Longitudinal Flight Characteristics for Mini-Ultrastic UAV

Jyot Buch

April 1, 2019

Abstract

Longitudinal flight characteristics such as static margin, lift curve slope, elevator power, forward and aft travel of the center of gravity for any Unmanned Aerial Vehicle (UAV) are very important considerations for design. Moreover, these characteristics are crucial for verification, validation and certification for safe and reliable civilian use. There are several ways to characterize an aircraft systematically. In this report, an experimental approach is considered in which mini-Ultrastick aircraft mounted on a 6 Degrees-of-Freedom sting mount is placed in a closed return wind tunnel to obtain aerodynamic forces and moments acting on the aircraft body. In this work, the elevator deflection angle was changed to three different values $+18^o$, 0^o, -18^o and angle of attack to be -10^o to 22^o in the increments of 2^o. Stability and control derivatives are obtained from the experimental data which characterizes the stability of an aircraft in a longitudinal flight. Further, to accomplish the mission objective of loitering a 30^o banked turn for 30 minutes at 1000 ft altitude is considered. The practical energy requirement is computed and recommendation on electric motor and appropriate battery pack is made.

Contents

List of Figures

List of Tables

1 Introduction

An Unmanned Aerial Vehicle (UAV), commonly known as a drone, is an aircraft without a human pilot aboard. The flight of UAVs may operate with various degrees of autonomy: either under remote control by a human operator or autonomously by onboard computers. The use of UAV is rapidly expanding to commercial, scientific, recreational, agricultural, and other applications, such as policing, peacekeeping and surveillance, product deliveries, aerial photography, agriculture, smuggling, and drone racing [2]. As civilian use of UAV increases, ensuring user, aircraft & public safety becomes a top priority for the manufacturers and engineers. This is also enforced and required by *Federal Aviation Regulations* (FAR). Among many other metrics, longitudinal flight stability and control is a very important design and certification aspect for civilian UAVs such as the mini-Ultrastick.

Characterization of the longitudinal stability of such an aircraft is critical, and can be obtained by multiple methods. By doing so, important parameters of the aircraft can be derived and assembled into one location, providing clearly defined characteristics of the test body. The goal of this report is to describe the methodology, experimentation, and results of such characterization for the mini-Ultrastick, and to discuss critical design parameters, such as the forward and aft center of gravity limits and corresponding minimum velocities that arise from test data obtained.

Currently, much stability analysis is focused on determining how the aerodynamic forces and moments around a test body interact with the test body itself when subjected to air flow stream. By utilizing dimensionless parameters, scaled down models can be used in relatively small wind tunnels to accurately measure characteristics that can be attributed to varied sizes of aircraft, so long as the dimensionless values remain constant. A relatively new and sophisticated technique for such an analysis is Computational Fluid Dynamics (CFD), which utilizes a finite mesh of elements to create a computer model of the fluid properties. CFD provides numerically estimated values for dimensionless parameters and does not require the traditional linearizion of complex equations of motion. In this experiment however, wind tunnel testing was used to derive these dimensionless values, and is described in the section 2. There are 5 parameters of importance for this experiment, which are summarized bellow for an aircraft body.

- **Angle of Attack (α):** The angle of the body fixed x_B axis with respect to the incoming free-stream flow U_∞. (see Figure 2)
- **Elevator deflection angle (δ_e):** The angle of the elevator with respect to the horizontal tail of the aircraft.
- **Coefficient of Lift (C_L):** Non-dimensionalized lift force (L)
- **Coefficient of Drag (C_D):** Non-dimensionalized drag force (D)
- **Longitudinal Total Moment Coefficient (C_M):** Non-dimensionalized total moment about center of gravity

The α directly affects the measured lift and drag forces, as they are calculated perpendicular and parallel to the incoming flow, yet are measured relative to the angle of attack of the body, thus meaning the forces measured in the experiment must be decomposed into forces along respective directions.

The goal of this experiment is to characterize the longitudinal stability of the mini-Ultrastick through wind tunnel tests. Stability and control derivatives will be determined by creating multiple plots of measured data and analyzing key characteristics from each. Plots of importance will be the C_L vs. α, C_M vs. α, and C_M vs. C_L, as seen in the Figures section with some additional plots to support the discussion. Results of interests are the static margin, SM, pitch stiffness, $C_{M\alpha}$, elevator power, $C_{M\delta_e}$, location of neutral point, x_{NP} and h_{NP}, and the slope of the linear portion of the C_L vs. α curve, C_{L_α}. Additional design parameters will be determined, such as the forward and aft limits of the center of gravity, x_{CG}, and the minimum flight speeds at those limits. The key assumptions made during this work are listed below.

- Available maximum elevator deflection on the mini-Ultrastick is $\pm 18^o$
- The maximum weight of the mini-Ultrastick (airplane + payload) is $W_{max} = W = 0.8$ kg $\times$ 9.81 m/s^2 = 7.848 N

For more details on experiment and notations, the reader is refereed to the wind tunnel lab manual [4].

2 Apparatus and Methods

2.1 Apparatus

The experiment was conducted in a closed return (CR) wind tunnel at the University of Minnesota - Twin Cities at an altitude of 250 m above sea level. The CR tunnel was operated using a program, `crtunnel`, interfaced to a desktop computer which gave operational inputs specified by the user and received data values on the velocity, density and pressure of the free stream flow as well as the normal, axial, and transverse forces on the test body measured by the sting. The test section of the CR tunnel has a cross section of 1×1.25 m, and the tunnel operates by a 100 hp frequency controlled variable speed motor with P-38 feathering propeller, that produces a maximum flow speed of 38 m/s. Setting the velocity of the free stream required setting a frequency value on an interface outside of the tunnel. One free-stream velocity was used for this experiment, and thus only one frequency was used, $f = 22$ Hz, which produced a free stream speed of approximately $U_\infty = 8$ m/s. This velocity was used for all data samples collected. A mini-Ultrastick was placed on a strain gauge balanced sting located within the test section as shown in Figure 1. The sting was capable of measuring axial, normal, and transverse forces as well as their corresponding moments. Force and moment measurements were limited to up to 222 N in any direction and 2.82 Nm to 8.47 Nm dependent on the axis, respectively [3].

For this experiment, only the normal and axial forces, as well as transverse moments, were used in calculations. The sting itself was pre-positioned in the center of the wind tunnel by a lab technician and was not moved throughout the experiment. The only physical orientation that was manipulated after this was the angle of the test body relative to the free stream flow. This value, α, was changed using a US Digital T7-1 inclinometer [3]. The T7 would also report the actual value of α back to the user via a connection to the desktop computer and `crtunnel`. Additionally, the mini-Ultrastick had the capability of adjusting its elevator deflection, δ_e, through three values by use of RC controller. These values were $\delta_e = -18^o$, $\delta_e = 0^o$, and $\delta_e = 18^o$. Velocity and free stream measurements were obtained from a pitot tube located within the wind tunnel that provided values based upon ram air entering it. The values measured were then sent to the prompt and were available for users to see and log the data [4].

2.2 Methods

Experimentation began by ensuring that the mini-Ultrastick was secured within the CR wind tunnel, and that the door was latched properly. Angle of attack, α, was set to an initial value using `crtunnel` software. The sting was tared to ensure no force and moment contributions by the weight of the aircraft itself. Wind tunnel speed was set by increasing the frequency of the propeller to $f = 22$ Hz, which set the free stream velocity to approximately $U_\infty = 8$ m/s. At the set flow speed and α, elevator angle, δ_e, was varied over three values: $\delta_e = -18^o$, $\delta_e = 0^o$, and $\delta_e = 18^o$. At each elevator condition, normal and axial forces, F_N and F_A respectively, as well as the moment about the sting attachment, M_S, were measured by the sting and output by `crtunnel` to the experimenter via a desktop connection. Elevator deflection angles were also verified manually using protractor. It should be noted that the sign convention for sting axes and body axes of the mini-Ultrastick are not the same. Moreover, $F_A = -X$ and $F_N = -Z$, where X and Z are the forces aligned with the body axes.

The process of varying δ_e at different α was completed through a range of $-10^o < \alpha < 22^o$ at increments of 2^o. At each instance, the wind tunnel was turned off and the sting was tared after obtaining the new α. Free-stream velocity U_∞ was brought back to the same value of $U_\infty = 8$ m/s, where δ_e was again varied through $\delta_e = -18^o$, $\delta_e = 0^o$, and $\delta_e = 18^o$. In total, 3 sets of force data corresponding to each elevator deflection were obtained for each α.

In what follows, the equations of interests are listed which are used to generate the plots attached in the Figures section. Lift (L), Drag (D) and the y-axis component of moment about the center of gravity (M_{cg}) [4] are given by,

$$L = Xsin(\alpha) - Zcos(\alpha) \tag{1}$$
$$D = -Xcos(\alpha) - Zsin(\alpha) \tag{2}$$
$$M_{cg} = M_s + z_{cg}X - x_{cg}Z \tag{3}$$

Normalizing the above equations by appropriate quantities in SI units [1], we can obtain the following non-dimensionalized equations,

$$C_L = \frac{2L}{\rho U_\infty^2 S} \tag{4}$$

$$C_D = \frac{2D}{\rho U_\infty^2 S} \tag{5}$$

$$C_M = \frac{2M}{\rho U_\infty^2 S\bar{c}} \tag{6}$$

Static Margin SM can be obtained from the C_M vs C_L plot using the following relation,

$$SM = -\frac{dC_M}{dC_L} = -(h_{CG} - h_{NP}) \tag{7}$$

Where, h_{CG} and h_{NP} are the locations of the center of gravity and neutral point as a fractions of mean chord length ($\bar{c}$), respectively. Elevator power can be obtained using the following equation,

$$C_{M\delta_e} = \frac{dC_M}{d\delta_e} = \frac{C_{M_1} - C_{M_2}}{\delta_{e_1} - \delta_{e_2}} \tag{8}$$

The pitch stiffness is given by the slope of the C_M vs α curve,

$$C_{M_\alpha} = \frac{dC_M}{d\alpha} \tag{9}$$

Similarly, lift curve slope of the linear† region of C_L vs α curve is given by,

$$C_{L_\alpha} = \frac{dC_L}{d\alpha} \tag{10}$$

† Stall conditions will arise at the stall angle of attack, causing the curves to deviate from linearity. This phenomenon is discussed in section 3.

In determining the forward and aft limits of the center of gravity (CG), it is important to understand what is the role of each term contributing to the moment. In both cases, it is desired that the aircraft be trimmable at some elevator deflection. This means that the total moment coefficient be zero, or $C_M = 0$. Likewise, the slope of the C_M vs. α curve must be $C_{M_\alpha} \leq 0$, or else a positive change in angle of attack will cause a positive moment. A final constraint is the slope intercept, C_{M_0}, must be $C_{M_0} \geq 0$, or else the aircraft will not be trimmable at positive α. These terms, in addition to the elevator power term*, can be combined into an equation as follows,

$$C_M = C_{M_0} + (h_{CG} - h_{NP})C_L + C_{M\delta_e}\delta_e \tag{11}$$

*Elevator is a control surface attached to the horizontal tail of an aircraft, and the its contribution to the longitudinal moment is referred to as control derivative or elevator power.

To determine the minimum speed of the aircraft at each forward and aft CG limits, the following equation obtained from trim condition can be used,

$$U_{min} = \sqrt{\frac{2W}{\rho C_{L_{max}} S}} \tag{12}$$

Where, W is the weight of the aircraft in N, $C_{L_{max}}$ is the maximum coefficient of lift attained at the given flight condition, ρ is the air density in kg/m^3, S is the wing area in m^2. It is worth noting that for the following analysis, only the angle of attack and elevator deflection angle were varied keeping the other parameters of experiments constant within experimental accuracy.

3 Results and Discussions

From Figures 3, 4, and 5 it can be seen that, at higher angle of attack ($\alpha \approx 12^o$), the mini-Ultrastick approaches stall limits. This is evident from the error bar plot as the data becomes noisy (inaccurate) due to structural vibrations and nonlinearities [1]. As lower angle of attack results are more accurate, we use the MATLAB command `fit` to obtain the linear/polynomial model approximation as appropriate. Solid lines in the Figure suggest a fit curve. The key results of interest based on Figures [6-10] are summarized in the following table for different values of elevator deflection angle δ_e.

Table 1: Summary of Experimental Results

Characteristics	$\delta_e = -18^o$	$\delta_e = 0^o$	$\delta_e = +18^o$
Static Margin (SM)	0.1255	0.1177	0.1409
Lift Curve Slope (C_{L_α})	0.0873	0.0843	0.0829
Pitch Stiffness (C_{M_α})	-0.0110	-0.0100	-0.0117
Neutral Point (x_{NP}) (inches)	3.1472	3.0822	3.2762
Neutral Point (h_{NP}) (fraction of $\bar{c}$)	0.3755	0.3677	0.3909
Elevator Power ($C_{M\delta_e}$)	-0.0135	-0.0135	-0.0135

In understanding the forward and aft limits of the aircraft, consideration is placed on increasing or decreasing SM, which directly relates to changing the slope of the C_M vs. C_L curve, while also maintaining a stable and trimmable aircraft. To maintain the latter condition, the C_M vs. C_L curve for the mini-Ultrastick must maintain a negative slope, with a positive C_{M_0} for static longitudinal stability of the aircraft.

- For the **forward limit**, the intercept should be as large as possible. This can be done by increasing C_{M_0} to its largest possible value. Obtaining the largest C_{M_0} means flying at the most negative δ_e according to Figure 10, which would be $\delta_e = -18^o$ for the mini-Ultrastick. Equating the Eq. 11 to 0 for trim condition with $C_{L_{max}} \approx 1 = C_L$ at $\alpha \approx 12^o$, $\delta_e = -18^o$ and relevant $C_{M_0} = 0.2691$, we obtain the forward center of gravity travel limit as $h_{cg,FL} = -0.1366$ or $x_{cg,FL} = -0.029$ m or 1.1417 inches. Here the negative sign implies that forward limit is in front of the leading edge. Using Eq. 12 the minimum velocity at this forward limit is obtained as $U_{min,FL} = 8.1559$ m/s.

- For the **aft limit**, if the center of gravity is moved backward beyond the neutral point, then the slope of the curve C_M vs C_L curve will become positive and aircraft will not be statically stable as the static margin becomes negative. Thus aft limit distance will be same as neutral point, $h_{cg,AL} = 0.3755$ or $x_{cg,AL} = 0.0799$ m. This is positive that implies aft limit is after the leading edge. Using Eq. 12 the minimum velocity at this aft limit is obtained as $U_{min,AL} = 7.77$ m/s.

The design mission is to loiter in a $\phi = 20^o$ banked turn at an altitude of $h = 1000$ feet $= 304.08$ m and a speed of $V = 10$ m/s for $t = 30$ minutes. For this operating condition with $\rho_h = 1.347$ kg$/m^3$, the following equations can be used to recommend an electric engine (motor) and required battery pack.

The required lift (L) for trim condition is given by,

$$L = \frac{W}{cos(\phi)} \tag{13}$$

The lift coefficient (C_L) for this lift is thus given by,

$$C_L = \frac{2L}{\rho_h V^2 S} \tag{14}$$

The required angle of attack (α) from obtained linear fit data from C_L vs α curve is,

$$\alpha = \frac{C_L - 0.256}{0.08286} \tag{15}$$

The drag coefficient (C_D) obtained from C_D vs α curve at $\delta_e = 0^o$ is,

$$C_D = 0.0006421\alpha^2 - 0.001363\alpha + 0.05854 \tag{16}$$

Total drag (D) can be computed as,

$$D = \frac{1}{2}\rho_h V^2 S C_D \tag{17}$$

In a trim condition, $T = D$ and thus the power required is,

$$P = T \times V \tag{18}$$

As this requirement is for $t = 30$ minutes duration, the energy requirement is given by,

$$E = P \times t \tag{19}$$

Considering the efficiency of the motor with battery as $\eta = 80\% = 0.8$, energy requirement now becomes E/η, thus if 12V battery is used then mAh current requirement becomes,

$$I_{mAh} = \frac{1000E}{12\eta} \tag{20}$$

Considering the power requirement, weight, durability and cost the Zippy flightmax 2200 mAh 3S1P battery with Johnson RS-775 12V high speed electric DC motor is recommended.

4 Summary

In this report, wind tunnel test data for mini-Ultrastick UAV were analyzed. Flight stability and control derivatives were obtained for longitudinal flight. The forward and aft center of gravity limits were calculated to maintain the stable flight envelope. The static margin was found to be positive for the aircraft design which suggests that mini-Ultrastick is statically stable for longitudinal flight. That means in a trimmed longitudinal flight, if an aircraft is perturbed then it will have a tendency to come back to the trim condition. The possible sources of uncertainty are sting measurements, elevator angle values as well as assumed center of gravity locations and gross weight. Moreover, at higher angle of attack the nonlinearities and structural vibrations contributes to the uncertainty. Ultimately, characterization of the mini-Ultrastick has allowed important values and derivatives for the aircraft to be determined, which can now be used for safe development of future mini-Ultrasticks, and further testing on the flight envelopes of the vehicles.

References

[1] Robert C Nelson et al. *Flight stability and automatic control*, volume 2. WCB/McGraw Hill New York, 1998.

[2] Unmanned aerial vehicle wikipedia, `https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle`.

[3] Wind tunnel lab information, `https://www.aem.umn.edu/facilities/windtunnel/`.

[4] Wind tunnel lab manual, aem 4303, spring 2019.

Figures

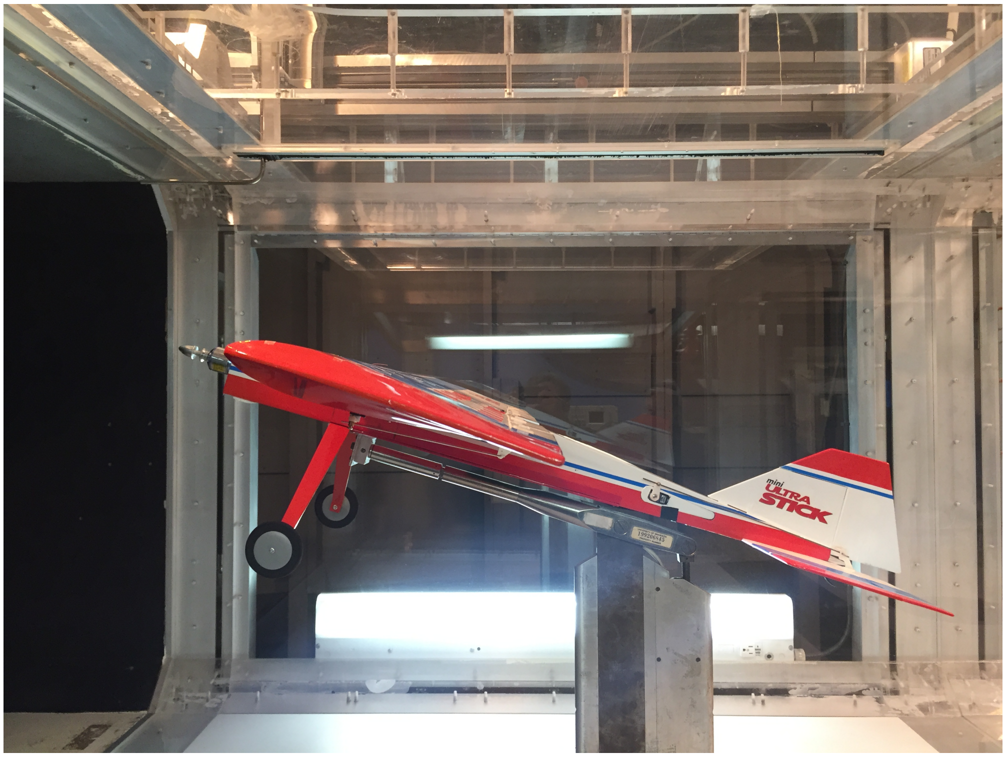

Figure 1: Experimental Setup

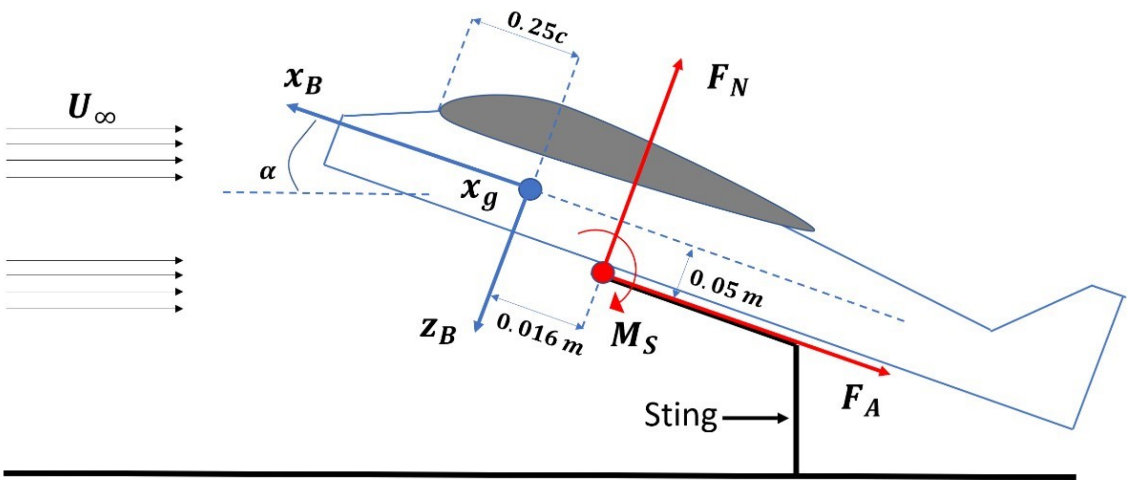

Figure 2: Body Fixed Coordinate System & Schematics

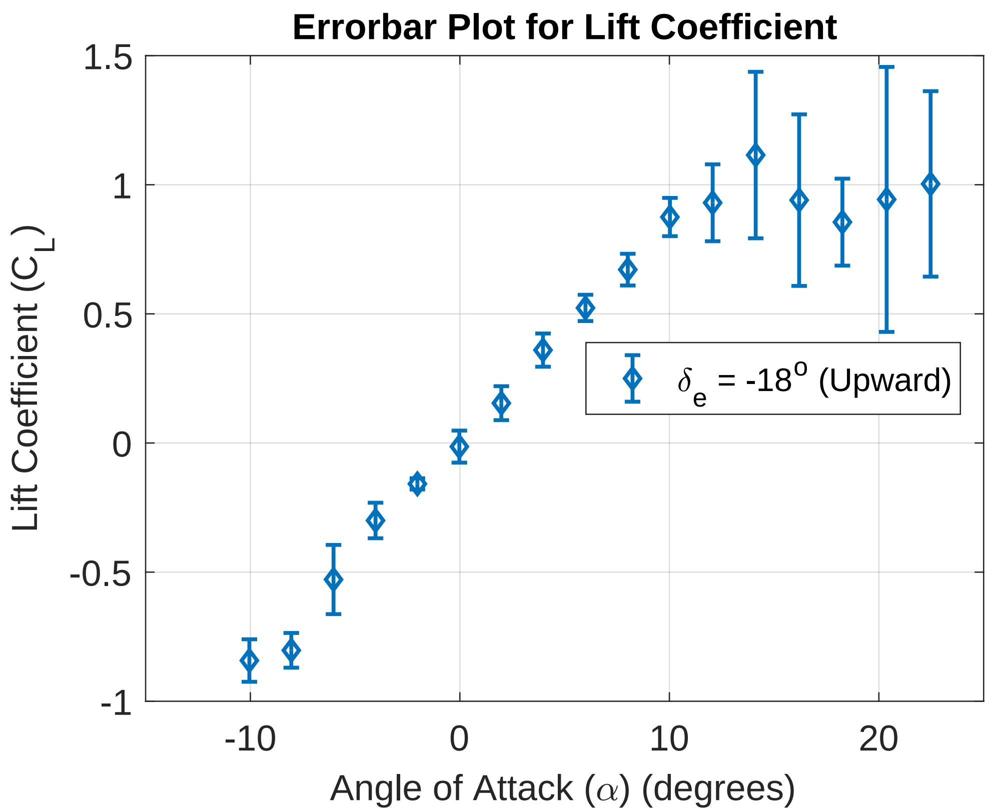

Figure 3: Errorbar Plot for Lift Coefficient

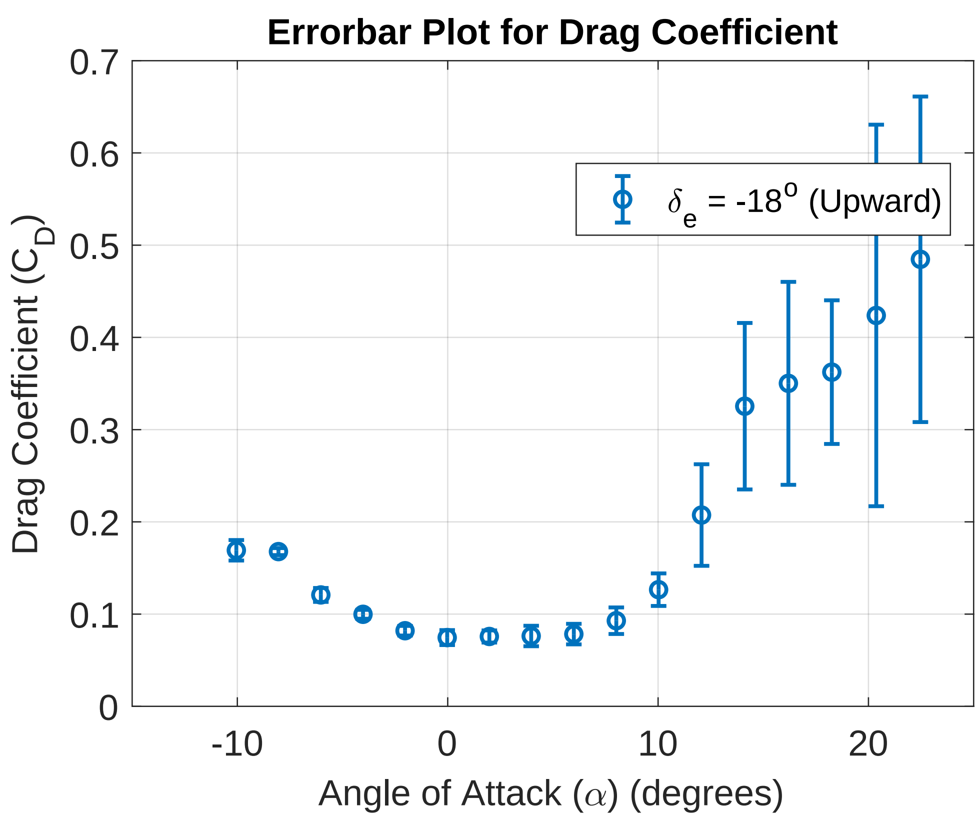

Figure 4: Errorbar Plot for Drag Coefficient

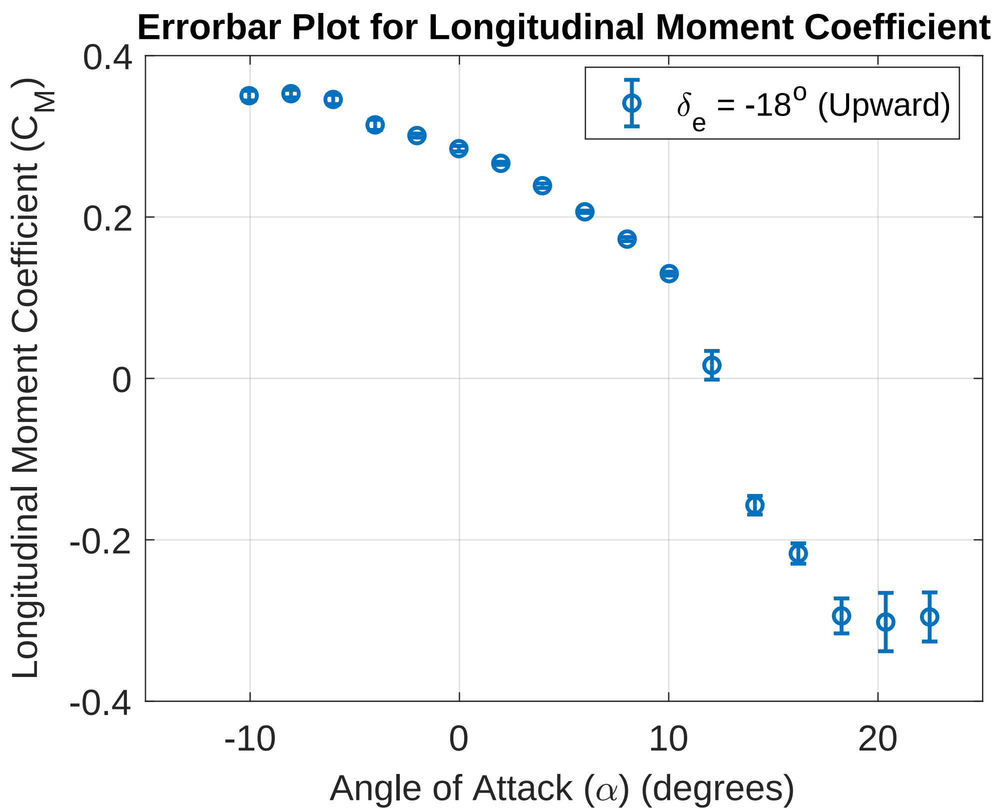

Figure 5: Errorbar Plot for Longitudinal Moment Coefficient

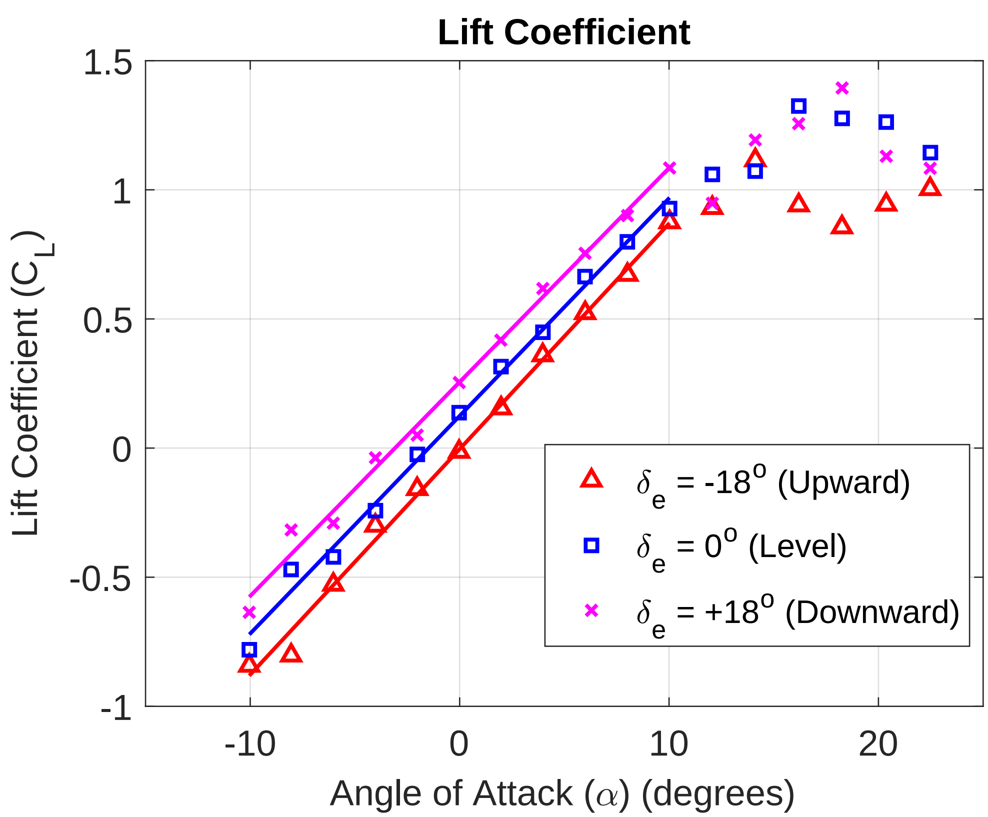

Figure 6: Lift Coefficient vs Anlge of Attack

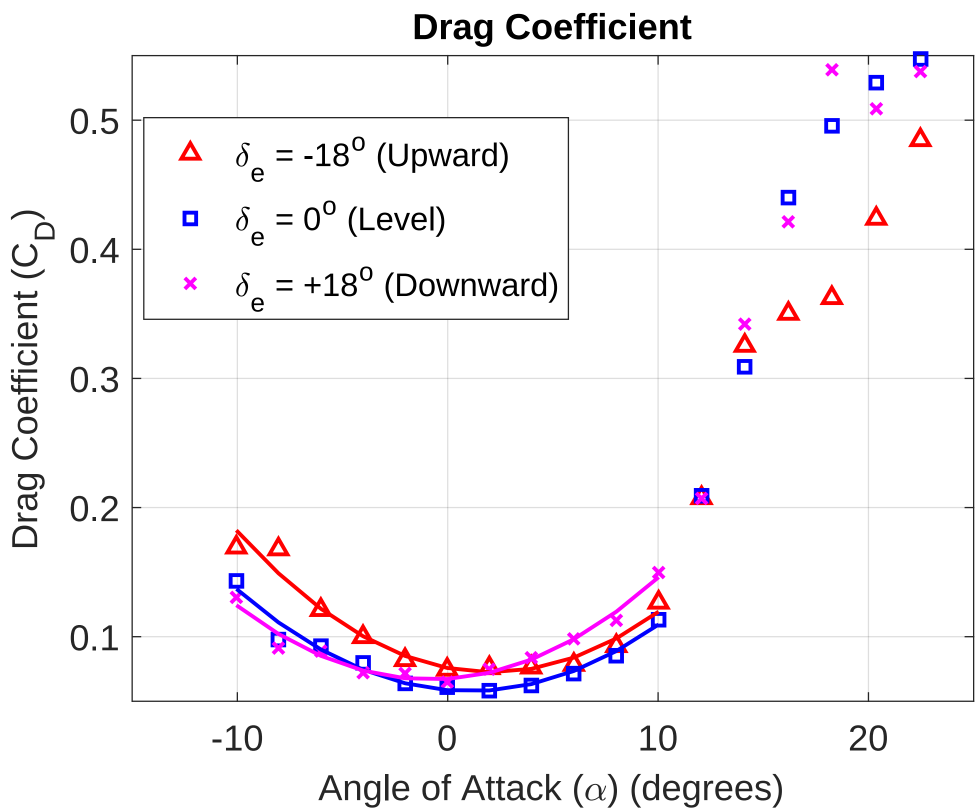

Figure 7: Drag Coefficient vs Angle of Attack

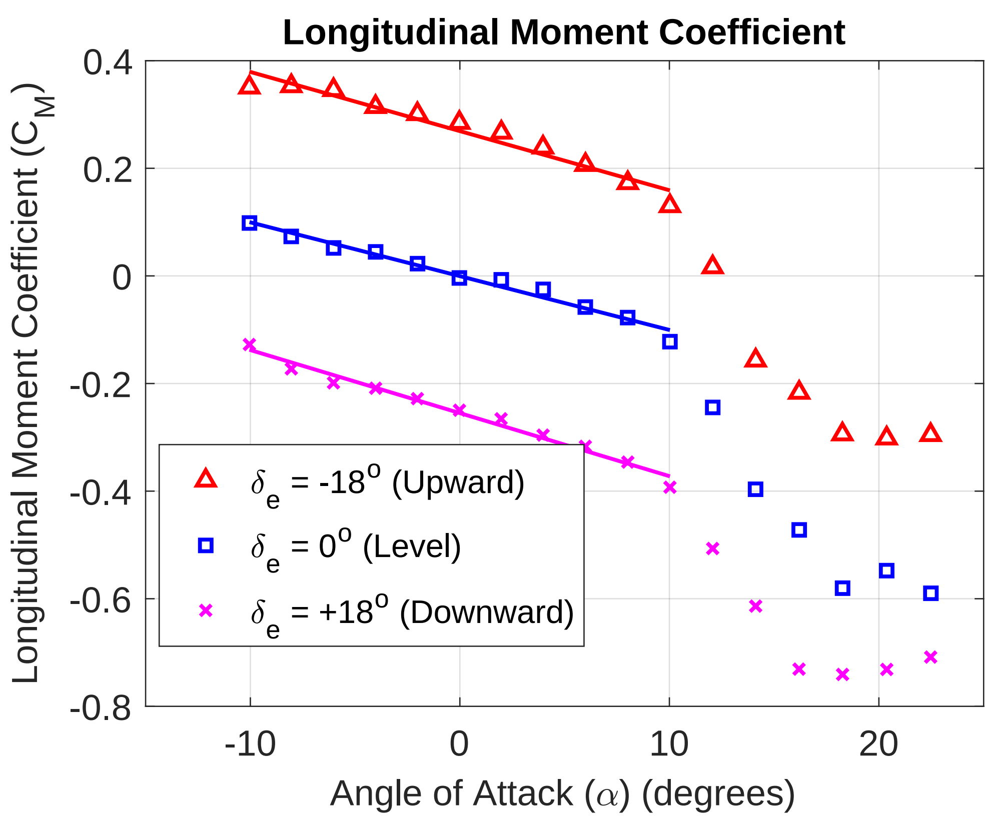

Figure 8: Longitudinal Moment Coefficient vs Angle of Attack

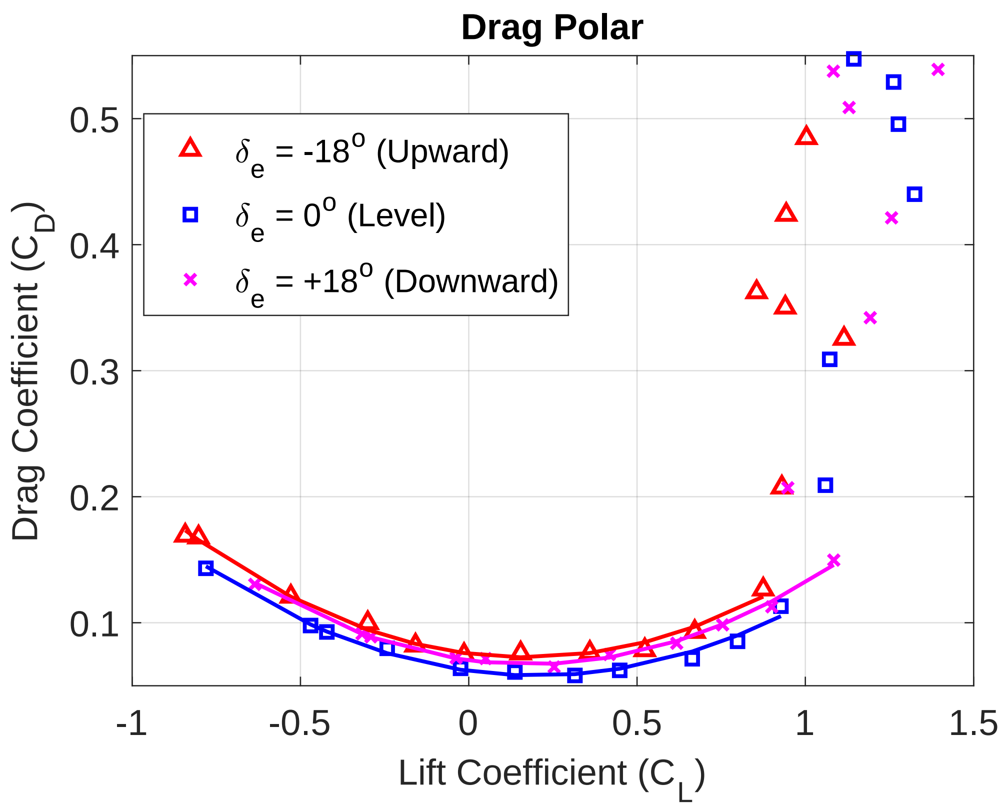

Figure 9: Drag Polar

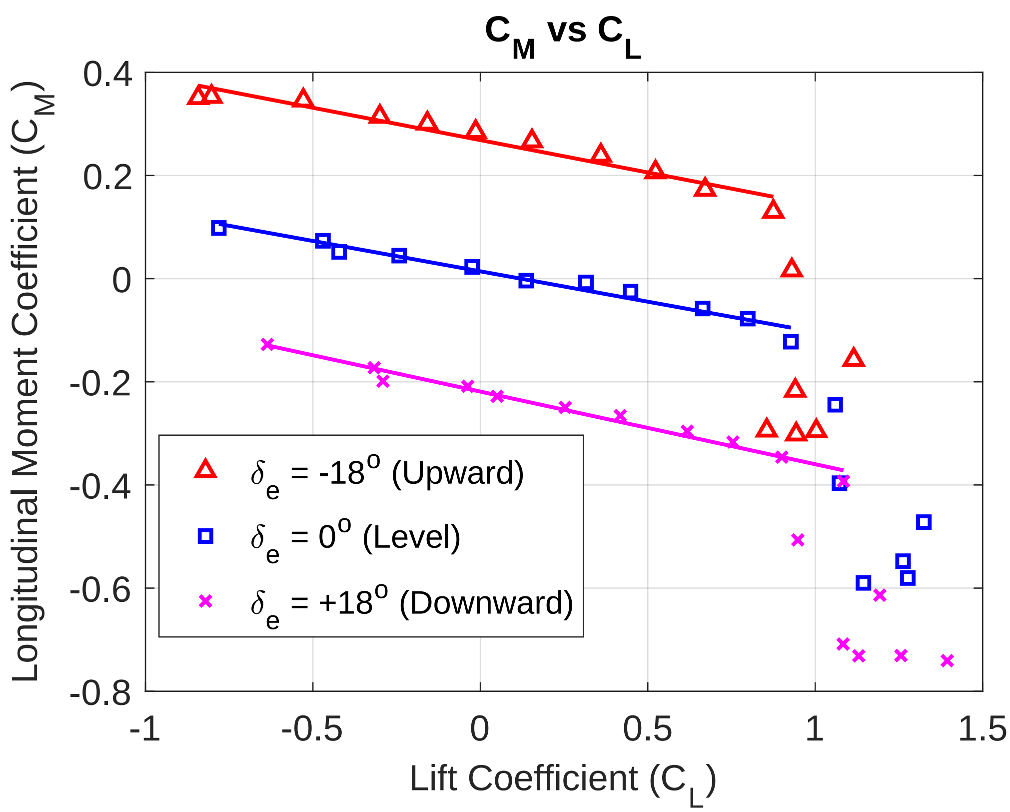

Figure 10: Longitudinal Moment Coefficient vs Lift Coefficient

Tunnel Data Analysis

Table of Contents

This file runs a CRT_data_parser_2019 file and plots the relevant data

Look for the MAT file generated by CRT_data_parser_2019.m file

```
MATFile = dir('CRT_data_*.mat');
if isempty(MATFile)
    try
        delete('*.mat');
        CRT_data_parser_2019
        clearvars -except data_matrix % clear everthing but the
 data_matrix we want
    catch ME
        throw(ME);
    end
end
```

Data Parsing

```
Column  1:      Row number
Column  2:      Angle of attack (rad)
Column  3:      Elevator deflection (rad)
Column  4:      Rudder deflection (rad)
Column  5:      Air density (kg/m^3)
Column  6:      Air speed (m/s)
Column  7:      Normal force (N)
Column  8:      Standard deviation of normal force (N)
Column  9:      Transverse force (N)
Column 10:      Standard deviation of transverse force (N)
Column 11:      Axial force (N)
```

```
Column 12:      Standard deviation of axial force (N)
Column 13:      Normal Moment (N-m)
Column 14:      Standard deviation of normal moment (N-m)
Column 15:      Transverse moment (N-m)
Column 16:      Standard deviation of transverse moment (N-m)
Column 17:      Axial moment (N-m)
Column 18:      Standard deviation of axial moment (N-m)

% Load MAT file
load(MATFile.name);

% Generate variables from data_matrix
rowN   = data_matrix(:,1);
alpha  = data_matrix(:,2); % Angle of attack
deltaE = data_matrix(:,3);
% We think we did experiment for +-15 degrees but as the lab manuel
 suggest
% we had +-18 maximum limits for deltaE, so modify the data
 accordingly
deltaE(deltaE==15) = 18;
deltaE(deltaE==-15) = -18;
deltaR = data_matrix(:,4);
rho    = data_matrix(:,5); % Air density
Vinf   = data_matrix(:,6); % Air velocity

Z      = -data_matrix(:,7); % Body-defined normal force
SdZ    = data_matrix(:,8);

FT     = data_matrix(:,9);
SdFT   = data_matrix(:,10);

X      = -data_matrix(:,11); % Body-defined axial force
SdX    = data_matrix(:,12);

MN     = data_matrix(:,13);
SdMN   = data_matrix(:,14);

Ms     = data_matrix(:,15); % Sting moment
SdMs   = data_matrix(:,16);

MA     = data_matrix(:,17);
SdMA   = data_matrix(:,18);

alpha = deg2rad(alpha);

% deltaEArray
deltaEArray = [18,0,-18];
```

Define Geomatric Parameters of an Aircraft

Mini Ultrastick model in the University of Minnesota closed-return wind tunnel.

```
xcg = -1.6*1e-2;            % [m]
zcg = 5*1e-2;               % [m]
```

```
cbar = 21.29*1e-2;          % Mean Aerodynamic Chord length [m]
bLE = 90*1e-2;              % Leading Edge Span [cm]
bTE = 98.50*1e-2;           % Trailing Edge Span [cm]
S = cbar*(bLE+bTE)/2;       % Wing Area [m^2]
deltaEmax = deg2rad(18);    % Trailing edge down [rad]
deltaEmin = deg2rad(-18);   % Trailing edge up [rad]
xMCCG = 1.6*1e-2;           % [m]
xac = 0.25*cbar;            % Aerodynamic Center [m]
alphamin = deg2rad(-10);
alphamax = deg2rad(22);
```

Priliminary Results

```
for i = 1:length(deltaEArray)
    % Get indexes for specific elevator deflection
    id = (deltaE==deltaEArray(i));
    Q = rho(id).*(Vinf(id).^2)/2;

    % Calculate the lift (L), drag (D) and the y-axis component of
 moment about
    % the center of gravity (Mcg) using the following:
    L = X(id).*sin(alpha(id)) - Z(id).*cos(alpha(id));
    D = -X(id).*cos(alpha(id)) - Z(id).*sin(alpha(id));
    Mcg = Ms(id) + zcg*X(id) - xcg*Z(id);

    % Propogate Standard Deviation
    SdL = SdX(id).*sin(alpha(id)) - SdZ(id).*cos(alpha(id));
    SdD = -SdX(id).*cos(alpha(id)) - SdZ(id).*sin(alpha(id));
    SdMcg = SdMs(id) + zcg*SdX(id) - xcg*SdZ(id);

    % AngleOfAttack
    AoA(:,i) = rad2deg(alpha(id));

    % Calculate the lift coefficient (CL); the drag coefficient (CD);
 the
    % lateral moment coefficient (Cl); the longitudinal moment
 coefficient
    % (CM); and the directional moment coefficient (CN) as follows
    CL(:,i) = L./(Q*S);
    CD(:,i) = D./(Q*S);
    CM(:,i) = Mcg./(Q*S*cbar);

    % Propogate Standard Deviation
    SdCL(:,i) = SdL./(Q*S);
    SdCD(:,i) = SdD./(Q*S);
    SdCM(:,i) = SdMcg./(Q*S);
end
```

Tunnel-Data-Plotting

NOTE: For plotting purpose (legend) we use the convention as shown in textbook as +ve delta_e for downward deflection and negative delta_e for upward elevator deflection. However in the experimental data we used opposite convention.

ErrorBar Plot

CL, CD, CM vs alpha

```
figure(1);errorbar(AoA(:,1),CL(:,1),SdCL(:,1),'d');
ylabel('Lift Coefficient (C_L)');
xlabel('Angle of Attack (\alpha) (degrees)');
title('Errorbar Plot for Lift Coefficient');
legend('\delta_e = -18^o (Upward)');

figure(2);errorbar(AoA(:,1),CD(:,1),SdCD(:,1),'o');
ylabel('Drag Coefficient (C_D)');
xlabel('Angle of Attack (\alpha) (degrees)');
title('Errorbar Plot for Drag Coefficient');
legend('\delta_e = -18^o (Upward)');

figure(3);errorbar(AoA(:,1),CM(:,1),SdCM(:,1),'o');
ylabel('Longitudinal Moment Coefficient (C_M)');
xlabel('Angle of Attack (\alpha) (degrees)');
title('Errorbar Plot for Longitudinal Moment Coefficient');
legend('\delta_e = -18^o (Upward)');
```

CL vs alpha

```
fprintf('===========================================\n');
figure(4);clf;hold on;Np=1:1:11;
plot(AoA(:,1),CL(:,1),'^','MarkerEdgeColor','r');
plot(AoA(:,2),CL(:,2),'s','MarkerEdgeColor','b');
plot(AoA(:,3),CL(:,3),'x','MarkerEdgeColor','m');
f1CL = fit(AoA(Np,1),CL(Np,1),'poly1');disp(f1CL);
f2CL = fit(AoA(Np,2),CL(Np,2),'poly1');disp(f2CL);
f3CL = fit(AoA(Np,3),CL(Np,3),'poly1');disp(f3CL);
plot(AoA(Np,1),f1CL(AoA(Np,1)),'-r');
plot(AoA(Np,2),f2CL(AoA(Np,2)),'-b');
plot(AoA(Np,3),f3CL(AoA(Np,3)),'-m');
ylabel('Lift Coefficient (C_L)');
xlabel('Angle of Attack (\alpha) (degrees)');
title('Lift Coefficient');
legend('\delta_e = -18^o (Upward)','\delta_e = 0^o (Level)','\delta_e
 = +18^o (Downward)');
hold off;grid on;
```

```
===========================================
     Linear model Poly1:
     f1CL(x) = p1*x + p2
     Coefficients (with 95% confidence bounds):
       p1 =     0.08732  (0.08284, 0.09179)
       p2 =    -0.00394  (-0.03232, 0.02444)
     Linear model Poly1:
     f2CL(x) = p1*x + p2
     Coefficients (with 95% confidence bounds):
       p1 =     0.08428  (0.07958, 0.08897)
```

```
  p2 =      0.1245  (0.09472, 0.1542)
Linear model Poly1:
f3CL(x) = p1*x + p2
Coefficients (with 95% confidence bounds):
  p1 =     0.08286  (0.07793, 0.0878)
  p2 =       0.256  (0.2247, 0.2873)
```

CD vs alpha

```
fprintf('===========================================\n');
figure(5);clf;hold on;Np=1:1:11;
plot(AoA(:,1),CD(:,1),'^','MarkerEdgeColor','r');
plot(AoA(:,2),CD(:,2),'s','MarkerEdgeColor','b');
plot(AoA(:,3),CD(:,3),'x','MarkerEdgeColor','m');
f1CD = fit(AoA(Np,1),CD(Np,1),'poly2');disp(f1CD);
f2CD = fit(AoA(Np,2),CD(Np,2),'poly2');disp(f2CD);
f3CD = fit(AoA(Np,3),CD(Np,3),'poly2');disp(f3CD);
plot(AoA(Np,1),f1CD(AoA(Np,1)),'-r');
plot(AoA(Np,2),f2CD(AoA(Np,2)),'-b');
plot(AoA(Np,3),f3CD(AoA(Np,3)),'-m');
legend('\delta_e = -18^o (Upward)','\delta_e = 0^o (Level)','\delta_e
 = +18^o (Downward)');
ylabel('Drag Coefficient (C_D)');
xlabel('Angle of Attack (\alpha) (degrees)');
title('Drag Coefficient');
hold off;grid on;

===========================================
     Linear model Poly2:
     f1CD(x) = p1*x^2 + p2*x + p3
     Coefficients (with 95% confidence bounds):
       p1 =    0.000746  (0.0005685, 0.0009234)
       p2 =   -0.003122  (-0.004117, -0.002126)
       p3 =     0.07571  (0.06617, 0.08524)
     Linear model Poly2:
     f2CD(x) = p1*x^2 + p2*x + p3
     Coefficients (with 95% confidence bounds):
       p1 =   0.0006421  (0.0005249, 0.0007593)
       p2 =   -0.001363  (-0.00202, -0.0007047)
       p3 =     0.05854  (0.05224, 0.06483)
     Linear model Poly2:
     f3CD(x) = p1*x^2 + p2*x + p3
     Coefficients (with 95% confidence bounds):
       p1 =   0.0006751  (0.0005636, 0.0007866)
       p2 =    0.001094  (0.0004678, 0.00172)
       p3 =     0.06723  (0.06124, 0.07322)
```

CM vs alpha

```
fprintf('===========================================\n');
figure(6);clf;hold on;Np=1:1:11;
plot(AoA(:,1),CM(:,1),'^','MarkerEdgeColor','r');
plot(AoA(:,2),CM(:,2),'s','MarkerEdgeColor','b');
```

```
plot(AoA(:,3),CM(:,3),'x','MarkerEdgeColor','m');
f1CM = fit(AoA(Np,1),CM(Np,1),'poly1');disp(f1CM);
f2CM = fit(AoA(Np,2),CM(Np,2),'poly1');disp(f2CM);
f3CM = fit(AoA(Np,3),CM(Np,3),'poly1');disp(f3CM);
plot(AoA(Np,1),f1CM(AoA(Np,1)),'-r');
plot(AoA(Np,2),f2CM(AoA(Np,2)),'-b');
plot(AoA(Np,3),f3CM(AoA(Np,3)),'-m');
legend('\delta_e = -18^o (Upward)','\delta_e = 0^o (Level)','\delta_e
 = +18^o (Downward)');
ylabel('Longitudinal Moment Coefficient (C_M)');
xlabel('Angle of Attack (\alpha) (degrees)');
title('Longitudinal Moment Coefficient');
hold off;grid on;
```

```
===========================================
     Linear model Poly1:
     f1CM(x) = p1*x + p2
     Coefficients (with 95% confidence bounds):
       p1 =    -0.01099  (-0.01288, -0.009107)
       p2 =      0.2691  (0.2572, 0.2811)
     Linear model Poly1:
     f2CM(x) = p1*x + p2
     Coefficients (with 95% confidence bounds):
       p1 =   -0.009998  (-0.01115, -0.008847)
       p2 =  -0.0004255  (-0.007726, 0.006875)
     Linear model Poly1:
     f3CM(x) = p1*x + p2
     Coefficients (with 95% confidence bounds):
       p1 =    -0.01172  (-0.01294, -0.0105)
       p2 =      -0.255  (-0.2627, -0.2472)
```

CD vs CL

```
fprintf('===========================================\n');
figure(7);clf;hold on;Np=1:1:11;
plot(CL(:,1),CD(:,1),'^','MarkerEdgeColor','r');
plot(CL(:,2),CD(:,2),'s','MarkerEdgeColor','b');
plot(CL(:,3),CD(:,3),'x','MarkerEdgeColor','m');
f1CLCD = fit(CL(Np,1),CD(Np,1),'poly2');disp(f1CLCD);
f2CLCD = fit(CL(Np,2),CD(Np,2),'poly2');disp(f2CLCD);
f3CLCD = fit(CL(Np,3),CD(Np,3),'poly2');disp(f3CLCD);
plot(CL(Np,1),f1CLCD(CL(Np,1)),'-r');
plot(CL(Np,2),f2CLCD(CL(Np,2)),'-b');
plot(CL(Np,3),f3CLCD(CL(Np,3)),'-m');
legend('\delta_e = -18^o (Upward)','\delta_e = 0^o (Level)','\delta_e
 = +18^o (Downward)');
ylabel('Drag Coefficient (C_D)');
xlabel('Lift Coefficient (C_L)');
title('Drag Polar');
hold off;grid on;
```

```
===========================================
     Linear model Poly2:
     f1CLCD(x) = p1*x^2 + p2*x + p3
```

```
     Coefficients (with 95% confidence bounds):
       p1 =     0.09779  (0.08681, 0.1088)
       p2 =    -0.03381  (-0.03928, -0.02834)
       p3 =     0.07545  (0.07091, 0.07999)
     Linear model Poly2:
     f2CLCD(x) = p1*x^2 + p2*x + p3
     Coefficients (with 95% confidence bounds):
       p1 =     0.08969  (0.07799, 0.1014)
       p2 =    -0.03631  (-0.04235, -0.03026)
       p3 =     0.06173  (0.05728, 0.06619)
     Linear model Poly2:
     f3CLCD(x) = p1*x^2 + p2*x + p3
     Coefficients (with 95% confidence bounds):
       p1 =     0.09671  (0.08925, 0.1042)
       p2 =    -0.03508  (-0.04019, -0.02998)
       p3 =     0.07015  (0.06749, 0.07282)
```

CM vs CL

```
fprintf('===========================================\n');
figure(8);clf;hold on;Np=1:1:11;
plot(CL(:,1),CM(:,1),'^','MarkerEdgeColor','r');
plot(CL(:,2),CM(:,2),'s','MarkerEdgeColor','b');
plot(CL(:,3),CM(:,3),'x','MarkerEdgeColor','m');
f1CLCM = fit(CL(Np,1),CM(Np,1),'poly1');disp(f1CLCM);
f2CLCM = fit(CL(Np,2),CM(Np,2),'poly1');disp(f2CLCM);
f3CLCM = fit(CL(Np,3),CM(Np,3),'poly1');disp(f3CLCM);
plot(CL(Np,1),f1CLCM(CL(Np,1)),'-r');
plot(CL(Np,2),f2CLCM(CL(Np,2)),'-b');
plot(CL(Np,3),f3CLCM(CL(Np,3)),'-m');
legend('\delta_e = -18^o (Upward)','\delta_e = 0^o (Level)','\delta_e
 = +18^o (Downward)');
ylabel('Longitudinal Moment Coefficient (C_M)');
xlabel('Lift Coefficient (C_L)');
title('C_M vs C_L');
hold off;grid on;

===========================================
     Linear model Poly1:
     f1CLCM(x) = p1*x + p2
     Coefficients (with 95% confidence bounds):
       p1 =     -0.1255  (-0.1475, -0.1034)
       p2 =      0.2686  (0.2564, 0.2809)
     Linear model Poly1:
     f2CLCM(x) = p1*x + p2
     Coefficients (with 95% confidence bounds):
       p1 =     -0.1177  (-0.134, -0.1015)
       p2 =     0.01423  (0.005285, 0.02318)
     Linear model Poly1:
     f3CLCM(x) = p1*x + p2
     Coefficients (with 95% confidence bounds):
       p1 =     -0.1409  (-0.1561, -0.1256)
       p2 =     -0.2189  (-0.2279, -0.21)
```

Table Data

```
fprintf('===========================================\n');
fprintf('### Table Data:\n');
fprintf(newline);

fprintf('Static Margin (-dCM/dCL):\n');
fprintf('Elevator Deflection = -18 deg, %.4f\n',-f1CLCM.p1);
fprintf('Elevator Deflection =   0 deg, %.4f\n',-f2CLCM.p1);
fprintf('Elevator Deflection = +18 deg, %.4f\n',-f3CLCM.p1);
fprintf(newline);

fprintf('Slope of CL vs Alpha Curve:\n');
fprintf('Elevator Deflection = -18 deg, %.4f\n',f1CL.p1);
fprintf('Elevator Deflection =   0 deg, %.4f\n',f2CL.p1);
fprintf('Elevator Deflection = +18 deg, %.4f\n',f3CL.p1);
fprintf(newline);

fprintf('Slope of CM vs Alpha Curve:\n');
fprintf('Elevator Deflection = -18 deg, %.4f\n',f1CM.p1);
fprintf('Elevator Deflection =   0 deg, %.4f\n',f2CM.p1);
fprintf('Elevator Deflection = +18 deg, %.4f\n',f3CM.p1);
fprintf(newline);

fprintf('Neutral Point (as fraction of chord length):\n');
fprintf('Elevator Deflection = -18 deg, %.4f\n',0.25-f1CLCM.p1);
fprintf('Elevator Deflection =   0 deg, %.4f\n',0.25-f2CLCM.p1);
fprintf('Elevator Deflection = +18 deg, %.4f\n',0.25-f3CLCM.p1);
fprintf(newline);

fprintf('Neutral Point (inches):\n');
fprintf('Elevator Deflection = -18 deg, %.4f\n',(0.25-
f1CLCM.p1)*21.29*0.3937);
fprintf('Elevator Deflection =   0 deg, %.4f\n',(0.25-
f2CLCM.p1)*21.29*0.3937);
fprintf('Elevator Deflection = +18 deg, %.4f\n\n',(0.25-
f3CLCM.p1)*21.29*0.3937);

CMdeltaE = (f1CLCM(0)-f3CLCM(0))/((-18)-18);
fprintf('Elevator Power: %.4f\n',CMdeltaE);

deltaEforPositiveIntercept = 18-(f3CLCM(0)/CMdeltaE);
fprintf('Any elevator angle less then %.4f deg will \nresult in
 positive intercept of CM vs CL curve.\n',deltaEforPositiveIntercept);
```

```
===========================================
### Table Data:

Static Margin (-dCM/dCL):
Elevator Deflection = -18 deg, 0.1255
Elevator Deflection =   0 deg, 0.1177
Elevator Deflection = +18 deg, 0.1409
```

```
Slope of CL vs Alpha Curve:
Elevator Deflection = -18 deg, 0.0873
Elevator Deflection =   0 deg, 0.0843
Elevator Deflection = +18 deg, 0.0829

Slope of CM vs Alpha Curve:
Elevator Deflection = -18 deg, -0.0110
Elevator Deflection =   0 deg, -0.0100
Elevator Deflection = +18 deg, -0.0117

Neutral Point (as fraction of chord length):
Elevator Deflection = -18 deg, 0.3755
Elevator Deflection =   0 deg, 0.3677
Elevator Deflection = +18 deg, 0.3909

Neutral Point (inches):
Elevator Deflection = -18 deg, 3.1472
Elevator Deflection =   0 deg, 3.0822
Elevator Deflection = +18 deg, 3.2762

Elevator Power: -0.0135
Any elevator angle less then 1.8350 deg will
result in positive intercept of CM vs CL curve.
```

Battery Requirement

```
V = 10;
t = 30/60;
h = 304.08;
rhoh = 1.347;
m = 0.8;
g = 9.81;
W = m*g;
phi = 20;
eta = 0.8;

Lreq = m*g/cosd(phi);
CLreq = 2*Lreq/(rhoh*V^2*S);
alphareq = (CLreq - f2CL.p2)/f2CL.p1;
CDreq = f2CD(alphareq);
Treq = 0.5*rhoh*V^2*S*CDreq;
P = Treq*V;
Ereq = P*t/eta;
I_mAh = Ereq/12*1000;

fprintf('Current Requirements (I_mAh):');
disp(I_mAh);
```

```
Current Requirements (I_mAh):  510.8476
```

Save Plots

```
fprintf('===========================================\n');
```

```
if exist('pp','file') && exist('p','file')
    % Code to clean up and save figure.
    % These functions are not in the standard Matlab release.
    % pp is abbreviated for "Pretty Plot" and p for "Print"
    ppOpt = ppOptions;
    ppOpt.Box = 'on';
    ppOpt.AxisGrid = 'on';
    ppOpt.MarkerSize = 6;
    ppOpt.LineWidth = 1.5;
    pp('all',ppOpt);p;
end

===========================================
```

Published with MATLAB® R2019a