Codici

Arianna Pascale

Aprile 2024

1 Geometria semplificata

```
%% Dati
% Dimensioni VTP
c_r = 270e-3; % corda alla radice (m)
c_p = 200e-3; % corda alla punta (m)
h_v = 165e-3; % altezza VTP (m)

V = 20; % Velocit  d’incidenza (m/s)
av = 5.194391; % Pendenza Cl_alpha VTP
Z_AC = 0.175;

% Dimensioni ala
% INFERIORE
b_1 = 2.000;
Cr_1 = 0.332;
Cp_1 = 0.220;
d_1 = 0.131;
Sw_1 = 0.859;
Sm_1 = 0.772;

% POSTERIORE
b_2 = 2.000;
Cr_2 = 0.2;
Cp_2 = 0.150;
d_2 = 0;
Sw_2 = 0.4;
Sm_2 = 0.4;

V_eq = 20;

%% Rappresentazione
% Coordinate dei vertici del trapezio
x1 = [10, 270+10, 270+10, 70+10];
y1 = [0, 0, 165, 165];

x = [0, 270, 270, 70];
y = [0, 0, 165, 165];

% Calcolare le coordinate del baricentro
x_bar = (1/3) * (x(1) + x(2) + x(3));
z_bar = (1/3) * (y(1) + y(2) + y(3));
cdg = [x_bar, z_bar];
```

```
% Calcolare l'altezza del trapezio
h = abs(y(3) - y(1));

% Calcolare le lunghezze delle basi del trapezio
b1 = abs(x(2) - x(1)); % mm
b2 = abs(x(3) - x(4)); % mm

% Calcolare l'area del trapezio
area = (h / 2) * (b1 + b2)/(1000^2); % Area in m^2

figure;
hold on;
plot(x1, y1, 'bo-');
fill(x1, y1, 'r');
plot(x_bar, z_bar, 'bx', 'MarkerSize', 10, 'LineWidth', 2);
title('Geometria dell\'impennaggio verticale');
xlabel('Asse X (mm)');
ylabel('Asse Z (mm)');
grid on;
xlim([0 300]);
ylim([0 300]);
hold off;

%% Rappresentazione delle Ali
% Coordinate dei vertici del trapezio per l'ala superiore
x1 = [0, 0.132, 0.332, 0.332]*1000+600;
x11 = [0, 0.132, 0.332, 0.332]*1000+600;
y1 = [0.155, 1, 1, 0.155]*1000;
y11 = [-0.155, -1, -1, -0.155]*1000;

% Coordinate dei vertici del trapezio per l'ala inferiore
x2 = [0, 0.2-0.15, 0.2, 0.2]*1000+660+660;
x22 = [0, 0.2-0.15, 0.2, 0.2]*1000+660+660;
y2 = [0.025, 1, 1, 0.025]*1000;
y22 = [-0.025, -1, -1, -0.025]*1000;

figure;
hold on;
plot(x1, y1);
fill(x1, y1, 'b');
hold on
plot(x11, y11);
fill(x11, y11, 'b');
hold on;
plot(x2, y2);
fill(x2, y2, 'g');
hold on
plot(x22, y22);
fill(x22, y22, 'g');
title('Geometria dell\'ala posteriore e frontale');
xlabel('Asse X (mm)');
ylabel('Asse Y (mm)');
grid on;
xlim([0 2000])
ylim([-1100 1100])
```

```
hold off;
```

Listing 1: Geometria semplificata.

2 Mesh Sensitivity

```
%% COEFFICIENTI - DIMENSIONI BASE
BS = [1, 2, 4, 5];
CL = [0.077887274824039
      0.078128423927313
      0.081100344736512
      0.081391297748964];

figure(1)
err_CL = [4.6, 2.2, 3, 1.3] * 10^(-4);
errorbar(BS, CL, err_CL, 'r', 'linewidth', 2)
grid on;
title('Coefficiente di portanza');
xlabel('Dimensione della base (m)');
ylabel('CL');

CD = [0.0590207388855366
      0.0319408568538362
      0.0333249848302900
      0.0336263006742696];

figure(2)
err_CD = [4.1, 1.3, 2, 3.1] * 10^(-3);
errorbar(BS, CD, err_CD, 'r', 'linewidth', 2)
grid on;
title('Coefficiente di resistenza');
xlabel('Dimensione della base (m)');
ylabel('CD');

CM = [0.0093145300229892
      0.0092771839159540
      0.0083061576207280
      0.0081046485849526];

figure(3)
err_CM = [4.1, 1.3, 2, 3.1] * 10^(-5);
errorbar(BS, CM, err_CM, 'r', 'linewidth', 2)
grid on;
title('Coefficiente di momento');
xlabel('Dimensione della base (m)');
ylabel('CM');

%% Variazione percentuale
A = [CL, CD]; % Colonna 1 CL, colonna 2 CD.

Var_CL = [];

for i = 1:length(CL)
    Var_CL = [Var_CL; abs(A(1, 1) - A(i, 1)) / A(1, 1)];
end
```

```
format long g, Var_CL * 100

Var_CD = [];

for i = 1:length(CD)
    Var_CD = [Var_CD; abs(A(1, 2) - A(i, 2)) / A(1, 2)];
end
format long g, Var_CD * 100

Var_CM = [];

for i = 1:length(CM)
    Var_CM = [Var_CM; abs(CM(1) - CM(i)) / CM(1)];
end
format long g, Var_CM * 100
```

Listing 2: Mesh sensitivity.

3 Derivate su Star CCM+

```
%% DERIVATE in Q
q = [0 0.05 0.1];
cm = [0.0092771839159540
    0.0074238018960057
    0.0055744482779273
    ]/0.3;

c = polyfit(q,cm,1);

figure(1)
plot(q, cm, 'b-', 'linewidth', 2)
hold on;
scatter(q, cm, 'bo', 'MarkerFaceColor', 'b') % Cerchi blu con
    riempimento

slope_text = sprintf('Pendenza: %.4f', c(1));
text(q(2)+0.02, cm(2)+0.003, slope_text, 'FontSize', 10, '
    HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');

hold off;
grid on;
title('Cmq');
xlabel('q (rad/s)');
ylabel('CM');

cx = [0.0319408568538362 0.031922889731102 0.031919045887018];
d = polyfit(q, cx, 1);

figure(2)
plot(q, cx, 'b-', 'linewidth', 2)
hold on;
scatter(q, cx, 'bo', 'MarkerFaceColor', 'b') % Cerchi blu con
    riempimento

slope_text = sprintf('Pendenza: %.4f', d(1));
```

```
text(q(2), cx(2)+0.00002, slope_text, 'FontSize', 10, 'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');

hold off;
grid on;
title('Cxq');
xlabel('q (rad/s)');
ylabel('CX');

cz = [0.078128423927313 0.08239354420290657 0.08663177061789876];
e = polyfit(q, cz, 1);

figure(3)
plot(q, cz, 'b-', 'linewidth', 2)
hold on;
scatter(q, cz, 'bo', 'MarkerFaceColor', 'b') % Cerchi blu con riempimento

slope_text = sprintf('Pendenza: %.4f', e(1));
text(q(2)+0.02, cz(2)+0.0002, slope_text, 'FontSize', 10, 'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');

hold off;
grid on;
title('Czq');
xlabel('q (rad/s)');
ylabel('CZ');

derivate = [c(1) d(1) e(1)];
adim = derivate * 2 * 20 / 0.3;


%% DERIVATE in P
p = [0 0.05 0.1];
Clp = [-0.0000383200808147 -0.001623982571221884 -0.0032097251374068685];
Cnp = [0.0000367575866534
0.0000315414829104
0.0000261556814045
 ];

CYp = [0.0000391385074484

0.0001591783632986
0.0002792375120757
    ];

% Clp
Cl = polyfit(p, Clp, 1);
figure(4)
plot(p, Clp, 'r-', 'linewidth', 2)
hold on;
scatter(p, Clp, 'ro', 'MarkerFaceColor', 'r') % Red circles with filling
slope_text = sprintf('Pendenza: %.4f', Cl(1));
text(p(1)+0.02, Clp(1)-0.003, slope_text, 'FontSize', 10, 'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
```

```
hold off;
grid on;
title('Clp');
xlabel('p (rad/s)');
ylabel('CL');

% Cnp
Cn = polyfit(p, Cnp, 1);
figure(5)
plot(p, Cnp, 'r-', 'linewidth', 2)
hold on;
scatter(p, Cnp, 'ro', 'MarkerFaceColor', 'r') % Red circles with
    filling
slope_text = sprintf('Pendenza: %.4f', Cn(1));
text(p(2)+0.02, Cnp(2), slope_text, 'FontSize', 10, '
    HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
hold off;
grid on;
title('Cnp');
xlabel('p (rad/s)');
ylabel('CN');

% CYp
CY = polyfit(p, CYp, 1);
figure(6)
plot(p, CYp, 'r-', 'linewidth', 2)
hold on;
scatter(p, CYp, 'ro', 'MarkerFaceColor', 'r') % Red circles with
    filling
slope_text = sprintf('Pendenza: %.4f', CY(1));
text(p(2)+0.02, CYp(2), slope_text, 'FontSize', 10, '
    HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
hold off;
grid on;
title('CYp');
xlabel('p (rad/s)');
ylabel('CY');

derivate_p = [Cl(1) Cn(1) CY(1)]
adim_p = derivate_p * 2 * 20 / 2

%% DERIVATE in R
r = -[0 0.05 0.1];
cy = [0.0000391385074484
    0.0004948243566669
    0.0009510865296281
];
c = polyfit(r, cy, 1);

figure(1)
plot(r, cy, '-o', 'linewidth', 2, 'color', [0, 0.6, 0], '
    MarkerFaceColor', [0, 0.6, 0])
hold on;
grid on;
title('Cyr');
xlabel('r (rad/s)');
ylabel('CY');
```

```
text(r(2)+0.002, cy(2), sprintf('Pendenza: %.4f', c(1)), 'FontSize', 10, 'Color', [0, 0.6, 0]);

cl = -[0.0000383200808147
    0.0001147901329164
    0.0001920266249104
];
d = polyfit(r, cl, 1);

figure(2)
plot(r, cl, '-o', 'linewidth', 2, 'color', [0, 0.6, 0], 'MarkerFaceColor', [0, 0.6, 0])
hold on;
grid on;
title('Clr');
xlabel('r (rad/s)');
ylabel('Cl');
text(r(2)+0.002, cl(2), sprintf('Pendenza: %.4f', d(1)), 'FontSize', 10, 'Color', [0, 0.6, 0]);

cn = [0.0000367575866534
    0.0001776734264252
    0.0003185162053411
];
e = polyfit(r, cn, 1);

figure(3)
plot(r, cn, '-o', 'linewidth', 2, 'color', [0, 0.6, 0], 'MarkerFaceColor', [0, 0.6, 0])
hold on;
grid on;
title('Cnr');
xlabel('r (rad/s)');
ylabel('Cn');
text(r(2)+0.002, cn(2), sprintf('Pendenza: %.4f', e(1)), 'FontSize', 10, 'Color', [0, 0.6, 0]);

derivate_r = [c(1) d(1) e(1)];
adim_r = derivate_r * 2 * 20 / 2;

%% Derivate in beta

beta =[0 1 5]*2*pi/360;
cy =[0.0000391385074484 -0.010940723238362138 -0.047074366742012576];
c= polyfit(beta, cy,1)

cl =[-0.0000383200808147 4.3341134779422574E-4 0.002051098010677344];
d= -polyfit(beta, cl,1)
%
cn =[0.0000367575866534 -4.3742281335062483E-4 -6.800232571175142E-4];
e= -polyfit(beta, cn,1)

figure(1)
plot(beta, cy, 'm-', 'linewidth', 2)
```

```
hold on;
scatter(beta, cy, 'mo', 'MarkerFaceColor', 'm') % Cerchi magenta
    con riempimento

slope_text = sprintf('Pendenza: %.4f', c(1));
text(beta(2)+0.02, cy(2)+0.003, slope_text, 'FontSize', 10, '
    HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');

hold off;
grid on;
title('Cyb');
xlabel('beta (rad)');
ylabel('CY');

figure(2)
plot(beta, cl, 'm-', 'linewidth', 2)
hold on;
scatter(beta, cl, 'mo', 'MarkerFaceColor', 'm') % Cerchi magenta
    con riempimento

slope_text = sprintf('Pendenza: %.4f', d(1));
text(beta(2), cl(2)+0.0004, slope_text, 'FontSize', 10, '
    HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');

hold off;
grid on;
title('Clb');
xlabel('beta (rad)');
ylabel('Cl');

figure(3)
plot(beta, cn, 'm-', 'linewidth', 2)
hold on;
scatter(beta, cn, 'mo', 'MarkerFaceColor', 'm') % Cerchi magenta
    con riempimento

slope_text = sprintf('Pendenza: %.4f', e(1));
text(beta(2)+0.02, cn(2)+0.0002, slope_text, 'FontSize', 10, '
    HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');

hold off;
grid on;
title('Cnb');
xlabel('b (rad)');
ylabel('Cn');
```

Listing 3: Calcolo derivate.

4 Calcolo matematico delle derivate

```
%% Derivate con formule
%dati
av = 3;
db = 0.3; %
dr = 0.1;
```

```
etav = 0.95; %
sv = 0.0388; %
s = 0.7 ; %
b = 2; %
zv =  0.15;
lv = 0.65;
c = 0.3; %
a = 5.22; %
p = [0 0.05 0.1];
r = [-0.1 -0.05 0];
beta = [0 5]*2*pi/360;
%% Derivate latero-direzionali

%Cyb (solo vtp)
cyb = -av*(1-db)*etav*sv/s;
cy =[
    -4.8022411466058305E-6
-0.008827052018452887
 ];
cyb2 = polyfit(beta,cy,1);

%clb (solo vtp) OK
clb = -etav*sv/s*zv/b*av*(1-db);

cl =[
    5.011540898113128E-7
    6.102051612000775E-4
 ];
clb2 = -polyfit(beta,cl,1);

%cnb (solo vtp) OK
cnb = etav*sv/s*lv/b*av*(1-db);
cn =[
    3.4396611393645147E-6
    -0.00438305705016204
 ];
cnb2 = -polyfit(beta,cn,1);

%cyp (solo vtp) OK
cyp = -2*sv/s*zv/b*av;
cy =[
    -4.8022411466058305E-6
    -1.0083626984074161E-4
 -1.9679621867864383E-4
 ];
cyp2 = polyfit(p,cy,1)*20;

%clp OK
clp_w = -4/s/b^2*a*c*integral(@(y) y.^2, 0, b/2);
clp_v= -2*sv/s*(zv/b)^2*av;
clp = clp_w+clp_v;

%cnp (solo vtp) OK
cnp = -cyp*lv/b;
cn =[
   0.0000367575866534
    1.97646826998092 3E-6
```

```
    5.079366393633893E-7
 ];
cnp2 =- polyfit(p,cn,1)*20;
%plot(p,cn)

%cyr (solo vtp) ??? SEGNO
cyr = sv/s*av*etav*(2*lv/b-dr);
cy =[3.61659326269009E-4
 1.7824361480322777E-4
 -4.8022411466058305E-6];
cyr2 = polyfit(r,cy,1)*20;

%clr (solo vtp)  OK
clr = av*sv/s*zv/b*(2*lv/b-dr);
cl =[-2.4655118841995452E-5
 -1.206533552377859E-5
 5.011540898113128E-7];
clr2 = polyfit(r,cl,1)*20;


%cnr (solo vtp) OK
cnr = -cyr*lv/b;
cn =[8.796947593782375E-5
 4.32926755360921E-5
 -1.2843015415843904E-6];
cnr2 = polyfit(r,cn,1)*20;
```

Listing 4: Calcolo delle derivate.

5 Calcolo degli autovalori

```
%% AUTOVALORI LONGITUDINALI
%INPUT
m = 12; % [Kg]
u0 = 20; % [m/s]
S = 0.7; %[m2]
c = 0.3; %[m]
b = 2; % [m]
rho = 1.225; % [kg/m3]
qc = 0.5*rho*u0^2;
g0 = 9.81;

Ix = 0.5^2*2; % [kg*m2]
Iy = 0.35^2*m; % [kg*m2]
Iz = 0.35^2*m; % [kg*m2]

% lt = 0.5; % [m]
% at = 5;
% St = (0.3+0.2)*1;
%
CD0 = 0.0319; %valore CFD
CDa = 0.178771;
CDu = 0;
%
CLa = 5.248757;
```

```
CLu = 0;
CZq = 11.3378 ;% -aero_data.CL_q; %OpenVSP:6.17; -2 * at * lt * St
    / c / S;
%
CMa = -2.240018;
% openVSP: -16.6687330
CMap = 0;
CMq = -16.4566; %-2*at * lt / c / lt * St / c / S;
%
CZde = 0;%
CMde = 0;%
%
% CYb = aero_data.CY_beta;
% CYp = aero_data.CY_p;
% CYr = aero_data.CY_r;
% Clb = aero_data.Cl_beta;
% Clp = aero_data.Cl_p;
% Clr = aero_data.Cl_r;
% Cnb = aero_data.Cn_beta;
% Cnp = aero_data.Cn_p;
% Cnr = aero_data.Cn_r;
%
% CLda = -0.134;
% CLdr = 0.107;
% CNda = -0.0035;
% CNdr = -0.072;
% CYdr = 0.157;
% CYda = 0.0;

% derived
%qc = 0.5*rho*u0^2;
CL0 = 2*m*g0/S/rho/u0^2;
%
CXu = -(CDu + 2*CD0);
CXw = -(CDa - CL0);
CXq = -0.0219;
CXwp = 0;
%
CZu = -(CLu + 2*CL0);
CZw = -(CLa + CD0);
CZwp = 0;

%
CMu = 0;

%
Xu = CXu/u0*qc*S/m;
Xw = CXw/u0*qc*S/m;
Zu = CZu/u0*qc*S/m;
Zw = CZw*qc*S/m/u0;
Zwp = CZwp*c/2/u0*qc*S/m;
Zq = CZq*c/2/u0*qc*S/m;
Zde = CZde*qc*S/m;
Mu = CMu/u0*qc*S*c/Iy;
Ma = CMa*qc*S*c/Iy;
Mw = CMa*qc*S*c/Iy/u0;
Mwp = CMap*c/2/u0*qc*S*c/Iy/u0;
```

```
Mq = CMq*c/2/u0*qc*S*c/Iy;
Mde = CMde*qc*S*c/Iy;
%
% Yb = CYb*qc*S/m;
% Yp = CYp*qc*S/m;
% Yr = CYr*qc*S/m;
% Lb = Clb*qc*S*b/Ix;
% Lp = Clp*qc*S*b/Ix;
% Lr = Clr*qc*S*b/Ix;
% Nb = Cnb*qc*S*b/Iz;
% Np = Cnp*qc*S*b/Iz;
% Nr = Cnr*qc*S*b/Iz;
%
% Lda = CLda*qc*S*b/Ix;
% Ldr = CLdr*qc*S*b/Ix;
% Nda = -CNda*qc*S*b/Iz;
% Ndr = CNdr*qc*S*b/Iz;
% Yda = CYda*qc*S/m;
% Ydr = CYdr*qc*S/m;

% Equilibrio
alpha0 = 0;
ub0 = u0;
wb0 = 0;


% matrice che moltiplica il vento
E = [-Xu -Xw 0;...
    -Zu -Zw 0;...
    -(Mu+Mwp*Zu) -(Mw+Mwp*Zw) -(Mq+Mwp*u0);...
    0   0   0];

A = [Xu Xw 0 -g0;...
    Zu Zw u0 0;...
    (Mu+Mwp*Zu) (Mw+Mwp*Zw) (Mq+Mwp*u0) 0;...
    0  0  1  0];

[lambda] = eig(A);


% Grafico
figure('Color', 'w'); % Sfondo bianco
h1 = plot(real(lambda(3:4)), imag(lambda(3:4)), '*r', 'MarkerSize',
     10, 'LineWidth', 2);
hold on
h2 = plot(real(lambda(1:2)), imag(lambda(1:2)), '*g', 'MarkerSize',
     10, 'LineWidth', 2);

% Assi x e y
xlim([-5, max(real(lambda))+2]);
plot(xlim, [0, 0], 'k--', 'LineWidth', 0.5);
plot([0, 0], ylim, 'k--', 'LineWidth', 0.5);

% Decorazioni aggiuntive
title('Autovalori della dinamica longitudinale', 'FontSize', 16);
legend('Fugoide', 'Corto Periodo', 'FontSize', 12);
xlabel('Parte Reale', 'FontSize', 14);
```

```
ylabel('Parte Immaginaria', 'FontSize', 14);
grid on
box on

% Imposta manualmente i limiti dell'asse x per spostarlo verso sinistra


% Miglioramento della leggibilit  degli assi
set(gca, 'FontSize', 12, 'LineWidth', 1.2);

% Personalizzazione dei marker
h1.MarkerEdgeColor = 'r';
h2.MarkerEdgeColor = 'g';

% Aggiunta delle coordinate vicino ai punti del grafico
text(real(lambda) + 0.1, imag(lambda), ...
    arrayfun(@(x) sprintf('%.2f + %.2fi', real(x), imag(x)), lambda, 'UniformOutput', false), ...
    'FontSize', 10, 'VerticalAlignment', 'middle', 'HorizontalAlignment', 'left');


approx_p_omega = sqrt(-Zu*g0/u0);
approx_p_zeta = -Xu/2/approx_p_omega;
approx_sp_omega = sqrt(Zw*Mq - Ma);
approx_sp_zeta = -(Mq + Mwp/u0 + Zw)/2/approx_sp_omega;

fprintf('Modo Fugoide Approssimato: omega = %f [rad/s], zeta = %f\nModo Corto Periodo approssimato: omega = %f [rad/s], zeta = %f\n',...
    approx_p_omega,...
    approx_p_zeta,...
    approx_sp_omega,...
    approx_sp_zeta);

fprintf( 'Modo 1 = Periodo = %f, t/2 = %f, omega = %f [rad/s], zeta = %f\n',...
    2*pi/abs(imag(lambda(end))),...
    -0.69/real(lambda(end)),...
    abs(lambda(end)),...
    sqrt( 1 / (1 + (imag(lambda(end))/real(lambda(end)))^2) ));
fprintf( 'Modo 2 = Periodo = %f, t/2 = %f, omega = %f [rad/s], zeta = %f\n',...
    2*pi/abs(imag(lambda(1))),...
    -0.69/real(lambda(1)),...
    abs(lambda(1)),...
    sqrt( 1 / (1 + (imag(lambda(1))/real(lambda(1)))^2) ));

tausp = m/(qc*S*CLa)
cap = m*g0*approx_sp_omega^2/(qc*S*CLa)
na =9.7^2/cap

%% TIME RESPONSE FOR corto periodo
% System matrix A for phugoid mode
A_phugoid = [0 1; 0 lambda(1)];

```

```
% Time vector
t = 0:0.1:20;

% Compute the response using sinusoidal functions
omega = imag(lambda(1)); % Frequency of oscillation
phi = atan2(imag(lambda(1)), real(lambda(1))); % Phase angle
% Mode 1
y1 = exp(real(lambda(1))*t) .* (cos(omega*t + phi));

% Mode 2 (conjugate)
y2 = exp(real(lambda(2))*t) .* (cos(omega*t + phi));

% Combine both modes
y_phugoid = y1 + y2;

% Plot the response
figure;
plot(t, real(y_phugoid), 'LineWidth', 2);
title('Short Period Mode - Time Response');
xlabel('Time');
ylabel('Amplitude');
grid on;

%% TIME RESPONSE FOR fugoide
% System matrix A for short period mode
A_short_period = [0 1; 0 lambda(3)];

% Time vector
t = 0:1:500;

% Compute the response using sinusoidal functions
omega = imag(lambda(3)); % Frequency of oscillation
phi = atan2(imag(lambda(3)), real(lambda(3))); % Phase angle

% Mode 1
y1_short_period = exp(real(lambda(3))*t) .* (cos(omega*t + phi));

% Mode 2 (conjugate)
y2_short_period = exp(real(lambda(4))*t) .* (cos(omega*t + phi));

% Combine both modes
y_short_period = y1_short_period + y2_short_period;

% Plot the response
figure;
plot(t, real(y_short_period), 'LineWidth', 2);
title('Phugoid Mode - Time Response');
xlabel('Time');
ylabel('Amplitude');
grid on;

%% AUTOVALORI LATERO-DIREZIONALI
% INPUT
m = 12; % [Kg]
u0 = 20; % [m/s]
S = 0.7; %[m2]
c = 0.3; %[m]
```

```
b = 2; % [m]
rho = 1.225; % [kg/m3]
qc = 0.5*rho*u0^2;
g0 = 9.81;
theta0 = 0;

Ix = 0.5^2*2; % [kg*m2]
Iy = 0.35^2*m; % [kg*m2]
Iz = 0.35^2*m; % [kg*m2]

% lt = 0.5; % [m]
% at = 5;
% St = (0.3+0.2)*1;
%
CD0 = 0.0319; %valore CFD
CDa = 0.178771;
CDu = 0;
%
CLa = 5.194391;
CLu = 0;
CZq = 11.3378 %OpenVSP:6.17; -2 * at * lt * St / c / S;
%
CMa = -2.224921;
% openVSP: -16.6687330
CMap = 0;
CMq = -16.4566; %-2*at * lt / c / lt * St / c / S;
%
CZde = 0;%
CMde = 0;%

CYb =-0.5335;
CYp = 0.0480;
CYr = -0.1824;
Clb = -0.0237; %aero_data.Cl_beta; %da calcolare con CFD
Clp = -0.6343;
Clr = 0.0307;
Cnb = 0.0069; %aero_data.Cn_beta;
Cnp = -0.0021;
Cnr = -0.0564;

CLda = -0.134;
CLdr = 0.107;
CNda = -0.0035;
CNdr = -0.072;
CYdr = 0.157;
CYda = 0.0;

% derived
qc = 0.5*rho*u0^2;
CL0 = 2*m*g0/S/rho/u0^2;
%
CXu = -(CDu + 2*CD0);
CXw = -(CDa - CL0);
CXq = 0;
CXwp = 0;
%
CZu = -(CLu + 2*CL0);
```

```
CZw = -(CLa + CD0);
CZwp = 0;

%
CMu = 0;

%
Xu = CXu/u0*qc*S/m;
Xw = CXw/u0*qc*S/m;
Zu = CZu/u0*qc*S/m;
Zw = CZw*qc*S/m/u0;
Zwp = CZwp*c/2/u0*qc*S/m;
Zq = CZq*c/2/u0*qc*S/m;
Zde = CZde*qc*S/m;
Mu = CMu/u0*qc*S*c/Iy;
Ma = CMa*qc*S*c/Iy;
Mw = CMa*qc*S*c/Iy/u0;
Mwp = CMap*c/2/u0*qc*S*c/Iy/u0;
Mq = CMq*c/2/u0*qc*S*c/Iy;
Mde = CMde*qc*S*c/Iy;
%
Yb = CYb*qc*S/m;
Yp = CYp*qc*S/m*b/2/u0;
Yr = CYr*qc*S/m*b/2/u0;
Lb = Clb*qc*S*b/Ix;
Lp = Clp*qc*S*b/Ix*b/2/u0;
Lr = Clr*qc*S*b/Ix*b/2/u0;
Nb = Cnb*qc*S*b/Iz;
Np = Cnp*qc*S*b/Iz*b/2/u0;
Nr = Cnr*qc*S*b/Iz*b/2/u0;

Lda = CLda*qc*S*b/Ix;
Ldr = CLdr*qc*S*b/Ix;
Nda = -CNda*qc*S*b/Iz;
Ndr = CNdr*qc*S*b/Iz;
Yda = CYda*qc*S/m;
Ydr = CYdr*qc*S/m;

% Equilibrio
alpha0 = CL0/CLa;
ub0 = u0*cos(alpha0);
wb0 = u0*sin(alpha0);
%
% vettore di stato X = [u, w, q, pitch, pn_p, pe_p, pd_p]

E = [-Xu -Xw 0;...
    -Zu -Zw 0;...
    -(Mu+Mwp*Zu) -(Mw+Mwp*Zw) -(Mq+Mwp*u0);...
    0   0   0];

A = [Yb/u0 Yp/u0 -(1 - Yr/u0) g0*cos(theta0)/u0;...
    Lb Lp Lr 0;...
    Nb Np Nr 0;...
    0  1  0  0];
lambda = eig(A);
% Grafico
figure('Color', 'w'); % Sfondo bianco
```

```
h1 = plot(real(lambda(1)), imag(lambda(1)), '*r', 'MarkerSize', 10, 'LineWidth', 2);
hold on
h2 = plot(real(lambda(2:3)), imag(lambda(2:3)), '*g', 'MarkerSize', 10, 'LineWidth', 2);
hold on
h3 = plot(real(lambda(4)), imag(lambda(4)), '*m', 'MarkerSize', 10, 'LineWidth', 2);

% Assi x e y
xlim([-25, max(real(lambda))+2]);
plot(xlim, [0, 0], 'k--', 'LineWidth', 0.5);
plot([0, 0], ylim, 'k--', 'LineWidth', 0.5);

% Decorazioni aggiuntive
title('Autovalori della dinamica latero-direzionale', 'FontSize', 16);
legend('Roll mode', 'Dutch roll','Modo spirale', 'FontSize', 12);
xlabel('Parte Reale', 'FontSize', 14);
ylabel('Parte Immaginaria', 'FontSize', 14);
grid on
box on

% Imposta manualmente i limiti dell'asse x per spostarlo verso sinistra


% Miglioramento della leggibilit  degli assi
set(gca, 'FontSize', 12, 'LineWidth', 1.2);

% Personalizzazione dei marker
h1.MarkerEdgeColor = 'r';
h2.MarkerEdgeColor = 'g';

% Aggiunta delle coordinate vicino ai punti del grafico
text(real(lambda) + 0.1, imag(lambda), ...
    arrayfun(@(x) sprintf('%.2f + %.2fi', real(x), imag(x)), lambda, 'UniformOutput', false), ...
    'FontSize', 10, 'VerticalAlignment', 'middle', 'HorizontalAlignment', 'left');


fprintf( 'Modo 1 = Periodo = %f, t/2 = %f\n',...
    2*pi/abs(imag(lambda(end))),...
    -0.69/real(lambda(end)) );
fprintf( 'Modo 2 = Periodo = %f, t/2 = %f\n',...
    2*pi/abs(imag(lambda(1))),...
    -0.69/real(lambda(1)) );
fprintf( 'Modo 3 = Periodo = %f, t/2 = %f, omega = %f, zeta = %f\n',...
    2*pi/abs(imag(lambda(2))),...
    -0.69/real(lambda(2)),...
    abs(lambda(2)),...
    sqrt( 1 / (1 + (imag(lambda(2))/real(lambda(2)))^2) ) );

tau_CB = -1/lambda(1)
```

```
%% TIME RESPONSE FOR roll mode
% System matrix A for roll mode
A_phugoid = [0 1; 0 lambda(1)];

% Time vector
t = 0:0.1:3;

% Compute the response using sinusoidal functions
omega = imag(lambda(1)); % Frequency of oscillation
phi = atan2(imag(lambda(1)), real(lambda(1))); % Phase angle
% Mode 1
y1 = exp(real(lambda(1))*t) .* (cos(omega*t + phi));

% Combine both modes
y_roll = y1;

% Plot the response
figure;
plot(t, real(y_roll), 'LineWidth', 2);
title('Roll Mode - Time Response');
ylim([-1.1 0.6])
xlabel('Time');
ylabel('Amplitude');
grid on;

%% TIME RESPONSE FOR dutch roll
% System matrix A for dutch roll
A_phugoid = [0 1; 0 lambda(2)];

% Time vector
t = 0:0.1:20;

% Compute the response using sinusoidal functions
omega = imag(lambda(2)); % Frequency of oscillation
phi = atan2(imag(lambda(2)), real(lambda(2))); % Phase angle
% Mode 1
y1 = exp(real(lambda(2))*t) .* (cos(omega*t + phi));

% Mode 2 (conjugate)
y2 = exp(real(lambda(3))*t) .* (cos(omega*t + phi));

% Combine both modes
y_dr = y1 + y2;

% Plot the response
figure;
plot(t, real(y_dr), 'LineWidth', 2);
title('Dutch Roll Mode - Time Response');
xlabel('Time');
ylabel('Amplitude');
grid on;

%% TIME RESPONSE FOR spiral mode
% System matrix A for spiral mode
A_phugoid = [0 1; 0 lambda(4)];

% Time vector
```

```
t = 0:0.1:200;

% Compute the response using sinusoidal functions
omega = imag(lambda(4)); % Frequency of oscillation
phi = atan2(imag(lambda(4)), real(lambda(4))); % Phase angle
% Mode 1
y1 = exp(real(lambda(4))*t) .* (cos(omega*t + phi));

% Combine both modes
y_s = y1;

% Plot the response
figure;
plot(t, real(y_s), 'LineWidth', 2);
title('Spiral Mode - Time Response');
ylim([-2 1])
xlabel('Time');
ylabel('Amplitude');
grid on;


tauSM = -1/real(lambda(4))
tauRM = -1/real(lambda(1))
tauSM/tauRM
```

Listing 5: Calcolo degli autovalori.