

A Gauss Pseudospectral Collocation for Rapid Trajectory Prediction and Guidance*

Bradley T. Burchett[†]

Rose-Hulman Institute of Technology, Terre Haute, IN, 47803

The flight of symmetric projectiles is modeled by linear and quasi-linear ODEs known respectively as projectile linear theory and modified projectile linear theory. A Gauss pseudo-spectral collocation may be used to discretize both linear and non-linear ODE models, converting the problem into a set of coupled algebraic equations. Since the approximation is exact at the collocation points, accurate trajectory predictions may be rendered using a small number of points, resulting in very rapid solution. The method allows for solution of high launch elevation trajectories and can account for varying aerodynamic coefficients as well. Results which are compared to a full 6DOF simulation are shown for standard linear, modified linear, and modified linear with varying aero coefficients. By also discretizing the cost function for optimal control, the problem of optimal trajectory design is rendered as an algebraic cost function with algebraic equality constraints. Such a problem is solved by appending equality constraints to the cost function integrand with Lagrange multipliers. The resulting large set of non-linear algebraic equations is then numerically solved. Feasibility of the optimal trajectories was demonstrated by commanding forward canards by a gain scheduled LQR inner loop. The projectile tracked desired trajectories with very little error resulting in a large reduction in dispersion at the target.

*Research was sponsored in part by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-16-2-0086. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

[†]Professor, Department of Mechanical Engineering, burchett@rose-hulman.edu, Associate Fellow, AIAA

Nomenclature

I_{xx}, I_{yy}	roll and pitch inertia expressed in the projectile reference frame (sl-ft ²)
\mathbf{A}, \mathbf{B}	Linear State Space Matrices
C_{NA}	normal force aerodynamic coefficient
C_{X0}	axial force aerodynamic coefficient
C_{LP}	roll rate damping moment aerodynamic coefficient
C_{LDD}	fin rolling moment aerodynamic coefficient
C_{MA}	pitch moment due to AOA aerodynamic coefficient
C_{MQ}	pitch rate damping moment aerodynamic coefficient
D	projectile characteristic length (ft)
\mathbf{f}	vector of non-linear algebraic constraints
g	gravitational constant = 32.2 (ft/s ²)
\mathbf{I}	identity matrix
L_N	N th order Legendre polynomial
m	projectile mass (sl)
N	number of collocation points
p, q, r	angular velocity vector components expressed in the fixed plane reference frame (rad/s)
S	= $\pi D^2/4$, projectile reference area (ft ²)
SL_{cg}	stationline of the projectile c.g. location (ft)
SL_{cp}	stationline of the projectile c.p. location (ft)
s	downrange distance (calibers)
u, v, w	translation velocity components of the projectile center of mass resolved in the fixed plane reference frame (ft/s)
u_1, u_2	= $C_{N\alpha, can} \delta_{canz}, C_{N\alpha, can} \delta_{cany}$ canard commands in no roll frame without effectiveness scaling
V	= $\sqrt{u^2 + v^2 + w^2}$, magnitude of the mass center velocity (ft/s)
x, y, z	position vector components of the projectile mass center expressed in the inertial reference frame
<i>Greek</i>	
ρ	air density (sl/ft ³)
ψ, θ, ϕ	Euler yaw, pitch, and roll angles (rad)
Δ	Legendre-Gauss differentiation matrix
Λ	cost function for optimality
<i>Subscript</i>	
i, j	row i , column j of matrix
f	horizon or target
can	canard

I. Introduction

FOR decades, ballisticians have sought methods to rapidly predict the free flight trajectory of a projectile. Such predictions are useful in identifying aerodynamic coefficients of projectiles, and providing guidance, navigation and control to guided missiles. Prior to the widespread use of digital computers, a standard linear model (PLT) was developed, allowing for closed-form solution of a set of ten coupled ODEs.¹ This method is only valid for ‘flat fire’ trajectories where the launch elevation is low. With widespread deployment of digital computers, numerical solution of fully non-linear ODEs is possible, but still time consuming, and impractical for real time guidance and control. In 2005, Hainz and Costello² introduced Modified Projectile Linear Theory (MPLT) as a synthesis of linearization and numerical integration. This technique allows for high launch elevation while taking advantage of roll-yaw decoupling such that 12 ODEs are solved in a piecewise fashion. In order to account for a widely varying pitch, the solution must be propagated forward in time with a relatively small time step.

Pseudospectral methods have been used for many years to numerically solve partial differential equations. They have recently come into prominence for discretizing non-linear optimal control problems, and providing a direct way to a numerical solution. After the seminal work by Elnagar, Kazemi, and Razzaghi,³ pseudo-spectral methods were rapidly deployed to many non-linear optimal control problems. Two commercially available extensions to Matlab are now available under the names ‘GPOPS’,⁴ and ‘DIDO’.⁵

In this work, we will show how a Gauss pseudo-spectral (GPS) collocation may be used to discretize both linear and non-linear models, converting the problem into a set of coupled algebraic equations. Since the approximation used is exact at the collocation points, accurate trajectory predictions may be found using a small number of points, resulting in a very rapid solution.

The method is then extended to the design of optimal trajectories for guidance of indirect shots by introducing two controls in the no-roll frame. A cost function is chosen and discretized by a Gauss quadrature. The discretized dynamics are then treated as equality constraints resulting in a problem that can be solved as a large set of coupled algebraic equations. These equations are differentiated such that they are solved using a gradient based search. No special software is required for the solution.

Flyability of the optimal trajectories is demonstrated on a high fidelity 6DOF simulation assuming two pair of forward canards. A rudimentary inner control loop is designed by LQR with gain scheduling based upon predicted optimal trajectories. The method is able to find optimal trajectories for a wide range of launch elevations. Simulation demonstrates a remarkable reduction in dispersion.

II. Linear and Modified Linear Theory Models

The Modified Linear Theory Model consists of the following ten quasi-linear ODEs.

$$\dot{y} = c_\theta V \psi + v \quad (1)$$

$$\dot{z} = -V s_\theta + c_\theta w \quad (2)$$

$$\dot{\theta} = q \quad (3)$$

$$\dot{\psi} = \frac{r}{c_\theta} \quad (4)$$

$$\dot{v} = -Av - Vr + b_1 u_2 \quad (5)$$

$$\dot{w} = -Aw + Vq + gc_\theta - b_1 u_1 \quad (6)$$

$$\dot{q} = Cw + Eq - \frac{I_{XX}}{I_{YY}} pr + b_2 u_1 \quad (7)$$

$$\dot{r} = -Cv + \frac{I_{XX}}{I_{YY}} pq + Er + b_2 u_2 \quad (8)$$

$$\dot{V} = -\frac{\pi \rho D^2}{8m} C_{X0} V^2 - gs_\theta \quad (9)$$

$$\dot{p} = \frac{\pi \rho V^2 D^3}{8I_{XX}} C_{LDD} + \frac{\pi \rho D^4 V}{16I_{XX}} p C_{LP} \quad (10)$$

Where

$$A = \frac{\rho SV}{2m} C_{NA} \quad (11)$$

$$C = \frac{\rho SV}{2I_{yy}} C_{MA} \quad (12)$$

$$E = \frac{\rho SV D^2}{4I_{yy}} C_{MQ} \quad (13)$$

$$C_{MA} = (SL_{COP} - SL_{CG}) C_{NA} \quad (14)$$

$$b_1 = \frac{\rho V^2 S_{can}}{2m} \quad (15)$$

$$b_2 = \frac{\rho V^2 S_{can}}{2I_{YY}} (SL_{can} - SL_{CG}) \quad (16)$$

The equations are written in scalar form to easily facilitate the collocation in the sequel. This formulation uses a ‘no roll’ frame intermediate to the traditional body fixed and ground fixed frames. This is permissible assuming the projectile is aerodynamically and inertially symmetric. Unlike customary linear theory formulations, the independent variable used here is time. Using time as the independent variable provides a scaling that keeps the equations well conditioned after collocation. This scaling also simplifies sensitivities with respect to total velocity which will be required for algebraic solution of the necessary conditions for optimality.

For open-loop solutions, Eqs. (1)–(10) are rewritten with non-dimensional arclength traveled s as the independent variable. This is accomplished by simply multiplying the RHS of each equation by the factor D/V , and using prime ($'$) rather than dot to denote the derivative wrt s . Standard linear theory (PLT) equations may then be found by subsequently assuming θ to be small such that $c_\theta = 1$ and $s_\theta = \theta$. For simplicity we assume zero winds. This study will consider only fin stabilized projectiles such that the Magnus term is neglected.

III. Gauss–Legendre Pseudo–spectral Collocation

Sets of coupled non-linear ODEs may be discretized using Legendre polynomials as the underlying basis functions. The presentation here follows closely from Elnagar, Kazemi and Razzaghi.³ Let $L_N(x)$ represent the Legendre polynomial of order N . The Legendre–Gauss–Lobatto (LGL) collocation points are defined as $x_0 = -1$, $x_N = 1$ with interior points x_m , $m = 1, \dots, N - 1$ found from the roots of \dot{L}_N , the first derivative of L_N . This choice allows boundary conditions to be enforced at the initial and final times desired in the solution.

If a Lagrange polynomial is constructed whose roots are the LGL points, it can be written as:

$$\lambda_l(x) = \frac{1}{N(N+1)L_N(x_l)} \bullet \frac{(x^2 - 1)\dot{L}_N(x)}{x - x_l}, \quad (l = 0, 1, \dots, N). \quad (17)$$

Lagrange polynomials have the Kronecker delta property such that

$$\lambda_l(x_j) = \begin{cases} 1 & \text{if } l = j \\ 0 & \text{if } l \neq j. \end{cases} \quad (18)$$

Any function $\Gamma(x)$ can then be approximated by the Lagrange interpolating polynomial as

$$\Gamma^N(x) = \sum_{l=0}^N \Gamma(x_l)\lambda_l(x) \quad (19)$$

Because $\lambda_l(x)$ has the Kronecker delta property

$$\Gamma^N(x_k) = \Gamma(x_k), \quad k = 0, 1, \dots, N. \quad (20)$$

The derivative of $\Gamma^N(x)$ can then be found by differentiating Eq. (19). The result is a matrix multiplication which can then be used to discretize differential equations.

$$\dot{\Gamma}^N(x_m) = \sum_{l=0}^N \Delta_{ml}\Gamma(x_l) \quad (21)$$

where $\Delta = (\Delta_{ml})$ is an $(N + 1) \times (N + 1)$ matrix given by

$$\Delta = (\Delta_{ml}) = \begin{cases} \frac{L_N(x_m)}{L_N(x_l)} \frac{1}{(x_m - x_l)} & m \neq l \\ -\frac{4}{N(N+1)} & m = l = 0 \\ \frac{4}{N(N+1)} & m = l = N \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

Here we use the symbol Δ for the differentiation matrix so as not to conflict with D the projectile characteristic length or diameter. Note that the LGL points, and the differentiation matrix will be computed to double precision on demand using built-in Matlab functions *roots*, *polyval*, and *polyder*. The method is limited to about 35 collocation points—beyond this the differentiation matrix becomes ill-conditioned due to the high order underlying interpolation polynomial. Also note that the differentiation matrix is automatically scaled assuming a time span of 2s ($-1 \leq t \leq 1$). Thus when discretizing ODEs, the scaling must be modified to reflect the actual time span. For instance, $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ becomes:

$$\frac{2}{t_f - t_0} \Delta X = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{u}$$

The Gauss-Lobatto quadrature approximates a continuous integral to a high degree of accuracy with a discrete sum. For instance:

$$\int_{t_0}^{t_f} f(t) dt \approx \frac{t_f - t_0}{2} \sum_{i=1}^N w_i \bullet f(\tau_i)$$

Where τ_i are the LGL abscissa points projected onto the actual time span by a linear mapping

$$\tau_i = \frac{t_f - t_0}{2} x_i + \frac{t_f + t_0}{2}$$

and w_i are the LGL quadrature weights found by:

$$w_i = \frac{2}{N(N+1)} \frac{1}{[L_N(x_i)]^2}, \quad i = 0, 1, \dots, N.$$

Most of the recent applications of Gauss Pseudospectral collocation are motivated by the solution to a non-linear optimal control problem. In that context, the dynamic constraints are discretized by using the differentiation matrix. The objective function is evaluated using a Gauss quadrature at the LGL points. The result is an optimization problem with algebraic constraints that can be solved using non-linear programming. This work will explore the use

of GPS discretization in rapidly solving for both flat fire and indirect open loop trajectories with a high degree of accuracy. Solving the algebraic constraints as an open-loop system is a valuable step toward solving the closed-loop problem for the purposes of 1) checking that dynamic constraints are properly represented, and 2) providing a reasonable starting guess for the non-linear programming solution of the optimal control problem. This paper will conclude by finding optimal closed-loop trajectories for indirect fire by using the LGL collocation to produce an augmented set of non-linear algebraic necessary conditions which can be solved using a Newton iteration. When only equality constraints are considered, the necessary conditions may be written as a large set of non-linear algebraic equations, and special software is not required to solve them.

IV. Gauss Pseudo-spectral Collocation of the PLT and MPLT Equations

After scaling the RHS of Eqs. (1)-(10) by D/V and changing the independent variable to non-dimensional arclength traveled, the differentiation matrix is applied to render an algebraic set of constraints in residual form.

$$\frac{2}{s_f}\Delta y - c_\theta\psi D - \frac{D}{V}v = 0 = f_1 \quad (23)$$

$$\frac{2}{s_f}\Delta z + Ds_\theta - \frac{Dc_\theta}{V}w = 0 = f_2 \quad (24)$$

$$\frac{2}{s_f}\Delta\theta - \frac{D}{V}q = 0 = f_3 \quad (25)$$

$$\frac{2}{s_f}\Delta\psi - \frac{D}{Vc_\theta}r = 0 = f_4 \quad (26)$$

$$\frac{2}{s_f}\Delta v + Av + Dr - b_1u_2 = 0 = f_5 \quad (27)$$

$$\frac{2}{s_f}\Delta w + Aw - Dq - \frac{Dgc_\theta}{V} + b_1u_1 = 0 = f_6 \quad (28)$$

$$\frac{2}{s_f}\Delta q - \frac{C}{D}w - Eq + Fr - b_2u_1 = 0 = f_7 \quad (29)$$

$$\frac{2}{s_f}\Delta r + \frac{C}{D}v - Fq - Er - b_2u_2 = 0 = f_8 \quad (30)$$

$$\frac{2}{s_f}\Delta V + \frac{\pi\rho D^3}{8m}C_{X0}V + \frac{Dg}{V}s_\theta = 0 = f_9 \quad (31)$$

$$\frac{2}{s_f}\Delta p - \frac{\pi\rho VD^4}{8I_{XX}}C_{LDD} - \frac{\pi\rho D^5}{16I_{XX}}C_{LPP} = 0 = f_{10} \quad (32)$$

Where s_f is the total arclength traveled from launch to target and $F = (D/V)(I_{XXp})/(I_{YY})$. Collocation maps the independent variable from $0 \leq s \leq s_f$ to $-1 \leq \tau \leq 1$.

Eqs. (23)–(32) constitute a non-linear algebraic set hereafter referred to as modified projectile linear theory (MPLT). Products and quotients of state variables such as $c_\theta\psi$ and v/V will be implemented using array arithmetic in Matlab. This will allow all states including p , V , and θ to vary over the entire trajectory. Unlike ref[2], aerodynamic coeffs. and atmospheric density will also vary over the trajectory. For ballistic trajectories, the controls u_1 , u_2 are set to zero.

IV.A. Standard Linear Implementation

For the standard flat-fire projectile linear theory (PLT), the state vector is limited to $[y \ z \ \theta \ \psi \ v \ w \ q \ r]^T$ while p , V , and aerodynamic coeffs. are held constant. This allows the remaining algebraic equations to be written in matrix form as:

$$\mathbf{Z}\mathbf{X} = \mathbf{Y} \quad (33)$$

Where

$$\mathbf{Z} = \begin{bmatrix} \frac{2}{s_f}\Delta & 0 & 0 & -D\mathbf{I} & -\frac{D}{V}\mathbf{I} & 0 & 0 & 0 \\ 0 & \frac{2}{s_f}\Delta & D\mathbf{I} & 0 & 0 & -\frac{D}{V}\mathbf{I} & 0 & 0 \\ 0 & 0 & \frac{2}{s_f}\Delta & 0 & 0 & 0 & -\frac{D}{V}\mathbf{I} & 0 \\ 0 & 0 & 0 & \frac{2}{s_f}\Delta & 0 & 0 & 0 & -\frac{D}{V}\mathbf{I} \\ 0 & 0 & 0 & 0 & \frac{2}{s_f}\Delta + A\mathbf{I} & 0 & 0 & D\mathbf{I} \\ 0 & 0 & 0 & 0 & 0 & \frac{2}{s_f}\Delta + A\mathbf{I} & -D\mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{C}{D}\mathbf{I} & \frac{2}{s_f}\Delta - E\mathbf{I} & F\mathbf{I} \\ 0 & 0 & 0 & 0 & \frac{C}{D}\mathbf{I} & 0 & -F\mathbf{I} & \frac{2}{s_f}\Delta - E\mathbf{I} \end{bmatrix}$$

Each entry in \mathbf{Z} is an $N \times N$ block where N is the number of collocation points chosen.

$$\mathbf{Y} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{Dg}{V} & 0 & 0 \end{bmatrix}^T.$$

Initial conditions are enforced by adding appropriate rows to \mathbf{Z} and \mathbf{Y} then deleting the first row of each state block. The state history \mathbf{X} is then solved for by the LU decomposition embedded in the Matlab ‘backslash’ (\backslash) operator.

IV.B. Modified Linear Implementation

For the non-linear case, an initial solution guess is found from the linear solution. Eqs. (23)–(32) are evaluated at the current guess. A Jacobian matrix is formed by differentiating Eqs. (23)–(32) with respect to the states, resulting in

$$J = \frac{\partial f_i}{\partial \chi_j} \quad (34)$$

Where the functions f_i are shown in Eqs. (23) through (32). The state vector is defined as

$$\chi = [y \quad z \quad \theta \quad \psi \quad v \quad w \quad q \quad r \quad V \quad p]^T.$$

Non-zero entries of the Jacobian are shown in Appendix A. Strictly speaking, column 9 of rows 5–10 would also require terms reflecting sensitivity to changes in aerodynamic coefficients due to changes in V . For instance

$$f_5 = f(A) = f(A(C_{NA}(M(V))))$$

or A is a function of C_{NA} which is a function of Mach and hence a function of V . This sensitivity is not included in the current Jacobian formulation, however it is included by default in the residual calculation as speed of sound is allowed to vary with altitude, and all aero coefficients are allowed to vary with Mach. Allowing this variance provides for a slight improvement in the trajectory prediction as will be seen in the sequel.

Starting with the solution to the corresponding PLT model, a Newton–Raphson iteration is used to find the non-linear solution as follows:

$$\mathbf{X}^{n+1} = \mathbf{X}^n - J^{-1}\mathbf{f} \quad (35)$$

where

$$\mathbf{f} = [f_1 \quad f_2 \quad \dots \quad f_{10}]^T|_{\mathbf{x}=\mathbf{x}^n}.$$

The search tends to converge in three to five iterations such that $\|\mathbf{f}\|_2 < 10^{-4}$. Each iteration requires the solution of an $n_S N \times n_S N$ linear system which is performed by the LU decomposition embedded in the Matlab ‘backslash’ (\backslash) operator.

IV.C. Optimal Guidance of Trajectories by MPLT

After discretization, the open-loop cases are fully constrained sets of algebraic equations. Introducing the controls provides $n_U N$ new free variables where n_U is the number of controls. In this work, two controls are assumed, corresponding to non-dimensional normal forces generated by sets of forward mounted canards in the no roll vertical and horizontal directions. The optimal trajectory from launch point to target can be determined by the following method.

Define a cost function for optimal control:

$$\Lambda = \frac{1}{2} \mathbf{x}^T(t_f) \mathbf{S} \mathbf{x}(t_f) + \frac{1}{2} \int_0^{t_f} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u} dt. \quad (36)$$

The integral can be converted to an algebraic expression through a Gauss quadrature.

$$\Lambda = \frac{1}{2} \mathbf{x}^T(t_f) \mathbf{S} \mathbf{x}(t_f) + \frac{1}{2} \frac{t_f}{2} \left(\mathbf{x}^T \mathbf{Q} \mathbf{W} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{W} \mathbf{u} \right) \quad (37)$$

where \mathbf{W} is a diagonal matrix of Gauss quadrature weights $\mathbf{W} = \text{diag}(w_k)$. Combining Eq. (37) with Eqs. (23)-(32), we have

$$\min_{x,u} \Lambda(x,u) \text{ such that } \mathbf{f}(x,u) = 0 \quad (38)$$

a problem that can be solved by adjoining the constraints to the cost function and taking derivatives to form the necessary conditions for a stationary point.⁶

The dynamic constraints for indirect fire are given by Eqs. (1)–(10). For the optimal guidance problem these are converted to algebraic constraints by application of the differentiation matrix similar to Eqs. (23)–(32), however keeping the original scaling with time as the independent variable. The scaling with arclength traveled is convenient for the previous tasks, however it results in an ill-conditioned set of constraints when working with an unspecified terminal time as is the case with indirect fire. For brevity, the revised equations are presented in Appendix B.

Since the independent variable in this formulation is time, the horizon distance t_f is not known *a priori* for an indirect shot with the target placed at $(x_f, 0, 0)$. Thus the horizon is treated as a free variable (unspecified terminal time), and an additional constraint is found by a Gauss quadrature which finds downrange distance in terms of total velocity and Euler angles. That is

$$x_f = \int_0^{t_f} V c_\theta c_\psi dt.$$

The integral is replaced by a Gauss quadrature and written as

$$x_f = \frac{t_f}{2} \sum_k w_k V_k \cos \theta_k \cos \psi_k$$

Thus, the additional constraint is written

$$\frac{t_f}{2} \sum_k w_k V_k \cos \theta_k \cos \psi_k - x_f = 0 = f_{11} \quad (39)$$

Now form a Hamiltonian by adjoining the constraints

$$H = \Lambda + \lambda^T \mathbf{f}$$

Form the necessary conditions

$$\frac{\partial H}{\partial \mathbf{x}} = \frac{\partial \Lambda}{\partial \mathbf{x}} + \lambda^T \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = 0 \quad (40)$$

$$\frac{\partial H}{\partial \mathbf{u}} = \frac{\partial \Lambda}{\partial \mathbf{u}} + \lambda^T \frac{\partial \mathbf{f}}{\partial \mathbf{u}} = 0 \quad (41)$$

$$\mathbf{f} = 0 \quad (42)$$

Eqs. (40)-(42) comprise $2n_x + n_u$ eqns. with $2n_x + n_u$ unknowns \mathbf{x} , λ , and \mathbf{u} . Including Eq. (39) adds one equation each to (42) and (40) with unknowns t_f and λ_{11} , for a total of $2(n_x + 1) + n_u$. For the discretized problem, $2n_x + n_u$ eqns. are formed at each collocation point for a grand total of $(2n_x + n_u)N + 2$ eqns. Note that $\partial \mathbf{f} / \partial \mathbf{x}$ is the Jacobian found in section IV.B but rescaled with time as independent variable and unpacked in Appendix B. A row and column are appended to the previous formulation to include f_{11} and derivatives wrt t_f .

$$\begin{aligned} \frac{\partial \Lambda}{\partial \mathbf{u}} &= \frac{t_f}{2} \mathbf{R} \mathbf{W} \mathbf{u} \\ \frac{\partial \Lambda}{\partial \mathbf{x}} &= \mathbf{S} \mathbf{x}(t_f) + \frac{t_f}{2} \mathbf{Q} \mathbf{W} \mathbf{x} \\ \frac{\partial \mathbf{f}}{\partial u_1} &= \left[0 \ 0 \ 0 \ 0 \ 0 \ -b_1 \ b_2 \ 0 \ 0 \ 0 \ 0 \right]^T \\ \frac{\partial \mathbf{f}}{\partial u_2} &= \left[0 \ 0 \ 0 \ 0 \ -b_1 \ 0 \ 0 \ -b_2 \ 0 \ 0 \ 0 \right]^T \end{aligned}$$

In order to solve Eqs. (40)-(42) by a Newton algorithm as per Eq. (37), derivatives of Eqs. (40)-(42) wrt free variables $(\mathbf{x}, \mathbf{u}, \lambda)$ are sought. Collecting equations, the full system

is:

$$\mathbf{P} = \begin{Bmatrix} \mathbf{f} \\ \Lambda_{\mathbf{u}} + \lambda^T \mathbf{f}_{\mathbf{u}} \\ \Lambda_{\mathbf{x}} + \lambda^T \mathbf{f}_{\mathbf{x}} \end{Bmatrix} = 0 \quad (43)$$

Taking derivatives wrt \mathbf{x} , \mathbf{u} , and λ , the overall Jacobian becomes

$$\mathbf{J} = \begin{bmatrix} \mathbf{f}_{\mathbf{x}} & \mathbf{f}_{\mathbf{u}} & 0 \\ \Lambda_{\mathbf{ux}} + \lambda \mathbf{f}_{\mathbf{ux}} + \lambda_{\mathbf{x}} \mathbf{f}_{\mathbf{u}} & \Lambda_{\mathbf{uu}} + \lambda_{\mathbf{u}} \mathbf{f}_{\mathbf{u}} + \lambda \mathbf{f}_{\mathbf{uu}} & \mathbf{f}_{\mathbf{u}} \\ \Lambda_{\mathbf{xx}} + \lambda \mathbf{f}_{\mathbf{xx}} + \lambda_{\mathbf{x}} \mathbf{f}_{\mathbf{x}} & \Lambda_{\mathbf{xu}} + \lambda \mathbf{f}_{\mathbf{ux}} + \lambda_{\mathbf{u}} \mathbf{f}_{\mathbf{x}} & \mathbf{f}_{\mathbf{x}} \end{bmatrix} \quad (44)$$

However, since λ is constant, $\Lambda_{\mathbf{ux}} = \Lambda_{\mathbf{xu}} = 0$, and $\mathbf{f}_{\mathbf{uu}} = 0$, \mathbf{J} simplifies to

$$\mathbf{J} = \begin{bmatrix} \mathbf{f}_{\mathbf{x}} & \mathbf{f}_{\mathbf{u}} & 0 \\ \lambda \mathbf{f}_{\mathbf{ux}} & \Lambda_{\mathbf{uu}} & \mathbf{f}_{\mathbf{u}} \\ \Lambda_{\mathbf{xx}} + \lambda \mathbf{f}_{\mathbf{xx}} & \lambda \mathbf{f}_{\mathbf{xu}} & \mathbf{f}_{\mathbf{x}} \end{bmatrix} \quad (45)$$

Note that $\mathbf{f}_{\mathbf{x}}$ and $\mathbf{f}_{\mathbf{u}}$ are matrices, therefore, terms such as $\mathbf{f}_{\mathbf{ux}}$ and $\mathbf{f}_{\mathbf{xx}}$ are 3rd order tensors. In order to avoid the use of tensors, consider differentiating the product $\lambda^T \mathbf{f}_{\mathbf{x}}$. For dimensional consistency, the product is transposed such that $\mathbf{P}_{12m+2:22m+1} = \mathbf{f}_{\mathbf{x}}^T \lambda + \Lambda_{\mathbf{x}}^T$. where

$$\mathbf{f}_{\mathbf{x}i,j} = \frac{\partial f_i}{\partial x_j}$$

Thus, we may break the product into columns of $\mathbf{f}_{\mathbf{x}}$.

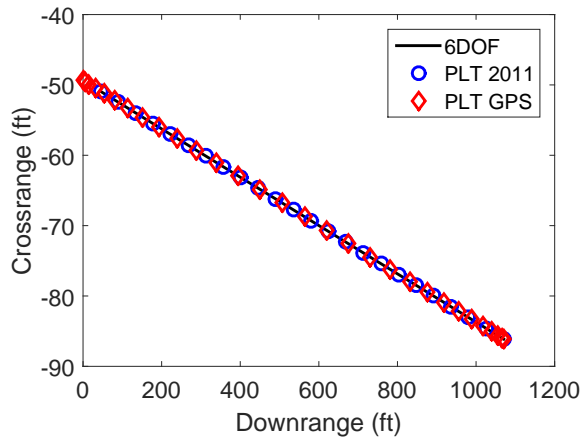
$$\mathbf{P}_{12m+2:13m+1} = \mathbf{f}_{\mathbf{x},:1:m}^T \lambda + \Lambda_{\mathbf{x}1:m}^T$$

Now, derivatives may be taken, invoking the product rule:

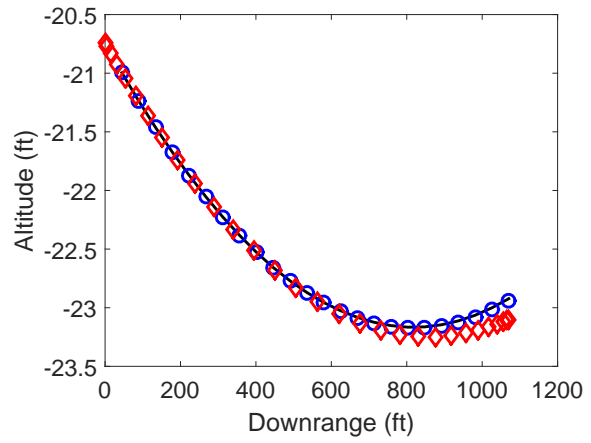
$$\frac{\partial \mathbf{P}_{12m+2:13m+1}}{\partial \mathbf{x}} = \frac{\partial}{\partial \mathbf{x}} \left(\mathbf{f}_{\mathbf{x}1:m}^T \right) \lambda + \frac{\partial}{\partial \mathbf{x}} \Lambda_{\mathbf{x}1:m}^T$$

and, for instance $\partial/\partial \mathbf{x} \left(\mathbf{f}_{\mathbf{x},:1}^T \right)$ is $[\partial/\partial \mathbf{x} (\partial \mathbf{f}/\partial y)]^T$ or in other words, $[\cdot]$ may be formed column wise by taking $\partial/\partial \mathbf{x}$ of the column $\partial \mathbf{f}/\partial y$. Since the product $\partial/\partial \mathbf{x} \left(\mathbf{f}_{\mathbf{x}1:m}^T \right) \lambda$ should result in a block $10m + 1$ rows by m columns, λ is block diagonalized to prevent summations across the collocation points. See Appendix C for programming specifics.

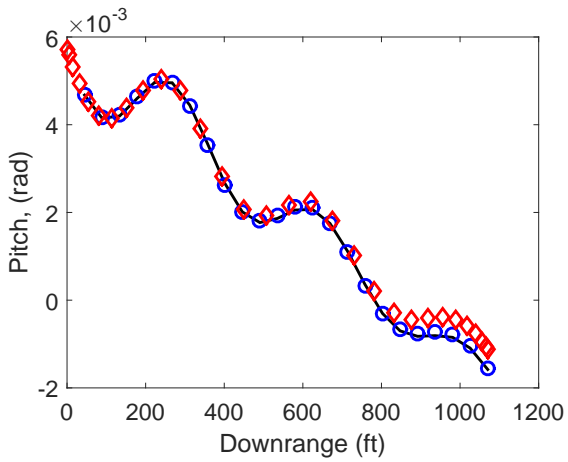
Once the full Jacobian is formed, the Newton iteration of Eq. 37 is used. Iterations proceed until $\|\mathbf{P}\|_2 < 10^{-8}$ or iterations reach 25.



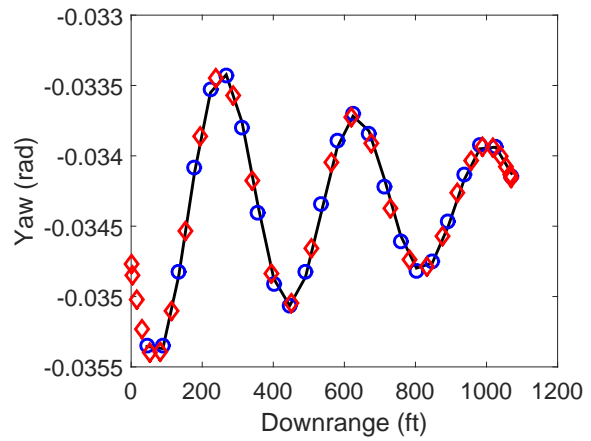
(a) Crossrange



(b) Altitude



(c) Pitch



(d) Yaw

Figure 1. Prediction Comparison for Low Launch Elevation Trajectory

IV.D. Gain Scheduled LQR Controller

In order to demonstrate the feasibility of trajectories designed by the Gauss PS method, a gain scheduled LQR controller was designed and implemented on a non-linear 6 DOF simulation. The control law takes the form

$$\mathbf{u} = -\mathbf{K}(\mathbf{x}_d - \mathbf{x})$$

where \mathbf{u} is a 2×1 vector of non-dimensional canard lift coefficients in the no roll frame. The state vector \mathbf{x} is defined as

$$\mathbf{x} = \begin{bmatrix} y & z & \theta & \psi & v & w & q & r \end{bmatrix}^T$$

and the vector of commanded states \mathbf{x}_d is found by interpolating the PS optimal controlled state history using downrange distance as the abscissa. That is, \mathbf{x}_d is found by a polynomial interpolation of each state from the optimally designed trajectory collocation points. A second order polynomial interpolation using Neville's algorithm⁷ is implemented using the waypoint immediately behind the projectile, and the next two downrange. Averaging three points avoids a small but disruptive transient each time a waypoint is passed.

Likewise, the gain matrix \mathbf{K} is found at each collocation point. Each time the control law is updated, the elements of \mathbf{K} are also interpolated using downrange distance as the abscissa. Each instance of \mathbf{K} is found by forming the eight state MPLT model from Eqs. (1) - (8) and collecting in linear, matrix form. The quantities s_θ , c_θ , V , p , density, aerodynamic coefficients, and parameters derived from these are treated as time varying parameters so that the model may be written

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

Where $\mathbf{u} = [u_1 \ u_2]^T$ and \mathbf{x} is as previously defined. Time varying parameters are evaluated at the altitudes, pitch angles, and mach numbers predicted by the optimal trajectory design. Then, the feedback gain \mathbf{K} may be found at each collocation point by solving the Ricatti Eq.:

$$\mathbf{A}^T \mathbf{G} + \mathbf{G}\mathbf{A} - \mathbf{G}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \mathbf{G} + \mathbf{Q} = 0 \quad (46)$$

and evaluating

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T \mathbf{G} \quad (47)$$

Where

$$\mathbf{Q} = \text{diag}\left(\begin{bmatrix} 75 & 75 & 25 & 25 & 1 & 1 & 1 & 1 \end{bmatrix}\right)$$

and $\mathbf{R} = 7500\mathbf{I}$ are chosen to encourage precise cross range and altitude tracking while

limiting control effort. Due to the guaranteed stability and robustness properties of LQR, the projectile should track close to the desired trajectory, making the linear model valid for such control. Note that guided state vector and LQR gains are computed offline based on initial conditions and target. The real time calculations for feedback only require interpolation for the commanded state vector and feedback gain matrix, and a matrix multiplication. Full state feedback is assumed. While state estimation is required for full implementation, it is beyond the scope of this work.

V. Results

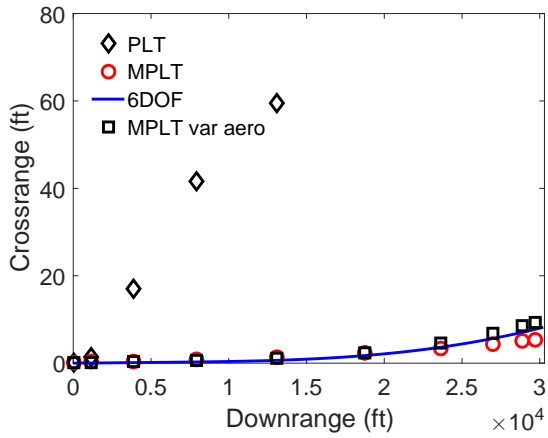
V.A. Linear Model

Figure (1) compares the predictions of the PLT using the linear implementation described above (PLT GPS) with similar predictions from PLT using the closed-form solution given in ref⁸ (PLT 2011). The solutions labeled ‘6DOF’ and ‘PLT 2011’ are discretized to the 24 equally spaced points plotted as ‘PLT 2011’. The 6DOF points plotted are linearly interpolated from a Runge Kutta simulation using a time step of 10^{-4} s. The PLT 2011 result computes the solution at every caliber of travel (4800 points for this case). 6DOF and PLT 2011 solutions are plotted for points beginning 200 calibers downrange at equal intervals separated by 200 calibers of downrange travel. 6DOF is plotted as a solid line in order to allow the reader to visually interpolate and compare the ‘PLT GPS’ solution.

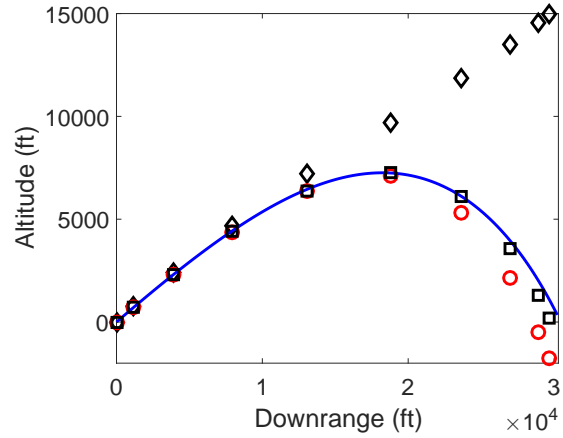
‘PLT GPS’ is the proposed pseudo-spectral numerical solution. The result shown uses 30 LGL collocation points as described above. One disadvantage of the technique is that the collocation points are dictated by the basis polynomials—the discretization is not valid for other points. The results in crossrange and yaw show a nearly exact match between all three solutions. Choosing 30 collocation points allows the pseudo-spectral solution to capture oscillations in pitch and yaw for this range of projectile travel. The pseudo-spectral solution diverges from the true solution in pitch and hence altitude at about 800 ft downrange. The previous PLT solution updates density and aerodynamic coefficients every caliber of downrange travel, and thus matches the true solution almost exactly. Overall the match for PLT GPS is quite remarkable considering that the equations are evaluated at only 30 points as opposed to 4800 for the previous method.

V.B. Non-Linear Model

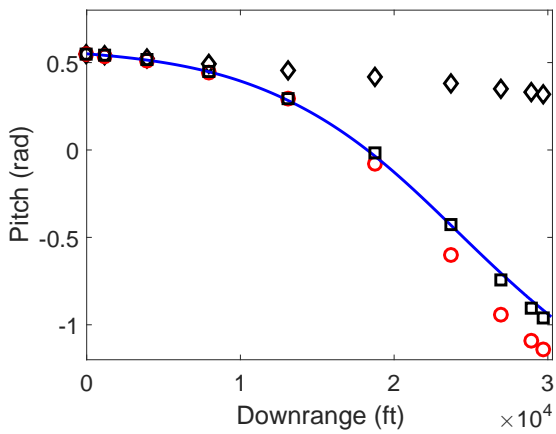
Figure 2 compares the predictions of PLT, MPLT, and MPLT with varying aero coefficients with a full non-linear 6DOF simulation. The 6DOF simulation uses Runge Kutta 4 with a fixed time step of 10^{-4} s over a flight of approximately 40.6 seconds. In order to emphasize



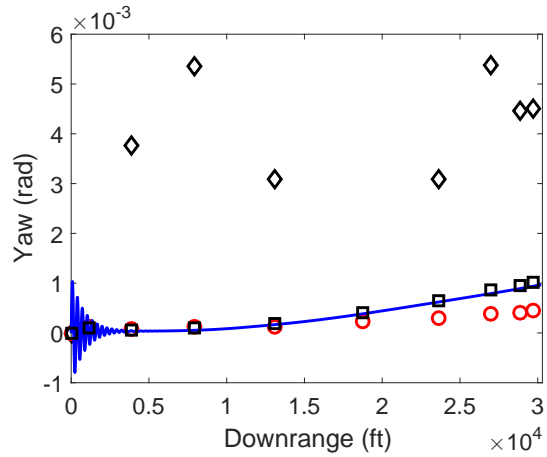
(a) Crossrange



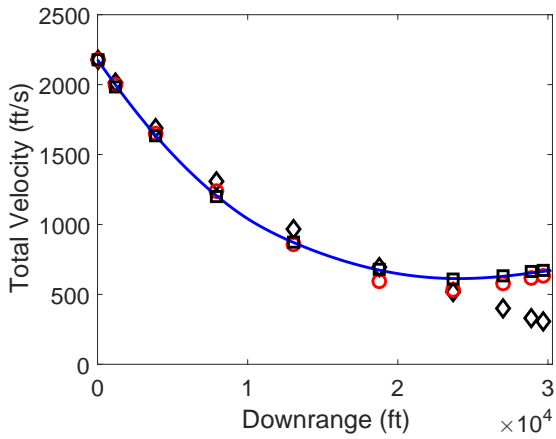
(b) Altitude



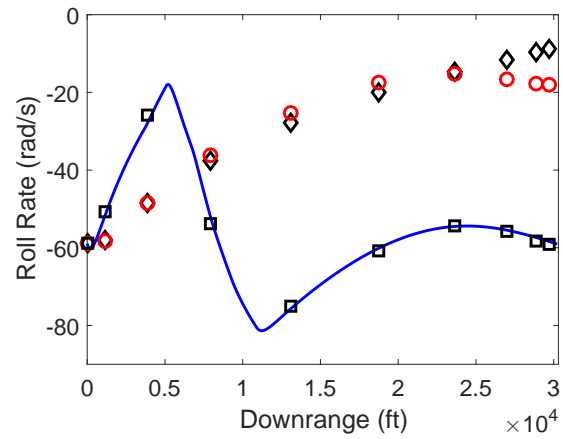
(c) Pitch



(d) Yaw

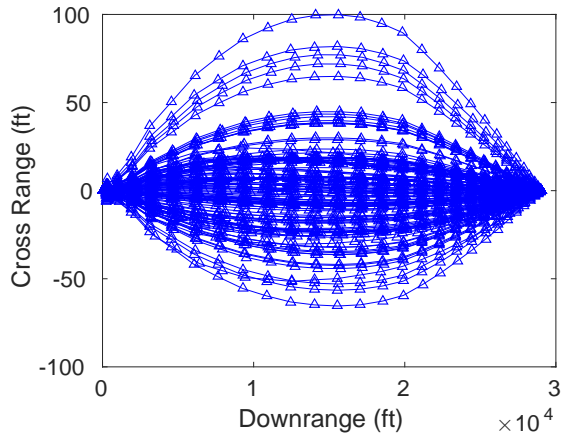


(e) Velocity

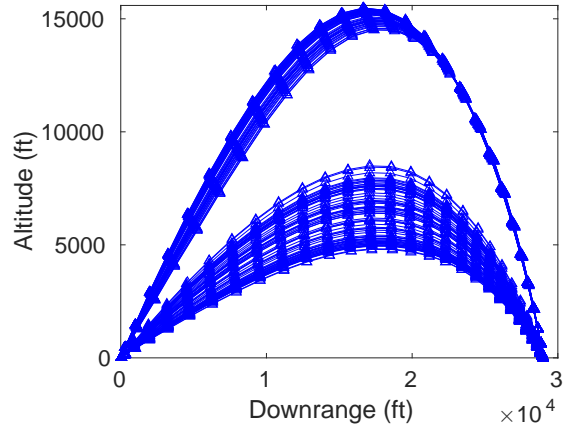


(f) Roll Rate

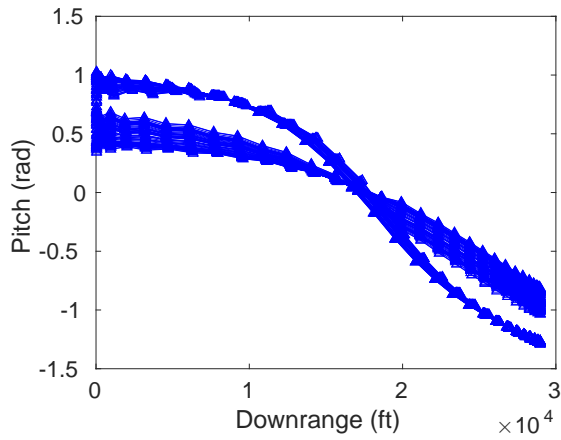
Figure 2. Prediction Comparison for High Launch Elevation Trajectory



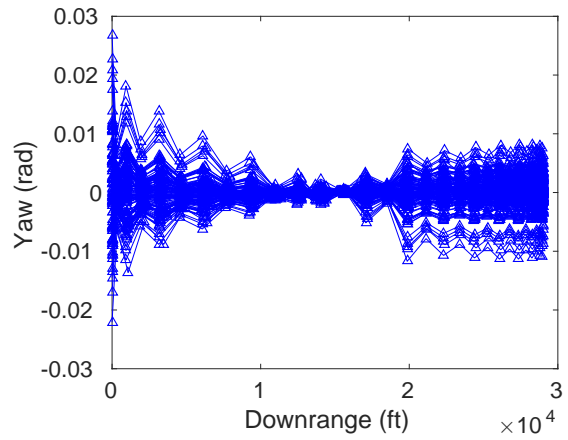
(a) Cross Range



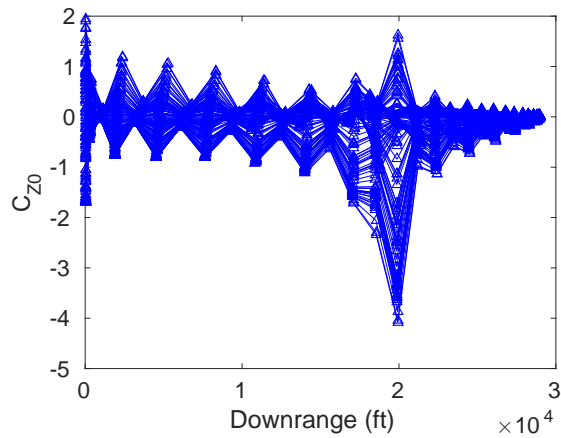
(b) Altitude



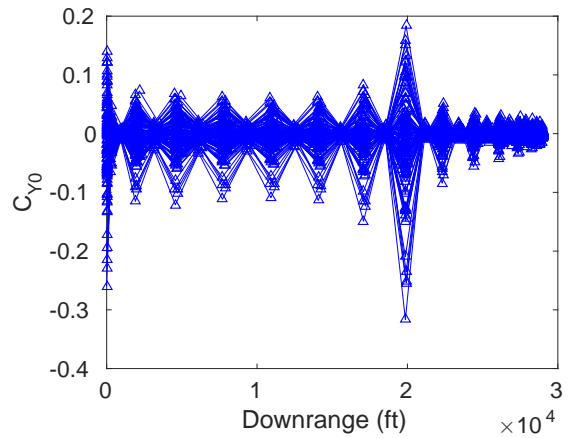
(c) Pitch



(d) Yaw



(e) Vertical Plane non-dimensional control



(f) Horizontal Plane non-dimensional control

Figure 3. Optimal Guidance Solution for Monte Carlo Set of 116 Poorly Aimed Indirect Shots

the power of Gauss PS, only ten collocation points are used for PLT and MPLT. Thus the PLT is found by solution of an 80x80 linear system. MPLT uses Newton–Raphson iteration with a Jacobian of dimension $n_s N \times n_s N$ or 100x100 for Figure (2).

Figure 2 illustrates predictions for a shot with $V_0 = 2178$ ft/s, $\theta_0 = 31.51$ degrees. The projectile returns to the ground at approximately 28000 ft. downrange. All methods perform equally well at predicting altitude during ascent. PLT predictions in altitude fail after actual apogee, apparently due to the constant velocity assumption. MPLT and MPLT with varying aero coefficients diverge at apogee with the constant aero prediction remaining below the truth model, and the varying aero prediction flying below but much closer to the truth model until the terminal point.

In cross range, the PLT greatly over predicts the projectile swerve due to assuming $\theta = 0$, and a poor total velocity estimate. MPLT with constant aero slightly under predicts cross range, and with varying aero appears to be in nearly perfect agreement with the 6DOF.

In pitch, the GPS methods fail to capture fast oscillations at the start of the trajectory due to coarse discretization. PLT fails to capture the drop in pitch due to inaccuracies in the gravity term and total velocity estimate. MPLT with constant aero follows the general trend of pitch but over predicts pitch magnitude at the terminal point. MPLT with varying aero matches closely to truth model for the entire flight. In yaw, rapid oscillation is much more prominent early in the 6DOF solution. PLT tends to oscillate wildly about 4.5×10^{-3} rad. MPLT constant aero under predicts the final yaw. With varying aero, MPLT slightly over predicts the final yaw and matches the general trend.

The total velocity estimates are greatly improved by accounting for the high launch elevation. This is shown by the MPLT and MPLT var aero velocities leveling off and then increasing after 20000 ft. downrange travel. The PLT velocity prediction continues to decrease for the full trajectory. PLT accuracy is also hindered by using a constant drag coefficient for the entire trajectory. MPLT with constant aero follows a similar trend early in the trajectory due to a constant drag coefficient. MPLT with var aero provides the best match by accounting for both large changes in pitch and high drag coefficient at transonic speeds.

Perhaps most remarkable is the contrast in roll rate predictions. Both PLT and MPLT constant aero fail to capture the general trend which is characterized by the reversal in \dot{p} around 5000 ft. downrange. Only MPLT with varying aero correctly matches the general trend which is caused by interactions between a falling total velocity reducing the effect of roll damping, and a stronger fin cant constant as Mach number decreases.

V.C. Optimal Guidance Solution

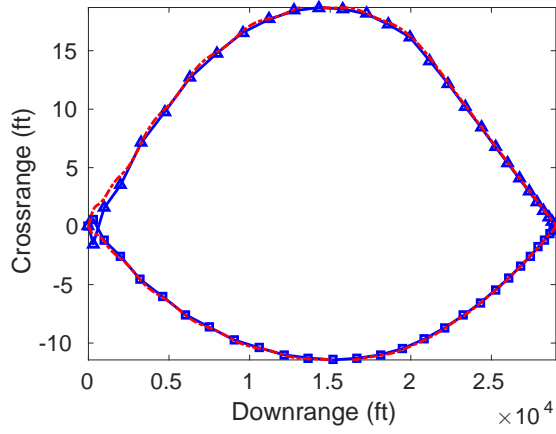
Figure 3 shows the trajectories found for the Monte Carlo set described in Nash.⁹ Initial velocity and roll rate are set at 2177.7 ft/s and -58.928 rad/s respectively. Initial y , z , v , q , and r are set to zero. As seen in Figs. 3c and d, initial pitch varies from 0.345 to 1.024 rad and yaw varies from $-2.21(10^{-2})$ to $2.68(10^{-2})$ rad. 30 collocation points are used. $\mathbf{R} = 50\mathbf{I}$, $\mathbf{S} = \text{diag}([249 \ 324 \ 0 \ \dots \ 0])$, and $\mathbf{Q} = \text{diag}([0 \ \dots \ 0 \ 100 \ 100 \ 0 \ 0])$. In other words, only terminal altitude and cross range, enroute controls and angular rates are penalized. Also, \mathbf{W} is modified such that

$$\mathbf{W} = \text{diag}(\sqrt{w_{15}} \ \sqrt{w_{14}} \ \dots \ \sqrt{w_1} \ \sqrt{w_{30}} \ \sqrt{w_{29}} \ \dots \ \sqrt{w_{16}})$$

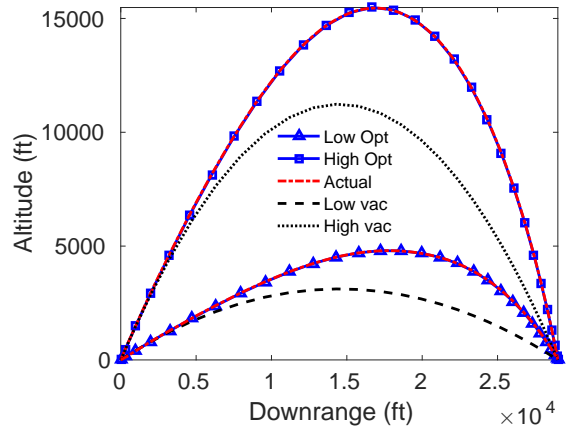
in order to penalize control effort and angular rates more severely at the initial and terminal points. Maneuver is encouraged near the middle collocation points instead. The method finds smooth trajectories for 116 of 125 initial conditions tested as shown in Figure 3. Nine of the initial conditions resulted in ill conditioning of the Jacobian and the solution was aborted. Predicted control effort remains below a non-dimensional value of 6 for all trajectories which should map to a deflection angle of 1 rad or less. It is well known that for a specified target range, there are always two ballistic launch solutions—using high and low launch elevations. The altitude plot shows a definite bifurcation where at a certain launch elevation, the algorithm selects a much higher path than previous solutions. This seems to be in keeping with the idea of minimal control effort for an optimal trajectory.

V.D. 6DOF Flight with Optimal Guidance Solution

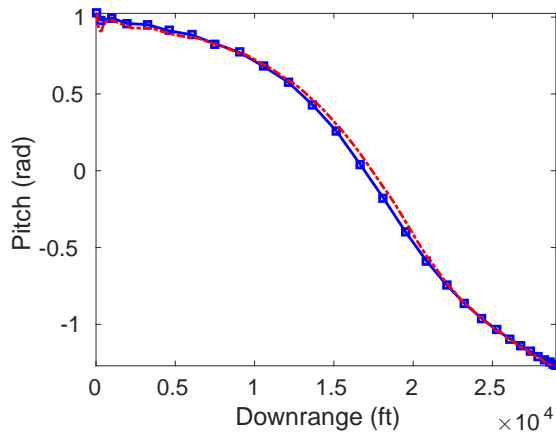
The gain scheduled LQR controller was designed and implemented on each of the successfully designed trajectories. Typical results are shown in Figure 4. Blue lines with triangles and diamonds indicate the optimal designed trajectory. Red dash-dot represent the path flown using gain scheduled LQR to command the canards. The projectile tracks the desired altitude and crossrange very well. Pitch and yaw tend to oscillate around the desired trajectory. These deviations are expected due to the low state penalties chosen on θ , ψ , v , w , q , and r . Also note the small y axis scaling on the yaw plot. Figure 4b shows the desired and actual altitude for the highest and lowest launch elevations successfully flown. In addition, point mass vacuum trajectories intersecting the third collocation point and target are shown for these two launch conditions. This was done as a basis of comparison with a previous work where the vacuum trajectory was assumed in order to provide limited state estimates and integrate the ‘optimal’ trajectory backward in time to produce an instantaneous canard command.⁹ The previous work had the advantage of circumventing an outer loop trajectory design, however, as the plot indicates, the point mass vacuum trajectory bears little resem-



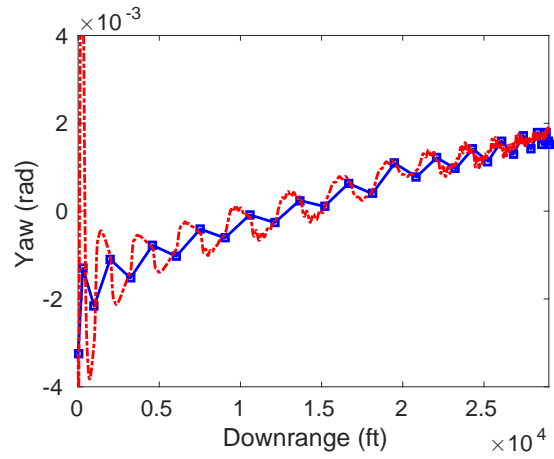
(a) Cross Range



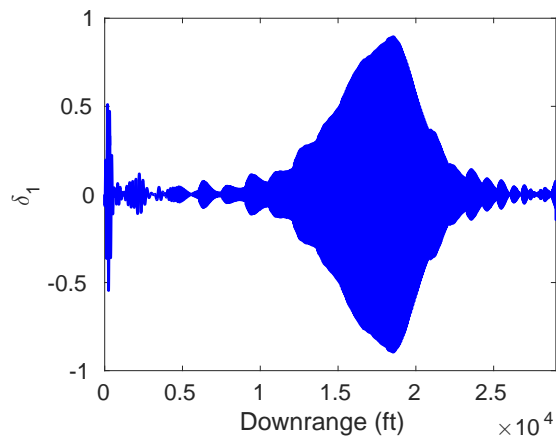
(b) Altitude



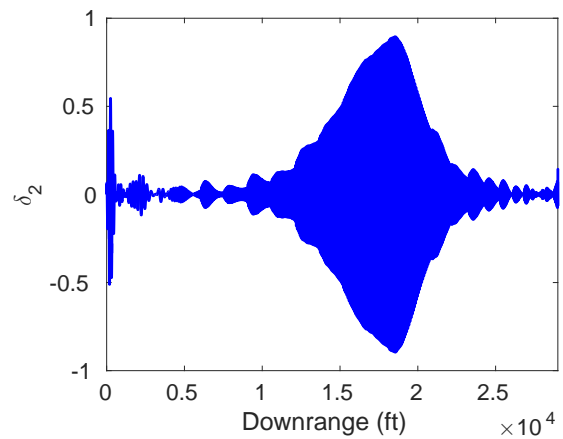
(c) Pitch High Only



(d) Yaw High Only



(e) Canard 1 High Only



(f) Canard 2 High Only

Figure 4. Optimal Guidance Solution with Closed-Loop 6DOF Simulated Flight for High and Low Launch Elevation Shot

blance to the optimal. The optimal flies much higher in all cases, and approaches the target from a steeper angle.

Deflections of canards 1 and 2 for the high launch elevation trajectory are shown in Figures 4e and 4f respectively. Since these are body frame deflections, the rapid oscillations are due to the projectile roll cycle. Deflections approach the ± 1 rad saturation limit, near 18000 ft downrange where the projectile is near apogee, total velocity is at a minimum, and maneuver is encouraged by the choice of control penalty weighting.

Dispersion near the target for all 116 trajectories is shown in Figure 5b. The target at (29000, 0, 0) is depicted as a red circle. Nearly all shots land within 0.2 ft in crossrange, however some shots fall more than fifteen feet beyond the target. In fact, the highest launch elevation shots land in a group about three feet beyond the target. As the launch angle is monotonically decreased, the miss distance monotonically increases beyond the target. The CEP for this set is 14.29 ft.

VI. Conclusions

A Gauss–Legendre pseudospectral method has been developed to predict uncontrolled trajectories of symmetric projectiles. The method provides very rapid and accurate prediction by discretizing the trajectory over a small number of points. Pseudospectral collocation provides for exact representation of system non-linearities and interactions between time varying quantities including atmospheric density and aerodynamic coefficients. This results in a method that can accurately represent and predict the dynamics of trajectories including ones with high launch elevations and transonic velocities.

By also discretizing the cost function for optimal control, the problem of optimal trajectory design is rendered as an algebraic cost function with algebraic equality constraints. Such a problem is solved by appending equality constraints to the cost function integrand with Lagrange multipliers. The resulting large set of non-linear algebraic equations is then numerically solved.

Feasibility of the optimal trajectories was demonstrated by commanding forward canards through a gain scheduled LQR inner loop. The projectile tracked desired trajectories with very little error resulting in a large reduction in dispersion at the target.

References

¹Murphy, C. H., “Data Reduction for the Free Flight Spark Ranges,” Ballistic Research Laboratories Report No. 900, 1954.

²Hainz, L. C. III, and Costello, M., “Modified Projectile Linear Theory for Rapid Trajectory Prediction,” *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 5, pp. 1006-1014, 2005.

³Elnagar, G., Kazemi, M. A., and Razzaghi, M., “The Pseudospectral Legendre Method for Discretizing Optimal Control Problems,” *IEEE Transactions on Automatic Control*, Vol. 40, No. 10, pp. 1793-1796, October, 1995.

⁴Rao, A. V., et al, “Algorithm 902: GPOPS, A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method,” *ACM Transactions on Mathematical Software*, Vol. 38, No. 2, Article 22, April 2010.

⁵Ross, I. M., *User’s Manual for DIDO: A MATLAB Application Package for Solving Optimal Control Problems*, Technical Report 04-01.0, Tomlab Optimization Inc, February 2004.

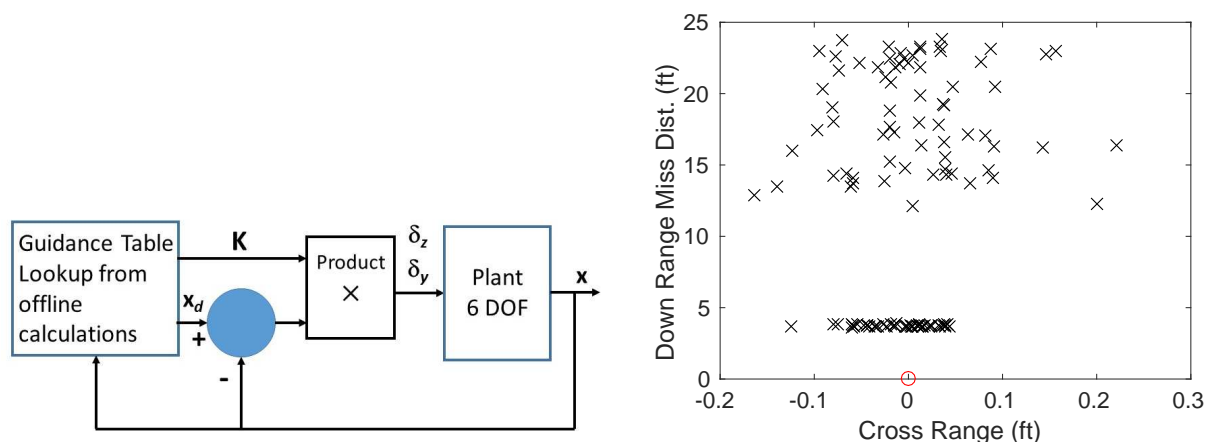
⁶Bryson, A. E., and Ho, Y., *Applied Optimal Control*, Ginn and Company, Waltham, MA, 1969.

⁷Press, W. H., et al, *Numerical Recipes in FORTRAN; The Art of Scientific Computing 2nd ed.*, Cambridge University Press, New York, NY, 1993.

⁸Burchett, B. T., “Aerodynamic Parameter Identification for Symmetric Projectiles: An Improved Gradient Based Method”, AIAA Atmospheric Flight Mechanics Conference, Minneapolis, Minnesota, 13-16 August, 2012, AIAA 2012-4861.

⁹Nash, A. L., and Burchett, B. T., ”Euler-Lagrange Optimal Control of Indirect Fire Symmetric Projectiles,” AIAA Atmospheric Flight Mechanics Conference, San Diego, CA, 4-8 January, 2016, AIAA 2016-0539.

¹⁰McCoy, R. L., *Modern Exterior Ballistics*, Schiffer, Atglen, PA, 1999.



(a) Functional Block Diagram for Gain Scheduled LQR

(b) Dispersion of 116 Trajectories successfully designed by algebraic solution of Discretized Hamiltonian Necessary Conditions

Figure 5. LQR Functional Block Diagram and Final Dispersion Plot

Appendix A

The Jacobian of Eq (34) is sparse, thus only the non-zero terms in the Jacobian are shown in Table 1. Certain constants are differentiated wrt altitude and total velocity. The scaling of A , C , E , and F are as shown in section IV.

Where

$$\frac{\partial A}{\partial z} = \frac{\partial \rho}{\partial z} \frac{SDC_{NA}}{2m} \quad (48)$$

$$\frac{\partial C}{\partial z} = \frac{\partial \rho}{\partial z} \frac{SD^2 C_{MA}}{2I_{YY}} \quad (49)$$

$$\frac{\partial E}{\partial z} = \frac{\partial \rho}{\partial z} \frac{SD^3 C_{MQ}}{4I_{YY}} \quad (50)$$

$$\frac{\partial F}{\partial p} = \frac{D I_{XX}}{V I_{YY}} \quad (51)$$

$$\frac{\partial F}{\partial V} = -\frac{I_{XX}}{I_{YY}} \frac{D}{V^2} p \quad (52)$$

$$\frac{\partial b_1}{\partial z} = \frac{\partial \rho}{\partial z} \frac{SD}{2m} V \quad (53)$$

$$\frac{\partial b_2}{\partial z} = \frac{\partial \rho}{\partial z} \frac{SD}{2I_{YY}} (SL_{CAN} - SL_{CG}) V \quad (54)$$

$$\frac{\partial \rho}{\partial z} = (0.0000068789)(4.258)(0.0023784722)(1 + 0.0000068789 * z)^{3.258} \quad (55)$$

Appendix B

Using time as the independent variable, the MPLT equations after collocation are written as:

$$\frac{2}{t_f} \Delta y - V c_\theta \psi D - v = 0 = f_1 \quad (56)$$

$$\frac{2}{t_f} \Delta z + V s_\theta - c_\theta w = 0 = f_2 \quad (57)$$

$$\frac{2}{t_f} \Delta \theta - q = 0 = f_3 \quad (58)$$

$$\frac{2}{t_f} \Delta \psi - \frac{r}{c_\theta} = 0 = f_4 \quad (59)$$

$$\frac{2}{t_f} \Delta v + Av + Vr - b_1 u_2 = 0 = f_5 \quad (60)$$

$$\frac{2}{t_f} \Delta w + Aw - Vq - gc_\theta + b_1 u_1 = 0 = f_6 \quad (61)$$

$$\frac{2}{t_f} \Delta q - Cw - Eq + \frac{I_{XX}}{I_{YY}} pr - b_2 u_1 = 0 = f_7 \quad (62)$$

$$\frac{2}{t_f} \Delta r + Cv - \frac{I_{XX}}{I_{YY}} pq - Er - b_2 u_2 = 0 = f_8 \quad (63)$$

$$\frac{2}{t_f} \Delta V + \frac{\pi \rho D^2}{8m} C_{X0} V^2 + gs_\theta = 0 = f_9 \quad (64)$$

$$\frac{2}{t_f}\Delta p - \frac{\pi\rho V^2 D^3}{8I_{XX}}C_{LDD} - \frac{\pi\rho D^4 V}{16I_{XX}}pC_{LP} = 0 = f_{10} \quad (65)$$

where the constants A , C , E , etc. are scaled according to Eqs. (11)-(16).

The Jacobian of Eq. (45) is unpacked here. Non-zero first derivatives \mathbf{f}_x are shown in Table 1, although the scaling of Table 1 is that of section IV rather than section I.

Non-zero second derivatives of the \mathbf{f} constraints are shown in Table 2 where first derivatives of the constants are rescaled as.

$$\frac{\partial A}{\partial z} = \frac{\partial \rho}{\partial z} \frac{SV}{2m} C_{NA} \quad (66)$$

$$\frac{\partial A}{\partial V} = \frac{\rho S}{2m} C_{NA} \quad (67)$$

$$\frac{\partial C}{\partial z} = \frac{\partial \rho}{\partial z} \frac{SV}{2I_{YY}} C_{MA} \quad (68)$$

$$\frac{\partial C}{\partial V} = \frac{\rho S}{2I_{YY}} C_{MA} \quad (69)$$

$$\frac{\partial E}{\partial z} = \frac{\partial \rho}{\partial z} \frac{SVD^2}{4I_{YY}} C_{MQ} \quad (70)$$

$$\frac{\partial E}{\partial V} = \frac{\rho SD^2}{4I_{YY}} C_{MQ} \quad (71)$$

$$\frac{\partial b_1}{\partial z} = \frac{\partial \rho}{\partial z} \frac{V^2 S_{CAN}}{2m} \quad (72)$$

$$\frac{\partial b_1}{\partial V} = \frac{\rho V S_{CAN}}{m} \quad (73)$$

$$\frac{\partial b_2}{\partial z} = \frac{\partial \rho}{\partial z} \frac{V^2 S_{CAN}}{2I_{YY}} (SL_{CAN} - SL_{CG}) \quad (74)$$

$$\frac{\partial b_2}{\partial V} = \frac{\rho V S_{CAN}}{I_{YY}} (SL_{CAN} - SL_{CG}) \quad (75)$$

$$(76)$$

Note that transposed terms are omitted from table 2, since the order of differentiation does not change the result. For instance: $(f_7)_{pr} = (f_7)_{rp}$. Second derivatives of the constants are written as

$$\frac{\partial^2 A}{\partial z^2} = \frac{\partial^2 \rho}{\partial z^2} \frac{SV}{2m} C_{NA} \quad (77)$$

$$\frac{\partial^2 A}{\partial V \partial z} = \frac{\partial \rho}{\partial z} \frac{S}{2m} C_{NA} \quad (78)$$

$$\frac{\partial^2 C}{\partial z^2} = \frac{\partial^2 \rho}{\partial z^2} \frac{SV}{2I_{YY}} C_{MA} \quad (79)$$

$$\frac{\partial^2 C}{\partial V \partial z} = \frac{\partial \rho}{\partial z} \frac{S}{2I_{YY}} C_{MA} \quad (80)$$

Table 1. Non-zero Jacobian Matrix Terms

$\frac{\partial f_1}{\partial y} = \frac{2}{s_f} \Delta$	$\frac{\partial f_1}{\partial \psi} = -Dc_\theta$	$\frac{\partial f_1}{\partial \theta} = Ds_\theta \psi$	$\frac{\partial f_1}{\partial v} = \frac{-D}{V}$	$\frac{\partial f_1}{\partial V} = \frac{D}{V^2} v$
$\frac{\partial f_1}{\partial s_f} = -\frac{2}{s_f^2} \Delta y$	$\frac{\partial f_2}{\partial z} = \frac{2}{s_f} \Delta$	$\frac{\partial f_2}{\partial \theta} = \frac{D}{V} s_\theta w + Dc_\theta$	$\frac{\partial f_2}{\partial w} = \frac{-D}{V} c_\theta$	$\frac{\partial f_2}{\partial V} = \frac{D}{V^2} c_\theta w$
$\frac{\partial f_2}{\partial s_f} = -\frac{2}{s_f^2} \Delta z$	$\frac{\partial f_3}{\partial \theta} = \frac{2}{s_f} \Delta$	$\frac{\partial f_3}{\partial q} = -\frac{D}{V}$	$\frac{\partial f_3}{\partial V} = \frac{D}{V^2} q$	$\frac{\partial f_3}{\partial s_f} = -\frac{2}{s_f^2} \Delta \theta$
$\frac{\partial f_4}{\partial \theta} = -\frac{D}{Vc_\theta^2} s_\theta r$	$\frac{\partial f_4}{\partial \psi} = \frac{2}{s_f} \Delta$	$\frac{\partial f_4}{\partial r} = \frac{-D}{Vc_\theta}$	$\frac{\partial f_4}{\partial V} = \frac{D}{V^2 c_\theta} r$	$\frac{\partial f_4}{\partial s_f} = -\frac{2}{s_f^2} \Delta \psi$
$\frac{\partial f_5}{\partial z} = \frac{\partial A}{\partial z} v$	$\frac{\partial f_5}{\partial v} = \frac{2}{s_f} \Delta + A\mathbf{I}$	$\frac{\partial f_5}{\partial r} = D\mathbf{I}$	$\frac{\partial f_5}{\partial s_f} = -\frac{2}{s_f^2} \Delta v$	$\frac{\partial f_6}{\partial \theta} = \frac{Dg}{V} s_\theta$
$\frac{\partial f_6}{\partial w} = \frac{2}{s_f} \Delta + A\mathbf{I}$	$\frac{\partial f_6}{\partial q} = -D\mathbf{I}$	$\frac{\partial f_6}{\partial z} = \frac{\partial A}{\partial z} w$	$\frac{\partial f_6}{\partial s_f} = -\frac{2}{s_f^2} \Delta w$	$\frac{\partial f_7}{\partial s_f} = -\frac{2}{s_f^2} \Delta q$
$\frac{\partial f_7}{\partial w} = -\frac{C}{D}$	$\frac{\partial f_7}{\partial q} = \frac{2}{s_f} \Delta - E\mathbf{I}$	$\frac{\partial f_7}{\partial r} = F\mathbf{I}$	$\frac{\partial f_7}{\partial p} = \frac{\partial F}{\partial p} r$	$\frac{\partial f_8}{\partial v} = \frac{C}{D}$
$\frac{\partial f_8}{\partial q} = -F\mathbf{I}$	$\frac{\partial f_8}{\partial r} = \frac{2}{s_f} \Delta - E\mathbf{I}$	$\frac{\partial f_8}{\partial p} = -\frac{\partial F}{\partial p} q$	$\frac{\partial f_8}{\partial s_f} = -\frac{2}{s_f^2} \Delta r$	$\frac{\partial f_9}{\partial s_f} = -\frac{2}{s_f^2} \Delta V$
$\frac{\partial f_9}{\partial z} = \frac{\pi D^3}{8m} \frac{\partial \rho}{\partial z} C_{X0} V$	$\frac{\partial f_9}{\partial \theta} = \frac{Dg}{V} c_\theta$	$\frac{\partial f_{10}}{\partial V} = -\frac{\pi \rho D^4}{8I_{XX}} C_{LDD}$	$\frac{\partial f_{10}}{\partial s_f} = -\frac{2}{s_f^2} \Delta p$	
$\frac{\partial f_6}{\partial V} = \frac{Dg}{V^2} c_\theta + \frac{\partial b_1}{\partial V} u_1$		$\frac{\partial f_7}{\partial z} = -\frac{1}{D} \frac{\partial C}{\partial z} w - \frac{\partial E}{\partial z} q$		
$\frac{\partial f_7}{\partial V} = -\frac{1}{D} \frac{\partial C}{\partial V} w - \frac{\partial E}{\partial V} r + \frac{\partial F}{\partial V} r - \frac{\partial b_2}{\partial V} u_1$		$\frac{\partial f_8}{\partial V} = -\frac{\partial b_2}{\partial V} u + 2 + \frac{1}{D} \frac{\partial C}{\partial V} v - \frac{\partial F}{\partial V} q - \frac{\partial E}{\partial V} r$		
$\frac{\partial f_8}{\partial z} = \frac{1}{D} \frac{\partial C}{\partial z} v - \frac{\partial E}{\partial z} r$		$\frac{\partial f_9}{\partial V} = \frac{2}{s_f} \Delta - \frac{Dg}{V^2} s_\theta + \frac{\pi \rho D^3}{8m} C_{X0}$		
$\frac{\partial f_{10}}{\partial p} = \frac{2}{s_f} \Delta - \frac{\pi \rho D^5}{16I_{XX}} C_{LP}$		$\frac{\partial f_{11}}{\partial s_f} = D\mathbf{W}(c_\theta c_\psi V)/(\mathbf{W}V)$		
$\frac{\partial f_{11}}{\partial \theta} = -Ds_f \mathbf{W}(s_\theta c_\theta V)/(\mathbf{W}V)$		$\frac{\partial f_{11}}{\partial \psi} = -Ds_f \mathbf{W}(c_\theta s_\psi V)/(\mathbf{W}V)$		

Table 2. Non-zero 2nd Derivative Terms

$(f_5)_{zv} = \frac{\partial A}{\partial z}$	$(f_6)_{zz} = \frac{\partial^2 A}{\partial z^2} w + \frac{\partial^2 b_1}{\partial z^2} u_1$	$(f_6)_{zw} = \frac{\partial A}{\partial z}$	$(f_7)_{zw} = -\frac{\partial C}{\partial z}$
$(f_7)_{zq} = -\frac{\partial E}{\partial z}$	$(f_8)_{zv} = \frac{\partial C}{\partial z}$	$(f_8)_{zr} = \frac{\partial E}{\partial z}$	$(f_9)_{zz} = \frac{\partial^2 \rho}{\partial z^2} \frac{\pi D^2}{8m} C_{X0} V^2$
$(f_9)_{zV} = \frac{\partial \rho}{\partial z} \frac{\pi D^2}{4m} C_{X0} V$	$(f_{10})_{zp} = -\frac{\partial \rho}{\partial z} \frac{\pi D^4 V}{16 I_{XX}} C_{LP}$	$(f_1)_{\theta\theta} = V \psi c_\theta$	$(f_1)_{\theta\psi} = V s_\theta$
$(f_1)_{\theta V} = \psi s_\theta$	$(f_2)_{\theta\theta} = -V s_\theta + w c_\theta$	$(f_2)_{\theta w} = s_\theta$	$(f_2)_{\theta V} = c_\theta$
$(f_4)_{\theta r} = -\frac{s_\theta}{c_\theta^2}$	$(f_6)_{\theta w} = g c_\theta$	$(f_9)_{\theta\theta} = -g s_\theta$	$(f_{11})_{\theta\theta} = -\frac{t_f}{2} w_i c_{\theta i} c_{\psi i} V_i$
$(f_{11})_{\theta\psi} = \frac{t_f}{2} w_i s_{\theta i} s_{\psi i} V_i$	$(f_{11})_{\theta V} = -\frac{t_f}{2} w_i s_{\theta i} c_{\psi i}$	$(f_1)_{\psi V} = -c_\theta$	$(f_{11})_{\psi\psi} = -\frac{t_f}{2} w_i c_{\theta i} c_{\psi i} V_i$
$(f_{11})_{\psi V} = -\frac{t_f}{2} w_i c_{\theta i} s_{\psi i}$	$(f_7)_{wq} = -\frac{\partial C}{\partial z}$	$(f_6)_{qV} = -1$	$(f_8)_{qp} = -\frac{I_{XX}}{I_{YY}}$
$(f_5)_{rV} = 1$	$(f_8)_{rV} = -\frac{\partial E}{\partial V}$	$(f_7)_{rp} = \frac{I_{XX}}{I_{YY}}$	$(f_9)_{VV} = \frac{\rho \pi D^2}{4m} C_{X0}$
$(f_{10})_{VV} = -\frac{\rho \pi D^3}{4 I_{XX}} C_{LDD}$	$(f_{10})_{Vp} = -\frac{\rho \pi D^4}{16 I_{XX}} C_{LP}$		
$(f_5)_{zz} = \frac{\partial^2 A}{\partial z^2} v - \frac{\partial^2 b_1}{\partial z^2} u_2$		$(f_5)_{zV} = \frac{\partial^2 A}{\partial V \partial z} v - \frac{\partial^2 b_1}{\partial V \partial z} u_2$	
$(f_6)_{zV} = \frac{\partial^2 A}{\partial V \partial z} w + \frac{\partial^2 b_1}{\partial V \partial z} u_1$		$(f_7)_{zz} = -\frac{\partial^2 C}{\partial z^2} w - \frac{\partial^2 E}{\partial z^2} q - \frac{\partial^2 b_2}{\partial z^2} u_1$	
$(f_7)_{zV} = -\frac{\partial^2 C}{\partial V \partial z} w - \frac{\partial^2 E}{\partial V \partial z} q - \frac{\partial^2 b_2}{\partial V \partial z} u_1$		$(f_8)_{zz} = \frac{\partial^2 C}{\partial z^2} v - \frac{\partial^2 E}{\partial z^2} r - \frac{\partial^2 b_2}{\partial z^2} u_2$	
$(f_8)_{zV} = \frac{\partial^2 C}{\partial V \partial z} v - \frac{\partial^2 E}{\partial V \partial z} r - \frac{\partial^2 b_2}{\partial V \partial z} u_2$		$(f_{10})_{zz} = -\frac{\partial^2 \rho}{\partial z^2} \frac{\pi D^3}{8 I_{XX}} C_{LDD} V^2 - \frac{\partial^2 \rho}{\partial z^2} \frac{\pi D^4 V}{16 I_{XX}} p C_{LP}$	
$(f_{10})_{zV} = -\frac{\partial \rho}{\partial z} \frac{\pi D^3 V}{4 I_{XX}} C_{LDD} - \frac{\partial \rho}{\partial z} \frac{\pi D^4}{16 I_{XX}} p C_{LP}$		$(f_4)_{\theta\theta} = -2 \frac{s_\theta^2}{c_\theta^3} r - \frac{r}{c_\theta}$	
$(f_5)_{VV} = \frac{\partial^2 b_1}{\partial V^2} u_2 + \frac{\partial^2 A}{\partial V^2} v$		$(f_6)_{VV} = \frac{\partial^2 b_1}{\partial V^2} u_1 + \frac{\partial^2 A}{\partial V^2} w$	
$(f_7)_{VV} = -\frac{\partial^2 E}{\partial V^2} q - \frac{\partial^2 b_2}{\partial V^2} u_1 - \frac{\partial^2 C}{\partial V^2} w$		$(f_8)_{VV} = -\frac{\partial^2 b_2}{\partial V^2} u_2 + \frac{\partial^2 C}{\partial V^2} v - \frac{\partial^2 E}{\partial V^2} r$	

$$\frac{\partial^2 E}{\partial z^2} = \frac{\partial^2 \rho}{\partial z^2} \frac{SVD^2}{4I_{YY}} C_{MQ} \quad (81)$$

$$\frac{\partial^2 E}{\partial V \partial z} = \frac{\partial \rho}{\partial z} \frac{SD^2}{4I_{YY}} C_{MQ} \quad (82)$$

$$\frac{\partial^2 b_1}{\partial z^2} = \frac{\partial^2 \rho}{\partial z^2} \frac{V^2 S_{CAN}}{2m} \quad (83)$$

$$\frac{\partial^2 b_1}{\partial V^2} = \frac{\rho S_{CAN}}{m} \quad (84)$$

$$\frac{\partial^2 b_1}{\partial V \partial z} = \frac{\partial \rho}{\partial z} \frac{V S_{CAN}}{m} \quad (85)$$

$$\frac{\partial^2 b_2}{\partial z^2} = \frac{\partial^2 \rho}{\partial z^2} \frac{V^2 S_{CAN}}{2I_{YY}} (SL_{CAN} - SL_{CG}) \quad (86)$$

$$\frac{\partial^2 b_2}{\partial V \partial z} = \frac{\partial \rho}{\partial z} \frac{V S_{CAN}}{I_{YY}} (SL_{CAN} - SL_{CG}) \quad (87)$$

$$\frac{\partial^2 b_2}{\partial V^2} = \frac{\rho S_{CAN}}{I_{YY}} (SL_{CAN} - SL_{CG}) \quad (88)$$

$$\frac{\partial^2 \rho}{\partial z^2} = (0.0000068789)^2 (4.258)(3.258)(0.0023784722)(1 + 0.0000068789 * z)^{2.258} \quad (89)$$

Appendix C

As mentioned previously, the term $\mathbf{f}_{\mathbf{x}\mathbf{x}}$ is a third order tensor, which reduces to a matrix after multiplication by the vector of Lagrange multipliers, λ . To avoid the use of tensors, consider instead the differentiation of $\lambda^T \mathbf{f}_{\mathbf{x}}$, that is $\partial/\partial \mathbf{x}(\mathbf{f}_{\mathbf{x}}^T \lambda)$ or $\partial/\partial \mathbf{x}(\mathbf{f}_{\mathbf{x}}^T) \lambda$. For practical calculation, take each column of $\mathbf{f}_{\mathbf{x}}$ and differentiate by \mathbf{x} . Take the product of matrix $(\mathbf{f}_{xi})_{\mathbf{x}}^T \lambda$, then concatenate all instances i to form $(\mathbf{f}_{\mathbf{x}})_{\mathbf{x}}^T \lambda$. That is

$$\mathbf{f}_{\mathbf{x}\mathbf{x}}^T \lambda = \begin{bmatrix} (\mathbf{f}_y)_{\mathbf{x}}^T \lambda & (\mathbf{f}_z)_{\mathbf{x}}^T \lambda & \dots & (\mathbf{f}_p)_{\mathbf{x}}^T \lambda \end{bmatrix}$$

Also note in this case that λ must be expanded into a block diagonal to preserve independence of the individual instances of Eqs. (56)–(65) for each collocation point. For example $(\mathbf{f}_{\theta})_{\mathbf{x}}^T \lambda$

when expanded is

$$\begin{bmatrix} 0 & 0 & (\mathbf{f}_1)_{\theta\theta} & (\mathbf{f}_1)_{\theta\psi} & \cdots \\ 0 & 0 & (\mathbf{f}_2)_{\theta\theta} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ 0 & 0 & (\mathbf{f}_{11})_{\theta\theta} & (\mathbf{f}_{11})_{\theta\psi} & \cdots \end{bmatrix}^T \begin{bmatrix} \lambda_1 & & & & \\ & \ddots & & & \\ & & & & \lambda_m \\ \lambda_{m+1} & & & & \\ & \ddots & & & \\ & & & & \lambda_{2m} \\ & & \vdots & & \\ & & & & \lambda_{10m+1} \mathbf{I} \end{bmatrix}$$

Additionally, sensitivities wrt the final time are treated separately so the term $(\mathbf{f}_{tf})_{\mathbf{x}}^T \lambda + \Lambda_{tf\mathbf{x}}$

$$= \begin{bmatrix} (\mathbf{f}_{\mathbf{x}})_{tf}^T & (\mathbf{f}_{11})_{\mathbf{x}tf}^T \end{bmatrix} \lambda + \begin{bmatrix} 0 & \cdots & 0 & 25\mathbf{W}q & 25\mathbf{W}r & 0 & \cdots & 0 \end{bmatrix}^T$$

or

$$\mathbf{f}_{\mathbf{xx}}^T \lambda + \Lambda_{\mathbf{xx}} = \begin{bmatrix} \mathbf{f}_{\mathbf{xx}1:10}^T \lambda + \Lambda_{\mathbf{xx}1:10} & \begin{bmatrix} (\mathbf{f}_{\mathbf{x}})_{tf}^T & (\mathbf{f}_{11})_{\mathbf{x}tf}^T \end{bmatrix} \lambda + \Lambda_{\mathbf{x}tf}^T \\ \lambda^T \begin{bmatrix} (\mathbf{f}_{\mathbf{x}})_{tf} \\ (\mathbf{f}_{11})_{\mathbf{x}tf} \end{bmatrix} + \Lambda_{\mathbf{x}tf} & \lambda^T (\mathbf{f})_{tftf} \end{bmatrix}$$