X-43 Stabilizer Analysis: Dan Lawson

Perform Stability and Forces analysis for AoA = 4 deg

```
%% Constants and Variable Initialization
% Variables
wedge_angle = [2.7, 3.07, 11.5, 0, 13.9];
stab_ang = linspace(5, 40, 20);
x_23 = 1.83;
Y = 0.2225;
x_2 = x_23 - Y / tan(deg2rad(11.5));
x_3 = x_23 - x_2;

x = [3.66, x_2, x_3, 0.762, 1.068];
AoA_sweep = linspace(-2, 16, 19);
M_inf = 7;

% General Constants
gamma = 1.362;
T_inf = 225.5; % K
P_inf = 1397; % Pa
V_inf = 2107; % m/s
altitude = 28974; % m

for j = 7

    for i = 1:length(wedge_angle)

        if i == 1
            % Stage 1
            if AoA_sweep(j) <= wedge_angle(i)
                theta = (wedge_angle(i) - AoA_sweep(j));
                [P_2, T_2, M2, beta] = oblique(theta, M_inf, P_inf, T_inf, gamma);
                M(i) = M2;
                T(i) = T_2;
                P(i) = P_2;
            elseif AoA_sweep(j) > wedge_angle(i)
                theta = (AoA_sweep(j) - wedge_angle(i));
                [P_2, T_2, M2] = PM_sol(theta, M_inf, P_inf, T_inf, gamma);
                M(i) = M2;
                T(i) = T_2;
                P(i) = P_2;
            end

            L(i) = P(i) * cos(deg2rad(AoA_sweep(j) - wedge_angle(i))) * (x(i) /
cos(deg2rad(wedge_angle(i))));
            D(i) = P(i) * sin(deg2rad(AoA_sweep(j) - wedge_angle(i))) * (x(i) /
cos(deg2rad(wedge_angle(i))));
```

```
        elseif i == 2
            % Stage 2
            if AoA_sweep(j) < 0 && abs(AoA_sweep(j)) > wedge_angle(i)
                theta = (wedge_angle(i) + AoA_sweep(j));
                [P_2, T_2, M2] = PM_sol(theta, M_inf, P_inf, T_inf, gamma);
                M(i) = M2;
                T(i) = T_2;
                P(i) = P_2;
            else
                theta = (AoA_sweep(j) + wedge_angle(i));
                [P_2, T_2, M2, beta] = oblique(theta, M_inf, P_inf, T_inf, gamma);
                M(i) = M2;
                T(i) = T_2;
                P(i) = P_2;
            end

            L(i) = P(i) * cos(deg2rad(AoA_sweep(j) + wedge_angle(i))) * (x(i) /
cos(deg2rad(wedge_angle(i))));
            D(i) = P(i) * sin(deg2rad(AoA_sweep(j) + wedge_angle(i))) * (x(i) /
cos(deg2rad(wedge_angle(i))));
        elseif i == 3
            % Stage 3
            theta = (wedge_angle(i) - wedge_angle(i - 1));
            [P_2, T_2, M2, beta] = oblique(theta, M(i - 1), P(i - 1), T(i - 1),
gamma);
            M(i) = M2;
            T(i) = T_2;
            P(i) = P_2;

            L(i) = P(i) * cos(deg2rad(AoA_sweep(j) + wedge_angle(i))) * (x(i) /
cos(deg2rad(wedge_angle(i))));
            D(i) = P(i) * sin(deg2rad(AoA_sweep(j) + wedge_angle(i))) * (x(i) /
cos(deg2rad(wedge_angle(i))));
        elseif i == 4
            % Stage 4
            theta = (wedge_angle(i - 1));
            [P_2, T_2, M2] = PM_sol(theta, M(i - 1), P(i - 1), T(i - 1), gamma);
            M(i) = M2;
            T(i) = T_2;
            P(i) = P_2;

            L(i) = P(i) * cos(deg2rad(AoA_sweep(j) + wedge_angle(i))) * (x(i) /
cos(deg2rad(wedge_angle(i))));
            D(i) = P(i) * sin(deg2rad(AoA_sweep(j) + wedge_angle(i))) * (x(i) /
cos(deg2rad(wedge_angle(i))));
        elseif i == 5
            % Stage 5
            theta = (wedge_angle(i - 1) + wedge_angle(i));
            [P_2, T_2, M2] = PM_sol(theta, M(i - 1), P(i - 1), T(i - 1), gamma);
            M(i) = M2;
```

```
            T(i) = T_2;
            P(i) = P_2;

            L(i) = P(i) * cos(deg2rad(AoA_sweep(j) - wedge_angle(i))) * (x(i) /
cos(deg2rad(wedge_angle(i))));
            D(i) = P(i) * sin(deg2rad(AoA_sweep(j) - wedge_angle(i))) * (x(i) /
cos(deg2rad(wedge_angle(i))));
        end

    end

    for jj = 1:length(stab_ang)

        [P_2, T_2, M2, beta] = oblique(stab_ang(jj), M_inf, P_inf, T_inf, gamma);
        L_w(jj) = 0.5 * x(5) * P_2 * cos(deg2rad(stab_ang(jj))) - P_inf;
        D_w(jj) = 0.5 * x(5) * P_2 * sin(deg2rad(stab_ang(jj)));

    end

        Lift(j) = sum(L(2:5)) - L(1) + L_w(19);
        Drag(j) = sum(D(2:5)) - D(1) + D_w(19);


        for ii = 1:length(wedge_angle)
            if ii < 3
                moment(ii) = L(ii) * 0.5 * (x(ii) / cos(deg2rad(wedge_angle(ii))));
            elseif ii >= 3
                moment(ii) = L(ii) * 0.5 * (x(ii) / cos(deg2rad(wedge_angle(ii))))
+ sum(x(2:ii));
            end
        end
        Moment(j) = moment(1) - (moment(2) + moment(3) + moment(4) + moment(5)) +
(L_w(19)*(x_23 + x(4) + 0.75*x(5)));

        LC(j) = Moment(j) / Lift(j);
        CG = 1.83;
        SM(j) = (LC(j) - CG) / x(1);


end

fprintf(['To achieve positive Static Margin, a Wedge angle of ' ...
    ' %g deg is required for the stabilizer at AoA = 4 deg'], stab_ang(19))
```

```
To achieve positive Static Margin, a Wedge angle of  38.1579 deg is required for the stabilizer at AoA = 4 deg
```