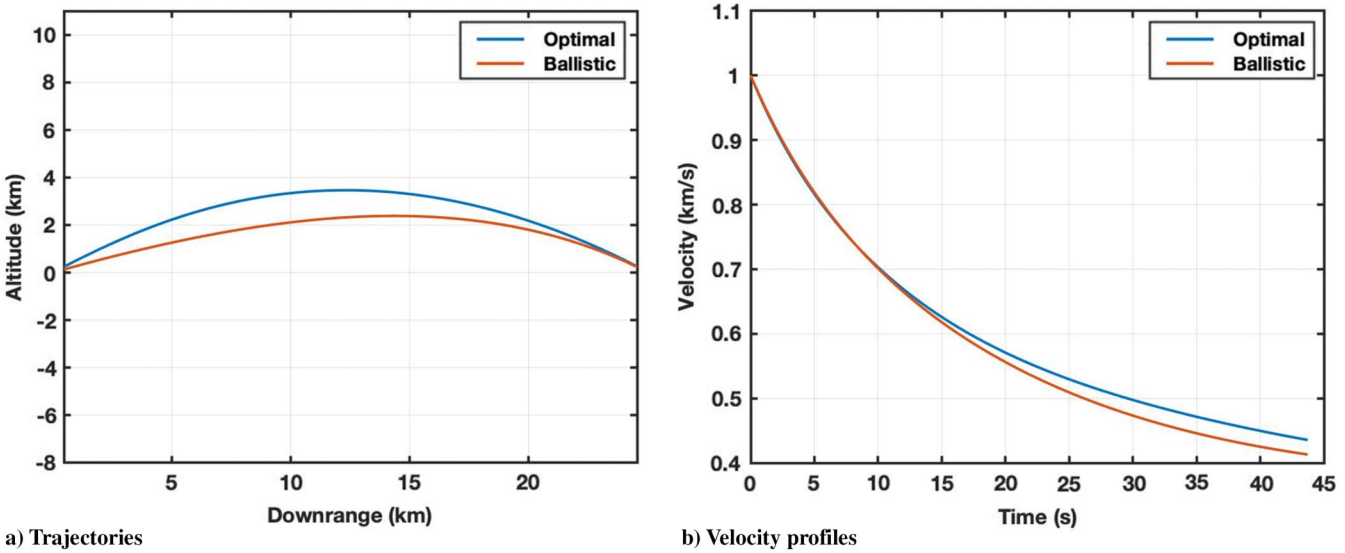


Fig. 5 Optimal angle-of-attack history.

although some of the costate estimates appear to be relatively far from the histories generated during the optimal trajectories, they are sufficiently close so as to be in the correct "valley" to allow the methodology to converge to an optimal solution.

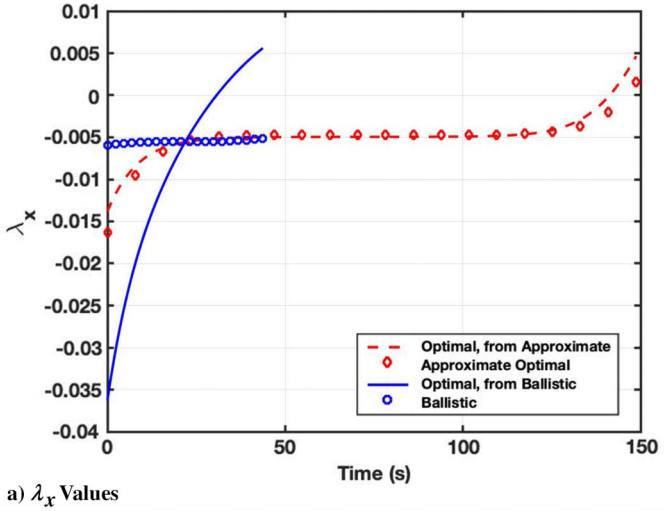
Subsequently, another case was chosen to explore the robustness of the methodology. By manipulating a longer-range (approximately 40 km) ballistic trajectory by manually altering the angle of attack history, a trajectory was developed that extended the range to approximately 56 km. This was done by specifying an angle of attack at each of the nodes and manually iterating until a longer range was achieved. The optimization methodology was applied using this manually derived control history and an optimal trajectory was achieved. The two trajectories are shown in Fig. 8a. As can be seen, both trajectories are essentially ballistic during the first half of their flight; the optimal trajectory is fired at a higher launch angle than the manual one and then pulls up slightly during the descent phase relative to the manual one. Also, the optimal trajectory is higher in the range extension portion of flight, allowing flight through less dense air and resulting in a significantly higher final velocity, as shown in Fig. 8b.

Figure 9a shows a comparison of the position costate estimates and optimal history for λ_x and λ_y, whereas Fig. 9b shows a comparison of the velocity costate estimates and optimal history for λ_{v_x} and λ_{v_y}. As shown earlier, although the estimates appear to be relatively distant from the integrated traces, they are sufficient to allow convergence to the optimal solution.

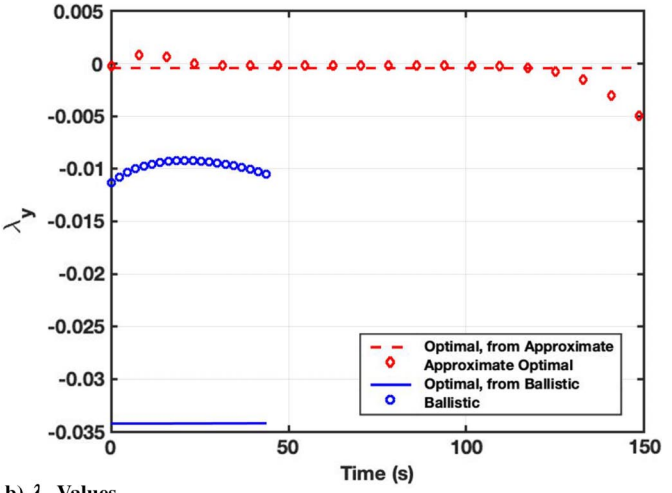


a) Trajectories **b) Velocity profiles**

Fig. 4 Comparison of ballistic and optimal trajectories.

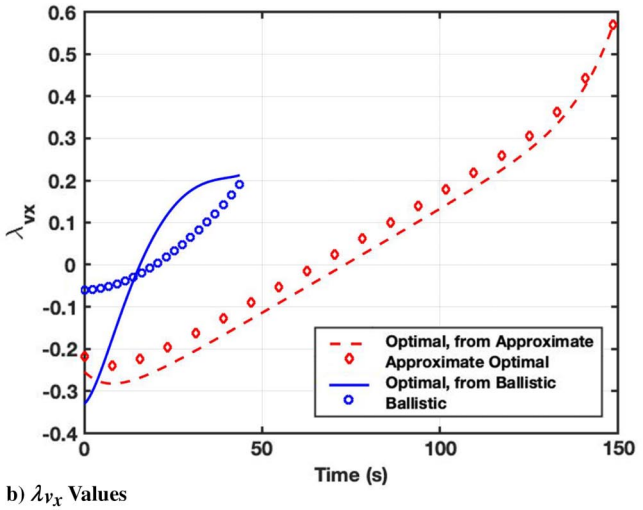


a) λ_x Values

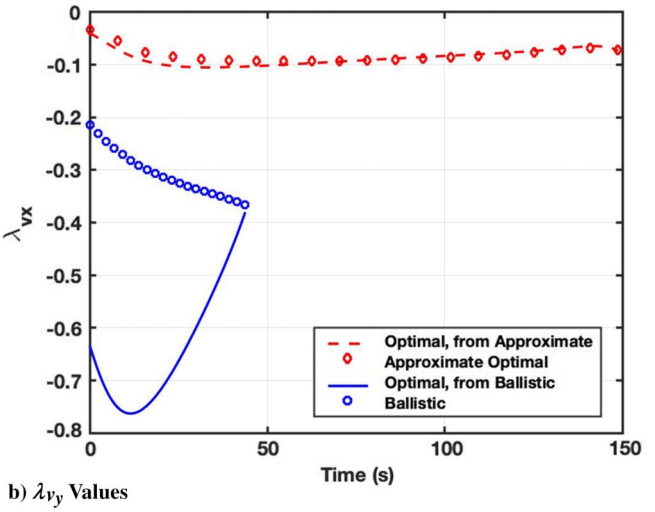


b) λ_y Values

Fig. 6 Comparison of λ_r for approximate, ballistic, and optimal trajectories.

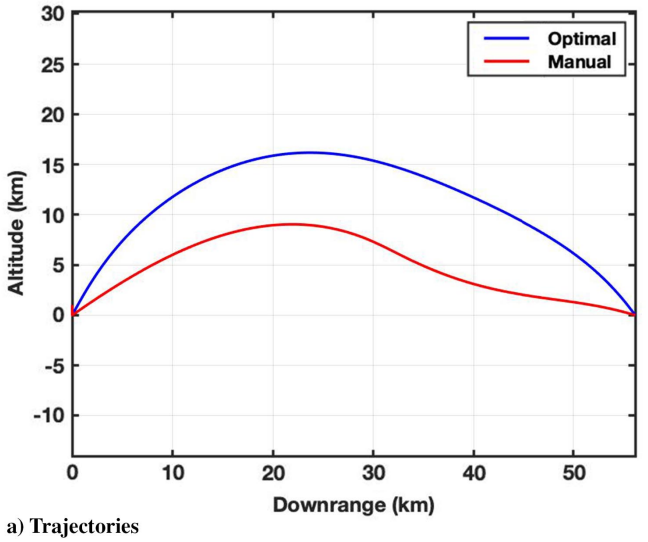


b) λ_{v_x} Values

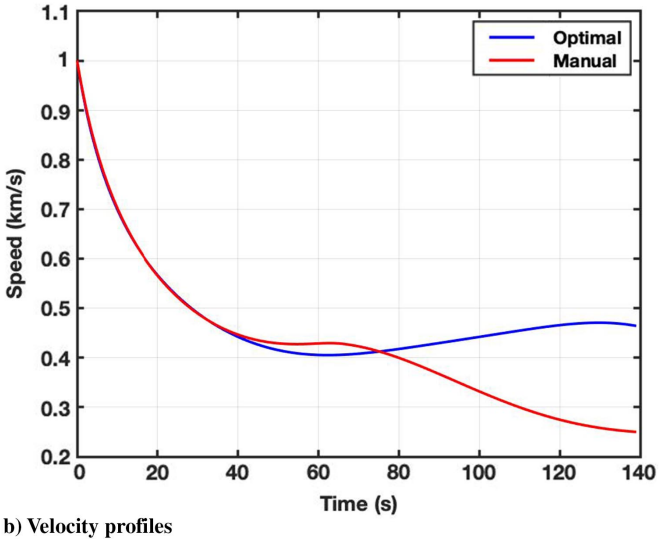


b) λ_{v_y} Values

Fig. 7 Comparison of λ_v for approximate, ballistic, and optimal trajectories.



a) Trajectories

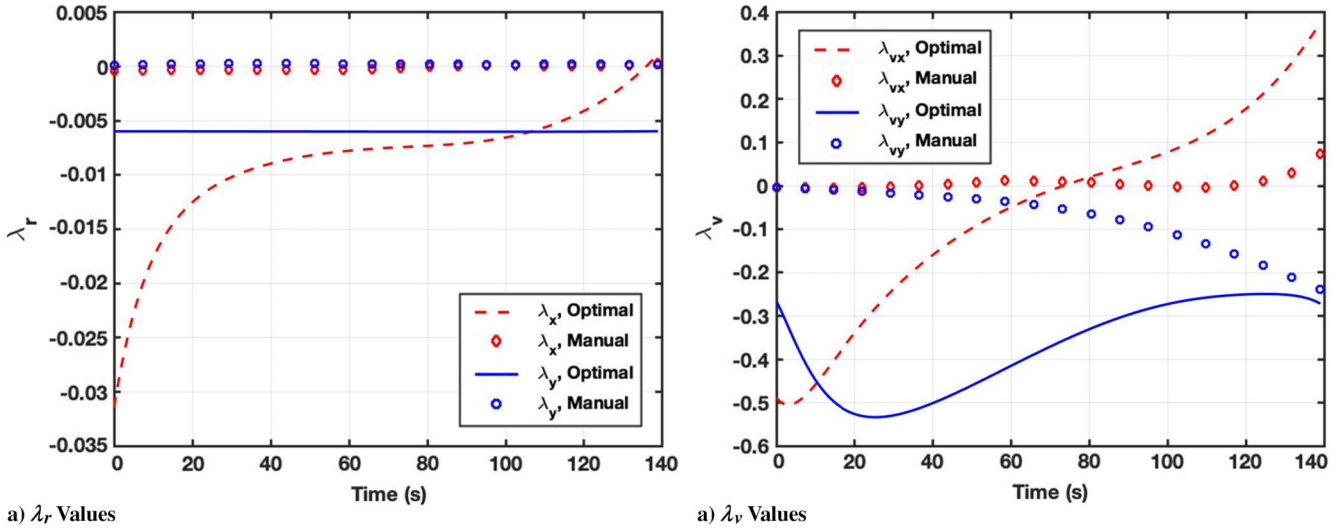


b) Velocity profiles

Fig. 8 Comparison of 56 km manual and optimal trajectories.

VII. Conclusions

A method was derived that uses a reference trajectory to analytically and noniteratively solve for estimates of the costates associated with the states of the reference trajectory, for a class of optimal control problems. The method finds the costates at the nodes along the reference trajectory by solving for the values that match Pontryagin's maximum principle as closely as possible in a specified least-squares sense. The particular advancement of this Note is that the reference



a) λ_r Values a) λ_v Values

Fig. 9 Comparison of costate estimates and optimal history for 56 km trajectory.

trajectory need not come from some optimization or near-optimal method. Therefore, the reference trajectory can be quite rough, as long as it generally contains the right features and shape in the ballpark of the ultimate solution.

To demonstrate the method, a guided-projectile optimal control problem was solved. Nearly optimal, ballistic, and rough "by-hand" reference trajectories were used in the method, and the subsequently-run multiple-shooting algorithm converged using the results of the method as input. Because the new process for estimating the costates is analytical, is noniterative, and needs only rough estimates of trajectories in the neighborhood of the final solution, it is a good tool to aid in solving optimal control problems via a multiple-shooting algorithm.

References

[1] Leitmann, G., *Optimization Techniques*, 1st ed., Academic Press, New York, 1962, pp. 255–279, Chap. 7.

[2] Betts, J. T., *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, 2nd ed., Soc. for Industrial and Applied Mathematics, Washington, D.C., 2010, pp. 109–111, Chap. 3.

[3] Hargraves, C. R., and Paris, S. W., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," *Journal of Guidance, Control, and Navigation*, Vol. 10, No. 4, July–Aug. 1987, pp. 338–342.

[4] Dennis, M. E., Hager, W. W., and Rao, A. V., "Computational Method for Optimal Guidance and Control Using Adaptive Gaussian Quadrature Collocation," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 9, Sept. 2019, pp. 2026–2041.

[5] Gath, P. F., and Well, K. H., "Trajectory Optimization Using a Combination of Direct Multiple Shooting and Collocation," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 2001-4047, Aug. 2001.

[6] 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, Nov.–Dec. 2006, pp. 1435–1440.

[7] Betts, J. T., *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, 2nd ed., Soc. for Industrial and Applied Mathematics, Washington, DC, 2010, p. 129, Chap. 4.

[8] Bryson, A. E., and Ho, Y.-C., *Applied Control Theory, Optimization, Estimation, and Control*, 1st ed., Taylor and Francis, New York, 1975, p. 214, Chap. 7.

[9] Bushong, P. M., and Lawton, J. A., "Fast Solution to an Optimal Trajectory Problem," *Conference Proceedings of the AAS/AIAA Spaceflight Mechanics Meeting*, American Astronautical Soc. Paper 94-131, Feb. 1994; also Advances in the Astronautical Sciences, Vol. 87, Univelt, San Diego, CA, 1994, pp. 973–980.

[10] Jiang, F., Tang, G., and Li, J., "Improving Low-Thrust Trajectory Optimization by Adjoint Shape-Based Path," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 12, Dec. 2017, pp. 3280–3287.

[11] Seywald, H., and Kumar, R. R., "Method for Automatic Costate Calculation," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 6, Nov.–Dec. 1996, pp. 1252–1261.

[12] Lee, D., and Bang, H., "Efficient Initial Costates Estimation for Optimal Spiral Orbit Transfer Trajectories Design," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 6, Nov.–Dec. 2009, pp. 1943–1947.

[13] Martell, C. A., and Lawton, J. A., "Adjoint Variable Solutions via an Auxiliary Optimization Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 6, Nov.–Dec. 1995, pp. 1267–1272.