

Codici

Arianna Pascale

Aprile 2024

1 Geometria semplificata

```
1 %% Dati
2 % Dimensioni VTP
3 c_r = 270e-3; % corda alla radice (m)
4 c_p = 200e-3; % corda alla punta (m)
5 h_v = 165e-3; % altezza VTP (m)
6
7 V = 20; % Velocit  d'incidenza (m/s)
8 av = 5.194391; % Pendenza Cl_alpha VTP
9 Z_AC = 0.175;
10
11 % Dimensioni ala
12 % INFERIORE
13 b_1 = 2.000;
14 Cr_1 = 0.332;
15 Cp_1 = 0.220;
16 d_1 = 0.131;
17 Sw_1 = 0.859;
18 Sm_1 = 0.772;
19
20 % POSTERIORE
21 b_2 = 2.000;
22 Cr_2 = 0.2;
23 Cp_2 = 0.150;
24 d_2 = 0;
25 Sw_2 = 0.4;
26 Sm_2 = 0.4;
27
28 V_eq = 20;
29
30 %% Rappresentazione
31 % Coordinate dei vertici del trapezio
32 x1 = [10, 270+10, 270+10, 70+10];
33 y1 = [0, 0, 165, 165];
34
35 x = [0, 270, 270, 70];
36 y = [0, 0, 165, 165];
37
38 % Calcolare le coordinate del baricentro
39 x_bar = (1/3) * (x(1) + x(2) + x(3));
40 z_bar = (1/3) * (y(1) + y(2) + y(3));
41 cdg = [x_bar, z_bar];
```

```

42
43 % Calcolare l'altezza del trapezio
44 h = abs(y(3) - y(1));
45
46 % Calcolare le lunghezze delle basi del trapezio
47 b1 = abs(x(2) - x(1)); % mm
48 b2 = abs(x(3) - x(4)); % mm
49
50 % Calcolare l'area del trapezio
51 area = (h / 2) * (b1 + b2)/(1000^2); % Area in m^2
52
53 figure;
54 hold on;
55 plot(x1, y1, 'bo-');
56 fill(x1, y1, 'r');
57 plot(x_bar, z_bar, 'bx', 'MarkerSize', 10, 'LineWidth', 2);
58 title('Geometria dell\'impennaggio verticale');
59 xlabel('Asse X (mm)');
60 ylabel('Asse Z (mm)');
61 grid on;
62 xlim([0 300]);
63 ylim([0 300]);
64 hold off;
65
66 %% Rappresentazione delle Ali
67 % Coordinate dei vertici del trapezio per l'ala superiore
68 x1 = [0, 0.132, 0.332, 0.332]*1000+600;
69 x11 = [0, 0.132, 0.332, 0.332]*1000+600;
70 y1 = [0.155, 1, 1, 0.155]*1000;
71 y11 = [-0.155, -1, -1, -0.155]*1000;
72
73 % Coordinate dei vertici del trapezio per l'ala inferiore
74 x2 = [0, 0.2-0.15, 0.2, 0.2]*1000+660+660;
75 x22 = [0, 0.2-0.15, 0.2, 0.2]*1000+660+660;
76 y2 = [0.025, 1, 1, 0.025]*1000;
77 y22 = [-0.025, -1, -1, -0.025]*1000;
78
79 figure;
80 hold on;
81 plot(x1, y1);
82 fill(x1, y1, 'b');
83 hold on
84 plot(x11, y11);
85 fill(x11, y11, 'b');
86 hold on;
87 plot(x2, y2);
88 fill(x2, y2, 'g');
89 hold on
90 plot(x22, y22);
91 fill(x22, y22, 'g');
92 title('Geometria dell\'ala posteriore e frontale');
93 xlabel('Asse X (mm)');
94 ylabel('Asse Y (mm)');
95 grid on;
96 xlim([0 2000])
97 ylim([-1100 1100])

```

```
98 hold off;
```

Listing 1: Geometria semplificata.

2 Mesh Sensitivity

```
1 %% COEFFICIENTI - DIMENSIONI BASE
2 BS = [1, 2, 4, 5];
3 CL = [0.077887274824039
4       0.078128423927313
5       0.081100344736512
6       0.081391297748964];
7
8 figure(1)
9 err_CL = [4.6, 2.2, 3, 1.3] * 10^(-4);
10 errorbar(BS, CL, err_CL, 'r', 'linewidth', 2)
11 grid on;
12 title('Coefficiente di portanza');
13 xlabel('Dimensione della base (m)');
14 ylabel('CL');
15
16 CD = [0.0590207388855366
17       0.0319408568538362
18       0.0333249848302900
19       0.0336263006742696];
20
21 figure(2)
22 err_CD = [4.1, 1.3, 2, 3.1] * 10^(-3);
23 errorbar(BS, CD, err_CD, 'r', 'linewidth', 2)
24 grid on;
25 title('Coefficiente di resistenza');
26 xlabel('Dimensione della base (m)');
27 ylabel('CD');
28
29 CM = [0.0093145300229892
30       0.0092771839159540
31       0.0083061576207280
32       0.0081046485849526];
33
34 figure(3)
35 err_CM = [4.1, 1.3, 2, 3.1] * 10^(-5);
36 errorbar(BS, CM, err_CM, 'r', 'linewidth', 2)
37 grid on;
38 title('Coefficiente di momento');
39 xlabel('Dimensione della base (m)');
40 ylabel('CM');
41
42 %% Variazione percentuale
43 A = [CL, CD]; % Colonna 1 CL, colonna 2 CD.
44
45 Var_CL = [];
46
47 for i = 1:length(CL)
48     Var_CL = [Var_CL; abs(A(1, 1) - A(i, 1)) / A(1, 1)];
49 end
```

```

50 format long g, Var_CL * 100
51
52 Var_CD = [];
53
54 for i = 1:length(CD)
55     Var_CD = [Var_CD; abs(A(1, 2) - A(i, 2)) / A(1, 2)];
56 end
57 format long g, Var_CD * 100
58
59 Var_CM = [];
60
61 for i = 1:length(CM)
62     Var_CM = [Var_CM; abs(CM(1) - CM(i)) / CM(1)];
63 end
64 format long g, Var_CM * 100

```

Listing 2: Mesh sensitivity.

3 Derivate su Star CCM+

```

1 %% DERIVATE in Q
2 q = [0 0.05 0.1];
3 cm = [0.0092771839159540
4       0.0074238018960057
5       0.0055744482779273
6       ]/0.3;
7
8 c = polyfit(q, cm, 1);
9
10 figure(1)
11 plot(q, cm, 'b-', 'linewidth', 2)
12 hold on;
13 scatter(q, cm, 'bo', 'MarkerFaceColor', 'b') % Cerchi blu con
14     riempimento
15 slope_text = sprintf('Pendenza: %.4f', c(1));
16 text(q(2)+0.02, cm(2)+0.003, slope_text, 'FontSize', 10, '
17     HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
18
19 hold off;
20 grid on;
21 title('Cmq');
22 xlabel('q (rad/s)');
23 ylabel('CM');
24
25 cx = [0.0319408568538362 0.031922889731102 0.031919045887018];
26 d = polyfit(q, cx, 1);
27
28 figure(2)
29 plot(q, cx, 'b-', 'linewidth', 2)
30 hold on;
31 scatter(q, cx, 'bo', 'MarkerFaceColor', 'b') % Cerchi blu con
32     riempimento
33 slope_text = sprintf('Pendenza: %.4f', d(1));

```

```

33 text(q(2), cx(2)+0.00002, slope_text, 'FontSize', 10, '
    HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
34
35 hold off;
36 grid on;
37 title('Cxq');
38 xlabel('q (rad/s)');
39 ylabel('CX');
40
41 cz = [0.078128423927313 0.08239354420290657 0.08663177061789876];
42 e = polyfit(q, cz, 1);
43
44 figure(3)
45 plot(q, cz, 'b-', 'linewidth', 2)
46 hold on;
47 scatter(q, cz, 'bo', 'MarkerFaceColor', 'b') % Cerchi blu con
    riempimento
48
49 slope_text = sprintf('Pendenza: %.4f', e(1));
50 text(q(2)+0.02, cz(2)+0.0002, slope_text, 'FontSize', 10, '
    HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
51
52 hold off;
53 grid on;
54 title('Czq');
55 xlabel('q (rad/s)');
56 ylabel('CZ');
57
58 derivate = [c(1) d(1) e(1)];
59 adim = derivate * 2 * 20 / 0.3;
60
61 %% DERIVATE in P
62 p = [0 0.05 0.1];
63 Clp = [-0.0000383200808147 -0.001623982571221884
    -0.0032097251374068685];
64 Cnp = [0.0000367575866534
    0.0000315414829104
    0.0000261556814045
    ];
65
66 CYp = [0.0000391385074484
    0.0001591783632986
    0.0002792375120757
    ];
67
68 % Clp
69 C1 = polyfit(p, Clp, 1);
70 figure(4)
71 plot(p, Clp, 'r-', 'linewidth', 2)
72 hold on;
73 scatter(p, Clp, 'ro', 'MarkerFaceColor', 'r') % Red circles with
    filling
74 slope_text = sprintf('Pendenza: %.4f', C1(1));
75 text(p(1)+0.02, Clp(1)-0.003, slope_text, 'FontSize', 10, '
    HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');

```

```

84 hold off;
85 grid on;
86 title('Clp');
87 xlabel('p (rad/s)');
88 ylabel('CL');
89
90 % Cnp
91 Cn = polyfit(p, Cnp, 1);
92 figure(5)
93 plot(p, Cnp, 'r-', 'linewidth', 2)
94 hold on;
95 scatter(p, Cnp, 'ro', 'MarkerFaceColor', 'r') % Red circles with
    filling
96 slope_text = sprintf('Pendenza: %.4f', Cn(1));
97 text(p(2)+0.02, Cnp(2), slope_text, 'FontSize', 10, '
    HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
98 hold off;
99 grid on;
100 title('Cnp');
101 xlabel('p (rad/s)');
102 ylabel('CN');
103
104 % CYp
105 CY = polyfit(p, CYp, 1);
106 figure(6)
107 plot(p, CYp, 'r-', 'linewidth', 2)
108 hold on;
109 scatter(p, CYp, 'ro', 'MarkerFaceColor', 'r') % Red circles with
    filling
110 slope_text = sprintf('Pendenza: %.4f', CY(1));
111 text(p(2)+0.02, CYp(2), slope_text, 'FontSize', 10, '
    HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
112 hold off;
113 grid on;
114 title('CYp');
115 xlabel('p (rad/s)');
116 ylabel('CY');
117
118 derivate_p = [Cl(1) Cn(1) CY(1)]
119 adim_p = derivate_p * 2 * 20 / 2
120
121 %% DERIVATE in R
122 r = -[0 0.05 0.1];
123 cy = [0.0000391385074484
124       0.0004948243566669
125       0.0009510865296281
126 ];
127 c = polyfit(r, cy, 1);
128
129 figure(1)
130 plot(r, cy, '-o', 'linewidth', 2, 'color', [0, 0.6, 0], '
    MarkerFaceColor', [0, 0.6, 0])
131 hold on;
132 grid on;
133 title('Cyr');
134 xlabel('r (rad/s)');
135 ylabel('CY');

```

```

136 text(r(2)+0.002, cy(2), sprintf('Pendenza: %.4f', c(1)), 'FontSize'
    , 10, 'Color', [0, 0.6, 0]);
137
138 c1 = [-0.0000383200808147
139       0.0001147901329164
140       0.0001920266249104
141 ];
142 d = polyfit(r, c1, 1);
143
144 figure(2)
145 plot(r, c1, '-o', 'linewidth', 2, 'color', [0, 0.6, 0], '
    MarkerFaceColor', [0, 0.6, 0])
146 hold on;
147 grid on;
148 title('Clr');
149 xlabel('r (rad/s)');
150 ylabel('Cl');
151 text(r(2)+0.002, c1(2), sprintf('Pendenza: %.4f', d(1)), 'FontSize'
    , 10, 'Color', [0, 0.6, 0]);
152
153 cn = [0.0000367575866534
154       0.0001776734264252
155       0.0003185162053411
156 ];
157 e = polyfit(r, cn, 1);
158
159 figure(3)
160 plot(r, cn, '-o', 'linewidth', 2, 'color', [0, 0.6, 0], '
    MarkerFaceColor', [0, 0.6, 0])
161 hold on;
162 grid on;
163 title('Cnr');
164 xlabel('r (rad/s)');
165 ylabel('Cn');
166 text(r(2)+0.002, cn(2), sprintf('Pendenza: %.4f', e(1)), 'FontSize'
    , 10, 'Color', [0, 0.6, 0]);
167
168 derivate_r = [c(1) d(1) e(1)];
169 adim_r = derivate_r * 2 * 20 / 2;
170
171 %% Derivate in beta
172
173 beta = [0 1 5]*2*pi/360;
174 cy = [0.0000391385074484 -0.010940723238362138
175       -0.047074366742012576];
176
177 c = polyfit(beta, cy, 1)
178
179 c1 = [-0.0000383200808147 4.3341134779422574E-4
180       0.002051098010677344];
181 d = -polyfit(beta, c1, 1)
182 %
183 cn = [0.0000367575866534 -4.3742281335062483E-4 -6.800232571175142E
184       -4];
185 e = -polyfit(beta, cn, 1)
186
187 figure(1)
188 plot(beta, cy, 'm-', 'linewidth', 2)

```

```

185 hold on;
186 scatter(beta, cy, 'mo', 'MarkerFaceColor', 'm') % Cerchi magenta
    con riempimento
187
188 slope_text = sprintf('Pendenza: %.4f', c(1));
189 text(beta(2)+0.02, cy(2)+0.003, slope_text, 'FontSize', 10, '
    HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
190
191 hold off;
192 grid on;
193 title('Cyb');
194 xlabel('beta (rad)');
195 ylabel('CY');
196
197 figure(2)
198 plot(beta, cl, 'm-', 'linewidth', 2)
199 hold on;
200 scatter(beta, cl, 'mo', 'MarkerFaceColor', 'm') % Cerchi magenta
    con riempimento
201
202 slope_text = sprintf('Pendenza: %.4f', d(1));
203 text(beta(2), cl(2)+0.0004, slope_text, 'FontSize', 10, '
    HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
204
205 hold off;
206 grid on;
207 title('Clb');
208 xlabel('beta (rad)');
209 ylabel('Cl');
210
211 figure(3)
212 plot(beta, cn, 'm-', 'linewidth', 2)
213 hold on;
214 scatter(beta, cn, 'mo', 'MarkerFaceColor', 'm') % Cerchi magenta
    con riempimento
215
216 slope_text = sprintf('Pendenza: %.4f', e(1));
217 text(beta(2)+0.02, cn(2)+0.0002, slope_text, 'FontSize', 10, '
    HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom');
218
219 hold off;
220 grid on;
221 title('Cnb');
222 xlabel('b (rad)');
223 ylabel('Cn');

```

Listing 3: Calcolo derivate.

4 Calcolo matematico delle derivate

```

1 %% Derivate con formule
2 %dati
3 av = 3;
4 db = 0.3; %
5 dr = 0.1;

```



```

6 etav = 0.95; %
7 sv = 0.0388; %
8 s = 0.7 ; %
9 b = 2; %
10 zv = 0.15;
11 lv = 0.65;
12 c = 0.3; %
13 a = 5.22; %
14 p = [0 0.05 0.1];
15 r = [-0.1 -0.05 0];
16 beta = [0 5]*2*pi/360;
17 %% Derivate latero-direzionali
18
19 %Cyb (solo vtp)
20 cyb = -av*(1-db)*etav*sv/s;
21 cy = [
22     -4.8022411466058305E-6
23     -0.008827052018452887
24 ];
25 cyb2 = polyfit(beta,cy,1);
26
27 %clb (solo vtp) OK
28 clb = -etav*sv/s*zv/b*av*(1-db);
29
30 cl = [
31     5.011540898113128E-7
32     6.102051612000775E-4
33 ];
34 clb2 = -polyfit(beta,cl,1);
35
36 %cnb (solo vtp) OK
37 cnb = etav*sv/s*lv/b*av*(1-db);
38 cn = [
39     3.4396611393645147E-6
40     -0.00438305705016204
41 ];
42 cnb2 = -polyfit(beta,cn,1);
43
44 %cyp (solo vtp) OK
45 cyp = -2*sv/s*zv/b*av;
46 cy = [
47     -4.8022411466058305E-6
48     -1.0083626984074161E-4
49     -1.9679621867864383E-4
50 ];
51 cyp2 = polyfit(p,cy,1)*20;
52
53 %clp OK
54 clp_w = -4/s/b^2*a*c*integral(@(y) y.^2, 0, b/2);
55 clp_v = -2*sv/s*(zv/b)^2*av;
56 clp = clp_w+clp_v;
57
58 %cnp (solo vtp) OK
59 cnp = -cyp*lv/b;
60 cn = [
61     0.0000367575866534
62     1.976468269980923E-6

```

```

63     5.079366393633893E-7
64 ];
65 cnp2 =- polyfit(p,cn,1)*20;
66 %plot(p,cn)
67
68 %cyr (solo vtp) ??? SEGNO
69 cyr = sv/s*av*etav*(2*lv/b-dr);
70 cy =[3.61659326269009E-4
71     1.7824361480322777E-4
72     -4.8022411466058305E-6];
73 cyr2 = polyfit(r,cy,1)*20;
74
75 %clr (solo vtp) OK
76 clr = av*sv/s*zv/b*(2*lv/b-dr);
77 cl =[-2.4655118841995452E-5
78     -1.206533552377859E-5
79     5.011540898113128E-7];
80 clr2 = polyfit(r,cl,1)*20;
81
82
83 %cnr (solo vtp) OK
84 cnr = -cyr*lv/b;
85 cn =[8.796947593782375E-5
86     4.32926755360921E-5
87     -1.2843015415843904E-6];
88 cnr2 = polyfit(r,cn,1)*20;

```

Listing 4: Calcolo delle derivate.

5 Calcolo degli autovalori

```

1 %% AUTOVALORI LONGITUDINALI
2 %INPUT
3 m = 12; % [Kg]
4 u0 = 20; % [m/s]
5 S = 0.7; %[m2]
6 c = 0.3; %[m]
7 b = 2; % [m]
8 rho = 1.225; % [kg/m3]
9 qc = 0.5*rho*u0^2;
10 g0 = 9.81;
11
12 Ix = 0.5^2*2; % [kg*m2]
13 Iy = 0.35^2*m; % [kg*m2]
14 Iz = 0.35^2*m; % [kg*m2]
15
16 % lt = 0.5; % [m]
17 % at = 5;
18 % St = (0.3+0.2)*1;
19 %
20 CD0 = 0.0319; %valore CFD
21 CDa = 0.178771;
22 CDu = 0;
23 %
24 CLa = 5.248757;

```

```

25 CLu = 0;
26 CZq = 11.3378 ;% -aero_data.CL_q; %OpenVSP:6.17; -2 * at * lt * St
    / c / S;
27 %
28 CMa = -2.240018;
29 % openVSP: -16.6687330
30 CMap = 0;
31 CMq = -16.4566; % -2*at * lt / c / lt * St / c / S;
32 %
33 CZde = 0;%
34 CMde = 0;%
35 %
36 % CYb = aero_data.CY_beta;
37 % CYp = aero_data.CY_p;
38 % CYr = aero_data.CY_r;
39 % Clb = aero_data.Cl_beta;
40 % Clp = aero_data.Cl_p;
41 % Clr = aero_data.Cl_r;
42 % Cnb = aero_data.Cn_beta;
43 % Cnp = aero_data.Cn_p;
44 % Cnr = aero_data.Cn_r;
45 %
46 % CLda = -0.134;
47 % CLdr = 0.107;
48 % CNda = -0.0035;
49 % CNdr = -0.072;
50 % CYdr = 0.157;
51 % CYda = 0.0;
52 %
53 % derived
54 %qc = 0.5*rho*u0^2;
55 CL0 = 2*m*g0/S/rho/u0^2;
56 %
57 CXu = -(CDu + 2*CD0);
58 CXw = -(CDA - CL0);
59 CXq = -0.0219;
60 CXwp = 0;
61 %
62 CZu = -(CLu + 2*CL0);
63 CZw = -(CLA + CD0);
64 CZwp = 0;
65 %
66 %
67 CMu = 0;
68 %
69 %
70 Xu = CXu/u0*qc*S/m;
71 Xw = CXw/u0*qc*S/m;
72 Zu = CZu/u0*qc*S/m;
73 Zw = CZw*qc*S/m/u0;
74 Zwp = CZwp*c/2/u0*qc*S/m;
75 Zq = CZq*c/2/u0*qc*S/m;
76 Zde = CZde*qc*S/m;
77 Mu = CMu/u0*qc*S*c/Iy;
78 Ma = CMa*qc*S*c/Iy;
79 Mw = CMa*qc*S*c/Iy/u0;
80 Mwp = CMap*c/2/u0*qc*S*c/Iy/u0;

```

```

81 Mq = CMq*c/2/u0*qc*S*c/Iy;
82 Mde = CMde*qc*S*c/Iy;
83 %
84 % Yb = CYb*qc*S/m;
85 % Yp = CYp*qc*S/m;
86 % Yr = CYr*qc*S/m;
87 % Lb = Clb*qc*S*b/Ix;
88 % Lp = Clp*qc*S*b/Ix;
89 % Lr = Clr*qc*S*b/Ix;
90 % Nb = Cnb*qc*S*b/Iz;
91 % Np = Cnp*qc*S*b/Iz;
92 % Nr = Cnr*qc*S*b/Iz;
93 %
94 % Lda = CLda*qc*S*b/Ix;
95 % Ldr = CLdr*qc*S*b/Ix;
96 % Nda = -CNda*qc*S*b/Iz;
97 % Ndr = CNdr*qc*S*b/Iz;
98 % Yda = CYda*qc*S/m;
99 % Ydr = CYdr*qc*S/m;
100
101 % Equilibrio
102 alpha0 = 0;
103 ub0 = u0;
104 wb0 = 0;
105
106
107 % matrice che moltiplica il vento
108 E = [-Xu -Xw 0;...
109      -Zu -Zw 0;...
110      -(Mu+Mwp*Zu) -(Mw+Mwp*Zw) -(Mq+Mwp*u0);...
111      0 0 0];
112
113 A = [Xu Xw 0 -g0;...
114      Zu Zw u0 0;...
115      (Mu+Mwp*Zu) (Mw+Mwp*Zw) (Mq+Mwp*u0) 0;...
116      0 0 1 0];
117
118 [lambda] = eig(A);
119
120
121 % Grafico
122 figure('Color', 'w'); % Sfondo bianco
123 h1 = plot(real(lambda(3:4)), imag(lambda(3:4)), '*r', 'MarkerSize',
124           10, 'LineWidth', 2);
125 hold on
126 h2 = plot(real(lambda(1:2)), imag(lambda(1:2)), '*g', 'MarkerSize',
127           10, 'LineWidth', 2);
128
129 % Assi x e y
130 xlim([-5, max(real(lambda))+2]);
131 plot(xlim, [0, 0], 'k--', 'LineWidth', 0.5);
132 plot([0, 0], ylim, 'k--', 'LineWidth', 0.5);
133
134 % Decorazioni aggiuntive
135 title('Autovalori della dinamica longitudinale', 'FontSize', 16);
136 legend('Fugoid', 'Corto Periodo', 'FontSize', 12);
137 xlabel('Parte Reale', 'FontSize', 14);

```

```

136 ylabel('Parte Immaginaria', 'FontSize', 14);
137 grid on
138 box on
139
140 % Imposta manualmente i limiti dell'asse x per spostarlo verso
    sinistra
141
142
143 % Miglioramento della leggibilit degli assi
144 set(gca, 'FontSize', 12, 'LineWidth', 1.2);
145
146 % Personalizzazione dei marker
147 h1.MarkerEdgeColor = 'r';
148 h2.MarkerEdgeColor = 'g';
149
150 % Aggiunta delle coordinate vicino ai punti del grafico
151 text(real(lambda) + 0.1, imag(lambda), ...
152      arrayfun(@(x) sprintf('%.2f + %.2fi', real(x), imag(x)), lambda
153      , 'UniformOutput', false), ...
154      'FontSize', 10, 'VerticalAlignment', 'middle', '
    HorizontalAlignment', 'left');
155
156 approx_p_omega = sqrt(-Zu*g0/u0);
157 approx_p_zeta = -Xu/2/approx_p_omega;
158 approx_sp_omega = sqrt(Zw*Mq - Ma);
159 approx_sp_zeta = -(Mq + Mwp/u0 + Zw)/2/approx_sp_omega;
160
161 fprintf('Modo Fugoide Approssimato: omega = %f [rad/s], zeta = %f\
    nModo Corto Periodo approssimato: omega = %f [rad/s], zeta = %f
    \n',...
162      approx_p_omega,...
163      approx_p_zeta,...
164      approx_sp_omega,...
165      approx_sp_zeta);
166
167 fprintf( 'Modo 1 = Periodo = %f, t/2 = %f, omega = %f [rad/s], zeta
    = %f\n',...
168      2*pi/abs(imag(lambda(end))),...
169      -0.69/real(lambda(end)),...
170      abs(lambda(end)),...
171      sqrt( 1 / (1 + (imag(lambda(end))/real(lambda(end)))^2) ));
172 fprintf( 'Modo 2 = Periodo = %f, t/2 = %f, omega = %f [rad/s], zeta
    = %f\n',...
173      2*pi/abs(imag(lambda(1))),...
174      -0.69/real(lambda(1)),...
175      abs(lambda(1)),...
176      sqrt( 1 / (1 + (imag(lambda(1))/real(lambda(1)))^2) ));
177
178 tausp = m/(qc*S*CLa)
179 cap = m*g0*approx_sp_omega^2/(qc*S*CLa)
180 na =9.7^2/cap
181
182 %% TIME RESPONSE FOR corto periodo
183 % System matrix A for phugoid mode
184 A_phugoid = [0 1; 0 lambda(1)];
185

```

```

186 % Time vector
187 t = 0:0.1:20;
188
189 % Compute the response using sinusoidal functions
190 omega = imag(lambda(1)); % Frequency of oscillation
191 phi = atan2(imag(lambda(1)), real(lambda(1))); % Phase angle
192 % Mode 1
193 y1 = exp(real(lambda(1))*t) .* (cos(omega*t + phi));
194
195 % Mode 2 (conjugate)
196 y2 = exp(real(lambda(2))*t) .* (cos(omega*t + phi));
197
198 % Combine both modes
199 y_phugoid = y1 + y2;
200
201 % Plot the response
202 figure;
203 plot(t, real(y_phugoid), 'LineWidth', 2);
204 title('Short Period Mode - Time Response');
205 xlabel('Time');
206 ylabel('Amplitude');
207 grid on;
208
209 %% TIME RESPONSE FOR fugoide
210 % System matrix A for short period mode
211 A_short_period = [0 1; 0 lambda(3)];
212
213 % Time vector
214 t = 0:1:500;
215
216 % Compute the response using sinusoidal functions
217 omega = imag(lambda(3)); % Frequency of oscillation
218 phi = atan2(imag(lambda(3)), real(lambda(3))); % Phase angle
219
220 % Mode 1
221 y1_short_period = exp(real(lambda(3))*t) .* (cos(omega*t + phi));
222
223 % Mode 2 (conjugate)
224 y2_short_period = exp(real(lambda(4))*t) .* (cos(omega*t + phi));
225
226 % Combine both modes
227 y_short_period = y1_short_period + y2_short_period;
228
229 % Plot the response
230 figure;
231 plot(t, real(y_short_period), 'LineWidth', 2);
232 title('Phugoid Mode - Time Response');
233 xlabel('Time');
234 ylabel('Amplitude');
235 grid on;
236
237 %% AUTOVALORI LATERO-DIREZIONALI
238 % INPUT
239 m = 12; % [Kg]
240 u0 = 20; % [m/s]
241 S = 0.7; % [m2]
242 c = 0.3; % [m]

```

```

243 b = 2; % [m]
244 rho = 1.225; % [kg/m3]
245 qc = 0.5*rho*u0^2;
246 g0 = 9.81;
247 theta0 = 0;
248
249 Ix = 0.5^2*2; % [kg*m2]
250 Iy = 0.35^2*m; % [kg*m2]
251 Iz = 0.35^2*m; % [kg*m2]
252
253 % lt = 0.5; % [m]
254 % at = 5;
255 % St = (0.3+0.2)*1;
256 %
257 CD0 = 0.0319; %valore CFD
258 CDa = 0.178771;
259 CDu = 0;
260 %
261 CLa = 5.194391;
262 CLu = 0;
263 CZq = 11.3378 %OpenVSP:6.17; -2 * at * lt * St / c / S;
264 %
265 CMa = -2.224921;
266 % openVSP: -16.6687330
267 CMap = 0;
268 CMq = -16.4566; %-2*at * lt / c / lt * St / c / S;
269 %
270 CZde = 0;%
271 CMde = 0;%
272
273 CYb = -0.5335;
274 CYp = 0.0480;
275 CYr = -0.1824;
276 Clb = -0.0237; %aero_data.Cl_beta; %da calcolare con CFD
277 Clp = -0.6343;
278 Clr = 0.0307;
279 Cnb = 0.0069; %aero_data.Cn_beta;
280 Cnp = -0.0021;
281 Cnr = -0.0564;
282
283 CLda = -0.134;
284 CLdr = 0.107;
285 CNda = -0.0035;
286 CNdr = -0.072;
287 CYdr = 0.157;
288 CYda = 0.0;
289
290 % derived
291 qc = 0.5*rho*u0^2;
292 CL0 = 2*m*g0/S/rho/u0^2;
293 %
294 CXu = -(CDu + 2*CD0);
295 CXw = -(CDa - CL0);
296 CXq = 0;
297 CXwp = 0;
298 %
299 CZu = -(CLu + 2*CL0);

```

```

300 CZw = -(CLa + CD0);
301 CZwp = 0;
302
303 %
304 CMu = 0;
305
306 %
307 Xu = CXu/u0*qc*S/m;
308 Xw = CXw/u0*qc*S/m;
309 Zu = CZu/u0*qc*S/m;
310 Zw = CZw*qc*S/m/u0;
311 Zwp = CZwp*c/2/u0*qc*S/m;
312 Zq = CZq*c/2/u0*qc*S/m;
313 Zde = CZde*qc*S/m;
314 Mu = CMu/u0*qc*S*c/Iy;
315 Ma = CMa*qc*S*c/Iy;
316 Mw = CMa*qc*S*c/Iy/u0;
317 Mwp = CMap*c/2/u0*qc*S*c/Iy/u0;
318 Mq = CMq*c/2/u0*qc*S*c/Iy;
319 Mde = CMde*qc*S*c/Iy;
320 %
321 Yb = CYb*qc*S/m;
322 Yp = CYp*qc*S/m*b/2/u0;
323 Yr = CYr*qc*S/m*b/2/u0;
324 Lb = Clb*qc*S*b/Ix;
325 Lp = Clp*qc*S*b/Ix*b/2/u0;
326 Lr = Clr*qc*S*b/Ix*b/2/u0;
327 Nb = Cnb*qc*S*b/Iz;
328 Np = Cnp*qc*S*b/Iz*b/2/u0;
329 Nr = Cnr*qc*S*b/Iz*b/2/u0;
330
331 Lda = CLda*qc*S*b/Ix;
332 Ldr = CLdr*qc*S*b/Ix;
333 Nda = -CNda*qc*S*b/Iz;
334 Ndr = CNdr*qc*S*b/Iz;
335 Yda = CYda*qc*S/m;
336 Ydr = CYdr*qc*S/m;
337
338 % Equilibrio
339 alpha0 = CL0/CLa;
340 ub0 = u0*cos(alpha0);
341 wb0 = u0*sin(alpha0);
342 %
343 % vettore di stato X = [u, w, q, pitch, pn_p, pe_p, pd_p]
344
345 E = [-Xu -Xw 0;...
346      -Zu -Zw 0;...
347      -(Mu+Mwp*Zu) -(Mw+Mwp*Zw) -(Mq+Mwp*u0);...
348      0 0 0];
349
350 A = [Yb/u0 Yp/u0 -(1 - Yr/u0) g0*cos(theta0)/u0;...
351      Lb Lp Lr 0;...
352      Nb Np Nr 0;...
353      0 1 0 0];
354 lambda = eig(A);
355 % Grafico
356 figure('Color', 'w'); % Sfondo bianco

```



```

357 h1 = plot(real(lambda(1)), imag(lambda(1)), '*r', 'MarkerSize', 10,
    'LineWidth', 2);
358 hold on
359 h2 = plot(real(lambda(2:3)), imag(lambda(2:3)), '*g', 'MarkerSize',
    10, 'LineWidth', 2);
360 hold on
361 h3 = plot(real(lambda(4)), imag(lambda(4)), '*m', 'MarkerSize', 10,
    'LineWidth', 2);
362
363 % Assi x e y
364 xlim([-25, max(real(lambda))+2]);
365 plot(xlim, [0, 0], 'k--', 'LineWidth', 0.5);
366 plot([0, 0], ylim, 'k--', 'LineWidth', 0.5);
367
368 % Decorazioni aggiuntive
369 title('Autovalori della dinamica latero-direzionale', 'FontSize',
    16);
370 legend('Roll mode', 'Dutch roll', 'Modo spirale', 'FontSize', 12);
371 xlabel('Parte Reale', 'FontSize', 14);
372 ylabel('Parte Immaginaria', 'FontSize', 14);
373 grid on
374 box on
375
376 % Imposta manualmente i limiti dell'asse x per spostarlo verso
    sinistra
377
378
379 % Miglioramento della leggibilit degli assi
380 set(gca, 'FontSize', 12, 'LineWidth', 1.2);
381
382 % Personalizzazione dei marker
383 h1.MarkerEdgeColor = 'r';
384 h2.MarkerEdgeColor = 'g';
385
386 % Aggiunta delle coordinate vicino ai punti del grafico
387 text(real(lambda) + 0.1, imag(lambda), ...
388     arrayfun(@(x) sprintf('%0.2f + %0.2fi', real(x), imag(x)), lambda,
    'UniformOutput', false), ...
389     'FontSize', 10, 'VerticalAlignment', 'middle', '
    HorizontalAlignment', 'left');
390
391
392 fprintf( 'Modo 1 = Periodo = %f, t/2 = %f\n', ...
393     2*pi/abs(imag(lambda(end))), ...
394     -0.69/real(lambda(end)) );
395 fprintf( 'Modo 2 = Periodo = %f, t/2 = %f\n', ...
396     2*pi/abs(imag(lambda(1))), ...
397     -0.69/real(lambda(1)) );
398 fprintf( 'Modo 3 = Periodo = %f, t/2 = %f, omega = %f, zeta = %f\n'
    , ...
399     2*pi/abs(imag(lambda(2))), ...
400     -0.69/real(lambda(2)), ...
401     abs(lambda(2)), ...
402     sqrt( 1 / (1 + (imag(lambda(2))/real(lambda(2)))^2) ) );
403
404 tau_CB = -1/lambda(1)
405

```

```

406 %% TIME RESPONSE FOR roll mode
407 % System matrix A for roll mode
408 A_phugoid = [0 1; 0 lambda(1)];
409
410 % Time vector
411 t = 0:0.1:3;
412
413 % Compute the response using sinusoidal functions
414 omega = imag(lambda(1)); % Frequency of oscillation
415 phi = atan2(imag(lambda(1)), real(lambda(1))); % Phase angle
416 % Mode 1
417 y1 = exp(real(lambda(1))*t) .* (cos(omega*t + phi));
418
419 % Combine both modes
420 y_roll = y1;
421
422 % Plot the response
423 figure;
424 plot(t, real(y_roll), 'LineWidth', 2);
425 title('Roll Mode - Time Response');
426 ylim([-1.1 0.6])
427 xlabel('Time');
428 ylabel('Amplitude');
429 grid on;
430
431 %% TIME RESPONSE FOR dutch roll
432 % System matrix A for dutch roll
433 A_phugoid = [0 1; 0 lambda(2)];
434
435 % Time vector
436 t = 0:0.1:20;
437
438 % Compute the response using sinusoidal functions
439 omega = imag(lambda(2)); % Frequency of oscillation
440 phi = atan2(imag(lambda(2)), real(lambda(2))); % Phase angle
441 % Mode 1
442 y1 = exp(real(lambda(2))*t) .* (cos(omega*t + phi));
443
444 % Mode 2 (conjugate)
445 y2 = exp(real(lambda(3))*t) .* (cos(omega*t + phi));
446
447 % Combine both modes
448 y_dr = y1 + y2;
449
450 % Plot the response
451 figure;
452 plot(t, real(y_dr), 'LineWidth', 2);
453 title('Dutch Roll Mode - Time Response');
454 xlabel('Time');
455 ylabel('Amplitude');
456 grid on;
457
458 %% TIME RESPONSE FOR spiral mode
459 % System matrix A for spiral mode
460 A_phugoid = [0 1; 0 lambda(4)];
461
462 % Time vector

```

```

463 t = 0:0.1:200;
464
465 % Compute the response using sinusoidal functions
466 omega = imag(lambda(4)); % Frequency of oscillation
467 phi = atan2(imag(lambda(4)), real(lambda(4))); % Phase angle
468 % Mode 1
469 y1 = exp(real(lambda(4))*t) .* (cos(omega*t + phi));
470
471 % Combine both modes
472 y_s = y1;
473
474 % Plot the response
475 figure;
476 plot(t, real(y_s), 'LineWidth', 2);
477 title('Spiral Mode - Time Response');
478 ylim([-2 1])
479 xlabel('Time');
480 ylabel('Amplitude');
481 grid on;
482
483
484 tauSM = -1/real(lambda(4))
485 tauRM = -1/real(lambda(1))
486 tauSM/tauRM

```

Listing 5: Calcolo degli autovalori.