ARL-MR-1041 ● SEP 2021

DEVCOM ARMY RESEARCH LABORATORY

A First Look at Using System Identification Programs for AirCraft (SIDPAC) to Reduce Data from Ballistic Shots

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-MR-1041 ● SEP 2021

A First Look at Using System Identification Programs for AirCraft (SIDPAC) to Reduce Data from Ballistic Shots

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

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED (From - To)
September 2021	Memorandum Report	05 August–05 September 2021

4. TITLE AND SUBTITLE
A First Look at Using System Identification Programs for AirCraft (SIDPAC) to Reduce Data from Ballistic Shots

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-MR-1041

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
NASA's System Identification Programs for AirCraft (SIDPAC) is used to reduce data generated by a coupled Computational Fluid Dynamics/Rigid Body Dynamics (CFD/RBD) simulation. First, the CFD force predictions are used in a direct regression to determine force and moment coefficients for a conventional aircraft-type aerodynamic model of the Laboratory Technology Vehicle. Second, RBD motion predictions of Cartesian position and Euler angle orientation are used as reference for a motion-reconstruction study using the SIDPAC output error (oe.m) routine. Finally, we explore motion reconstruction with sparse measurements mimicking what would be captured in an actual spark-range test.

15. SUBJECT TERMS
spark range, projectile aerodynamics, trajectory reconstruction, system identification, System Identification Programs for AirCraft, SIDPAC

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	31	19b. TELEPHONE NUMBER (Include area code) (812) 201-0390

Contents

List of Figures

List of Tables

1. Introduction

System identification is a process that uses measured inputs and outputs to a dynamic system to devise a model of the system behavior. Typically, the system is assumed to be *dynamic* meaning that system behavior depends somewhat on the history of the system's internal state. Physically, dynamic behavior can typically be explained by the exchange of potential and kinetic energy in the system: the oscillation of a mass-spring damper system is a good example of this. When the analyst has no insight as to the underlying physics of the system, a *black box* model is formed whose structure in no way reflects the actual internal workings of the system. If the analyst knows something about the underlying physics, then a *gray box* model may be formed where a mathematic formalism such as sets of ordinary differential equations are used to describe system behavior. If such a model is adequate to capture system behavior without modification of the model structure, it may be termed a *white box* model. The process of matching a white-box model to data may be described as "parameter identification" since the model, apart from physical constants, was fully determined through physics.

Aircraft system identification is typically a case of gray-box modeling. Certain causes and correlations are well documented in the literature such that a great deal of the model structure may be inferred from aerodynamics and rigid body dynamics (RBD). Also, several standard models of the forces and moments acting on aircraft are well established, so many vehicles can be adequately modeled using known models.

Morelli and Klein have provided a comprehensive history of system identification of aircraft at the NASA Langley Research Center (LaRC)[1] so there is no need to review such history here. Morelli and Klein also authored a leading textbook[2] in the field, and the System Identification Programs for AirCraft (SIDPAC) software.[3]

Parameter identification for projectiles has been done through wind tunnel and free flight tests. More recently, aerodynamic prediction has become much more precise using computational fluid dynamics (CFD) tools. There are several tools already available to reduce data from spark-range free flight tests. However, as new and controlled configurations are developed, these tools need to be updated. In this work, we show that SIDPAC allows the user to tailor algorithms to new configurations and models with minimal effort.

In this report, we provide a first look at using SIDPAC in place of other tools for regression and motion reconstruction of short-range ballistic shots similar to spark-range experiments. Two sets of data generated using a coupled CFD/RBD simulation will be used to illustrate the process and performance of two methods

built into SIDPAC. First, direct regressions are performed using the "swr.m" stepwise regression tool. Motion reconstruction is attempted using several models of varying fidelity to illustrate the software utility and flexibility. Finally, we demonstrate motion reconstruction using a very sparse data set as may be encountered in actual spark-range testing.

2. Direct Regression from CFD Force Predictions

Two sets of data were generated by coupling the Metacomp CFD++[4] solver with the BOOM rigid body ballistics tool.[5] Twenty quantities were stored at a rate of 10 kHz, including point number, time, 12 states, 3 forces, and 3 moments in the body frame.

Unlike actual flight tests where the forces and moments cannot be directly measured, CFD gives us these quantities. Thus, we are able to apply regression directly to the forces and moments, choosing inputs to match an expected aerodynamic model. We choose a model that is linear in the unknown constants such as

$$y = \beta_0 x_0 + \beta_2 x_1 + \beta_3 x_3 + \cdots + \beta_m \cdot 1 \tag{1}$$

where $\beta_0, \ldots, \beta_{m-1}$ are unknown constants, $x_0, \ldots, x_m$ are the regressors, and where $x_m = 1$ and the final term β_m is a bias that SIDPAC automatically assigns to the end of the parameter vector. Given the model of Eq. 1 and data vector $\mathbf{y}$ of length $> m$, and matrix of regressors $\mathbf{X}$, the least squares estimate for the coefficient vector $\boldsymbol{\beta} = [\beta_0, \beta_1, \beta_2, \ldots, \beta_m]$ can be found by[6]

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{y}) \tag{2}$$

When using SIDPAC, Eq. 2 is transparent to the user. SIDPAC solves Eq. 2 using the Moore–Penrose singular value decomposition-based inverse and has built-in remediation for ill-conditioning.

We propose the following simple aerodynamic models, and examine the performance of individual terms in the results following. For body *x* force:

$$F_x = -\bar{q}S(C_{x2}s_{\bar{\alpha}}^2 + C_{x4}s_{\bar{\alpha}}^4 + C_{x0}) \tag{3}$$

For body *y* force[8]:

$$F_y = \bar{q}S\left(-C_{N\alpha}s_\beta - C_{N3}s_\beta^3 + C_{Y\phi\alpha}s_{\bar{\alpha}}^2 s_\alpha \sin N\phi_A + C_{Y\alpha}s_\alpha\right) \tag{4}$$

For body z force:

$$F_z = -\bar{q}S\left(C_{N\alpha}s_\alpha + C_{N3}s_\alpha^3 + C_{Y\phi\alpha}s_{\bar{\alpha}}^2 s_\beta \sin N\phi_A + C_{Y\alpha}s_\beta\right) \tag{5}$$

For body x moment:

$$M_x = \bar{q}SD\left(C_{l\alpha}s_{\bar{\alpha}} + C_{l\alpha 2}s_{\bar{\alpha}}^2 + C_{l\delta}\delta + \frac{pD}{2V_\infty}C_{lp}\right) \tag{6}$$

For body y moment:

$$M_y = \bar{q}SD\left(C_{m\alpha}s_\alpha + C_{m3}s_\alpha^3 + C_{n\phi\alpha}s_{\bar{\alpha}}^2 s_\beta \sin N\phi_A + C_{n\alpha}s_\beta + \frac{qD}{2V_\infty}C_{mq}\right) \tag{7}$$

For body z moment:

$$M_z = \bar{q}SD\left(-C_{m\alpha}s_\beta - C_{m3}s_\beta^3 + C_{n\phi\alpha}s_{\bar{\alpha}}^2 s_\alpha \sin N\phi_A + C_{n\alpha}s_\alpha + \frac{rD}{2V_\infty}C_{mq}\right) \tag{8}$$

Where, assuming small sideslip, $s_\alpha = \frac{w}{V_\infty}$, also

$$\phi_A = \tan^{-1}\frac{v}{w}, \qquad s_\beta = \frac{v}{V_\infty}, \qquad s_{\bar{\alpha}} = \sqrt{s_\alpha^2 + s_\beta^2} = \frac{\sqrt{v^2 + w^2}}{V_\infty}$$

SIDPAC's swr.m function provides a menu-driven user interface, and dynamically updated figure.[2,7] The user calls the function from the MATLAB command line as

```
[y,p] = swr(X,C,1);
```

Where `X` is a matrix of regressors, `C` is the target data, and `y` is the prediction using the vector `p` of parameters. The user needs to assemble the input data prior to this call, such as:

```
X = [alfabar.^2, alfabar.^4];
```

The swr function automatically assumes a bias term, so no columns are added to the regressor matrix X for bias. Upon calling swr, the function prompts the user with the menu shown in Fig. 1.

```
                                                          Squared
      Parameters                          F ratio      Part. Corr.
No.    Estimate        Change              In            Out
---    --------        ------              --            ---
 1   -3.6447e+00    -1.6648e+00     7.3343e+03      0.00000
 2    3.9920e+02     3.9920e+02     1.6221e+03      0.00000
  constant term  = -4.3884e-01     F cut-off value =  20.00
  dependent variable rms value =   4.4474e-01
  fit error  =  8.159100e-04  or   0.18 percent
  R squared  =  88.87 %        PRESS =  3.3228e-03
                                 PSE =  6.6890e-07

  NUMBER OF REGRESSOR TO MOVE (0 to quit)
```

Fig. 1 SIDPAC swr.m menu after completion of the Axial Force Regression

The user can toggle which regressors are included by pressing 1 or 2 in this example. The bias term is included by default, and is listed as "constant term". Starting from all regressors toggled off, the user should toggle them on from greatest to least "Squared Part. Corr. Out". After adding a regressor, check that the "F ratio In" is above the "F cut-off value". It not, remove it permanently from the solution.[7]

Note that the menu gives a current fit error in percent and an R squared value. The tool also creates a plot that is dynamically updated with each user change to the regressor matrix. The result for Axial force for the supersonic shot is shown in Fig. 2. The plot in Fig. 2 corresponds one-to-one with the text output shown in Fig. 1. We present the remaining five plots next and reserve the remaining estimated coefficients for Table 1.

Figure 3 shows the final quality of fit for Z force prediction using the model shown in Eq. 5. The R squared value for this case is 99.98%, indicating very strong correlation between the data and model. Figure 4 shows the final quality of fit for Y force prediction using the model of Eq. 4. The R squared value for this case is 99.98% as well.

Figure 5 shows the final fit quality for axial moment prediction using the model shown in Eq. 6. The R squared value for this fit is 99.48%. The general downward trend in axial moment is mainly due to the damping term (C_{lp}). Figure 6 displays the quality of fit for pitch moment using the model shown in Eq. 7. Figure 7 shows the final fit for yaw moment using the model shown in Eq. 7. Pitch and yaw moments are strongly correlated to angle of attack/sideslip cubed.

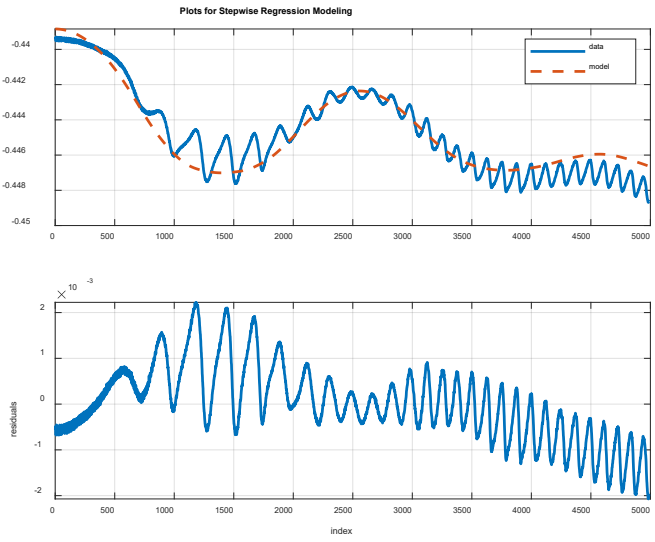


Fig. 2 SIDPAC swr.m figure window, Body Axial Force regression V0 = 535 m/s, final fit error = 0.18%

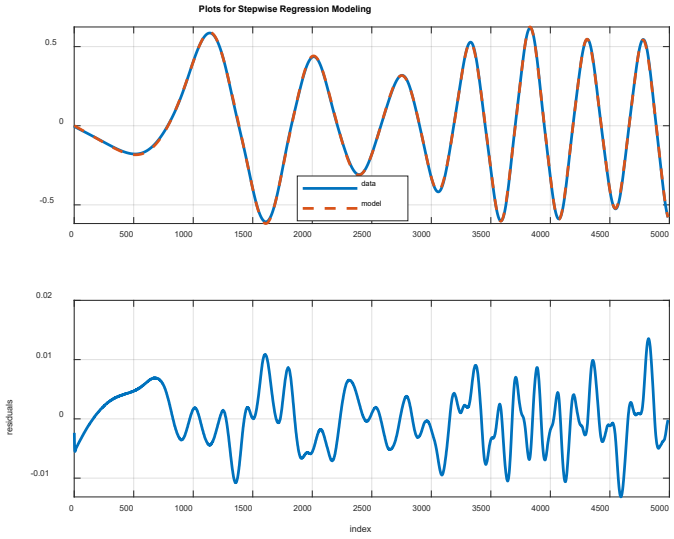


Fig. 3 Body *Z* Force regression V0 = 535 m/s, final fit error = 1.43%

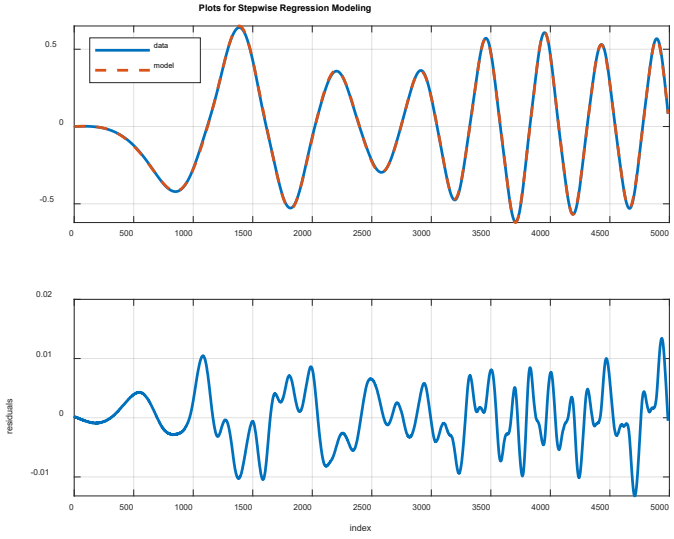


Fig. 4 Body *Y* Force regression V0 = 535 m/s, final fit error = 1.47%

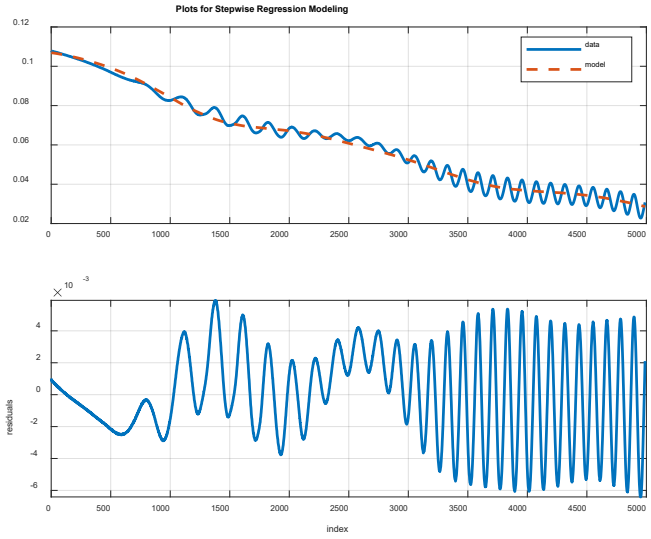


Fig. 5 Body Axial Moment regression V0 = 535 m/s, final fit error = 4.36%

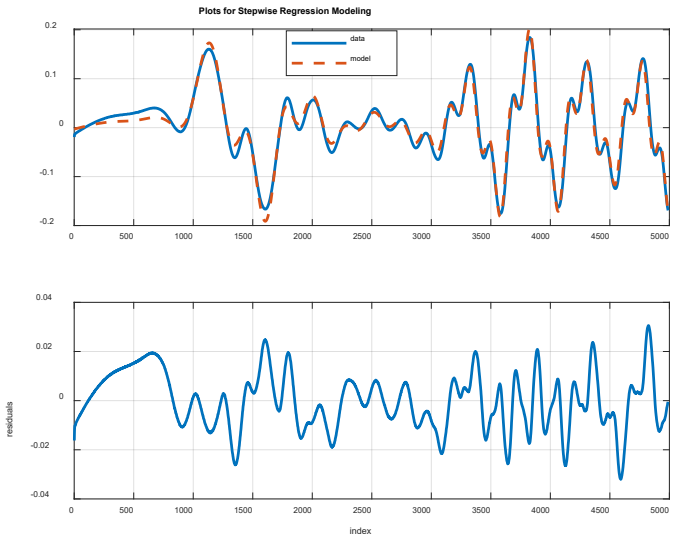


Fig. 6 Body Pitch Moment regression, V0 = 535 m/s, final fit error = 17.02%

Fig. 7 Body Yaw Moment regression, V0 = 535 m/s, final fit error = 15.65%

3. Motion Reconstruction Using Output Error (oe.m)

SIDPAC provides a time-domain output error search method through the function oe.m. The function is called by

```
[y,p] = oe('mname',p0,u,t,x0,c,z);
```

Where “mname” is the function name of a user-defined model. Such models can be linear or nonlinear. The parameter vector p is assumed to be time invariant. The

user must also supply an initial guess of the parameter vector, p0; an array of the input history, u; the time vector, t; the initial state, x0; any constants to be passed to the model, c; and the target output array, z. After iteration, the function will return the predicted output array, y, and the best fit parameters, p.

We tried several models in seeking a match to the data sets. These include a projectile linear theory model, a nonlinear, body-fixed, 6 degrees-of-freedom (DOF) model, and a Simulink implementation of a 6DOF model. The algorithm uses finite differencing to approximate the gradient vector in parameter search space. Such finite differencing means that the search run time is directly proportional to the number of adjustable parameters. This caused the Simulink implementation to run extremely slowly. Also, we ran into memory issues with iterating the Simulink model if it contained more than 12 adjustable parameters. The model itself was set up to accommodate up to 33.

Note that this method assumes the initial vehicle state is known—an invalid assumption for gun-launched tests such as spark ranges. For the virtual CFD/RBD data, we do in fact know the initial state and will take advantage of that fact. For a practical spark-range algorithm, it is necessary to estimate the initial state as well as the parameters. The oe.m has built-in checks for divergence. If the gradient-based method diverges, it automatically reverts to a simplex method. Note that simplex methods require a number of function evaluations one greater than the dimension of the search space; however, they do not require gradient information. Since the finite differencing approach uses centered differencing, it actually requires a number of function evaluations equal to twice the dimension of the search space plus one.

These features cause the algorithm to run quite slowly; however, from a user standpoint there is no need to code analytic gradients. Also, the user can invoke many MATLAB features including the ode suite such that nonlinear models are integrated using robust, variable time-step methods. This tends to speed up the algorithm in comparison to fixed-step, user-defined Runge–Kutta methods. Also, the oe.m algorithm is called in a single line of MATLAB and needs one (linear) or two (nonlinear) user-defined functions for the model. Thus, the user needs to write about 200 lines of code to use this method, as opposed to about 2000 to code up a full gradient search from scratch.

3.1 Motion Reconstruction from Full Simulated Data

We attempted to reconstruct the motion using two levels of data density and two sets of data. The low-speed example used a 0.99-s time horizon with a sampling rate of 10 kHz for a total of 9900 samples. The high-speed example had a time

horizon of 0.499 for a total of 4989 samples. Using the entire history in each case would have been computationally prohibitive, so for the dense data we downsampled each of these to 1 kHz for total data lengths of 990 and 499, respectively.

A body-fixed nonlinear 6DOF model was also used to fit both the high and low speed shots downsampled to 1 kHz. That model is shown in Eqs. 9–20.

$$\dot{x} = V\, c_\theta\, c_\psi + v(s_\phi s_\theta c_\psi - c_\phi s_\psi) + w(c_\phi s_\theta c_\psi + s_\phi s_\psi) \tag{9}$$

$$\dot{y} = V\, c_\theta\, s_\psi + v(s_\phi s_\theta s_\psi - c_\phi c_\psi) + w(c_\phi s_\theta s_\psi - s_\phi c_\psi) \tag{10}$$

$$\dot{z} = -V s_\theta + v s_\phi c_\theta + w c_\phi c_\theta \tag{11}$$

$$\dot{\phi} = p + q s_\phi t_\theta + r c_\phi t_\theta \tag{12}$$

$$\dot{\theta} = q c_\phi - r s_\phi \tag{13}$$

$$\dot{\psi} = \frac{q s_\phi}{c_\theta} + \frac{r c_\phi}{c_\theta} \tag{14}$$

$$\dot{u} = \frac{F_x}{m} + rv - qw - g s_\theta \tag{15}$$

$$\dot{v} = \frac{F_y}{m} - ru + pw - c_\theta s_\phi g \tag{16}$$

$$\dot{w} = \frac{F_z}{m} + qu - pv - c_\theta c_\phi g \tag{17}$$

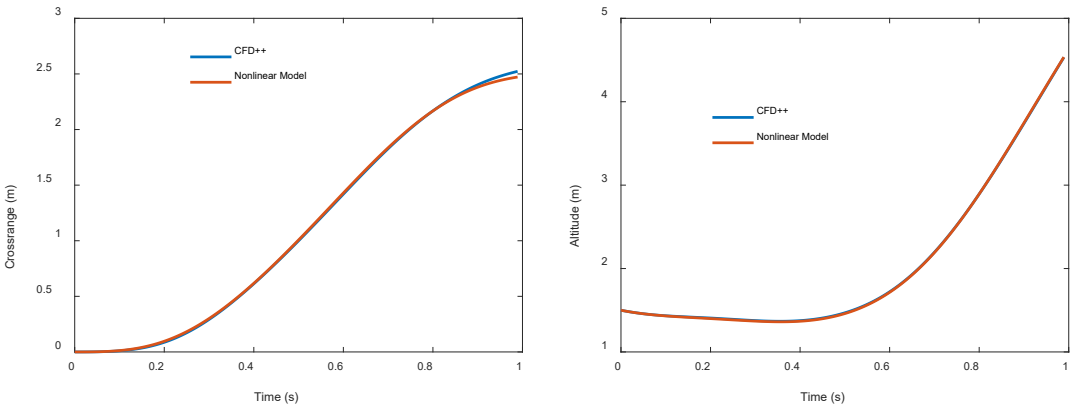
$$\dot{p} = \frac{M_x}{I_{xx}} \tag{18}$$

$$\dot{q} = \frac{M_y}{I_{yy}} + rp\left(1 - \frac{I_{XX}}{I_{YY}}\right) \tag{19}$$

$$\dot{r} = \frac{M_z}{I_{YY}} + pq\left(\frac{I_{XX}}{I_{YY}} - 1\right) \tag{20}$$

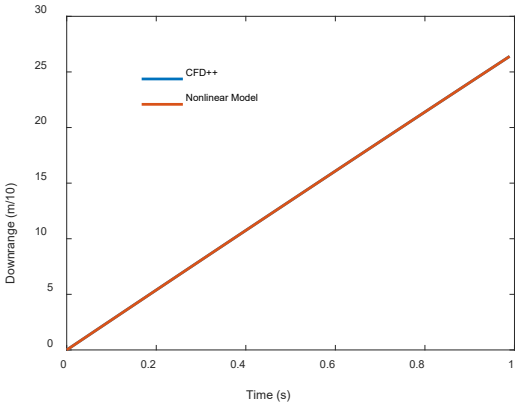
Where the force and moment terms are defined in Eqs. 3–8.

After 499 iterations, the algorithm was able to reconstruct the motion to the degree of accuracy shown in Figs. 8 and 9 and Table 1.

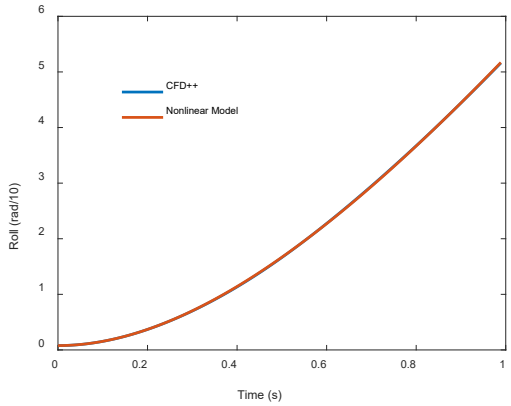


a) Crossrange

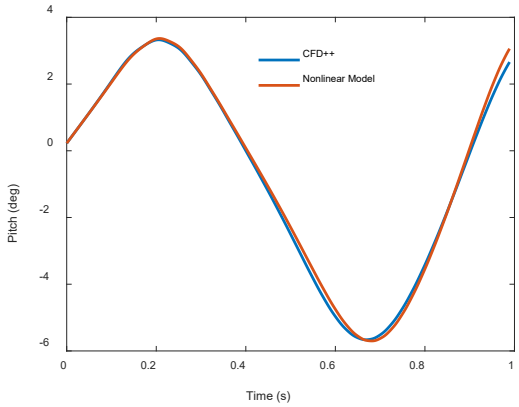
b) Altitude



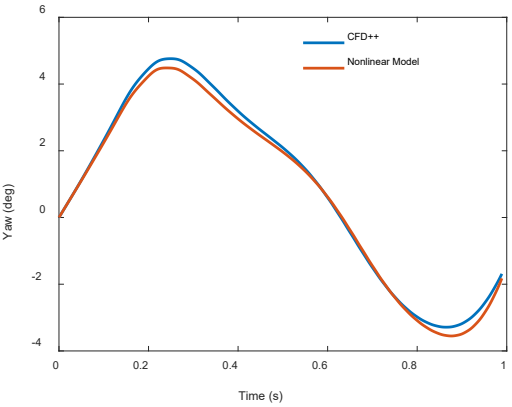
c) Downrange



d) Roll

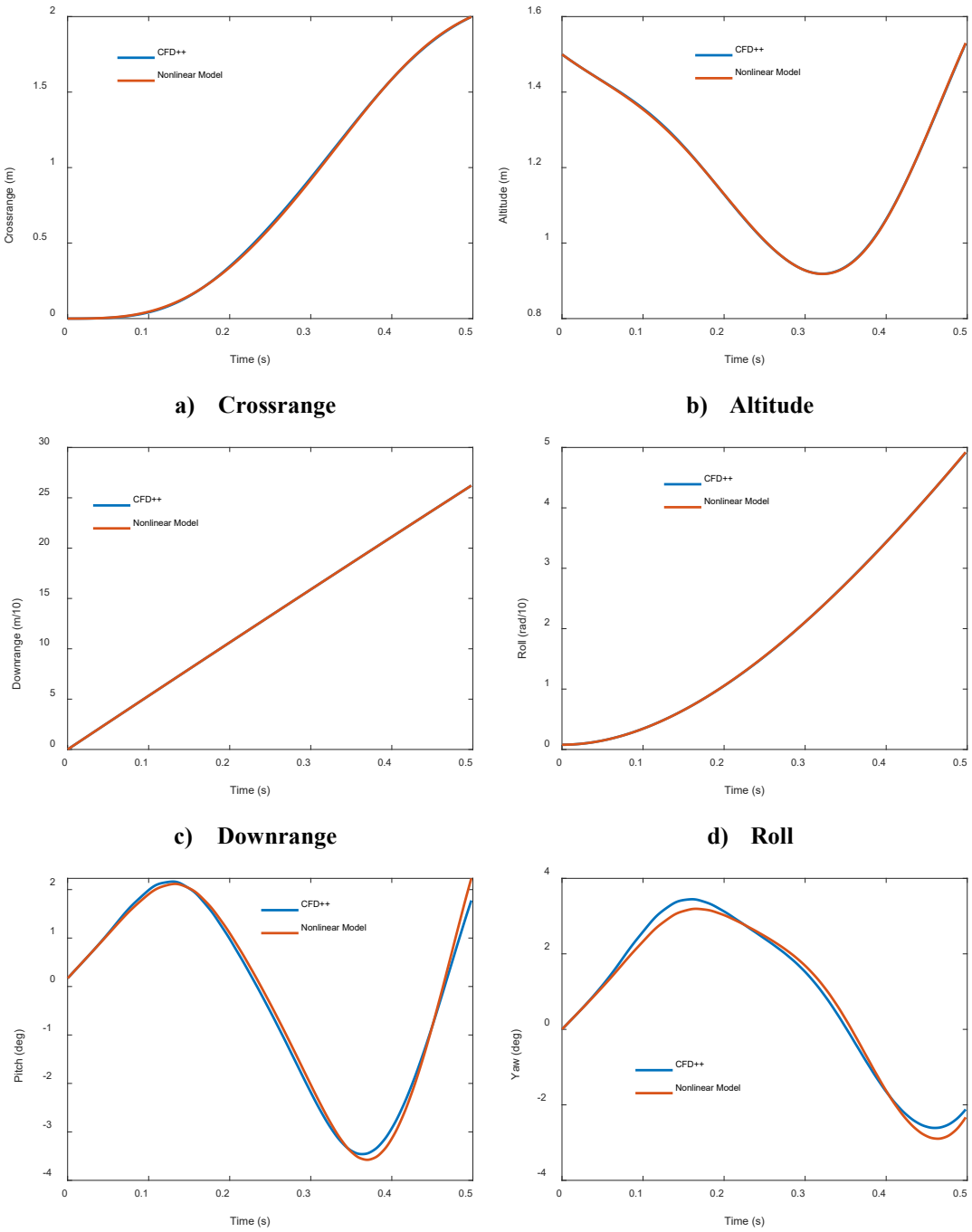


e) Pitch



f) Yaw

Fig. 8 Nonlinear model motion matching result, V0 = 270 m/s



a) Crossrange **b) Altitude**

c) Downrange **d) Roll**

e) Pitch **f) Yaw**

Fig. 9 Nonlinear model motion matching result, V0 = 535 m/s

Table 1 Motion-reconstruction rms prediction errors

	Root-mean-square error					
Case.	X **[m]**	Y **[m]**	Z **[m]**	ϕ **[deg]**	θ **[deg]**	ψ **[deg]**
Nonlin 270m/s	0.00018	0.013567	0.00603	0.04632	0.14567	0.19028
Nonlin 535m/s	0.00009	0.00679	0.00142	0.02624	0.1361	0.1802
SR 270m/s	0.00150	0.00393	0.00580	0.03751	0.12968	0.14672
SR 535m/s	0.00026	0.00476	0.00500	0.01883	0.13630	0.08906
ICs 270m/s	0.00131	0.00211	0.00511	0.08788	0.13242	0.041112
ICs 535m/s	0.00025	0.00370	0.00487	0.02594	0.03215	0.06382

Table 1 shows the root-mean-square (rms) prediction errors for all of the motion reconstruction cases. The rms errors for the nonlinear (Nonlin) cases rival those of established spark-range methods. The standard deviation of measurement noise for the Transonic Experimental Facility (TEF) at the US Army Combat Capabilities Development Command Army Research Laboratory is considered to be σ = 3.3 mm for position and 0.15° for angles. The nonlinear prediction errors are close to those benchmarks. However, note that the results shown here benefit from several unfair advantages including 1) a very dense data set on the order of hundreds of points rather than 20, 2) known initial conditions, and 3) no measurement noise. The rows labeled "SR" are the results when simulating spark range data, which is very sparse compared to the full set. These results are discussed in the next section. Those labeled "ICs" are from the algorithm described in Section 3.3.

The estimated aerodynamic coefficients for each method are summarized in Tables 2 and 3. The first two columns compare the aero coefficient estimates for direct regression and output error using the full data set for the high-speed shot. The third (SR) shows estimates using only measurements at 25 virtual spark stations, but assuming known initial conditions. The fourth column (ICs) shows the estimates for an algorithm that assumes unknown initial conditions and measurements at only 25 virtual spark stations. Table 3 compares the estimates for the low-speed shot. Nearly all of the coefficients are the same order of magnitude—and typically quite close in magnitude with the exception of C_{mq}—close enough to offer an extra degree of confidence in the nonlinear motion model. Roll-dependent side force, $C_{Y\phi\alpha}$, and side moment, $C_{n\phi\alpha}$, tend to be weakly correlated to the measurements and vary greatly from method to method. These terms were dropped

from the final analysis (ICs) to reduce the DOF since the final search also included 12 unknown initial conditions.

Table 2 SIDPAC aerodynamic coefficient estimates from CFD++ predictions, V0 = 535 m/s

Coeff:	Method			
	Regression	Output error (full)	Output error (SR)	Output error (ICs)
$C_{N\alpha}$	8.4741	7.8253	9.8386	8.4408
$C_{m\alpha}$	1.0137	1.1646	0.97858	0.8308
C_{mq}	16.6679	–52.211	–37.454	–122.1
$C_{Y\alpha}$	–0.1748	12.622	–10.56	–15.995
$C_{Y\phi\alpha}$	12.9062	19.573	6.5524	. . .
$C_{n\phi\alpha}$	–36.7073	106.92	1083.8	. . .
$C_{n\alpha}$	0.6019	0.56018	0.55045	0.6695
C_{X0}	0.43872	0.44372	0.44283	0.43884
C_{lp}	–4.8514	–4.7849	–4.6291	–4.3154
C_{l0}	0.10691	0.1090	0.10789	0.10494
C_{N3}	474.8230	539.07	334.59	476.71
C_{m3}	–1132.1	–1129.1	–1230.9	–1023.4

Table 3 SIDPAC aerodynamic coefficient estimates from CFD++ predictions, V0 = 270 m/s

Coeff:	Method			
	Regression	Output error (Full)	Output error (SR)	Output error (ICs)
$C_{N\alpha}$	8.1801	7.1599	5.0795	6.3028
$C_{m\alpha}$	1.0367	1.3373	1.3438	1.3204
C_{mq}	–64.2074	–183.63	–186.34	–249.1
$C_{Y\alpha}$	–0.1741	12.66	–4.2303	–11.073
$C_{Y\phi\alpha}$	17.291	17.622	–188.72	. . .
$C_{n\phi\alpha}$	–31.5301	–28.3	–226.21	. . .
$C_{n\alpha}$	0.6118	0.59537	0.5604	0.74143
C_{X0}	0.28164	0.28385	0.27884	0.27745
C_{lp}	–5.6366	–5.4186	–5.2896	–5.2991
C_{l0}	0.11901	0.11861	0.11782	0.1173
C_{N3}	316.1749	320.26	1067.5	721.87
C_{m3}	–613.2058	–644.94	–645.48	–652.98

3.2 Motion Reconstruction with Simulated Spark-Range Data, Assuming Known Initial Conditions

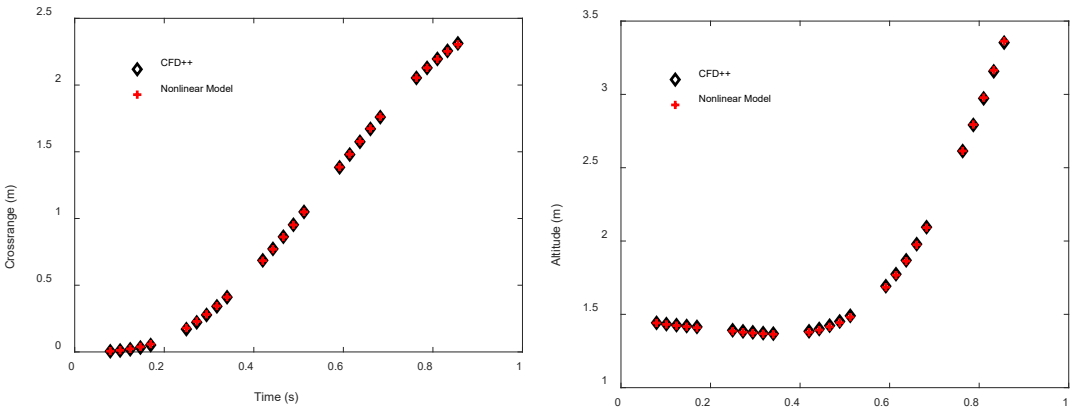
Having demonstrated the 6DOF model and obtained sets of aerodynamic coefficients that match the 1-kHz sampled motion data within accepted tolerances, we want to further exercise the model and algorithm by removing all but 25 sets of

measurements that correspond to the spark-range stations in the TEF. In order to contrive this data set, we merely interpolate the full set of data at downrange distances that correspond to the spark-range stations. Note that we must also interpolate the time vector to get the times at which the projectile crosses each downrange point that defines a station. Then, we apply the output error algorithm by having the nonlinear model return predictions only at the times specified in the interpolated time vector.

We seed the algorithm with estimates from direct regression as shown in Tables 2 and 3. Table 1 shows the final prediction errors for both shots. The errors are very similar in magnitude to those found using the full data set. Tables 2 and 3 show the aero coefficients under the heading *Output Error (SR)* in comparison with the two previous methods.

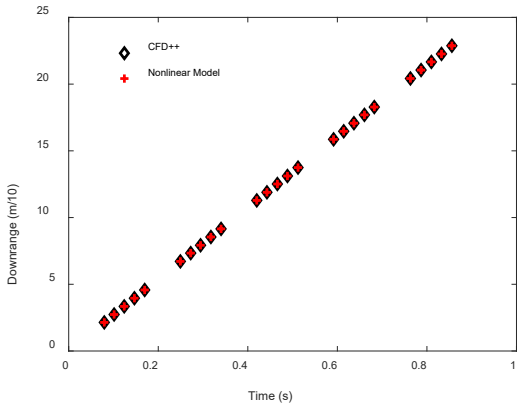
Figures 10 and 11 show the quality of match between simulated spark-range data and predictions.

In order to test the robustness of the algorithm, we seeded the search with parameters perturbed up to 50% from the regression estimates. As long as the initial conditions were known, the algorithm had little trouble finding a set of parameters that would fit the data within acceptable limits.

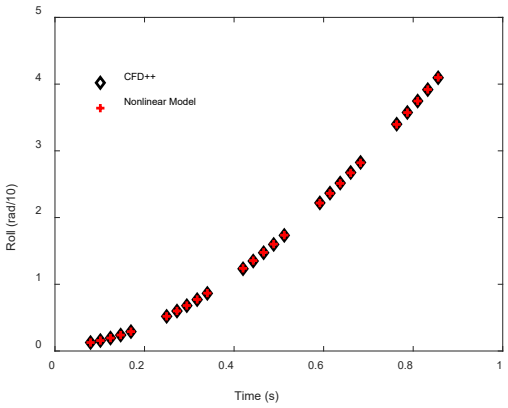


a) Crossrange

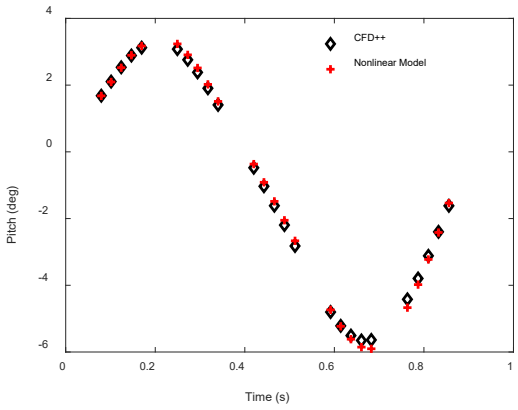
b) Altitude



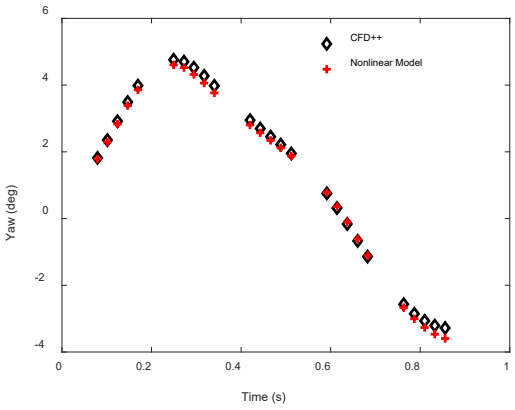
c) Downrange



d) Roll

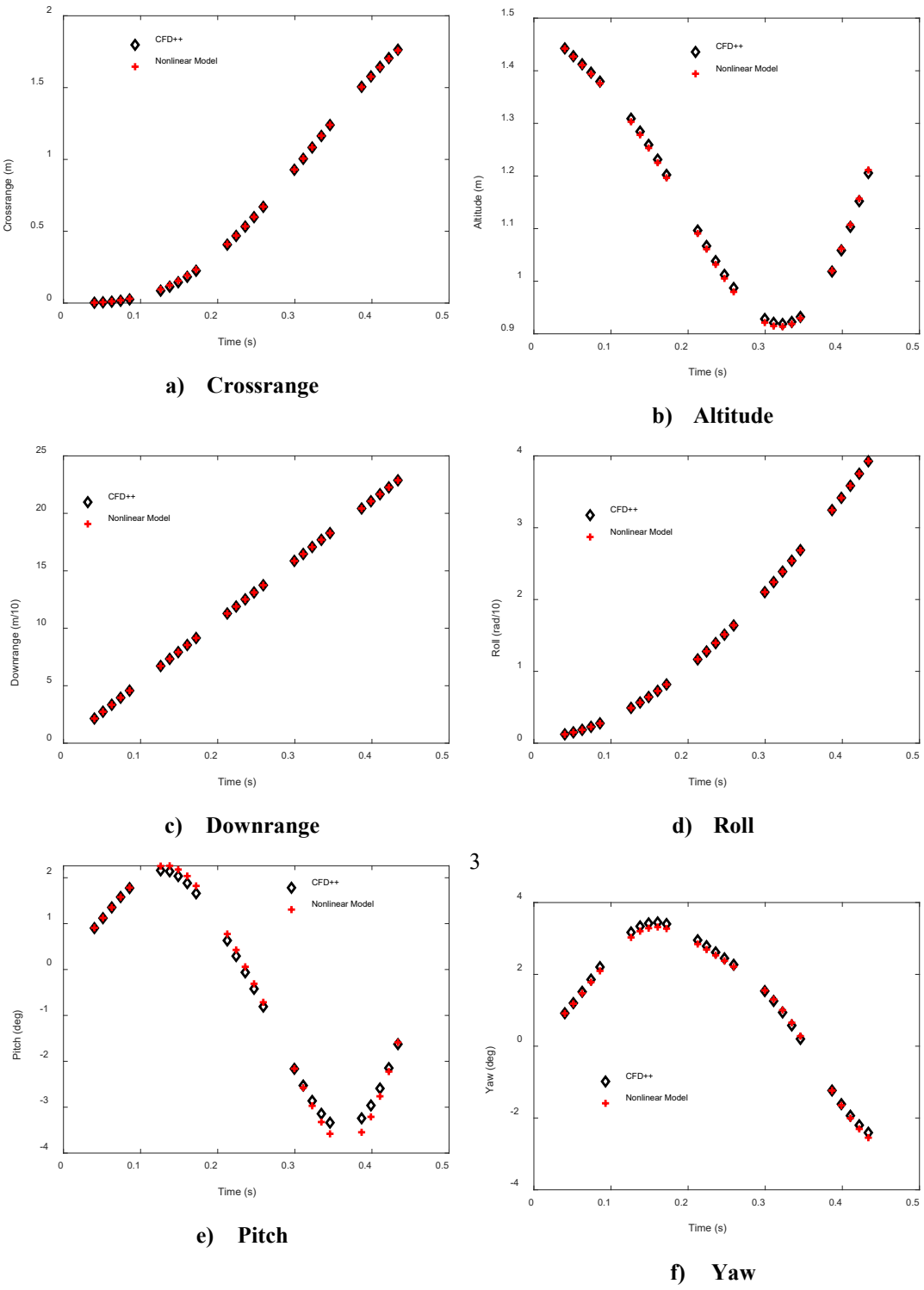


e) Pitch



f) Yaw

Fig. 10 Nonlinear model motion matching result, simulated spark range, V0=270 m/s, known initial conditions



3

Fig. 11 Nonlinear model motion matching result, simulated spark range, V0 = 535 m/s, known initial conditions

3.3 Motion Reconstruction with Simulated Spark-Range Data, Seeding Initial Conditions from Linear Fit

In order to relax the assumption that initial conditions are known, the user must modify the user-defined model such that the unknown initial conditions are part of the parameter vector. SIDPAC can find the additional unknowns if the initial guess is close enough to the actual values. Thus, to seed the nonlinear search, we first apply this method to a linear search, as is done in established spark range reduction methods. The linear model is given in Eqs. 21–30.

$$\dot{y} = V\psi + v \tag{21}$$

$$\dot{z} = V\theta + w \tag{22}$$

$$\dot{\theta} = q \tag{23}$$

$$\dot{\psi} = r \tag{24}$$

$$\dot{v} = -Av - rV \tag{25}$$

$$\dot{w} = -Aw + qV + g \tag{26}$$

$$\dot{q} = \frac{C}{D}w + Eq + Fr \tag{27}$$

$$\dot{r} = -\frac{C}{D}v + Fq + Er \tag{28}$$

$$\dot{V} = -\frac{\bar{q}S}{m}C_{x0} \tag{29}$$

$$\dot{p} = \frac{\bar{q}SD^2}{2I_{XX}V}C_{lp}p + \frac{\bar{q}SD}{I_{XX}}C_{L\delta}\delta \tag{30}$$

where the convenience variables are

$$A = \frac{\bar{q}S}{mV}C_{N\alpha}$$

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

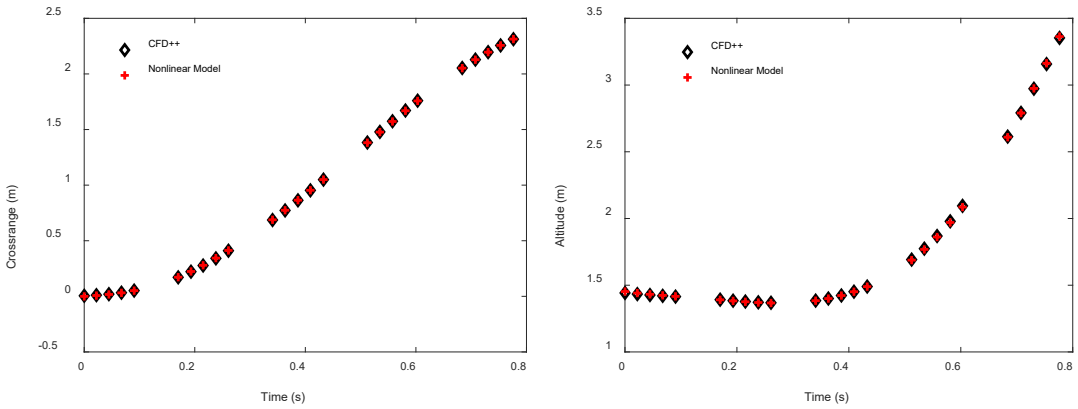
$$E = \frac{\bar{q}SD^2}{2I_{YY}V}C_{mq}$$

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

By seeding the linear model with the values measured at the first spark-range station for $\{x, y, z\ \phi, \theta, \psi\}$, assuming that the muzzle velocity and initial roll rate can be accurately estimated from simple models, and using a uniform random distribution

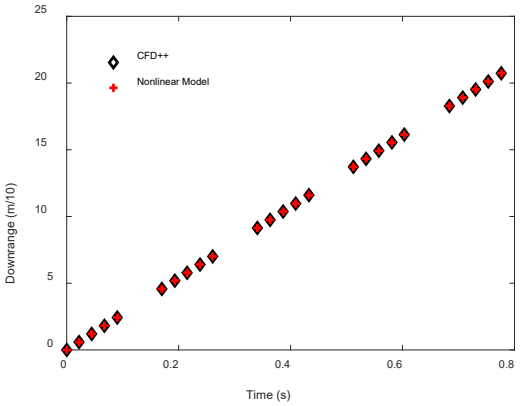
for the epicyclic velocities such that $\{v, w, q, r\} \in [-0.5, 0.5]$, the linear algorithm is able to converge to initial condition guesses that are satisfactory to seed the nonlinear search. The last two rows of Table 1 show the statistics for this method. As long as the linear case converges adequately, this method provides a final prediction error smaller than the other methods. The added flexibility of adjustable initial conditions allows the algorithm to outperform ones where the initial conditions are fixed. Figures 12 and 13 show the quality of match for the four quantities measured in the spark range. In both cases, the model predictions are indistinguishable from the measurements.

Tables 2 and 3 show the coefficients for this method in the column *Output Error (ICs)* in comparison with the previous ones. Since the roll-dependent terms $C_{Y\phi\alpha}$ and $C_{n\phi\alpha}$ had inconsistent values across the previous methods, they were dropped from this analysis. This reduced the number of DOF in the search and allowed for reliable convergence. The remaining coefficients were consistent with previous estimates.

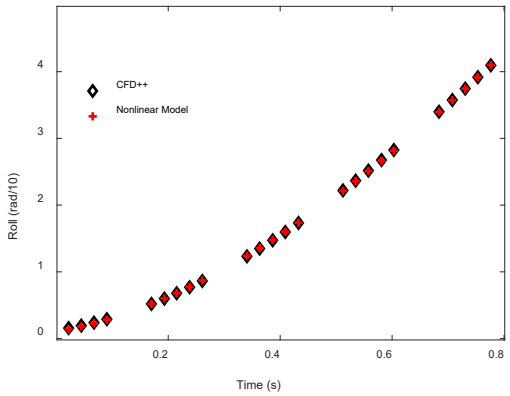


a) Crossrange

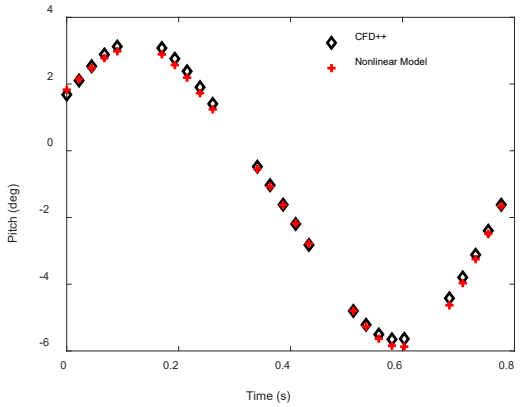
b) Altitude



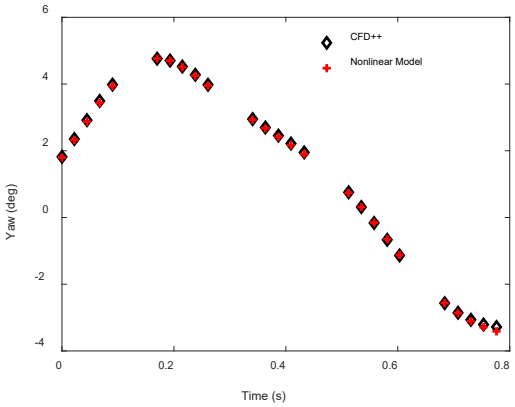
c) Downrange



d) Roll

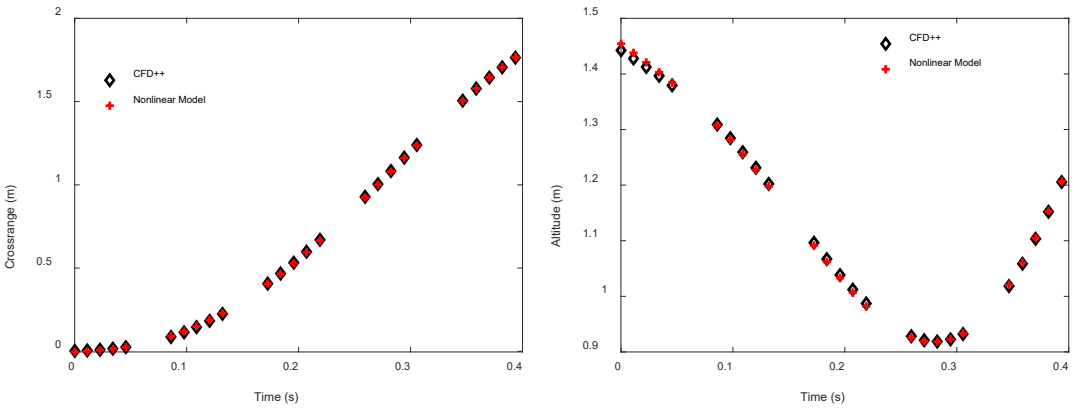


e) Pitch



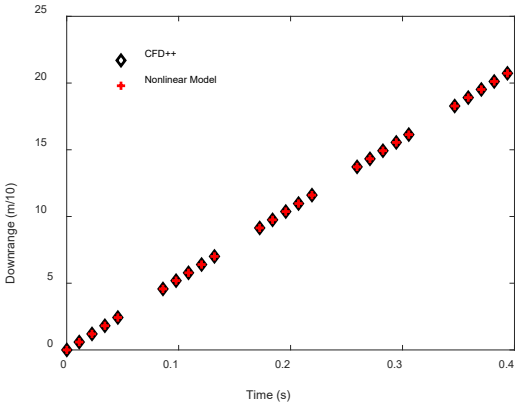
f) Yaw

Fig. 12 Nonlinear model motion matching result with initial conditions estimated by fitting the linear model, V0 = 270 m/s

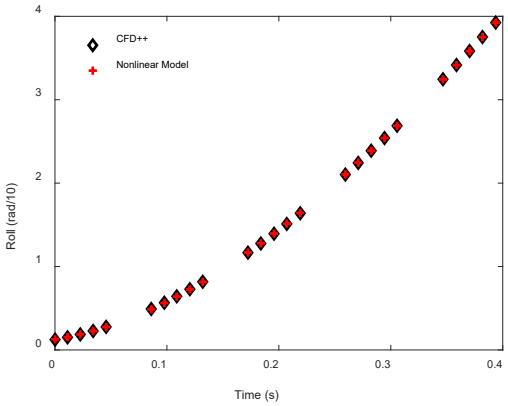


a) Crossrange

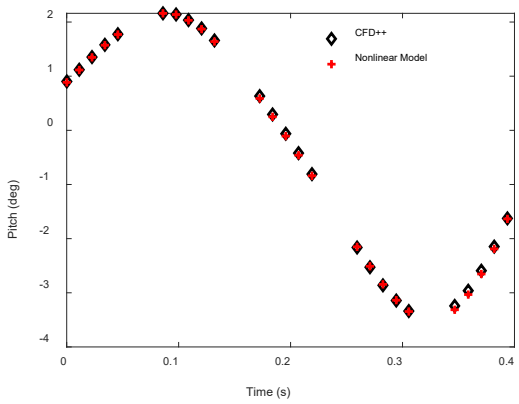
b) Altitude



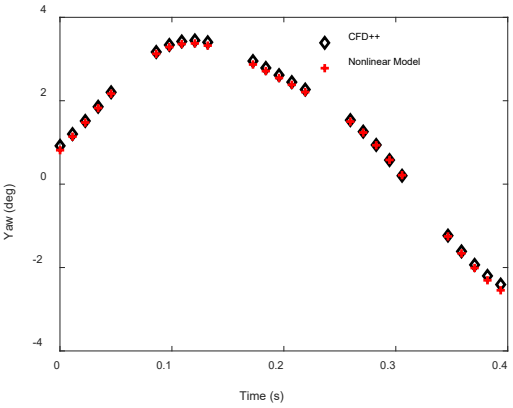
c) Downrange



d) Roll



e) Pitch



f) Yaw

Fig. 13 Nonlinear model motion matching result with initial conditions estimated by fitting the linear model, V0 = 535 m/s

4. Conclusion

This report has explored the application of NASA's SIDPAC toolbox to the reduction of data from CFD++ simulation and spark-range experiments. Spark-range experiments have long been regarded as the most accurate way to estimate aerodynamic coefficients for projectiles. Determining these coefficients differs from the typical application of SIDPAC because the data were much sparser, there is no control input, and initial conditions are unknown.

However, by including the unknown initial conditions in the parameter set, and fitting the linear model first, a successful algorithm has been contrived. Note that the user only needed to write a total of about 400 lines of MATLAB code, not including plotting, to get this result. A full user-defined algorithm that includes numerical integration, finite difference sensitivities, and multiple models would typically require 2000 or more lines of code.

The stepwise regression tool in SIDPAC was very capable of backing out aerodynamic coefficients from CFD simulations that provide the forces, moments, and 12 states of a 6DOF motion model. In this report, we have studied free response only (controls fixed). Additional work should be devoted to determine whether the tool would also serve well for controlled cases.

5. References

1. Morelli EA, Klein, V. Application of system identification to aircraft at NASA Langley Research Center. J Aircraft. 2005;42(1):12–25.

2. Morelli EA, Klein V. Aircraft system identification: theory and practice. Sunflyte Enterprises; 2016.

3. Morelli EA. System identification programs for AirCraft (SIDPAC). Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit; 2002 Aug 5–8; Monterey, CA. Paper AIAA-2002-4704.

4. Metacomp Technologies Inc. CFD++ user manual. Agoura Hills (CA): Metacomp Technologies Inc.; 2014.

5. Costello M, Rogers J. BOOM: A computer-aided engineering tool for exterior ballistics of smart projectiles. Army Research Laboratory (US); 2011 June Report No.: ARL-CR-670.

6. Allen J, Ghoreyshi M. Forced motions design for aerodynamic identification and modeling of a generic missile configuration, Aerospace Sci Tech. 2018;77:742–754.

7. Kutluay Ü, Mahmutyazıcıoğlu G, Platin B. An application of equation error method to aerodynamic model identification and parameter estimation of a gliding flight vehicle. Proceedings of the AIAA Atmospheric Flight Mechanics Conference; 2009 Aug 10–13; Chicago, IL. Paper AIAA-2009-5724.

8. Whyte R, Hathaway W, Winchenbach G. Analysis of free flight trajectory data for a complex asymmetric missile configuration at subsonic Mach numbers. Proceedings of the 5th Atmospheric Flight Mechanics Conference for Future Space Systems; 1979 Aug 6–9; Boulder, CO. Paper AIAA 1979-1689.

List of Symbols, Abbreviations, and Acronyms

ARL	Army Research Laboratory
AOA	angle of attack
CFD	computational fluid dynamics
CG	center of gravity
DEVCOM	US Army Combat Capabilities Development Command
DOF	degrees of freedom
LaRC	Langley Research Center
NASA	National Aeronautics and Space Administration
RBD	rigid body dynamics
SIDPAC	System Identification Programs for AirCraft
TEF	Transonic Experimental Facility
{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]
m	projectile mass [kg]
I	inertia matrix [kg-m2] or identity matrix
V	total velocity $\sqrt{u^2 + v^2 + w^2}$ [m/s]
ρ	atmospheric density [kg/m3]
S	area of projectile cross section [m2]
D	projectile diameter [m]
C_{x0}	zero angle of attack axial force coefficient
$C_{N\alpha}$	normal force due to angle of attack coefficient
$C_{m\alpha}$	pitch moment due to AOA coefficient
$C_{l\delta}\delta$	static roll moment coefficient

C_{mq}	pitch damping coefficient
C_{lp}	roll damping coefficient
$C_{Y\alpha}$	side force coefficient
$C_{n\alpha}$	side moment coefficient
C_{N3}	normal force due to AOA cubed coefficient
C_{m3}	pitch moment due to AOA cubed coefficient
Subscript	
l	roll moment
m	pitch moment
n	yaw moment
X	body axial (+X) force
y	body side (y) force
N	body normal (-Z) force
0	zero AOA term
α	linear in AOA term
3	cubic in AOA term
p	roll damping term
q	pitch damping term
$p\alpha$	Magnus term
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 L STROHM VA BHAGWANDIN J BRYSON B BURCHETT I CELMINS J DESPIRITO LD FAIRFAX B GRUENWALD J PAUL JD VASILE