ARL-TR-9281 ● SEP 2021

DEVCOM ARMY RESEARCH LABORATORY

Spark Range Data Reduction Using a Gauss Collocation Method

by Bradley T Burchett

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-9281 ● SEP 2021

Spark Range Data Reduction Using a Gauss Collocation Method

Bradley T Burchett
Weapons and Materials Research Directorate,
DEVCOM Army Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
September 2021	Technical Report	1 May–31 July 2021

4. TITLE AND SUBTITLE
Spark Range Data Reduction Using a Gauss Collocation Method

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Bradley T Burchett

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
DEVCOM Army Research Laboratory
ATTN: FCDD-RLW-WD
Aberdeen Proving Ground, MD 21005

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-9281

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCID ID: Bradley T Burchett, 0000-0002-1934-0537

14. ABSTRACT

A new method is proposed for reducing spark range data using a Gauss pseudospectral collocation. Existing methods use numerical integration to propagate the differential equation model and its sensitivities forward in time. Here a collocation method is used to transcribe the differential equations into algebraic ones that are easily solved. Realizing that the auxiliary equations used to find model sensitivities are linear, a second transcription renders a set of linear algebraic equations that solve exactly for the needed sensitivities without iteration. Thus, measurement sensitivities with respect to initial conditions and aerodynamic parameters are easily found through solution of a set of linear algebraic equations. The method is demonstrated on a set of actual data from the M898 155-mm projectile. For models using fixed plane coordinates, and only the 10 most influential aerodynamic coefficients, the new method produces results that rival established commercial codes in accuracy.

15. SUBJECT TERMS
spark range, projectile aerodynamics, trajectory reconstruction, initial conditions, pseudospectral collocation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Bradley T Burchett
Unclassified	Unclassified	Unclassified	UU	50	19b. TELEPHONE NUMBER (Include area code) (410) 278-2479

Contents

List of Figures

List of Tables

1. Introduction

The aeroballistic spark range is a facility where projectiles are launched in free flight in a dark, enclosed space. As the projectile flies past each spark station, a short-duration light pulse is activated. In a small-scale range (such as the Aerodynamics Experimental Facility at Aberdeen Proving Ground), the projectile flies between the spark source and an open film plate, thus throwing a shadow directly on the film. In larger-scale ranges, the shadow is thrown on a backdrop, which is photographed by an open-shutter large-format camera. A second photographic plate captures a second image on a plane perpendicular to the first, and thus the projectile position in 3-D space may be established from the two shadowgraphs. By capturing the position at several (15–75) stations along the trajectory, the projectile's path through space is known. Aeroballistic spark ranges are the best method to estimate aerodynamic coefficients of projectiles. Unlike the wind tunnel, free flight tests do not require corrections for sting force and tunnel interference effects. Spark ranges have been in operation since World War II, and the methods for data reduction have been established nearly as long.[1,2] This process reconstructs the projectile trajectory through the range, while refining estimates of the aerodynamic coefficients until the model trajectory fits the data within a specified tolerance. For the past 50 years, spark range data reduction has been accomplished by a Chapman–Kirk method where the model trajectory and a set of auxiliary equations are numerically integrated forward in time.[3,4] The model parameters are then iteratively corrected using a Marquardt search until the model predictions are within a specified tolerance of the data or parametric corrections fall below a specified threshold. This method is embedded in the ARFDAS commercial software developed during the 1970s and 1980s by Hathaway and Whyte.[5] Bradley also developed a general nonlinear ordinary differential equation (ODE)-solving Fortran engine,[6] and his report provided an important building block for the current work.

The aforementioned methods work well for axisymmetric projectiles of all sizes. However, as programs branch out into asymmetric and actively controlled designs, it becomes necessary for large portions of the existing code to be modified to include more-complex models. When these methods are implemented apart from compiled code, they can be extremely slow.

In this work, we explore the application of collocation methods to this problem. We seek to build an algorithm that is faster than numerical integration and allows for easier modification of the chosen aerodynamic model. Existing methods require a significant amount of computational effort to find the sensitivities required for a gradient-based search, either through the propagation of a set of costates in parallel

with the trajectory prediction or through finite differencing of the solution. Collocation provides a very fast way to find sensitivities of the solution with respect to system parameters and initial conditions. The relationship between these two types of sensitivities can be exploited to find sensitivities with respect to the aerodynamic parameters without the large analytic burden required by existing methods.

Collocation methods have recently come to the forefront in the aerospace control community, specifically to solve nonlinear optimal control problems.[7–9] Collocation allows for ODEs both linear and nonlinear to be rapidly solved by transcribing the problem onto a coarse time grid. The grid or abscissa points are specified as a set of roots of the underlying basis polynomial chosen for transcription. However, the transcription also provides an exact mapping from differential equations to algebraic ones. Linear ODEs are then solved as linear algebraic equations. Nonlinear ODEs are solved as nonlinear equations through a Newton method.

In this work, we explore the application of collocation methods to the reduction of spark range data. One significant issue arises immediately in that the downrange positions of the spark stations are fixed at locations that do not correspond to any set of Gauss collocation points. The points to be used in Gauss Pseudospectral (GPS) collocation are mathematically established as roots of the underlying basis polynomials, and cannot be arbitrarily changed without inducing ill conditioning in the problem. Thus, the spark range data will need to be interpolated onto the collocation grid, which will introduce additional errors. However, the established methods in effect also interpolate the solution in time/distance to match the spark stations, just starting from a finer time grid. In the case of collocation, this complication can be overcome by breaking the collocation solution into several segments which begin and end at the spark range stations. Using Legendre–Gauss–Lobatto (LGL) collocation, the endpoints will be part of the transcribed abscissa. Small sets of LGL collocation points will be used between spark station observations.

Figure 1 illustrates this process. We have zoomed in on the GPS altitude prediction for a shot through the Transonic Experimental Facility such that only stations 2-4, 2-5, 3-1, and 3-2 are shown. Using seven LGL points per segment, the segment endpoints overlap at the spark range stations—thus the GPS prediction does not require further interpolation prior to comparison with the data. For this specific case, the remaining five points will interpolate the solution between stations. This is similar to solutions using time-based integration in that many time steps would be needed between stations to accurately predict the solution. Also, note in the figure that the stations are not evenly spaced—a longer gap exists between the

station “groups”. The LGL grid size will be automatically adjusted such that the [-1,1] abscissa in Legendre domain is mapped to the time and distance between spark range stations.

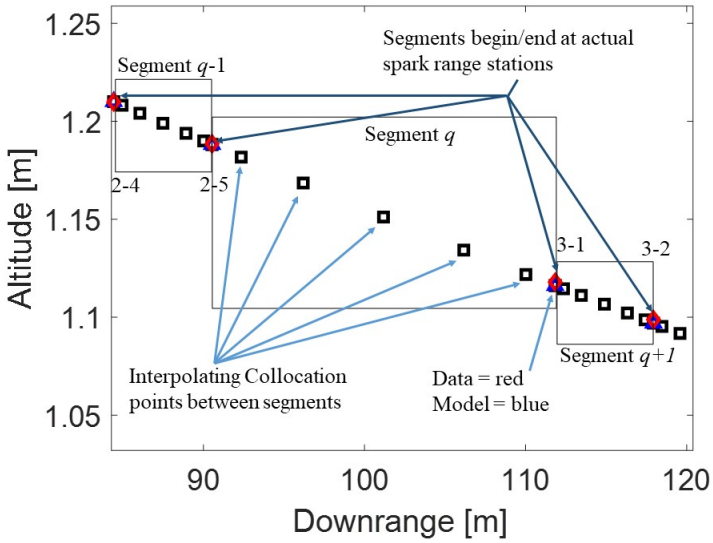


Fig. 1 Multi-segment Gauss collocation and interpolation illustrated

2. Approach: GPS Collocation

GPS collocation has been in use for many years; however, it was not widely known in the aerospace community when the Chapman–Kirk method was introduced and later adapted to spark range reductions.[3]

GPS methods came to the forefront in the optimal control community in the early 2000s to solve nonlinear optimal control problems.[7] Two notable commercial software packages came out of this—GPOPS and DIDO—both of which are implemented in the MATLAB environment.[8,9] Here we use collocation to transcribe nonlinear models of projectile flight and auxiliary models of paramic sensitivities into nonlinear algebraic equations, which may be solved iteratively. Optimization is not needed.

Sets of coupled nonlinear ODEs may be discretized using Legendre polynomials as the underlying basis functions. Let $L_N(x)$ represent the Legendre polynomial of order N. The LGL collocation points are defined as $x_0 = -1, x_N = 1$ with interior points $x_m, m = 1, \ldots, N-1$ found from the roots of $\dot{L}_N$, the first derivative of L_N. This choice will allow the end of each segment to correspond to a spark range station.

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

$$\lambda_l(x) = \frac{1}{N(N+1)L_N(x_l)} \cdot \frac{(x^2-1)\dot{L}_N(x)}{x - x_l}, \qquad (l = 0,1,\dots,N) \tag{1}$$

Lagrange polynomials have the Kronecker delta property such that

$$\lambda_l(x_j) = \begin{cases} 1 & \text{if } l = j \\ 0 & \text{if } l \neq j \end{cases} \tag{2}$$

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

$$\Gamma^N(x) = \sum_{l=0}^{N} \Gamma(x_l)\, \lambda_l(x) \tag{3}$$

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

$$\Gamma^N(x_k) = \Gamma(x_k), \;\; k = 0,1,\dots,N$$

The derivative of $\Gamma^N(x)$ can then be found by differentiating Eq. 3. The result is a matrix multiplication that can then be used to discretize differential equations.

$$\dot{\Gamma}^N(x_m) = \sum_{l=0}^{N} \Delta_{ml}\Gamma(x_l) \tag{4}$$

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

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

Here we use the symbol Δ for the differentiation matrix so as not to conflict with *D*, the projectile characteristic length. 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 the differentiation matrix is automatically scaled assuming a time span of 2 s ($-1 \leq t < 1$). Thus, when discretizing ODEs, the scaling must be modified to reflect the actual time span. For instance, $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$ becomes

$$\frac{2}{t_f - t_0}\mathbf{\Delta X} = \mathbf{AX} + \mathbf{Bu}$$

3. Gauss Collocation of Models of Projectile Flight

Historically, spark range reduction consists of the fitting of a deliberate sequence of models to specific dimensions of the data set to limit the problem dimensionality at each step. First, drag is estimated by considering the measurements of time and downrange distance. Next, roll characteristics are found by looking at roll decoupled from states other than velocity, roll rate, and downrange distance. Third, pitch and yaw angles are used to fit a model—typically linear—of the epicyclic motion and estimate linear pitch/yaw parameters. Finally, altitude and crossrange are added to the mix and a full model is fit to find remaining parameters.

Each of these steps is included in the Gauss collocation method, with the notable difference that we use parametric linear models where the traditional approach uses nondimensional parameters in the linear fit from which nonlinear parameters are backed-out after fitting.

The downrange–velocity model is

$$\dot{x} = V \tag{5}$$

$$\dot{V} = -\frac{\pi\rho D^2}{8\bar{m}} C_{X_0} V^2 \tag{6}$$

This model is used to match downrange distance and time to correct erroneous time measures and find an initial drag estimate.

The roll model is

$$\dot{\phi} = p \tag{7}$$

$$\dot{p} = -\frac{\pi\rho V^2 D^3}{8I_{XX}} C_{l_0} + \frac{\pi\rho D^4 V}{16I_{XX}} p C_{l_p} \tag{8}$$

When roll data is available, this model is used to match the roll data, provide an initial roll-rate estimate, and determine the roll parameters in Eq. 8. When roll data is not available, the initial spin rate is estimated from muzzle velocity and gun twist. Equations 7 and 8 are then integrated to provide an estimate of the roll-rate profile.

Two fixed-plane models of pitch, yaw, and swerve are used in this work. Fixed-plane models are preferred over body-fixed for this work because they provide accurate solutions while using a more coarse time grid. However, fixed-plane models have the disadvantage that they cannot resolve force and moment transients

that may occur over a roll cycle. Both fixed-plane models use an earth-fixed frame for position $\{x,y,z\}$ such that x points downrange, y to the gunner's right, and z is positive down. Pitch, θ, is the angle subtended in the vertical plane from the horizontal to the projectile axis of symmetry and is positive for nose-up flight. Yaw, ψ is positive for nose right rotation and is measured in the horizontal plane from the downrange vector to the projectile axis of symmetry. Both use a non-rolling reference frame for body-fixed velocities and angular rates such that q is the body angular rate in the vertical plane, and r is the body angular rate in the horizontal plane. The body-fixed non-rolling frame is found by first rotating through ψ about the earth fixed z axis, then from this frame rotating through θ about the intermediate y axis. Body-fixed rectilinear velocities v and w are along the y and z axes of the body-fixed non-rolling frame, respectively

The linear epicyclic model assumes small Euler angles (θ, ψ), small angle of attack and sideslip, flat fire, as well as mass and aerodynamic symmetry. We differ from traditional linear theory here in keeping time as the independent variable. The linear epicyclic model is given in Eqs. 9 through 16.

$$\dot{y} = V\psi + v \tag{9}$$

$$\dot{z} = V\theta + w \tag{10}$$

$$\dot{\theta} = q \tag{11}$$

$$\dot{\psi} = r \tag{12}$$

$$\dot{v} = -Av - rV \tag{13}$$

$$\dot{w} = -Aw + qV \tag{14}$$

$$\dot{q} = Bv + \frac{C}{D}w + Eq + Fr \tag{15}$$

$$\dot{r} = -\frac{C}{D}v + Bw + Fq + Er \tag{16}$$

where the convenience variables are

$$A = \frac{\bar{q}S}{\bar{m}V}C_{N_\alpha}$$

$$B = \frac{\bar{q}SDp}{I_{YY}V^2}C_{n_{p\alpha}}$$

$$C = \frac{\bar{q}SD}{I_{YY}V} C_{m_\alpha} =$$

$$E = \frac{\bar{q}SD^2}{2I_{YY}V} C_{m_q}$$

$$F = \frac{I_{XX}}{I_{YY}} p$$

To accurately seed the nonlinear fit with estimates of parameters and initial conditions, a linear epicyclic model is used for two preliminary fits. First, this model is fit to pitch and yaw measurements of the first full oscillation of motion (typically the first 6 to 10 stations) to determine initial angular rates and first guesses on transverse linear-motion parameters. Second, the full set of pitch, yaw, crossrange, and altitude measurements are fit to the complete 8-state linear epicyclic model to refine initial guesses on these states plus parameters prior to the nonlinear fit.

The fixed plane 6-degrees-of-freedom (6DOF) equations of motion use the non-rolling frame described such that body-fixed velocity *w* is always in the vertical plane and *v* is always in the horizontal plane. Unlike the linear epicyclic model, pitch and yaw can be large, and nonlinear aerodynamics such as $C_{N_{\alpha 3}}$ enter the mix. Roll, roll rate, total velocity, and downrange distance are treated as known quantities from the downrange–velocity–roll–spin solution. The fixed-plane 6DOF equations of motion are given in Eqs. 17–26.

$$\dot{y} = V\, c_\theta\, s_\psi + v c_\psi + w s_\theta s_\psi \tag{17}$$

$$\dot{z} = -V s_\theta + w c_\theta \tag{18}$$

$$\dot{\theta} = q \tag{19}$$

$$\dot{\psi} = \frac{r}{c_\theta} \tag{20}$$

$$\dot{u} = \frac{F_x}{\bar{m}} + rv - qw - g s_\theta \tag{21}$$

$$\dot{v} = \frac{F_y}{\bar{m}} - rV \tag{22}$$

$$\dot{w} = \frac{F_z}{\bar{m}} + qV + g c_\theta \tag{23}$$

$$\dot{p} = \frac{M_x}{I_{xx}} \tag{24}$$

$$\dot{q} = \frac{M_y}{I_{yy}} - rp\left(\frac{I_{XX}}{I_{YY}}\right) \tag{25}$$

$$\dot{r} = \frac{M_z}{I_{YY}} + pq\left(\frac{I_{XX}}{I_{YY}}\right) \tag{26}$$

We propose the two sets of force and moment models. For spin-stabilized rounds, aerodynamic symmetry is assumed with an out-of-plane moment due to the Magnus effect.

$$F_x = -\bar{q}SC_{X_0}$$

$$F_y = -\bar{q}S\left(C_{N_\alpha}\frac{v}{V} + C_{N_{\alpha 3}}\frac{v^3}{V^3}\right)$$

$$F_z = -\bar{q}S\left(C_{N_\alpha}\frac{w}{V} + C_{N_{\alpha 3}}\frac{w^3}{V^3}\right)$$

$$M_x = D\bar{q}S\left(C_{l_0} + DC_{l_p}\frac{p}{2V}\right)$$

$$M_y = \bar{q}SD\left(C_{m_\alpha}\frac{w}{V} + C_{m_{\alpha 3}}\frac{w^3}{V^3} + DC_{m_q}\frac{q}{2V} + DC_{n_{p\alpha}}\frac{pv}{V^2}\right)$$

$$M_z = \bar{q}SD\left(-C_{m_\alpha}\frac{v}{V} - C_{m_{\alpha 3}}\frac{v^3}{V^3} + DC_{m_q}\frac{r}{2V} + DC_{n_{p\alpha}}\frac{pw}{V^2}\right)$$

For fin-stabilized projectiles, transverse force and moment terms are somewhat different. Out-of-plane force and moment terms are included to account for slight, roll-dependent asymmetries, and Magnus is ignored.

$$F_y = \bar{q}S\left(-C_{N_\alpha}\frac{v}{V} - C_{N_{\alpha 3}}\frac{v^3}{V^3} + C_{Y_{\phi\alpha}}s_{\bar{\alpha}}^2\frac{w}{V}\sin N\phi_A + C_{Y_\alpha}\frac{w}{V}\right)$$

$$F_z = -\bar{q}S\left(C_{N_\alpha}\frac{v}{V}s_\alpha + C_{N_{\alpha 3}}\frac{v^3}{V^3} + C_{Y_{\phi\alpha}}s_{\bar{\alpha}}^2\frac{v}{V}\sin N\phi_A + C_{Y_\alpha}\frac{v}{V}\right)$$

$$M_y = \bar{q}SD\left(C_{m_\alpha}\frac{w}{V} + C_{m_{\alpha 3}}\frac{w^3}{V^3} + C_{n_{\phi\alpha}}s_{\bar{\alpha}}^2\frac{v}{V}\sin N\phi_A + C_{n_\alpha}\frac{v}{V} + \frac{qD}{2V}C_{m_q}\right)$$

$$M_z = \bar{q}SD\left(-C_{m_\alpha}\frac{v}{V} - C_{m_{\alpha 3}}\frac{v^3}{V^3} + C_{n_{\phi\alpha}}s_{\bar{\alpha}}^2\frac{w}{V}\sin N\phi_A + C_{n_\alpha}\frac{w}{V} + \frac{rD}{2V}C_{m_q}\right)$$

where $V = \sqrt{u^2 + v^2 + w^2}$, $s_{\bar{\alpha}}$=$\sqrt{v^2 + w^2}/V$, and $\phi_A = \tan^{-1}\left(\frac{v}{w}\right)$

Given the models in Eqs. 5–26, we propose using collocation instead of numerical integration to rapidly solve them. In Section 3.1, we describe the transcription and solution of these models. In Section 3.2, we describe how we can build on this technique to determine sensitivities with respect to model parameters. In Section 3.4, we combine these methods to refine model parameters for best fit to the data.

3.1 Collocation Solution of ODE Models

After collocation, the downrange-velocity and roll-spin models are

$$\frac{2}{t_f}\boldsymbol{\Delta}x_N - V_N = 0 = g_1 \tag{27}$$

$$\frac{2}{t_f}\boldsymbol{\Delta}V_N + \frac{\pi\rho D^2}{8\bar{m}}C_{X_0}V_N^2 = 0 = g_2 \tag{28}$$

$$\frac{2}{t_f}\boldsymbol{\Delta}\phi_N - p_N = 0 = g_3 \tag{29}$$

$$\frac{2}{t_f}\boldsymbol{\Delta}p_N + \frac{\pi\rho V_N^2 D^3}{8I_{XX}}C_{L_0} - \frac{\pi\rho D^4 V}{16I_{XX}}p_N C_{L_p} = 0 = g_4 \tag{30}$$

After collocation, the 8-state linear epicyclic model may be written and solved in matrix form as $\mathbf{ZX} = \mathbf{Y}$, where

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

$$\mathbf{Y} = [\mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \boldsymbol{g} \quad \mathbf{0} \quad \mathbf{0}]^{\boldsymbol{T}}$$

The linear formulation has the advantage that one may solve for the predicted trajectory simply by matrix inversion—no iteration is required and no initial guess of the trajectory is required. The linear solution does need to be iterated using the Marquardt algorithm, as described in Section 3.4, to estimate parameters and initial conditions. Also, the matrix equation needs to be augmented with rows that enforce initial conditions as are discussed in Section 3.3 on initial conditions and multiple segment problems.

After collocation, the nonlinear 6DOF pitch/yaw and swerve model is

$$\frac{2}{t_f}\boldsymbol{\Delta}x_N - V_N\, c_{\theta_N}\, c_{\psi_N} + v_N s_{\psi_N} - w_N s_{\theta_N} c_{\psi_N} = f_1 = 0 \tag{31}$$

$$\frac{2}{t_f}\boldsymbol{\Delta}y_N - V_N\, c_{\theta_N}\, s_{\psi_N} - v_N c_{\psi_N} - w_N s_{\theta_N} s_{\psi_N} = f_2 = 0 \tag{32}$$

$$\frac{2}{t_f}\boldsymbol{\Delta}z_N + s_{\theta_N}\, V_N - w_N c_{\theta_N} = f_3 = 0 \tag{33}$$

$$\frac{2}{t_f}\boldsymbol{\Delta}\phi_N - p_N - r_N t_{\theta_N} = f_4 = 0 \tag{34}$$

$$\frac{2}{t_f}\boldsymbol{\Delta}\theta_N - q_N = f_5 = 0 \tag{35}$$

$$\frac{2}{t_f}\boldsymbol{\Delta}\psi_N - r_N\frac{c_{\phi_N}}{c_{\theta_N}} = f_6 = 0 \tag{36}$$

$$\frac{2}{t_f}\boldsymbol{\Delta}u_N - \frac{F_x}{\bar{m}} - r_N v_N + q_N w_N + g s_{\theta_N} = f_7 = 0 \tag{37}$$

$$\frac{2}{t_f}\boldsymbol{\Delta}v_N - \frac{F_y}{\bar{m}} - p_N w_N + r_N V_N = f_8 = 0 \tag{38}$$

$$\frac{2}{t_f}\boldsymbol{\Delta}w_N - \frac{F_z}{\bar{m}} - q_N V_N + p_N v_N - g c_{\theta_N} = f_9 = 0 \tag{39}$$

$$\frac{2}{t_f}\boldsymbol{\Delta}p_N - \frac{M_x}{I_{XX}} = f_{10} = 0 \tag{40}$$

$$\frac{2}{t_f}\boldsymbol{\Delta}q_N - \frac{M_y}{I_{YY}} + r_N p_N\left(\frac{I_{XX}}{I_{YY}}\right) = f_{11} = 0 \tag{41}$$

$$\frac{2}{t_f}\boldsymbol{\Delta}r_N - \frac{M_z}{I_{YY}} - p_N q_N\left(\frac{I_{XX}}{I_{YY}}\right) = f_{12} = 0 \tag{42}$$

Given a set of parameters $\left(C_{X_0}, C_{L_0}, C_{L_p}\right)$, and initial conditions (x_0, V_0, ϕ_0, p_0), Eqs. 27–30 are a nonlinear algebraic set that may be solved by the Newton–Raphson method.

$$\mathbf{X}^{n+1} = \mathbf{X}^n - \mathbf{J}^{-1}\mathbf{f}$$

where

$$\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{X}}\Big|_{\vartheta}, \mathbf{f} = [f_1 \quad f_2 \quad \cdots]^T|_{\mathbf{X}=\mathbf{X}^n}, \ \mathbf{X} = [x_N \quad V_N \quad \cdots]^T$$

Eqs. 31–42 can be solved using the same method, simply redefining the matrix-vector quantities to match. The Jacobian matrix for Eqs. 31–42 is expanded in the Appendix.

3.2 A Key Insight: Paramic Equations

Given a set of nonlinear equations, such as 17–26, we can write compactly

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{x_0}, \vartheta) \tag{43}$$

where $\mathbf{x}$ is the vector of states, t is time, and $\boldsymbol{\vartheta}$ is a vector of unknown parameters. Since the launch conditions of a projectile are unknown, we must include the initial conditions, $\mathbf{x_0}$, in the set of unknown parameters. To avoid additional trajectory predictions and finite differencing, the sensitivity of the solution with respect to the initial conditions can be found through a set of auxiliary equations. In fact, this technique can be applied to the unknown parameters as well. Adopting a term from Bradley,[6] we call the full set of unknowns $(\mathbf{x_0}, \boldsymbol{\vartheta})$ "paramics". Taking partial derivatives of Eq. 43 with respect to the paramics,

$$\frac{\partial}{\partial \boldsymbol{\vartheta}}\left(\frac{d}{dt}\mathbf{x}\right) = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\vartheta}} \tag{44}$$

and reversing the order of differentiation we get

$$\frac{d}{dt}\left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\vartheta}}\right) = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\vartheta}} \tag{45}$$

such that when the number of parameters in $\boldsymbol{\vartheta}$ is less than the length of the state vector, Eq. 45 is a rectangular array of ODEs whose solution is the sensitivity of the trajectory to the paramics. If we substitute $\mathbf{x_0}$ for $\boldsymbol{\vartheta}$ in Eq. 45, we have a rectangular matrix of equations for the sensitivity of the state with respect to the initial states. Bradley[6] astutely observed that Eq. 45 always results in a *linear* set of matrix odes. Specifically, if we define

$$\mathbf{P} = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\vartheta}}, \qquad \mathbf{R} = \frac{\partial \mathbf{x}}{\partial \mathbf{x_0}} \tag{46}$$

then

$$\frac{d\mathbf{P}}{dt} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\vartheta} \mathbf{P} + \left.\frac{\partial \mathbf{f}}{\partial \boldsymbol{\vartheta}}\right|_{\mathbf{x}} \tag{47}$$

and

$$\frac{d\mathbf{R}}{dt} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\vartheta} \mathbf{R} \tag{48}$$

Since these equations are already linear, albeit matrix-valued, their Gauss transcriptions are linear, algebraic equations:

$$\frac{2}{t_f}\boldsymbol{\Delta}\mathbf{P} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\vartheta} \mathbf{P} + \left.\frac{\partial \mathbf{f}}{\partial \boldsymbol{\vartheta}}\right|_{\mathbf{x}} \tag{49}$$

which is easily solved as

$$\mathbf{P} = \left(\frac{2}{t_f}\boldsymbol{\Delta} - \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\vartheta}\right)^{-1} \left.\frac{\partial \mathbf{f}}{\partial \boldsymbol{\vartheta}}\right|_{\mathbf{x}} \tag{50}$$

with the initial conditions

$$\mathbf{P}(0) = \mathbf{0}$$

and

$$\frac{2}{t_f}\boldsymbol{\Delta}\mathbf{R} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\vartheta} \mathbf{R} \tag{51}$$

which is solved as

$$\left(\frac{2}{t_f}\boldsymbol{\Delta} - \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\vartheta}\right)^{-1} \mathbf{R} = \mathbf{0} \tag{52}$$

However, since

$$\mathbf{R}_{ml}(0) = \begin{cases} 1 & \text{if } m = l \\ 0 & \text{otherwise} \end{cases} \tag{53}$$

Equation 52 is a free response from a non-zero "initial" condition. The conditions of Eq. 53 must be fully enforced by inserting rows in Eq. 52 that constrain the initial value of *every* state. Since $\mathbf{R}$ is actually a matrix of unknowns, Eq. 52 can then be solved a column at a time as

$$\mathbf{R}_{:,l} = \left(\boldsymbol{\Xi} - \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\vartheta}\right)^{-1} \mathbf{e}_l \tag{54}$$

where $\mathbf{e}_l$ is the unit vector corresponding to state l as described in Eq. 53 and

$$\boldsymbol{\Xi} = \frac{2}{t_f}\begin{bmatrix} \boldsymbol{\Delta} & \mathbf{0} & \cdots \\ \mathbf{0} & \boldsymbol{\Delta} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

Note the inverted matrix in Eq. 54 is independent of l, so it needs to be constructed and inverted only once. Furthermore, note the Jacobian $\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\vartheta}$ is the same one used in the Newton solution to the collocation of the nonlinear system ODEs (Section 3.1). Thus, the exact sensitivities of the collocation solution to the system initial states are found with very little additional effort once the solution itself has been found. Moreover, the matrix to be inverted in Eq. 50 is the same as that in Eq. 54, and the derivatives $\left.\frac{\partial \mathbf{f}}{\partial \boldsymbol{\vartheta}}\right|_{\mathbf{x}}$ with respect to constant parameters will be sparse and relatively simple. Thus, these sensitivities are also found with little additional effort. Expressions for $\left.\frac{\partial \mathbf{f}}{\partial \boldsymbol{\vartheta}}\right|_{\mathbf{x}}$ for the most influential aerodynamic coefficients are shown in the Appendix. Note that due to the simplicity of these expressions, additional aerodynamic coefficients can be easily added to the model.

3.3 Multiphase Gauss Collocation

As previously noted, and shown in Fig. 1, the spark range stations do not correlate at all to collocation points. Thus, we have two options: 1) interpolate the data onto the collocation points and fit the numerical solution to the interpolated data, or 2) use the actual spark stations as limits of the collocation domain such that each interstitial space is a separate collocation "phase", and the phases are tied together by constraining adjoining state vectors to be equal at the stations. We demonstrate option 1 later using contrived data. For actual data, a few stations often provide no data or unusable data. Due to these dropouts, interpolation onto a single collocation grid caused significant deviation from the actual trajectory. Thus, the final algorithm uses option 2, which we illustrate here.

When reducing actual data, the trajectory is divided into phases where each phase begins and ends at two adjacent spark stations. Between stations, the pitch–yaw–swerve model is described by Eqs. 31–42. A set of residual equations and a Jacobian matrix are formed for each phase. For the first phase, the initial value of each state is determined by constraining it to the initial conditions. These constraints replace the first row of each state block in the residuals and Jacobian. For instance, for state y, we might have the initial condition constraint

$$y|_{\mathbf{1}} = y_0$$

Row 1 of the y block of the first phase of the Jacobian will contain a single one for

$$\frac{\partial y|_{\mathbf{1}}}{\partial y_0} = 1$$

For each subsequent phase, the initial value of each state is constrained to match its final value in the previous phase, in the case of state y

$$y|_m - y|_{m+1} = 0$$

Thus, row 1 of the y block of the second phase of the Jacobian will contain a +1 and a –1 corresponding to

$$\frac{\partial (y|_m - y|_{m+1})}{\partial y|_m} = 1 \quad \text{and} \quad \frac{\partial (y|_m - y|_{m+1})}{\partial y|_{m+1}} = -1$$

Figure 2 illustrates the Jacobian sparsity pattern for a 6-state model with two phases.

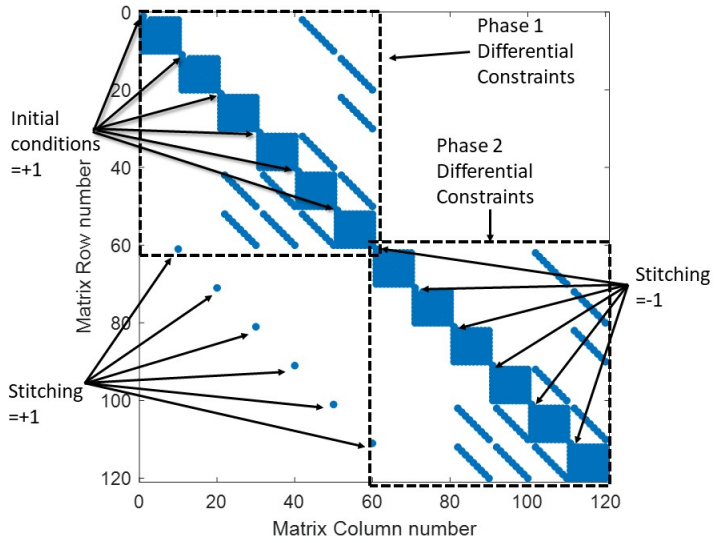


Fig. 2 Sparsity of the Jacobian matrix for a two-phase collocation with 6 states and m = 10

The upper left block contains the differential constraints or governing equations for the trajectory from initial to phase interface. The large blue blocks contain the $\mathbf{\Delta}$ matrix and any terms that multiply that particular state variable. For instance, the top section of Phase 1 Differential Constraints could represent $\frac{2}{t_f}\mathbf{\Delta}x_N - V_N = 0 = g_1$, where the blue block is the $\mathbf{\Delta}$ matrix ($=\partial g_1/\partial x_N$) and the diagonal section a few blocks to the right would represent $\frac{\partial g_1}{\partial V_N} = \mathbf{I}$. The bottom right block contains the differential constraints from phase interface to final condition. The first row of each differential constraint in the first phase is replaced with a 1 to provide the sensitivity of the solution with respect to the initial condition. The first row of each state block in the second phase is replaced with a row that has a +1 in the mth column of that state's domain (m = 10, 20, …), and a –1 in the "m+1" columns, (where "m+1" = 61, 81, 91,…).

3.4 Marquardt Search Algorithm

Hathaway, Whyte, Bradley, and others[4–6] recommend the use of a true Marquardt nonlinear search to provide the fastest convergence of parameter estimates. In this work, we use a two-step process with the second step consisting of a Marquardt iteration for the best model parameters and initial conditions. The full algorithm repeats this two-step process four times to gradually increase the problem scope in order to help focus the algorithm on highly correlated parameters and measurements.

The very first step is to design the set of collocation points based upon the measured times in the data set. For the contrived data sets, the total time is broken into q segments $q \in \{4, 6, 12, 24\}$ in the trade studies that follow. For the actual data, the number of segments is exactly one less than the number of usable spark range stations ($q \leq 24$). Each segment is mapped from time domain $t_i \leq t \leq t_F$ to Legendre domain $-1 < t < 1$. Within each segment, m collocation points are selected from the LGL set. Next, the cumulative time is reconstructed by mapping the times of each segment back to time domain and laying the segments end-to-end. This complete time vector of length qm is then used to interpolate all of the spark range data. In every case, the segment endpoints lie exactly on the spark range stations. All spark range stations lie on segment endpoints only in the $q = 24$ case. We must use the interpolated points in the algorithm for good conditioning of the search. The actual spark range data are included as end points of each segment. This causes prediction errors at the actual stations to be double penalized during model refinement. The final statistics shown for actual data cases ignore the interpolated points.

The final algorithm applied to actual data uses a series of five models of increasing complexity to fit subsets of the data in the following sequence:

1) Fit the model to measured time and downrange distance, adjusting initial velocity, and drag.

2) Estimate the initial roll rate, and roll rate profile using Eq. 8, and catalog values for roll coefficients. If roll data is available, fit to Eqs. 7 and 8 by adjusting initial conditions, and roll coefficients.

3) Fit the linear epicyclic model to the first full oscillation of motion (typically 6–10 stations), using pitch/yaw measurements only

4) Fit the full linear epicyclic model to the full trajectory, using measurements of crossrange, altitude, pitch, and yaw.

5) Fit the full nonlinear model to the trajectory, using Eqs. 31–42

The first step (drag fitting) is done using an established approach. If roll data is available, the second step uses multi-phase Gauss collocation. Each of the last three steps uses the multi-phase Gauss collocation algorithm described in this report. Each of these fits requires the following iteration:

1) Using the current estimates of parameters and initial conditions, solve the discretized system equations using Newton–Raphson until the 2-norm of the prediction vector **f** falls below 10^{-9}.

2) Using the solution from the previous step, use Eqs. 50 and 52 to find columns of the parameter search Jacobian $\mathbf{J}_S = [\mathbf{R} \quad \mathbf{P}]$.

3) Compute the difference between the predicted solution and the spark range data, $\bar{\boldsymbol{\rho}} = \mathbf{C}\mathbf{X}^{n+1} - \mathbf{y}$, where $\mathbf{C}$ is a sparse matrix used to pull out and scale the predictions of measured quantities—pitch, yaw, crossrange, and altitude.

4) Use the Marquardt formula to improve estimates of parameters and initial conditions

$$\boldsymbol{\vartheta}^{n+1} = \boldsymbol{\vartheta}^{n} - (\mathbf{J}_S^T\mathbf{J}_S + \boldsymbol{\mu}\mathbf{I})^{-1}\mathbf{J}_S^T\bar{\boldsymbol{\rho}}$$

5) Repeat the prediction step, and check for reduction of $\|\bar{\boldsymbol{\rho}}\|_2$.

   - If $\|\bar{\boldsymbol{\rho}}\|_2$ is reduced, accept $\boldsymbol{\vartheta}^{n+1}$, set $\boldsymbol{\mu} = \boldsymbol{\mu}/\mathbf{10}$, and return to the first step.
   - If $\|\bar{\boldsymbol{\rho}}\|_2$ is not reduced, set $\boldsymbol{\mu} = \boldsymbol{\mu} \cdot \mathbf{10}$ and repeat the parameter update up to five times or until $\|\bar{\boldsymbol{\rho}}\|_2$ is reduced.

The algorithm used to process simulated data differs in one significant way. Since roll data is available, the drag and roll parameters are found simultaneously, using the Gauss collocation approach with Eqs. 5–8.

4. Results

During its development, the algorithm was tested on one set of virtual data and three sets of actual data. We present results for one of each type.

4.1 Dense, Virtual Data

The proposed algorithm was exercised on a shot of the Laboratory Technology Vehicle simulated using the ARL6DOF Simulink model. This particular model has highly nonlinear, roll-dependent aerodynamics, with the linear terms matching Eqs. 9–16 identically. This preliminary case will have the advantage of providing data from all 25 stations of the Transonic Experimental Facility at ARL. Specific stations can be ignored later to extend the capability of this algorithm. The "data" can be modified with user-specified noise or left alone to demonstrate the best possible performance.

To determine the best possible performance, and tradeoffs between model-fitting error, parameter uncertainty, and computational cost, we performed a trade study that varied the number of collocation points and segments. The cost is computed in each case as the 2-norm of the final residual divided by the total number of

collocation points. This cost definition gives a qualitative assessment of performance, since for all collocation schemes, only the points between segments correspond to actual spark range data. The convergence of collocation points between stations to the *interpolated* data is included in the cost, putting schemes with more interpolated points at an advantage. Schemes with less than 24 segments do not include all of the actual data, but instead replace it with interpolated data. Table 1 shows the normalized cost as a function of number of segments, and number of collocation points.

Table 1 Segments trade study

m **points per seg.**	**4 segments**	**6 segments**	**12 segments**	**24 segments**
5	0.062215	0.05845	0.020065	0.0046641
7	0.059796	0.011457	0.0022827	0.0029719
9	0.018068	0.0064594	0.0033488	0.0023115
11	0.003908	0.0039179	0.0030742	0.0021072
13	0.0043935	0.0040792	0.0027461	0.0019244

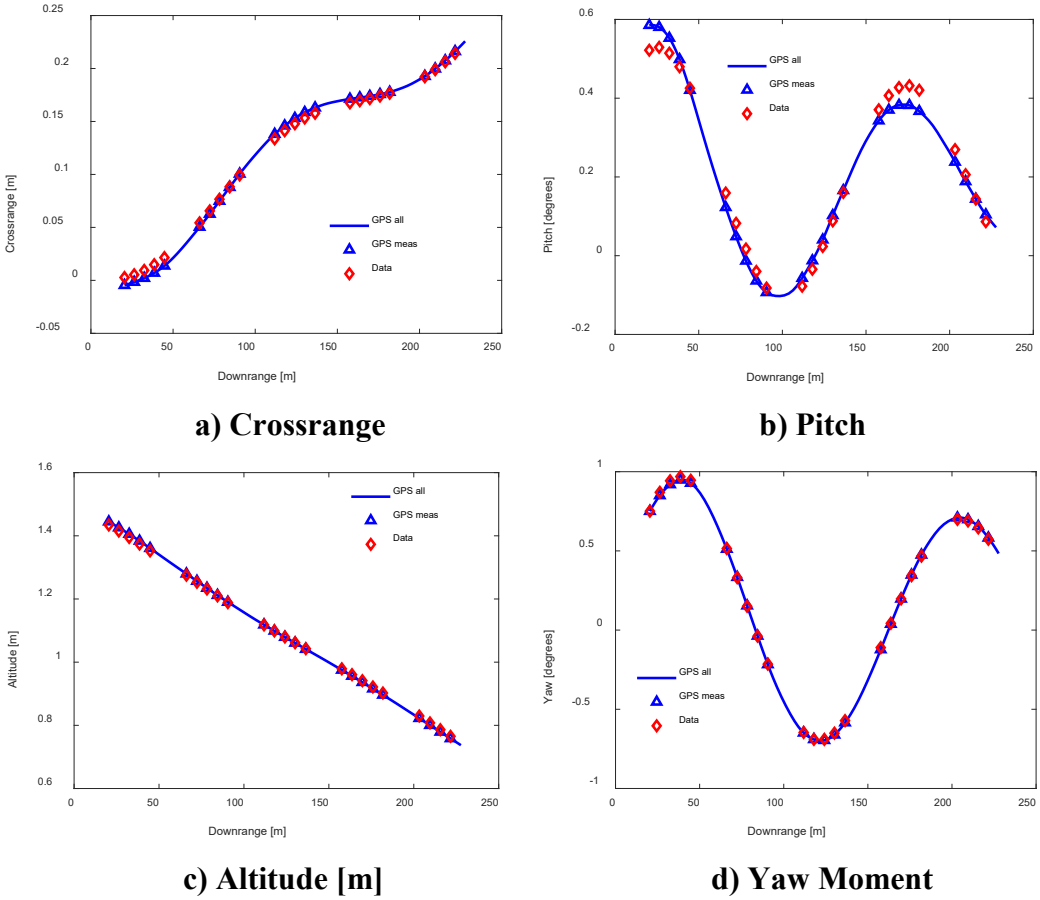
Since only the 24-segment case accounts for all of the actual data, it was selected for further study. A second case study was done varying the number of collocation points per segment and computing the final cost as the residuals at the actual stations only. Table 2 shows the root mean square (rms) error for each of the four measurements (pitch, yaw, crossrange, and altitude) as functions of the number of collocation points per segment. The final algorithm should be capable of reducing the estimation error to the standard deviation of the range measurement noise. For the ARL Transonic Range, that is 3.7 mm in position and 0.15° in pitch and yaw. The cases with 9 or more collocation points achieve both of these metrics.

Table 2 Segments trade study: holding number of segments constant at 24. Errors are calculated from points where the spark range data exists.

m	**rms Y, m**	**rms Z, m**	**rms θ, deg**	**rms ψ, deg**
5	0.0053335	0.0026768	0.0065365	0.0050804
7	0.0015565	0.0016222	0.0073105	0.0024513
9	0.0014062	0.0013019	0.0065356	0.0023014
11	0.0011853	0.0012849	0.0066027	0.002246
13	0.00089752	0.0012438	0.0065954	0.0021405

Thus, for fully dense data, we recommend using nine segments for fastest convergence, and more than nine for best accuracy.

In Fig. 3, a sample prediction is shown after the search has converged. For this case, 24 segments are used with 13 collocation points per segment. The rms errors for this case are shown in the bottom row of Table 2.



a) Crossrange **b) Pitch**

c) Altitude [m] **d) Yaw Moment**

Fig. 3 Gauss pseudospectral predictions compared with data for m = 13 collocation points per segment. Solid blue line is the entire trajectory prediction with 24 segments for collocation.

4.2 Actual M898 Data

Legacy spark range data from 14 shots of the M898[10] 155-mm projectile were used to test the algorithm described here. The final fitting statistics for this data set are shown in Table 4. Shots are sorted from smallest to largest muzzle velocity. Note that all the errors shown are less than two standard deviations of the measurement noise (σ = 3.3 mm for position and 0.15° for angles), and about half of the errors are less than one sigma.

Table 3 Nonlinear fitting statistics for actual data

Transonic Range Number (TRN)		$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x-\hat{x})^2}$			
	V_0 [m/s]	rms y [m]	rms z [m]	rms θ [deg]	rms ψ [deg]
30834	246.66	0.004697	0.003837	0.078736	0.101
30833	247.09	0.007285	0.008642	0.28646	0.24591
30831	290.41	0.002083	0.002496	0.085046	0.084406
30829	297.65	0.004245	0.004683	0.083428	0.12266
30835	332.87	0.03775	0.028724	0.29893	0.31735
30841	334.24	0.002448	0.004265	0.066561	0.1112
30820	351.04	0.001853	0.002658	0.17943	0.14349
30827	351.04	0.002081	0.002776	0.14309	0.12501
30814	448.59	0.005383	0.003621	0.091724	0.1058
30812	451.57	0.0024	0.0055	0.0992	0.0923
30818	536.16	0.004408	0.004259	0.070603	0.080512
30815	536.89	0.0054	0.0036	0.0917	0.1058
30825	780.6	0.002517	0.003686	0.21599	0.3004
30824	786.51	0.005366	0.005628	0.21755	0.23785

Figure 4 illustrates the quality of final fit for shot TRN 30812. Note the axis scaling, especially in crossrange. This particular shot has limited motion, which makes the fitting process difficult.

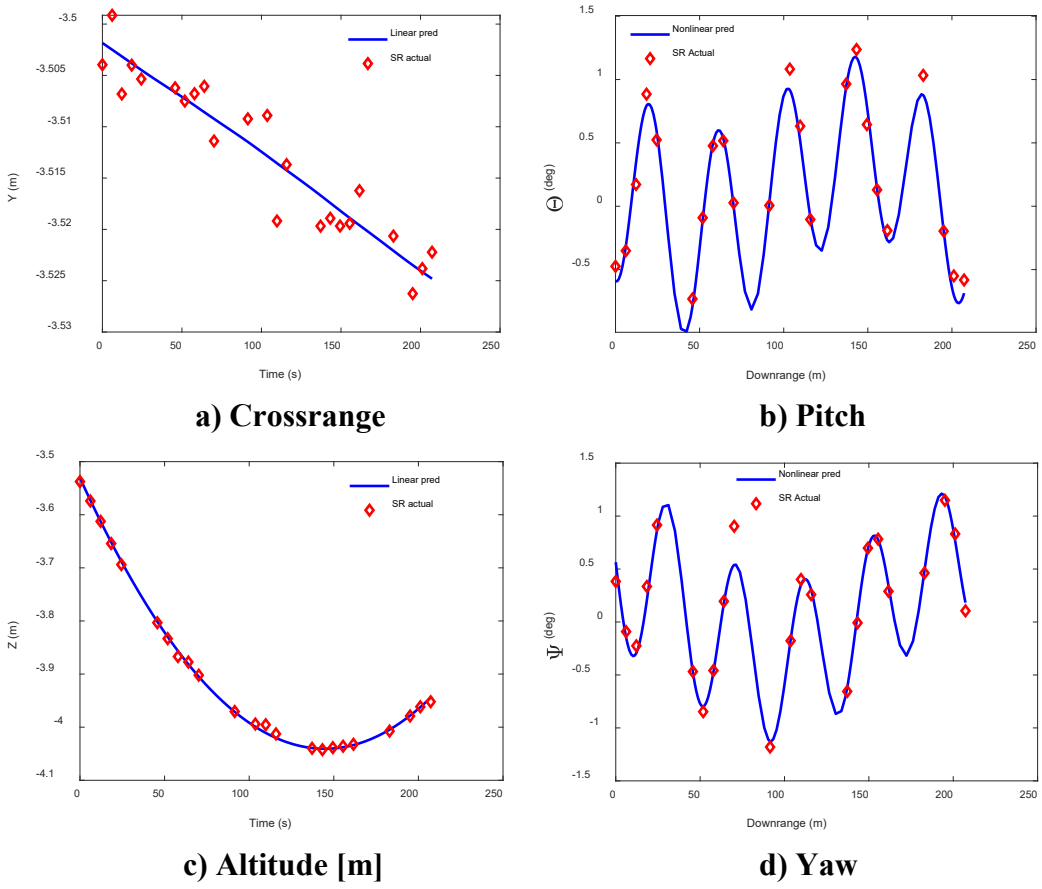


Fig. 4 Gauss pseudospectral predictions compared with data for m = 9 collocation points per segment. M898 Shot TRN 30812, nonlinear model. Solid blue line is the entire trajectory prediction with 23 segments—1 per spark range station.

The most sparse set of data, TRN 30835, containing just 21 valid stations, brought to our attention another disadvantage of the method. Notice the poor apparent quality of fit in Fig. 5. Since we have plotted only the actual spark range data for crossrange and altitude, we can see from Fig. 5a and 5c that only two of the stations in Group 3 (around 100 m downrange) captured any data. Thus, the sparse group of interpolated points in Fig. 5b and 5c just prior to 100 m downrange are the result of these missing data, and these points are not part of the actual state history. However, in building the residual and Jacobian, we include these points for good conditioning of the problem.

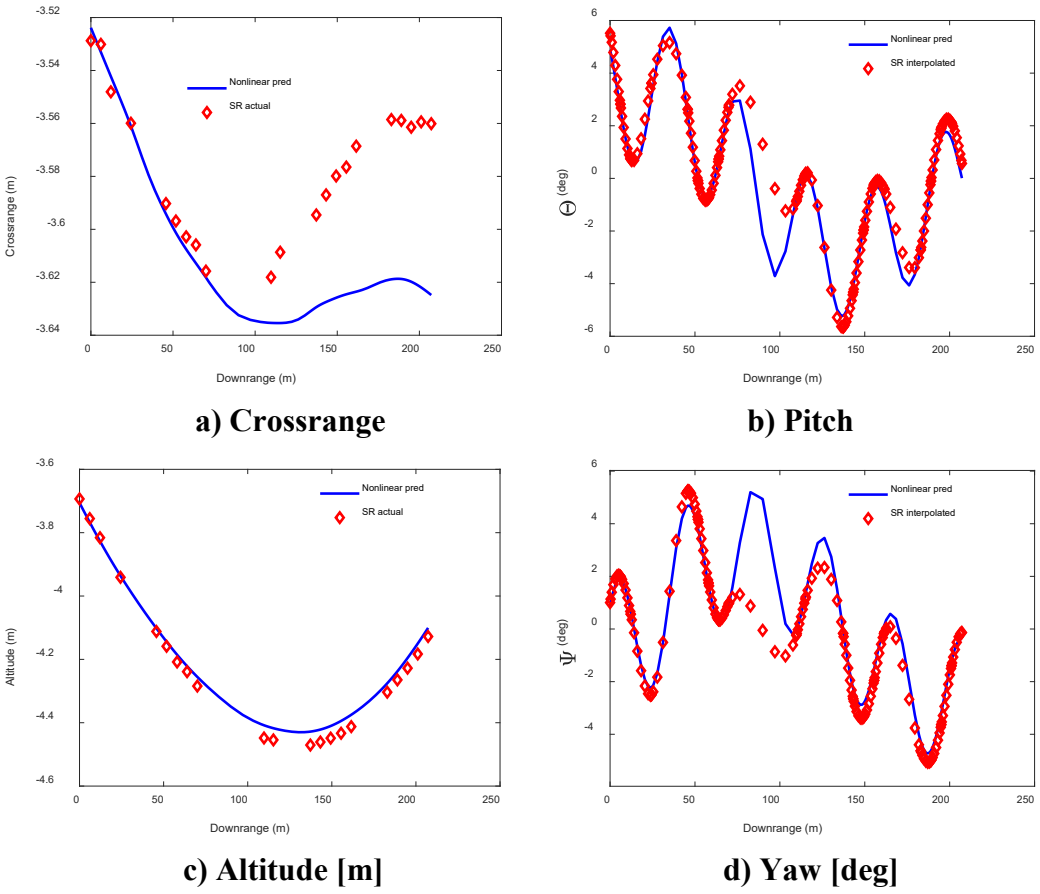
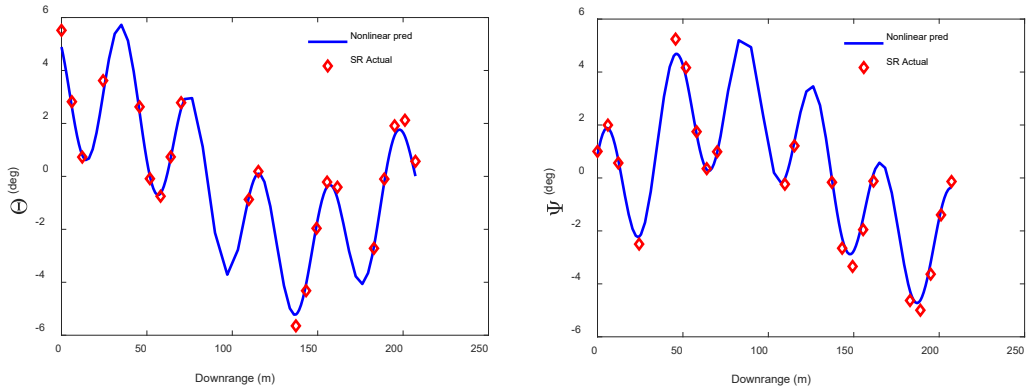


Fig. 5 Gauss pseudospectral predictions compared with data for m = 9 collocation points per segment. M898 Shot TRN 30835, linear model. Solid blue line is the entire trajectory prediction with 23 segments—1 per spark range station.

Figure 6 is updated to include only the predictions that correspond to actual spark range data. Notice the absence of spark range data in the vicinity of 100 m downrange. The pitch and yaw predictions are very close to all of the actual spark range data. The statistics in Table 4 are based only on where the trajectory intersects the actual measurements, not the interpolated values. Comparing Fig. 6 with Fig. 5, it should be apparent that the collocation algorithm is justified in ignoring the interpolated points shown in Fig. 5b and 5d. In fact, during the search, the actual spark range points are given double the weight of interpolated points since they are shared between segments. The interpolated data do not fit the differential constraints defined by the equations of motion, so the algorithm ignores them. Apparently, this transparent feature of the algorithm is enough to keep it from chasing interpolated points.



a) Pitch [deg] **b) Yaw [deg]**

Fig. 6 Gauss pseudospectral predictions compared with data for m = 9 collocation points per segment. M898 Shot TRN 30835, nonlinear model. Solid blue line is the entire trajectory prediction with 23 segments—1 per spark range station.

Table 4 shows the linear pitch/yaw aerodynamic coefficients found from the M898 data. The normal force and pitch moment coefficients are most reliable and consistent, which is typical for this type of test. Pitch damping shows a large amount of scatter, which is also typical in spark range tests due to the limited number of oscillations observed in each shot. These estimates were all found from 15 iterations of the algorithm without any user intervention to downselect specific parameters, and so on.

Table 4 Identified coefficients for actual data

TRN	Mach	C_{N_α}	C_{M_α}	C_{M_q}	$C_{n_{p\alpha}}$	C_{X_0}
30834	0.7255	2.017	4.573	–11.296	0.0130	0.1506
30833	0.7267	2.071	4.594	–5.717	0.0122	0.1732
30831	0.8541	2.050	4.856	–6.512	–0.0373	0.1454
30829	0.8754	2.083	4.908	1.056	–0.0307	0.1590
30835	0.9790	2.105	4.172	–7.165	–0.0421	0.2939
30841	0.9831	2.358	4.369	–14.942	–0.0160	0.2635
30820	1.0325	2.406	4.922	–17.186	–0.0122	0.3764
30827	1.0325	2.406	4.705	–13.802	–0.0153	0.3855
30814	1.3194	2.608	4.808	–14.249	0.00634	0.3652
30812	1.3281	1.043	5.132	–21.133	0.0306	0.3543
30818	1.5769	9.360	5.602	34.635	–0.300	0.3376
30815	1.5791	2.831	4.919	–80.298	0.206	0.3221
30825	2.2959	3.036	4.407	–31.202	0.0251	0.2907
30824	2.3133	3.250	4.328	–12.659	0.00756	0.3046

Shot 30818 has very small pitch/yaw motion as shown in Fig. A-3. This results in a high degree of uncertainty in $\boldsymbol{C_{N_\alpha}}$ and $\boldsymbol{C_{M_\alpha}}$, resulting in a $\boldsymbol{C_{N_\alpha}}$ of 9.360, which is clearly erroneous and outside of any trends perceived in the other data. Shot 30812 shown in Fig. 4, also has very small pitch / yaw amplitude, causing large uncertainty in the estimates. Since these two are shown on adjacent rows in the table, $\boldsymbol{C_{N_\alpha}}$ estimates may appear erratic. However, once these two are ignored, note the consistent trend in $\boldsymbol{C_{N_\alpha}}$ elsewhere.

We plotted the remaining shots in the Appendix for brevity here. Note the quality of fit as reflected in Table 4.

5. Conclusion

A new method for reduction of spark range data has been developed and demonstrated. The method uses Gauss collocation rather than numerical integration for trajectory prediction. By realizing that the trajectory sensitivities to changes in parameters and initial conditions are captured by linear ODEs, we have developed a novel method to compute the sensitivities by solving two sets of linear algebraic equations.

The method performs well without any user intervention in downselecting the parameters to be estimated. As of this writing, the code has been tested on virtual data of a fin-stabilized projectile, and actual data of three spin-stabilized projectiles. These examples demonstrated final estimation errors less than twice the standard deviation of the measurement noise in all cases while finding nine of the most influential aerodynamic parameters. Additional aerodynamic parameters can easily be added to the code because the programmer only needs to determine the corresponding values of $\left.\frac{\partial \mathbf{f}}{\partial \boldsymbol{\vartheta}}\right|_{\mathbf{x}}$ rather than working out the entire chain rule of differentiation from output back to parameter.

The method has one significant disadvantage. Since the amount of data from the spark range is notably sparse compared with the roll rate of a typical projectile, it is doubtful that the present algorithm could facilitate the use of body-fixed dynamic models. That is, the number of collocation points possible between spark stations is limited both by 1) the ill-conditioning of underlying high-order polynomials, and 2) the numerical difficulty of inverting a large, somewhat sparse array. By using only fixed-plane models, the roll angle is ignored and these difficulties are avoided.

6. References

1. Murphy CH. Data reduction for the free flight spark ranges. Ballistic Research Laboratory (US); 1954. Report No.: 900.

2. McCoy RL. Modern exterior ballistics: the launch and flight dynamics of symmetric projectiles. Schiffer Military History; 1999.

3. Chapman GT, Kirk DB. A method for extracting aerodynamic coefficients from free-flight data. AIAA Journal. 1970;8(4):753–758.

4. Whyte RH, Jeung A, Bradley W. Chapman-Kirk reduction of free-flight range data to obtain nonlinear aerodynamic coefficients. Ballistic Research Laboratory (US); 1973 May. Report No.: 2298.

5. Hathaway W, Whyte R. Aeroballistic research facility free flight data analysis using the maximum likelihood method. Air Force Armament Laboratory (US); 1979 Dec. Report No.: AFATL-79-98.

6. Bradley JW. FINLIE: A Fortran program for fitting ordinary differential equations with nonlinear parameters to data. Ballistic Research Laboratory (US); 1981 Feb. Report No.: ARBRL-TR-02290. https://apps.dtic.mil/sti/pdfs/ADA098038.pdf.

7. Elnagar G, Kazemi MA, Razzaghi M. The pseudospectral Legendre method for discretizing optimal control problems. IEEE Transactions on Automatic Control. 1995 Oct;40(10):1793–1796,

8. Rao AV, Benson DA, Darby C, Patterson MA, Francolin C, Sanders I, Huntington GT. Algorithm 902: GPOPS, a MATLAB software for solving multiple–phase optimal control problems using the Gauss pseudospectral method. ACM Transactions on Mathematical Software. 2010 Apr;38(2):Article 22.

9. Ross IM. User's manual for DIDO: a MATLAB application package for solving optimal control problems. Tomlab Optimization Inc; 2004 Feb. Report No.: 04-01.0.

10. Soencksen, KP, Oskay, V, An Investigation into the aerodynamics and structural integrity of the 155-mm M898 projectile. Army Research Laboratory (US); 1999 Nov. Report No.: ARL-MR-461.

Appendix. Sensitivities and Additional Results

Here we expand the non-zero terms in the Jacobian for Eqs. 31–42 in the main report.

$$\frac{\partial f_1}{\partial x} = \frac{2}{t_f}\boldsymbol{\Delta} \qquad \frac{\partial f_1}{\partial \theta} = u_N s_{\theta_N} c_{\psi_N} - w_N c_{\theta_N} c_{\psi_N}$$

$$\frac{\partial f_1}{\partial \psi} = u_N c_{\theta_N} s_{\psi_N} + v_N c_{\psi_N} + w_N s_{\theta_N} s_{\psi_N} \qquad \frac{\partial f_1}{\partial u} = -c_{\theta_N} c_{\psi_N}$$

$$\frac{\partial f_1}{\partial v} = s_{\psi_N} \qquad \frac{\partial f_1}{\partial w} = -s_{\theta_N} c_{\psi_N}$$

$$\frac{\partial f_2}{\partial y} = \frac{2}{t_f}\boldsymbol{\Delta} \qquad \frac{\partial f_2}{\partial \theta} = u_N s_{\theta_N} s_{\psi_N} - w_N c_{\theta_N} s_{\psi_N}$$

$$\frac{\partial f_2}{\partial \psi} = -u_N c_{\theta_N} c_{\psi_N} + v_N s_{\psi_N} - w_N s_{\theta_N} c_{\psi_N} \qquad \frac{\partial f_2}{\partial u} = -c_{\theta_N} s_{\psi_N}$$

$$\frac{\partial f_2}{\partial v} = -c_{\psi_N} \qquad \frac{\partial f_2}{\partial w} = -s_{\theta_N} s_{\psi_N}$$

$$\frac{\partial f_3}{\partial z} = \frac{2}{t_f}\boldsymbol{\Delta} \qquad \frac{\partial f_3}{\partial \theta} = u_N c_{\theta_N} + w_N s_{\theta_N}$$

$$\frac{\partial f_3}{\partial u} = s_{\theta_N} \qquad \frac{\partial f_3}{\partial w} = -c_{\theta_N}$$

$$\frac{\partial f_4}{\partial \phi} = \frac{2}{t_f}\boldsymbol{\Delta} \qquad \frac{\partial f_4}{\partial \theta} = -\frac{r_N}{c_{\theta_N}^2}$$

$$\frac{\partial f_4}{\partial p} = -\mathbf{I} \qquad \frac{\partial f_4}{\partial r} = -t_{\theta_N}$$

$$\frac{\partial f_5}{\partial \theta} = \frac{2}{t_f}\boldsymbol{\Delta} \qquad \frac{\partial f_5}{\partial q} = -\mathbf{I}$$

$$\frac{\partial f_6}{\partial \theta} = -\frac{r_N s_{\theta_N}}{c_{\theta_N}^2} \qquad \frac{\partial f_6}{\partial \psi} = \frac{2}{t_f}\boldsymbol{\Delta}$$

$$\frac{\partial f_6}{\partial r} = -\frac{1}{c_{\theta_N}} \qquad \frac{\partial f_7}{\partial \theta} = g c_{\theta_N}$$

$$\frac{\partial f_7}{\partial u} = \frac{2}{t_f}\boldsymbol{\Delta} + \frac{1}{\bar{m}}\frac{\partial F_x}{\partial u} \qquad \frac{\partial f_7}{\partial v} = -r_N$$

$$\frac{\partial f_7}{\partial w} = q_N \qquad \frac{\partial f_7}{\partial q} = w_N$$

$$\frac{\partial f_7}{\partial r} = -v_N \qquad \frac{\partial f_8}{\partial u} = r_N - \frac{1}{\bar{m}}\frac{\partial F_y}{\partial u}$$

$$\frac{\partial f_8}{\partial v} = \frac{2}{t_f}\boldsymbol{\Delta} + \frac{1}{\bar{m}}\frac{\partial F_y}{\partial v} \qquad \frac{\partial f_8}{\partial w} = -\frac{1}{\bar{m}}\frac{\partial F_y}{\partial w} - p_N$$

$$\frac{\partial f_8}{\partial p} = -w_N \qquad \frac{\partial f_8}{\partial r} = V_N$$

$$\frac{\partial f_9}{\partial \theta} = gs_{\theta_N} \qquad \frac{\partial f_9}{\partial u} = -q_N - \frac{1}{\bar{m}}\frac{\partial F_z}{\partial u}$$

$$\frac{\partial f_9}{\partial v} = -p_N - \frac{1}{\bar{m}}\frac{\partial F_z}{\partial v} \qquad \frac{\partial f_9}{\partial w} = \frac{2}{t_f}\boldsymbol{\Delta} - \frac{1}{\bar{m}}\frac{\partial F_z}{\partial w}$$

$$\frac{\partial f_9}{\partial p} = v_N - \frac{1}{\bar{m}}\frac{\partial F_y}{\partial w} \qquad \frac{\partial f_9}{\partial q} = -u_N$$

$$\frac{\partial f_{10}}{\partial u} = D\bar{q}S\left(DC_{L_p}\frac{p_N}{2I_{XX}V_N^2}\right) - \frac{\partial \bar{q}}{\partial u}SD\left(C_{L_0} + DC_{L_p}\frac{p_N}{2I_{XX}V_N}\right)$$

$$\frac{\partial f_{10}}{\partial p} = \frac{2}{t_f}\boldsymbol{\Delta} - D\bar{q}S\left(\frac{DC_{L_p}}{2I_{XX}V_N}\right) \qquad \frac{\partial f_{11}}{\partial u} = \frac{\partial M_y}{\partial u}$$

$$\frac{\partial f_{11}}{\partial v} = \frac{\partial M_y}{\partial v} \qquad \frac{\partial f_{11}}{\partial w} = \frac{\partial M_y}{\partial w}$$

$$\frac{\partial f_{11}}{\partial p} = -\frac{1}{I_{YY}}\frac{\partial M_y}{\partial p} - \frac{I_{XX}}{I_{YY}}r_N \qquad \frac{\partial f_{11}}{\partial q} = \frac{2}{t_f}\boldsymbol{\Delta} - \bar{q}S\left(\frac{D^2C_{M_q}}{2I_{YY}V_N}\right)$$

$$\frac{\partial f_{11}}{\partial r} = \frac{I_{XX}}{I_{YY}}p_N \qquad \frac{\partial f_{12}}{\partial p} = -\frac{1}{I_{YY}}\frac{\partial M_z}{\partial p} + \frac{I_{XX}}{I_{YY}}q_N$$

$$\frac{\partial f_{12}}{\partial q} = -\frac{I_{XX}}{I_{YY}}p_N \qquad \frac{\partial f_{12}}{\partial r} = \frac{2}{t_f}\boldsymbol{\Delta} - \bar{q}S\left(\frac{D^2C_{N_r}}{2I_{YY}V_N}\right)$$

$$\frac{\partial f_{11}}{\partial u} = -\frac{1}{I_{YY}}\frac{\partial M_y}{\partial u}$$

$$\frac{\partial f_{11}}{\partial v} = -\frac{1}{I_{YY}}\frac{\partial M_y}{\partial v} \qquad \frac{\partial f_{11}}{\partial w} = -\frac{1}{I_{YY}}\frac{\partial M_y}{\partial w}$$

$$\frac{\partial f_{12}}{\partial u} = -\frac{1}{I_{YY}}\frac{\partial M_z}{\partial u}$$

$$\frac{\partial f_{12}}{\partial v} = -\frac{1}{I_{YY}}\frac{\partial M_z}{\partial v} \qquad \frac{\partial f_{12}}{\partial w} = -\frac{1}{I_{YY}}\frac{\partial M_y}{\partial w}$$

Here are the expressions for $\left.\frac{\partial \mathbf{f}}{\partial \boldsymbol{\vartheta}}\right|_{\mathbf{x}}$ for the coefficients found for spin-stabilized examples in this study:

$$\frac{\partial \mathbf{f}}{\partial C_{N_\alpha}} = \begin{bmatrix} \mathbf{0}_{7\times 1} \\ \bar{q}S\dfrac{v_N}{\bar{m}V_N} \\ \bar{q}S\dfrac{w_N}{\bar{m}V_N} \\ \mathbf{0}_{3\times 1} \end{bmatrix}$$

$$\frac{\partial \mathbf{f}}{\partial C_{m_\alpha}} = \begin{bmatrix} \mathbf{0}_{10\times 1} \\ D\bar{q}S\dfrac{w_N}{I_{YY}V_N} \\ D\bar{q}S\dfrac{v_N}{I_{YY}V_N} \end{bmatrix}$$

$$\frac{\partial \mathbf{f}}{\partial C_{m_q}} = \begin{bmatrix} \mathbf{0}_{10\times 1} \\ -D^2\bar{q}S\dfrac{q_N}{2V_N I_{YY}} \\ -D^2\bar{q}S\dfrac{r_N}{2V_N I_{YY}} \end{bmatrix}$$

$$\frac{\partial \mathbf{f}}{\partial C_{n_{p\alpha}}} = \begin{bmatrix} \mathbf{0}_{10\times 1} \\ -D^2\bar{q}S\dfrac{v_N p_N}{2V_N^2 I_{YY}} \\ -D^2\bar{q}S\dfrac{w_N p_N}{2V_N^2 I_{YY}} \end{bmatrix}$$

$$\frac{\partial \mathbf{f}}{\partial C_{X_0}} = \begin{bmatrix} \mathbf{0}_{6\times 1} \\ \dfrac{\bar{q}S}{\bar{m}} \\ \mathbf{0}_{5\times 1} \end{bmatrix}$$

$$\frac{\partial \mathbf{f}}{\partial C_{l_p}} = \begin{bmatrix} \mathbf{0}_{9\times 1} \\ -D^2\bar{q}S\dfrac{p_N}{2V_N I_{XX}} \\ \mathbf{0}_{2\times 1} \end{bmatrix}$$

$$\frac{\partial \mathbf{f}}{\partial C_{l_0}} = \begin{bmatrix} \mathbf{0}_{9\times 1} \\ -D\dfrac{\bar{q}S}{I_{XX}} \\ \mathbf{0}_{2\times 1} \end{bmatrix}$$

In Figs. A-1 through A-11, we present plots of the fit quality for the 11 shots not included in the results section.

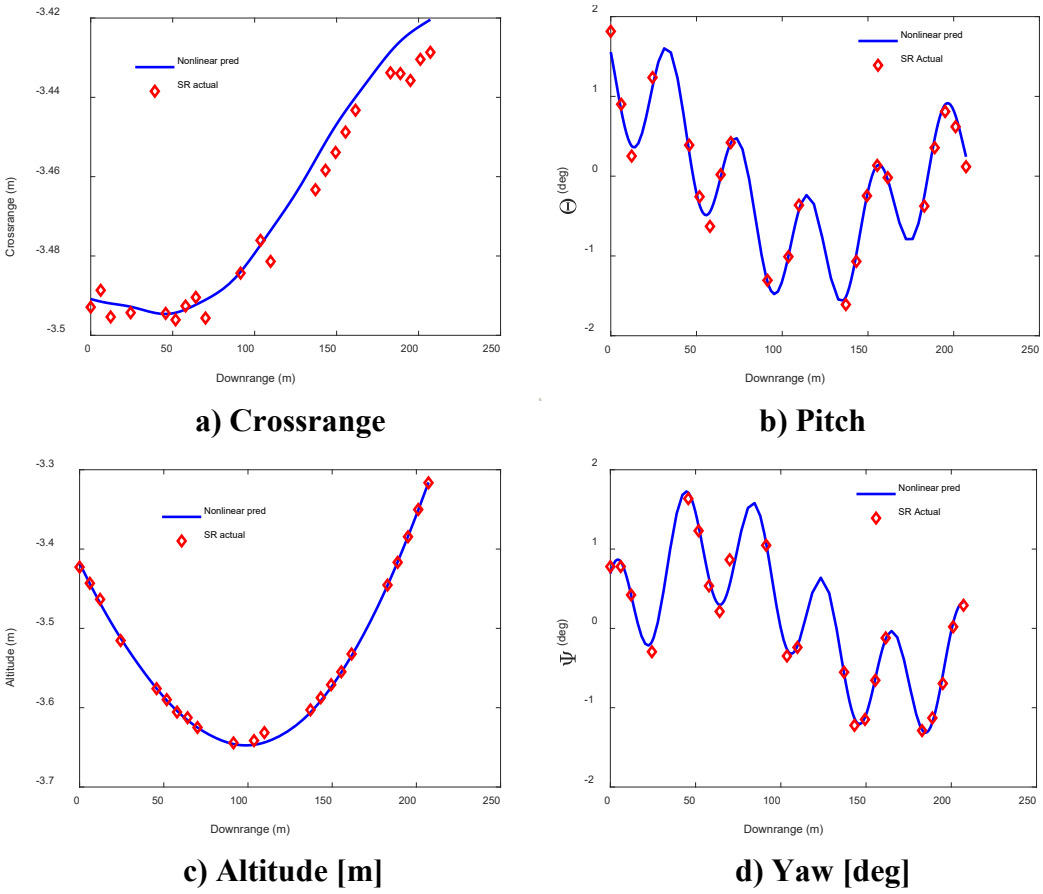


Fig. A-1 Gauss pseudospectral predictions compared with data for *m* = 9 collocation points per segment. M898 Shot TRN 30814, nonlinear model. Solid blue line is the entire trajectory prediction with 22 segments—1 per spark range station.

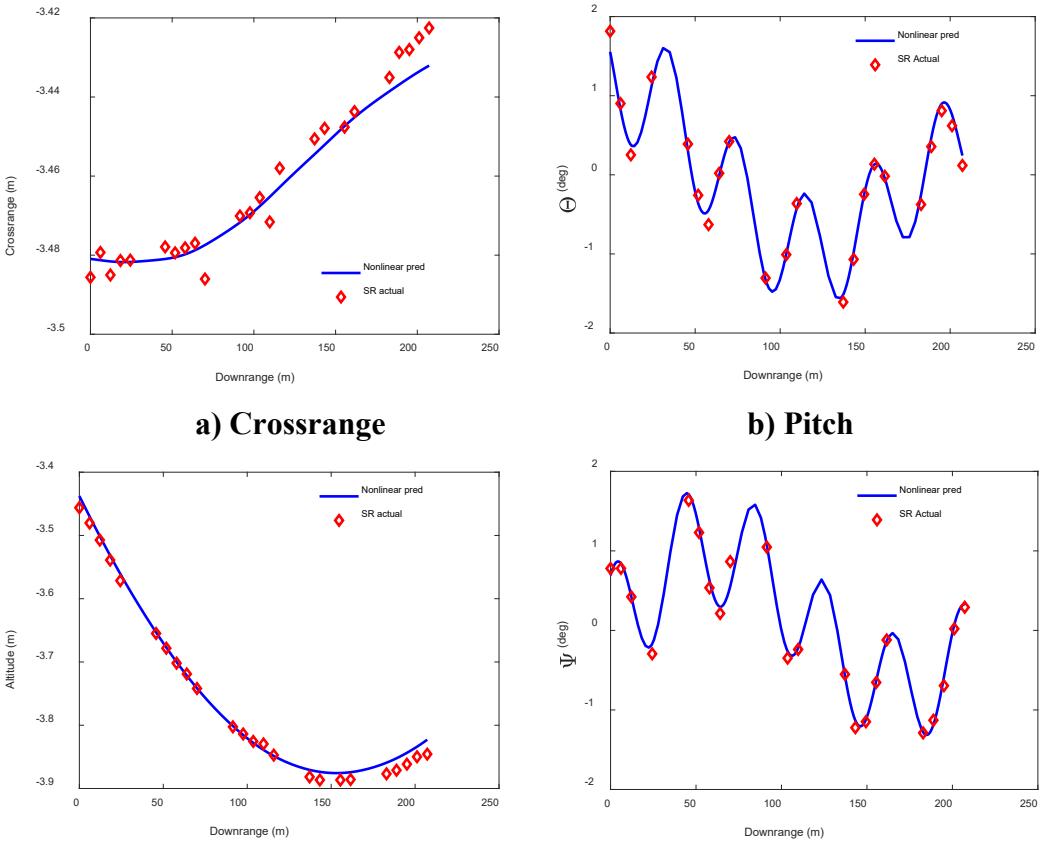


Fig. A-2 Gauss pseudospectral predictions compared with data for *m* = 9 collocation points per segment. M898 Shot TRN 30815, nonlinear model. Solid blue line is the entire trajectory prediction with 24 segments—1 per spark range station.

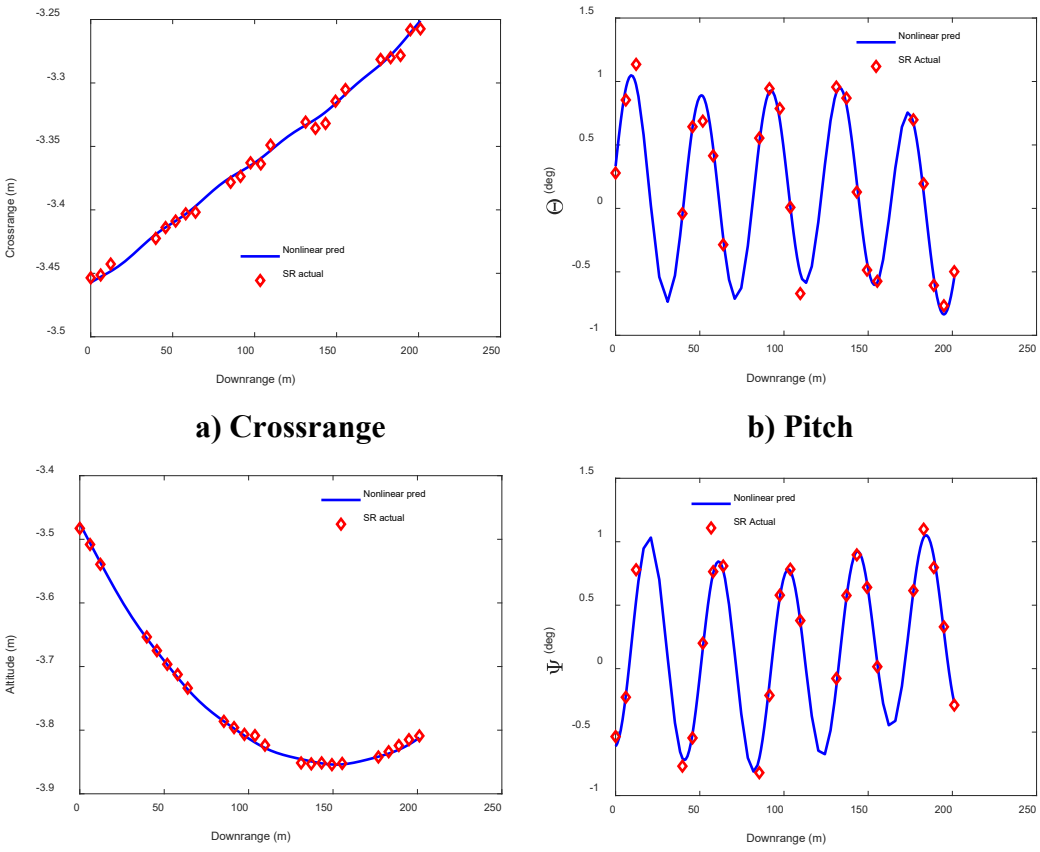


Fig. A-3 Gauss pseudospectral predictions compared with data for m = 9 collocation points per segment. M898 Shot TRN 30818, nonlinear model. Solid blue line is the entire trajectory prediction with 23 segments—1 per spark range station.

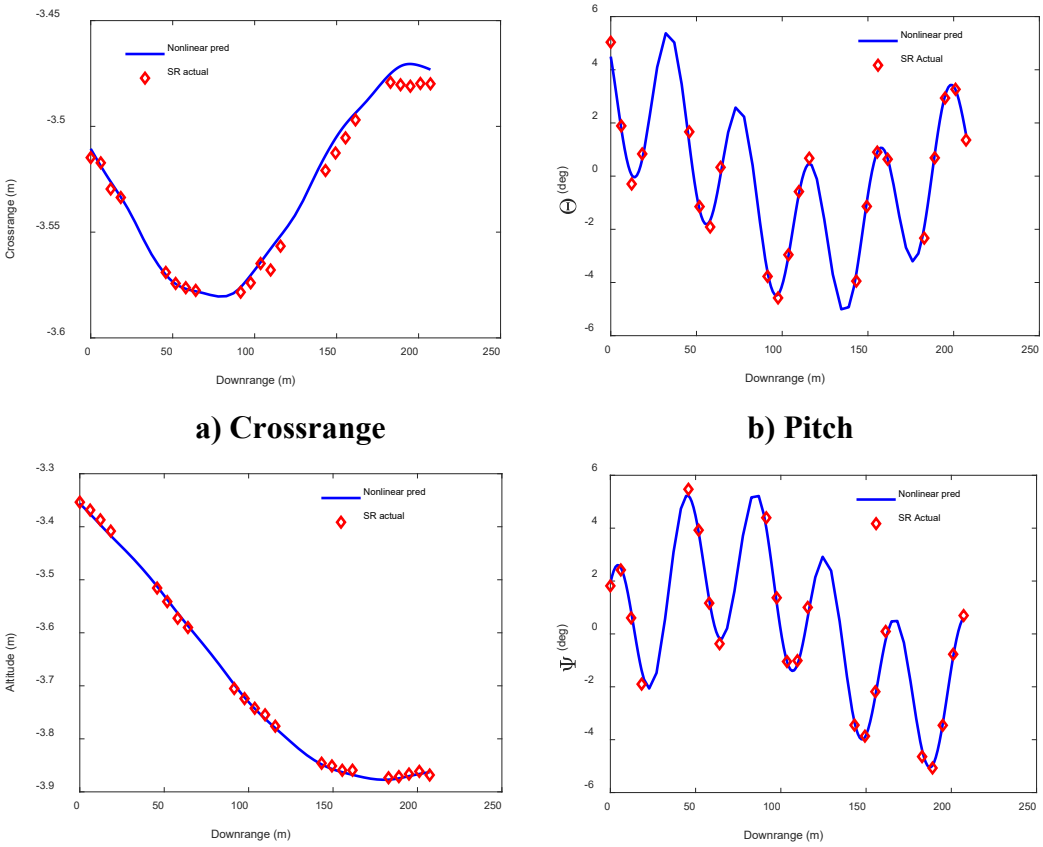


Fig. A-4 Gauss pseudospectral predictions compared with data for m = 9 collocation points per segment. M898 Shot TRN 30824, nonlinear model. Solid blue line is the entire trajectory prediction with 22 segments—1 per spark range station.

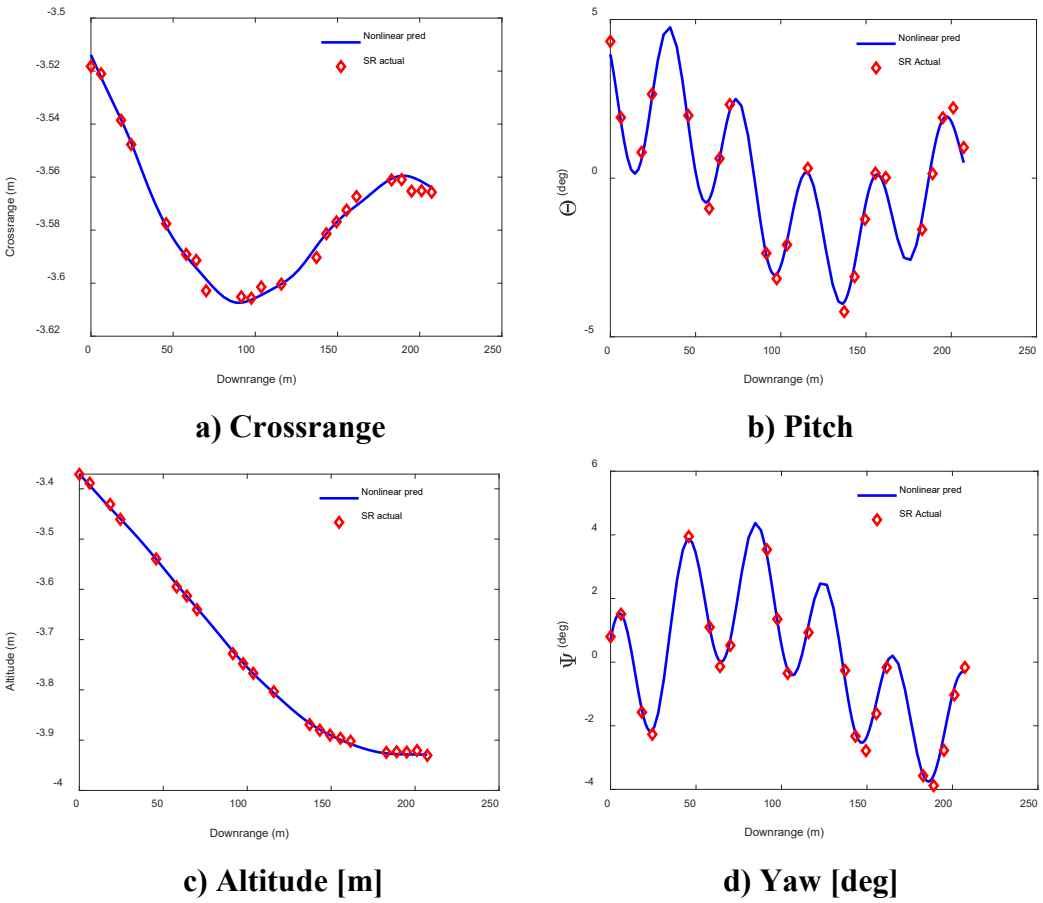


Fig. A-5 Gauss pseudospectral predictions compared with data for *m* = 9 collocation points per segment. M898 Shot TRN 30825, nonlinear model. Solid blue line is the entire trajectory prediction with 22 segments—1 per spark range station.

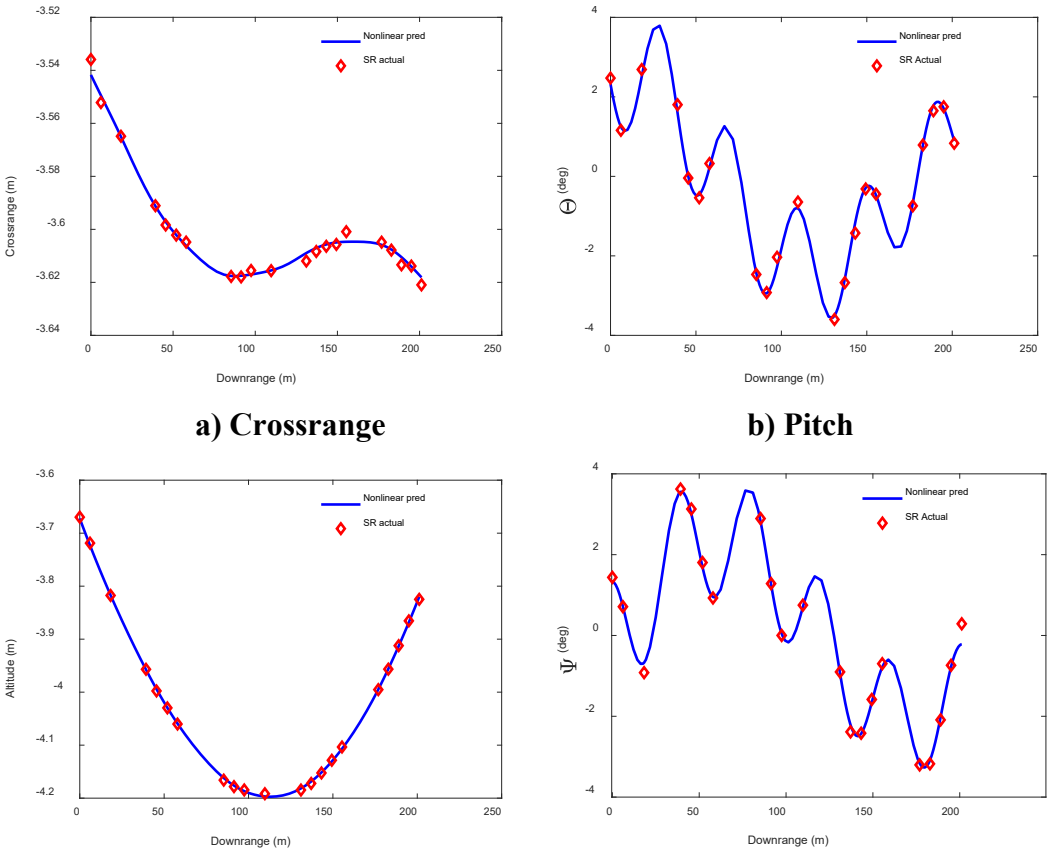


Fig. A-6 Gauss pseudospectral predictions compared with data for *m* = 9 collocation points per segment. M898 Shot TRN 30827, nonlinear model. Solid blue line is the entire trajectory prediction with 22 segments—1 per spark range station.

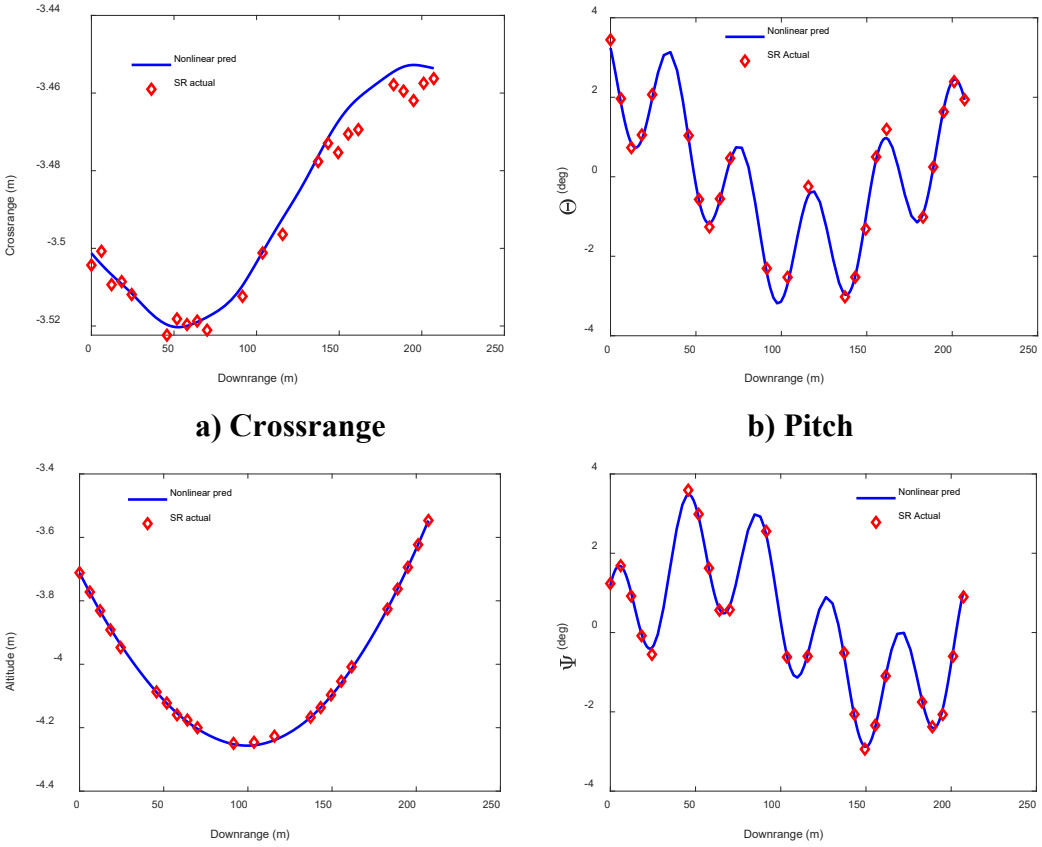


Fig. A-7 Gauss pseudospectral predictions compared with data for *m* = 9 collocation points per segment. M898 Shot TRN 30829, nonlinear model. Solid blue line is the entire trajectory prediction with 23 segments—1 per spark range station.

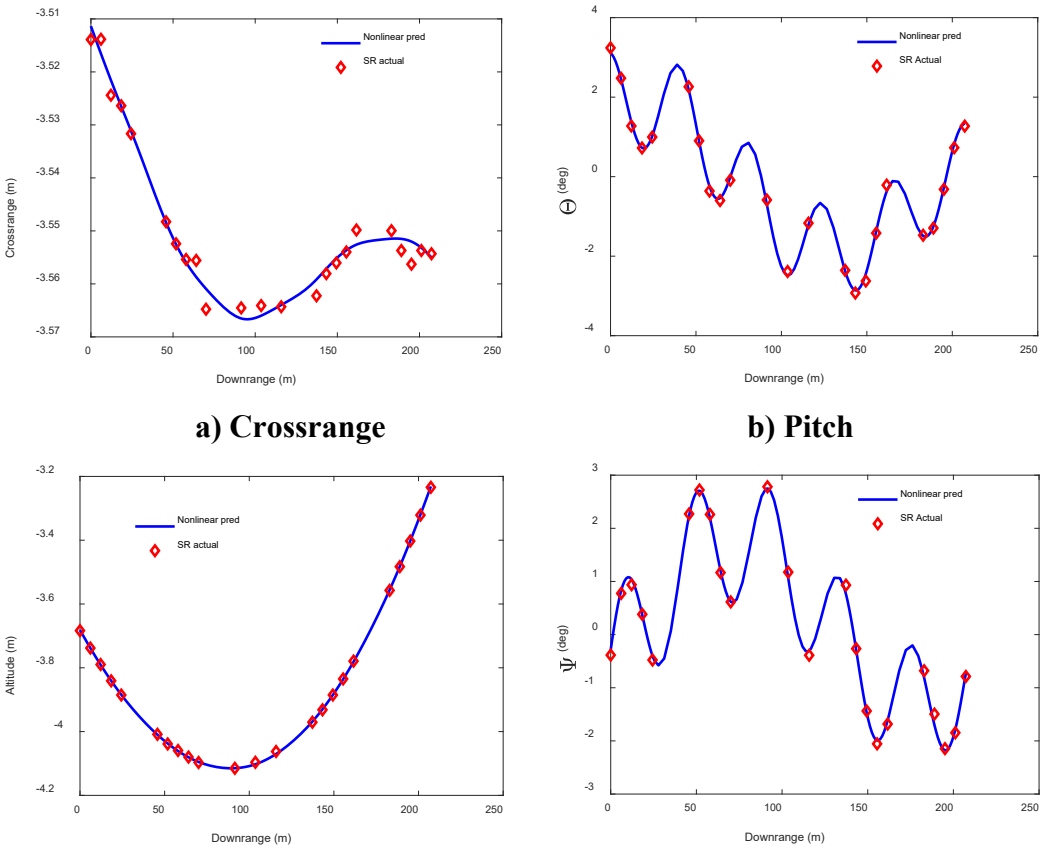


Fig. A-8 Gauss pseudospectral predictions compared with data for *m* = 9 collocation points per segment. M898 Shot TRN 30831, nonlinear model. Solid blue line is the entire trajectory prediction with 23 segments—1 per spark range station.

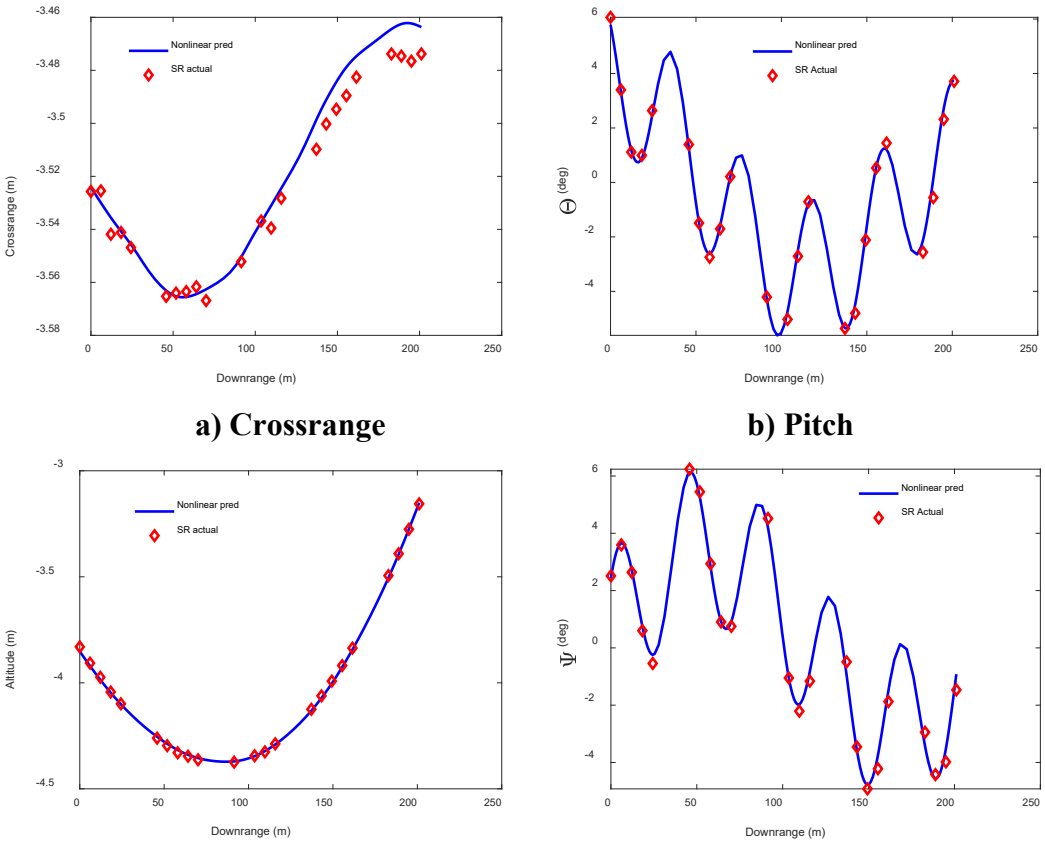


Fig. A-9 Gauss pseudospectral predictions compared with data for *m* = 9 collocation points per segment. M898 Shot TRN 30833, nonlinear model. Solid blue line is the entire trajectory prediction with 23 segments—1 per spark range station.

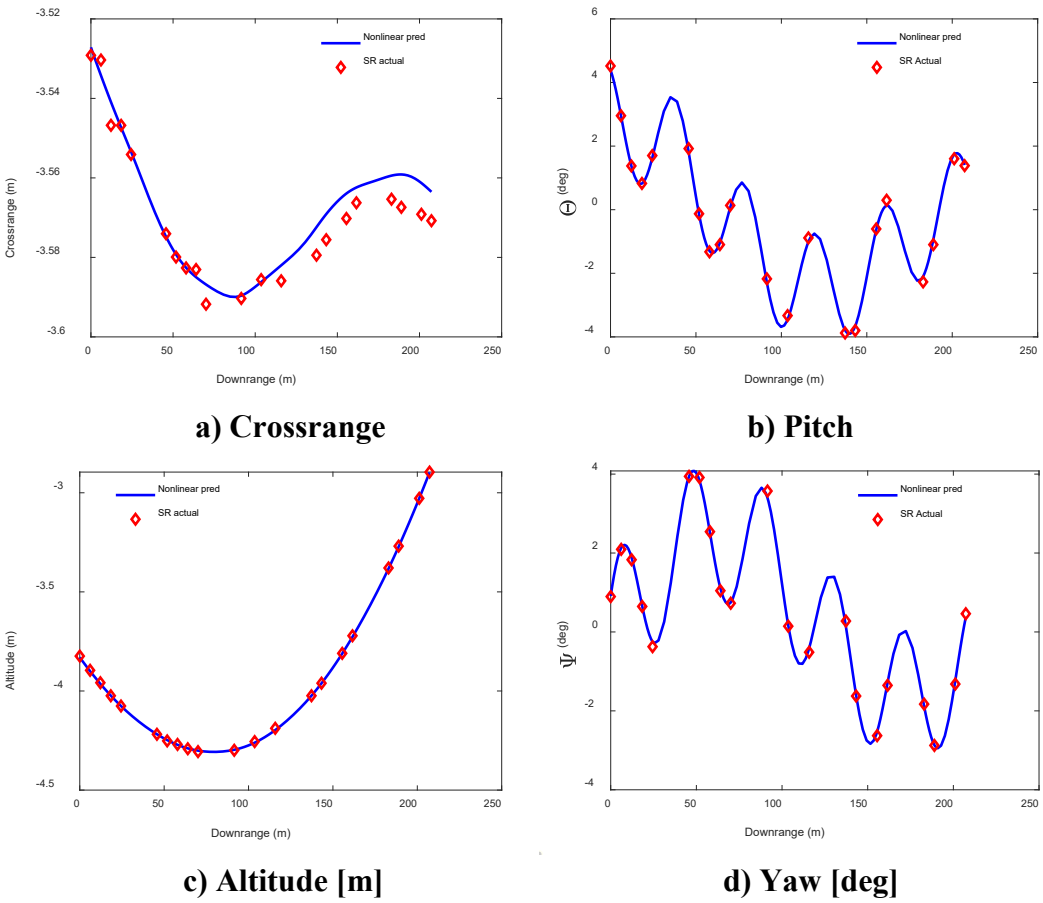


Fig. A-10 Gauss pseudospectral predictions compared with data for m = 9 collocation points per segment. M898 Shot TRN 30834, nonlinear model. Solid blue line is the entire trajectory prediction with 21 segments—1 per spark range station.

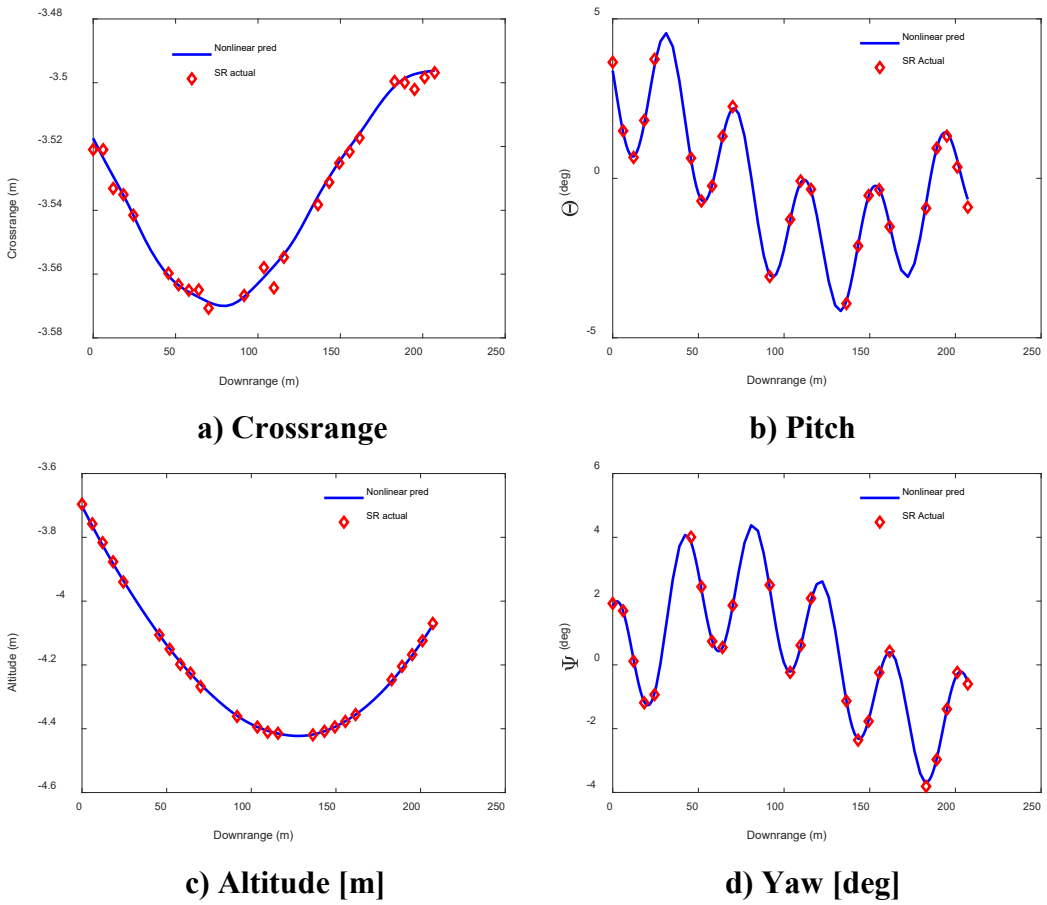


Fig. A-11 Gauss pseudospectral predictions compared with data for $m = 9$ collocation points per segment. M898 Shot TRN 30841, nonlinear model. Solid blue line is the entire trajectory prediction with 24 segments—1 per spark range station.

List of Symbols, Abbreviations, and Acronyms

3-D	three-dimensional
6DOF	6 degrees of freedom
AOA	angle of attack
cg	center of gravity
GPS	Gauss
LGL	Legendre–Gauss–Lobatto
ODE	ordinary differential equation
rms	root mean square
TRN	Transonic Range Number

Nomenclature

$\{x,y,z\}$	projectile cg position in gun tube frame [m]
$\{V_x, V_y, V_z\}$	projectile cg velocity in gun tube frame [m/s]
$\{\phi, \theta, \psi\}$	projectile roll, pitch and yaw in gun tube frame [rad]
$\{u,v,w\}$	projectile linear velocity in the body or no-roll frame [m/s]
$\{p,q,r\}$	projectile angular rates in the body or no-roll frame [rad/s]
$\{X,Y,Z\}$	total force vector in projectile body frame [N]
$\{l,m,n\}$	total moment vector in projectile body frame [N-m]
$\bar{m}$	projectile mass [kg]
m	number of collocation points per segment
T	rotation matrix
$\mathbf{I}$	inertia matrix [kg-m^2] or identity matrix
L	Legendre polynomial
V	total velocity $\sqrt{u^2 + v^2 + w^2}$ [m/s]
Δ	differentiation matrix
Γ	function to be approximated

λ	Lagrange polynomial
μ	Marquardt parameter, initial value = 0.001
ρ	atmospheric density [kg/m^3]
$\overline{\boldsymbol{\rho}}$	vector of residuals—model prediction minus measurements
S	area of projectile cross section [m^2]
D	projectile diameter [m]
C_{X_0}	zero AOA drag coefficient
C_{N_α}	normal force due to AOA coefficient
$C_{N_{\alpha 3}}$	normal force due to AOA cubedC_{m_α} pitch moment due to AOA coefficient
$C_{m_{\alpha 3}}$	pitch moment due to AOA cubed
C_{l_0}	static roll moment coefficient
C_{l_p}	roll damping coefficient
C_{m_q}	pitch damping coefficient
C_{n_α}	out-of-plane moment due to AOA coefficient
C_{Y_α}	out-of-plane force due to AOA coefficient
$C_{n_{p\alpha}}$	Magnus moment coefficient
$C_{n_{\phi\alpha}}$	out-of-plane roll-dependent moment due to AOA coefficient
$C_{Y_{\phi\alpha}}$	out-of-plane roll-dependent force due to AOA coefficient

Subscript

N	collocated states
l	order of the Lagrange polynomial, column of differentiation matrix
m	collocation point number, row of differentiation matrix

Superscript

T	matrix transpose

1 (PDF) DEFENSE TECHNICAL INFORMATION CTR DTIC OCA

1 (PDF) DEVCOM ARL FCDD RLD DCI TECH LIB

10 (PDF) DEVCOM ARL FCDD RLW WD B BURCHETT L STROHM VA BHAGWANDIN J BRYSON I CELMINS J DESPIRITO LD FAIRFAX B GRUENWALD J PAUL JD VASILE

1 (PDF) DEVCOM AC FCDD ACM FA G RODEBAUGH

1 (PDF) ARROW TECH ASSOCIATES R WHYTE