

Flight Dynamics with Matlab/Simulink

Elaborato di Dinamica
e simulazione di volo

Alessandro Lucignano



Indice

1 Cinematica del velivolo	7
1.1 Virata di Immelmann	7
1.1.1 Codice Matlab e Risultati	8
2 Non linear Aircraft Performance Equations (NAPE)	14
2.1 Introduzione	14
2.2 Codice matlab	14
2.3 Modello Simulink	17
2.4 Risultati	33
3 Equazioni del moto	36
3.1 Introduzione	36
3.2 Esercizio 7.3 con HARV	36
3.3 Implementazione del modello aerodinamico del Citation I	38
3.3.1 Funzioni matlab	41
3.4 Utilizzo del nuovo modello per simulare una virata	44
3.5 Ricerca delle condizioni di trim per moto a 6DOF per il Citation I	49
4 Superfici di governo	56
4.1 Volo longitudinal-simmetrico a comandi liberi	56
4.1.1 Modello aerodinamico 3DOF	56
4.1.2 Coefficiente di momento aerodinamico elevatore	58
4.1.3 Risultati	59
4.2 Visualizzazione dell'accoppiamento tra dinamica del velivolo e dinamica dell'equilibratore	62
4.2.1 Modello 6DOF	62
4.2.2 Variabili di volo e risultati	65
5 Piccole perturbazioni del moto di un velivolo	68
5.1 Risposta di un velivolo ad una raffica verticale	68
5.1.1 Codice Matlab	68
5.1.2 Risultati	72
5.2 Implementazione di un controllo in retroazione	76
5.2.1 Modello simulink e risultati	76

6 Digital Datcom	80
6.0.1 File di configurazione per il Citation I	80
6.0.2 Risultati	84

Elenco delle figure

1.1	Virata di Immleman	11
1.2	Coordinate del baricentro	12
1.3	Andamento nel tempo degli angoli di Eulero	12
1.4	Settaggio delle velocità angolari nel signal editor	13
2.1	Traiettoria di volo	33
2.2	Traiettoria di volo	34
2.3	Variazione delle variabili nel tempo	35
3.1	Output del programma	38
3.2	Coordinate ed angoli di Eulero	45
3.3	Parametri della velocità e velocità angolare del velivolo	46
3.4	Comandi impartiti	46
3.5	Manetta del motore	47
3.6	Visualizzazione della manovra, prima prospettiva	47
3.7	Visualizzazione della manovra, seconda prospettiva	48
3.8	Variabili minimizzate	50
3.9	Coordinate e angoli di Eulero	51
3.10	Visualizzazione di 20 secondi di volo	55
4.1	Parametri di volo	59
4.2	Parametri di volo	60
4.3	Traiettoria	61
4.4	Deflessione superfici di controllo	66
4.5	Cinematica del velivolo	66
4.6	Deflessione elevatore con e senza accoppiamento	67
5.1	Risposta libera del velivolo alle condizioni iniziali	72
5.2	Variazione degli autovalori	73
5.3	Variazione dell'angolo β	73
5.4	Variazione di p	74
5.5	Variazione di r	74
5.6	Variazione di ϕ	75
5.7	Schermata del tool PID Tuner	76
5.8	Dinamica del velivolo con controllo in retroazione	78
5.9	Risposta degli alettoni	79

6.1	Curva del CL	85
6.2	Curva del CD	86
6.3	Curva del momento di beccheggio al variare di α_B	87
6.4	Variazione del momento di beccheggio dovuta all'elevatore	87
6.5	Momento di rollio dovuto alla deflessione degli alettoni	88
6.6	Momento di imbardata dovuto alla deflessione del timone	89
6.7	Downwash visto dal piano di coda	90

Capitolo

1

Cinematica del velivolo

1.1 Virata di Immelmann

Nel corso di questa esercitazione vi 'e la descrizione cinematica del moto di un velivolo nel corso di una manovra detta "virata di Immelmann" calcolata utilizzando la formulazione con il quaternione dell'orientamento che permette di evitare il fenomeno del Gimbal Lock. Non viene simulata la dinamica del velivolo ma piuttosto vengono prescritte le velocità angolari in assi body: $\{p, q, r\}$.

Data la breve durata del volo e le relativamente piccole distanze percorse la curvatura della superficie terrestre è ignorata. Segue una descrizione high level dell'algoritmo.

1. Si assegnano le velocità angolari $\{p, q, r\}$ tramite il Simulink Signal Editor;
2. Si importano i segnali nel workspace di matlab e si trasformano in funzioni utili all'esecuzione dell'algoritmo
3. Si assegnano i valori iniziali delle coordinate spaziali, della velocità e degli angoli di Eulero
4. Usando la funzione *angle2quat* si ottiene il quaternione iniziale
5. Vengono implementate le equazioni (1.1) in una funzione matlab dedicata che sarà utilizzata da *ode45*
6. Si utilizza *ode45* per ottenere l'andamento temporale del quaternione
7. Vengono utilizzate le equazioni (3.28) discusse nel quaderno 3 che permettono di trasformare le velocità in assi body in assi terra utilizzando direttamente i valori del quaternione
8. Si utilizzano le funzioni di display del volo fornite e opportunamente modificate

$$\begin{Bmatrix} \dot{q}_0 \\ \dot{q}_x \\ \dot{q}_y \\ \dot{q}_z \end{Bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{Bmatrix} q_0 \\ q_x \\ q_y \\ q_z \end{Bmatrix} \quad (1.1)$$

1.1.1 Codice Matlab e Risultati

Listing 1.1

```

1 %% Esercitazione 3.3
2 %% Soluzione problema a valori iniziali, virata di Immlemann
3 clc; clear; close all;
4 global q p r VXE VYE VZE;
5 % angVel=[1,1,1];
6 % quat=[1,2,1,1];
7 % quatDer(angVel,quat)
8
9 %% Parametri simulazione
10
11
12 Nt=50;
13
14 %% Definizioni leggi velocit angolari
15 %Con questo metodo visualizzare manovre immediato perch si possono
16 %modificare i valori di pqr in maniera molto veloce nel signal editor
17
18
19 flightScenario = load('Scenario_Immlemann.mat');
20
21 T = max([max(flightScenario.p.Time),max(flightScenario.q.Time),max(
22     flightScenario.r.Time)]);%Durata della simulazione
23 p = @(t) convangvel(interp1(flightScenario.p.Time,flightScenario.p.Data,t
24     , 'previous','extrap'), 'deg/s','rad/s');
25 q = @(t) convangvel(interp1(flightScenario.q.Time,flightScenario.q.Data,t
26     , 'previous','extrap'), 'deg/s','rad/s');
27 r = @(t) convangvel(interp1(flightScenario.r.Time,flightScenario.r.Data,t
28     , 'previous','extrap'), 'deg/s','rad/s');
29 timeBrPoints=[0*T,0.25*T,0.5*T,0.75*T,1*T];
30
31 % qmax=convang(2.5,'deg','rad');
32 % qBrPoints=[0*qmax,1*qmax,1*qmax,1*qmax,0*qmax];
33 % q=@(t) interp1(timeBrPoints,qBrPoints,t,'pchip');
34 %
35 % pmax=convang(5,'deg','rad');
36 % pBrPoints=[0*pmax,1*pmax,1*pmax,1*pmax,0*pmax];
37 % p=@(t) interp1(timeBrPoints,pBrPoints,t,'pchip');
38 %
39 % t=linspace(0,T,60);
40 % leggeAng=figure(1);
41 % plot(t,q(t));
42 % title("Velocit angolare q (rad/s)");
43
44 %% Definizione leggi velocit assi body

```

```

41 u0 = convvel(380.0, 'km/h', 'm/s');
42 v0 = convvel( 0.0, 'km/h', 'm/s');
43 w0 = convvel( 0.0, 'km/h', 'm/s');
44
45 uBrPoints=[1*u0,0.7*u0,0.7*u0,0.7*u0,1*u0];
46 u=@(t) interp1(timeBrPoints,uBrPoints,t,'pchip');
47
48
49 %% Valori iniziali
50 %Posizioni
51 PositionsE0=[0, 0, -1000];
52 %Angoli
53 anglesE0=[deg2rad(0.1),0,0]; %Psi,theta,phi
54 anglesE0Q=angle2quat(anglesE0(1), anglesE0(2), anglesE0(3), 'ZYX');
55
56
57
58 %% Integrazione del problema
59 timeBrPointsRK=linspace(0,T,Nt);
60 VE=zeros(Nt,3);
61 options = odeset('RelTol',1e-15,'AbsTol',1e-15*ones(1,4));
62 [Times,quaternion]=ode89(@quatDer,timeBrPointsRK,anglesE0Q,options);
63 %Velocit
64 for it=1:Nt
65     t=Times(it);
66     q0=quaternion(it,1);
67     qx=quaternion(it,2);
68     qy=quaternion(it,3);
69     qz=quaternion(it,4);
70     MatV=zeros(3);
71     MatV(1,:)=[(q0^2 + qx^2 - qy^2 - qz^2), 2*(qx*qy - q0*qz), 2*(qz*qx + q0*qy)];
72     MatV(2,:)=[2*(qx*qy + q0*qz), (q0^2 - qx^2 + qy^2 - qz^2), 2*(qy*qz - q0*qx)];
73     MatV(3,:)=[2*(qz*qx - q0*qy), 2*(qy*qz + q0*qx), (q0^2 - qx^2 - qy^2 + qz^2)];
74     utemp=u(t);
75     VB=[utemp; 0; 0];
76     VE(it,:)=MatV*VB;
77 end
78
79 %Interpolazione e creazione delle leggi della velocit in assi terra
80 VXE=VE(:,1)';
81 VYE=VE(:,2)';
82 VZE=VE(:,3)';
83
84 VXE=@(t) interp1(Times,VXE,t,'');
85 VYE=@(t) interp1(Times,VYE,t,'pchip');
86 VZE=@(t) interp1(Times,VZE,t,'pchip');
87 %Integrazione con RK sulle VE per ottenere le posizioni
88
89 [Times,positions]=ode45(@posDer,timeBrPointsRK,PositionsE0);
90
91
92
93 %% Plotting

```

```
94 %Angoli
95 plotAngoli = figure(1);
96 title("Angoli assi terra")
97 [psi, theta, phi]=quat2angle(quaternion, 'ZYX');
98
99
100
101 % psi=convang(psi,'rad','deg');
102 % theta=convang(theta,'rad','deg');
103 % phi=convang(phi,'rad','deg');
104 plot(Times,rad2deg(psi), Times,rad2deg(theta), Times, rad2deg(phi));
105 legend("Psi","Theta","Phi");
106 grid on;
107 %Posizioni
108 plotPos = figure(2);
109 title("Posizione assi terra")
110 plot(Times,positions(:,1),Times,positions(:,2),Times,-positions(:,3));
111 legend("X","Y","Z");
112 grid on;
113
114 %% Traiettoria e modello
```

Come accennato in precedenza le leggi di $\{p, q, r\}$ sono state definite ed implementate utilizzando il simulink SignalEditor, questo permette di cambiare il tipo di manovra visualizzata in maniera veloce ed intuitiva.

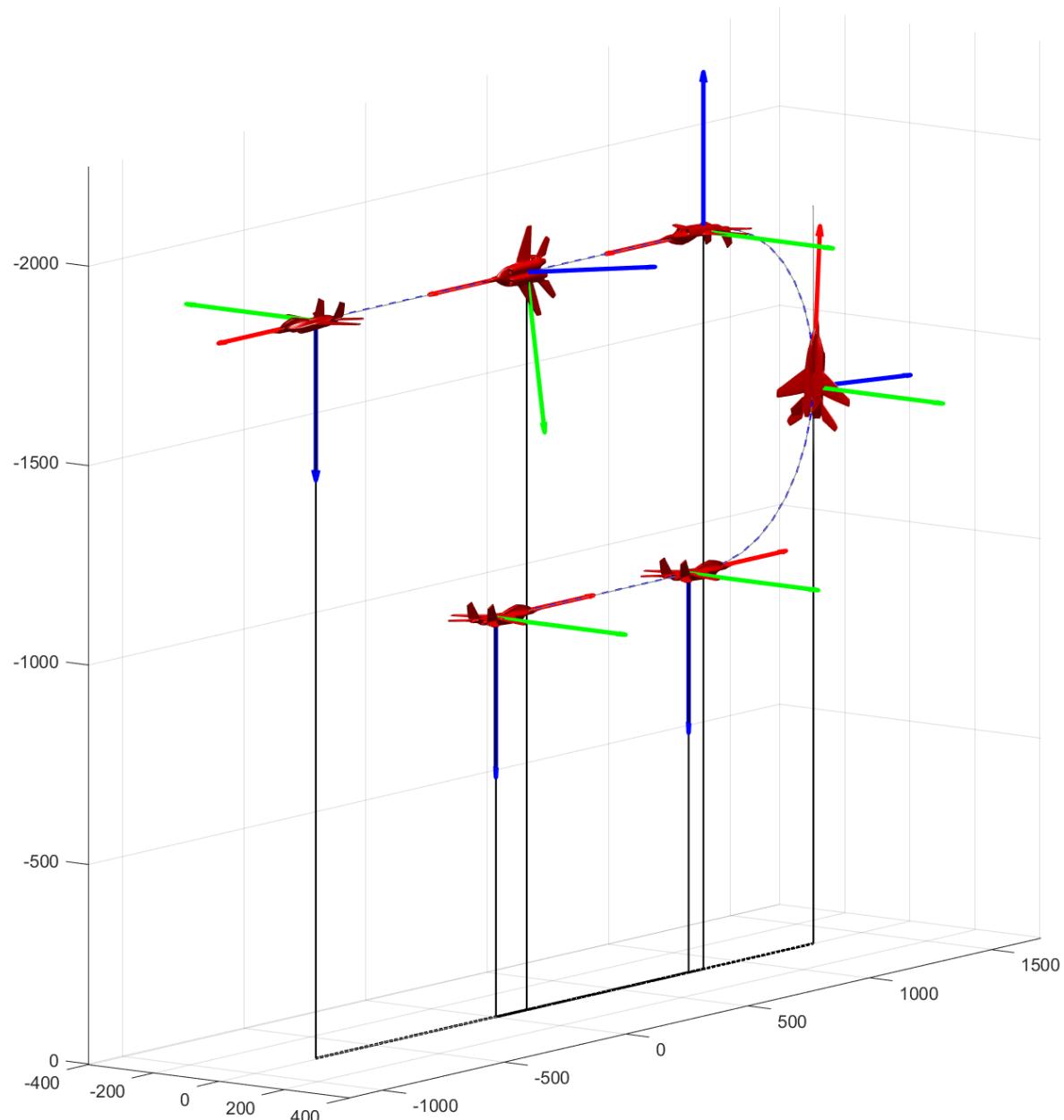


Figura 1.1 Virata di Immleman

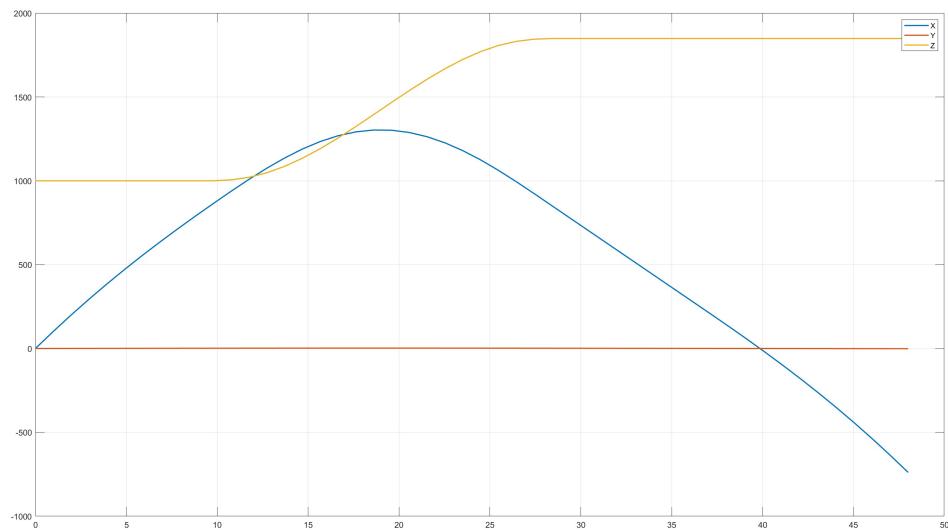


Figura 1.2 Coordinate del baricentro

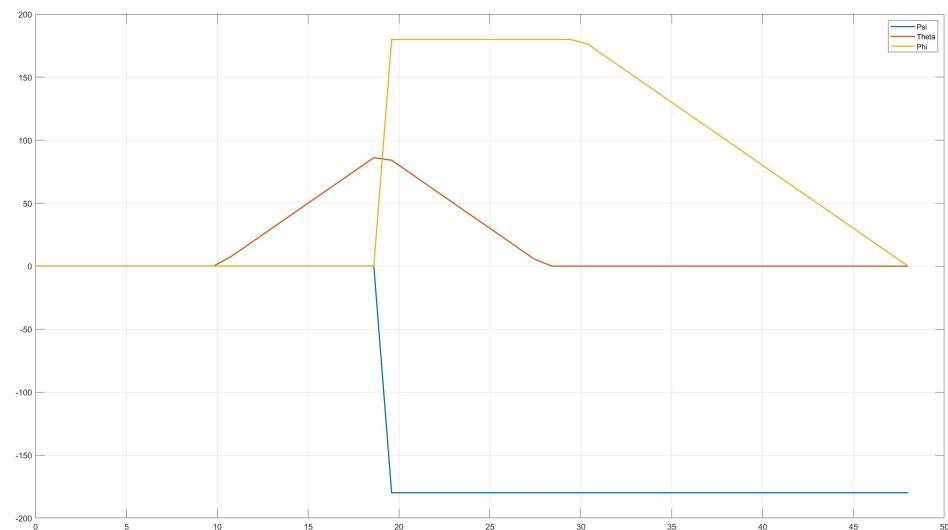


Figura 1.3 Andamento nel tempo degli angoli di Eulero

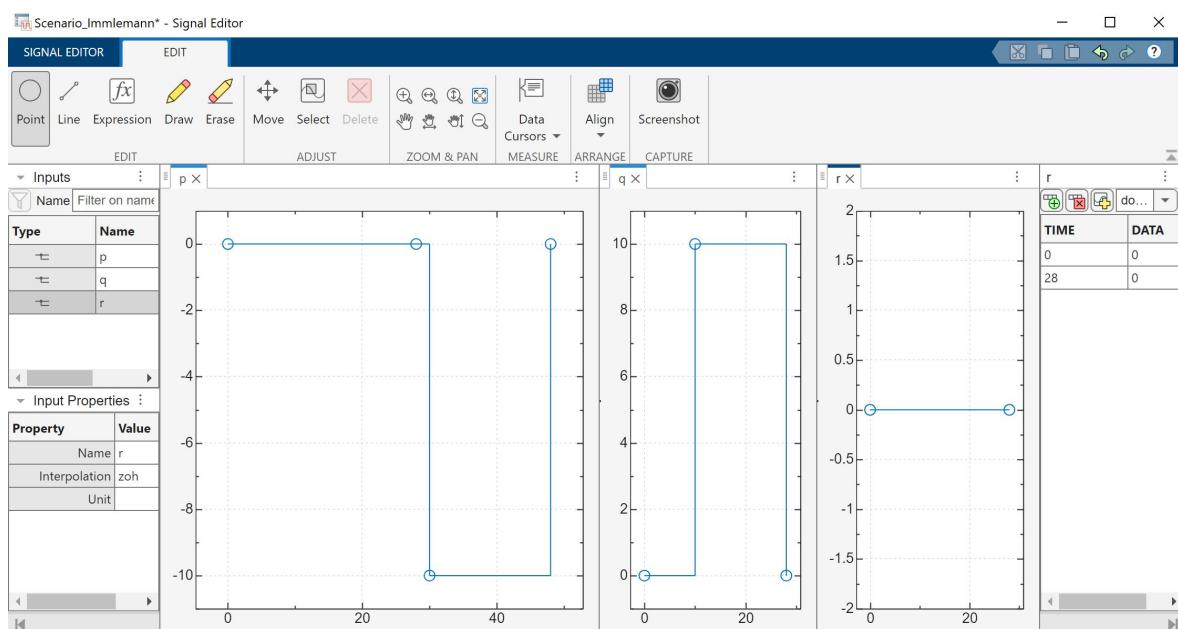


Figura 1.4 Settaggio delle velocità angolari nel signal editor

Capitolo

2

Non linear Aircraft Performance Equations (NAPE)

2.1 Introduzione

In questo capitolo viene riportato lo svolgimento dell'esercizio dell'esercizio 5 del quaderno 6 consistente nell'implementazione in simulink delle equazioni NAPE (2.4) per un Lockheed C-130 con V , γ , δ come parametri comandati variabili nel tempo.

Le leggi di variazione di T , L e v sono proporzionali all'errore tra la variabile comandata e lo stato effettivo della variabile, come mostrato nelle ??-(2.3).

$$\dot{T} = p_T(T_c - T) \quad (2.1)$$

$$\dot{L} = p_L(L_c - L) \quad (2.2)$$

$$\dot{v} = p_v(v_c - v) \quad (2.3)$$

$$\begin{cases} \dot{V} = \frac{T-D}{m} - g \sin(\gamma) \\ \dot{\gamma} = \frac{L \cos(v) - mg \cos(\gamma)}{mV} \\ \dot{\delta} = \frac{L \sin(v)}{mV \cos(\gamma)} \end{cases} \quad (2.4)$$

2.2 Codice matlab

Il seguente codice simula il modello simulink ed elabora i suoi output.

Listing 2.1

```
1 clc; clear; close all;
2
3 simulation = sim("E6_5.slx");
4 alphaB = simulation.yout.getElement('alphaB');
5 alphaB=alphaB.Values;
6 delta = simulation.yout.getElement('delta');
```

```

7 delta=delta.Values;
8 gamma = simulation.yout.getElement('gamma');
9 gamma=gamma.Values;
10 ni = simulation.yout.getElement('ni');
11 ni=ni.Values;
12 Xe = simulation.yout.getElement('xg');
13 Xe=Xe.Values;
14 Ye = simulation.yout.getElement('yg');
15 Ye=Ye.Values;
16 Ze = simulation.yout.getElement('hg');
17 Ze=Ze.Values;
18 T=simulation.yout.getElement('T');
19 T=T.Values;
20 L=simulation.yout.getElement('L');
21 L=L.Values;
22 V=simulation.yout.getElement('V');
23 V=V.Values;
24
25 Tc=simulation.yout.getElement('TComando');
26 Tc=Tc.Values;
27 Tmax=simulation.yout.getElement('Tmax');
28 Tmax=Tmax.Values;
29 D=simulation.yout.getElement('D');
30 D=D.Values;
31 Vinf=simulation.yout.getElement('Vinfinito');
32 Vinf=Vinf.Values;
33 Lc=simulation.yout.getElement('LComando');
34 Lc=Lc.Values;
35 Vc=simulation.yout.getElement('VComando');
36 Vc=Vc.Values;
37
38 %Devo convertire gli angoli dati dalla simulazione in phi theta e psi
39 %Rotazione 321
40 %Angoli in gradi
41 niInterp=@(t) interp1(ni.Time,ni.Data, t,"linear");
42 gammaInterp=@(t) interp1(gamma.Time,gamma.Data, t,"linear");
43 deltaInterp=@(t) interp1(delta.Time,delta.Data,t,"linear");
44 XeInterp=@(t) interp1(Xe.Time,Xe.Data,t,"linear");
45 YeInterp=@(t) interp1(Ye.Time,Ye.Data,t,"linear");
46 ZeInterp=@(t) interp1(Ze.Time,-Ze.Data,t,"linear");
47 LInterp=@(t) interp1(L.Time,L.Data,t,"linear");
48 TInterp=@(t) interp1(T.Time,T.Data,t,"linear");
49 VInterp=@(t) interp1(V.Time,V.Data,t,"linear");
50 alphaBInterp=@(t) interp1(alphaB.Time,alphaB.Data,t,"linear");
51 TcInterp=@(t) interp1(Tc.Time,Tc.Data,t,"linear");
52 TmaxInterp=Tmax.Data;
53 DInterp=@(t) interp1(D.Time,D.Data,t,"linear");
54 VinfInterp=@(t) interp1(Vinf.Time,Vinf.Data,t,"linear");
55 LcInterp=@(t) interp1(Lc.Time,Lc.Data,t,"linear");
56 VcInterp=@(t) interp1(Vc.Time,Vc.Data,t,"linear");
57
58
59
60 %% Plotting
61 t_fin = simulation.SimulationMetadata.ModelInfo.StopTime*0.99;
62 Nt=500;

```

```

63 timeSamplesDisplay = linspace(0,t_fin,Nt);
64
65 figure(7)
66 subplot(5,1,1);
67 title("Trust and drag");
68 hold on;
69 yline(TmaxInterp,LineStyle='--',LineWidth=1);
70 plot(timeSamplesDisplay, TIInterp(timeSamplesDisplay),LineWidth=1);
71 plot(timeSamplesDisplay, TcInterp(timeSamplesDisplay),LineStyle='--',
    LineWidth=1);
72 plot(timeSamplesDisplay, DIInterp(timeSamplesDisplay),LineWidth=1);
73 legend("Tmax","T","Tc","D");
74 grid on;
75
76 subplot(5,1,2);
77 title("Speed and airspeed");
78 hold on;
79 plot(timeSamplesDisplay, VcInterp(timeSamplesDisplay),LineStyle='--',
    LineWidth=1);
80 plot(timeSamplesDisplay, VIInterp(timeSamplesDisplay),LineWidth=1);
81 plot(timeSamplesDisplay, VinfInterp(timeSamplesDisplay),LineWidth=1);
82 legend("Vc","V","Vinf");
83 grid on;
84
85 subplot(5,1,3);
86 title("Lift");
87 hold on;
88 plot(timeSamplesDisplay, LcInterp(timeSamplesDisplay),LineStyle='--',
    LineWidth=1);
89 plot(timeSamplesDisplay, LIInterp(timeSamplesDisplay),LineWidth=1);
90 legend("Lc","L");
91 grid on;
92
93 subplot(5,1,4);
94 title("Angle of attach and gamma");
95 hold on;
96 plot(timeSamplesDisplay, rad2deg(alphaBInterp(timeSamplesDisplay)),
    LineWidth=1);
97 plot(timeSamplesDisplay, gammaInterp(timeSamplesDisplay),LineWidth=1);
98 legend("\alpha_B","\gamma");
99 grid on;
100
101 subplot(5,1,5);
102 title("Rate of climb");
103 hold on;
104 plot(timeSamplesDisplay, VIInterp(timeSamplesDisplay).*deg2rad(gammaInterp
    (timeSamplesDisplay)),LineWidth=1);
105 grid on;
106
107 % Setup the figure/scene
108 h_fig1 = figure(4);
109 light('Position',[3 3 -3], 'Style','local');
110 % Trick to have Ze pointing downward and correct visualization
111 set(gca,'XDir','reverse'); set(gca,'ZDir','reverse');
112 grid on; hold on;
113 %% Load aircraft shape

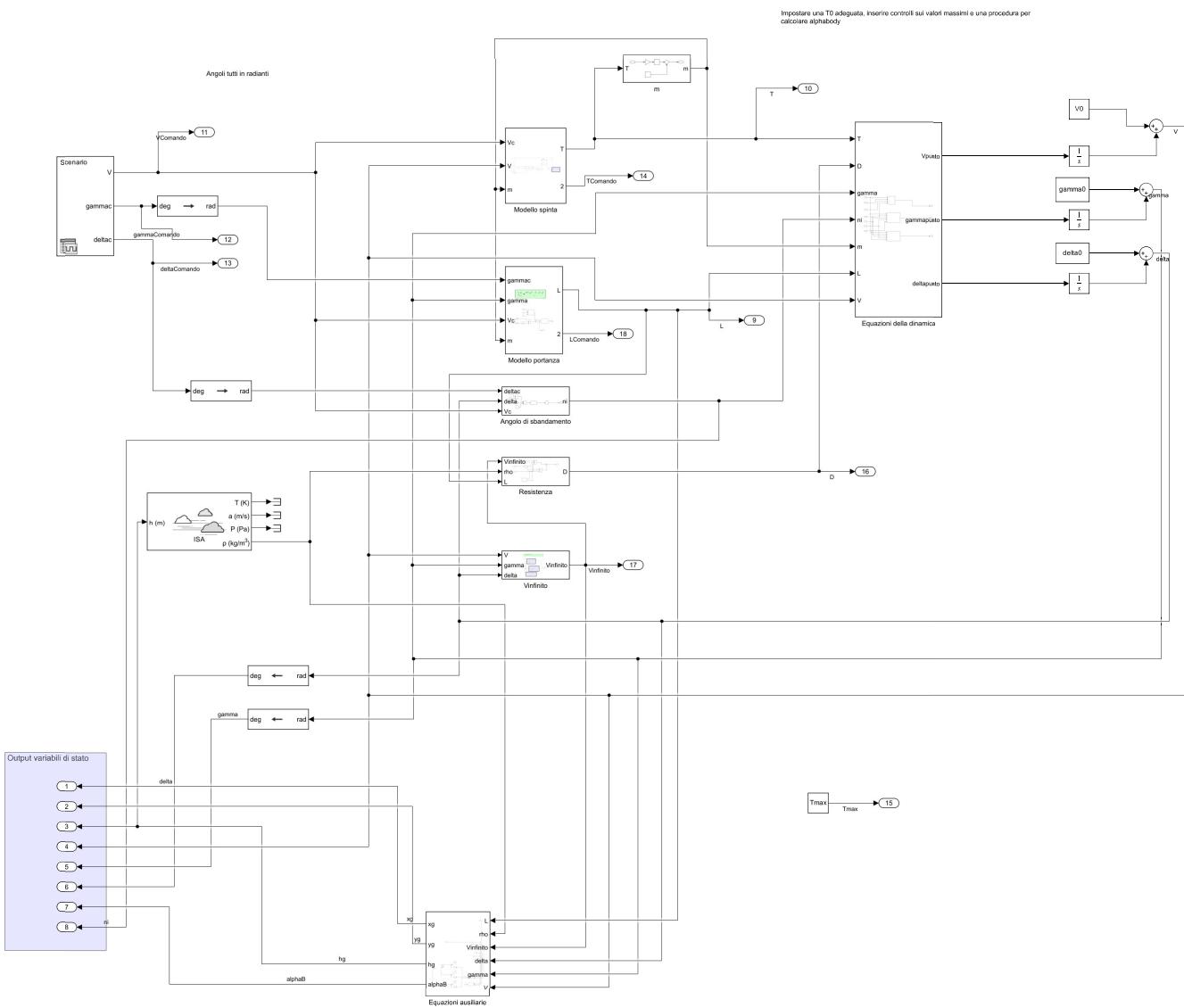
```

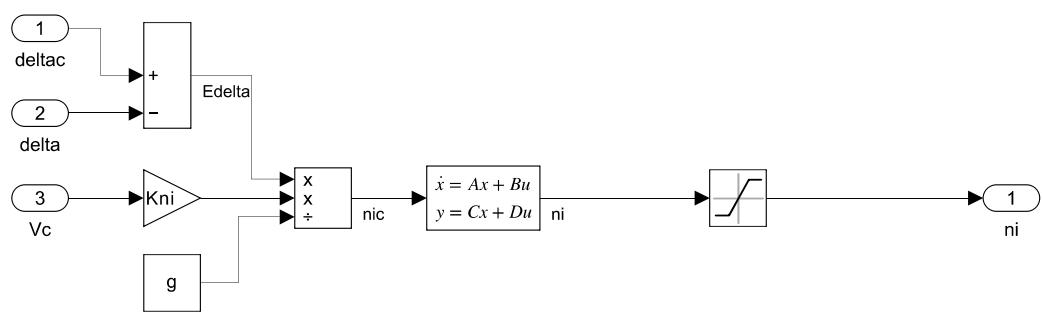
```

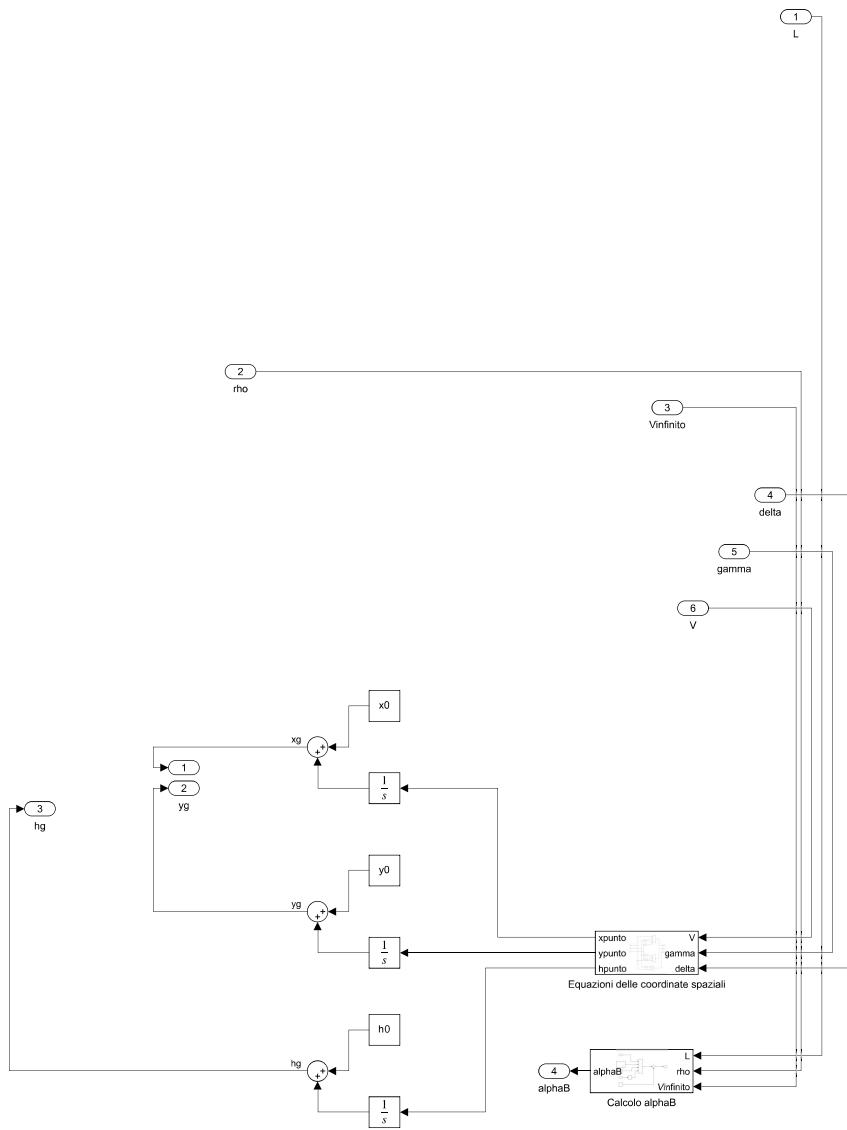
114 shapeScaleFactor = 600.0;
115 shape = loadAircraftMAT('Embraer-Phenom.mat', shapeScaleFactor);
116 % Set the aircraft in place
117 % Posision in Earth axes
118 Times=[0, 0.2*t_fin, 0.4*t_fin, 0.6*t_fin, 0.8*t_fin,t_fin]';
119
120 vXZe = [XeInterp(Times), YeInterp(Times), ZeInterp(Times)];
121 % psi, theta, phi -> 'ZYX'
122 %Il in gamma 180 serve perch il modello sfasato per qualche motivo
123 % di 180 gradi
124 vEulerAngles=deg2rad([deltaInterp(Times), gammaInterp(Times), -niInterp(
125 Times)+180]);
126 % Observer point-of-view
127 theView = [1005 150];
128 % body axes settings
129 bodyAxesOptions.show = true;
130 bodyAxesOptions.magX = 20.0*shapeScaleFactor;
131 bodyAxesOptions.magY = 20.0*shapeScaleFactor;
132 bodyAxesOptions.magZ = 10.5*shapeScaleFactor;
133 bodyAxesOptions.lineWidth = 2.5;
134
135 % Plot Earth axes
136 hold on;
137 xMax = 1;
138 yMax = 1;
139 zMax = 1; % max([abs(max(vXYZe(1))),0.18*xMax]);
140 vXYZ0 = [0,0,0];
141 plotEarthAxes(h_fig1, vXYZ0, vExtent);
142 options.samples=[1,2,3,4,5,6];
143 options.theView=theView;
144 options.bodyAxes.show = 1;
145 options.bodyAxes.magX = 2.0*shapeScaleFactor;
146 options.bodyAxes.magY = 2.0*shapeScaleFactor;
147 options.bodyAxes.magZ = 2.0*shapeScaleFactor;
148 options.bodyAxes.lineWidth = 2.5;
149 options.helperLines.show= 1;
150 options.helperLines.lineColor='k';
151 options.helperLines.lineWidth=1.0;
152 options.helperLines.lineStyle='--';
153 options.trajectory.show = 1;
154 options.trajectory.lineColor = 'blue';
155 options.trajectory.lineWidth=1.0;
156 options.trajectory.lineStyle = '--';
157 plotTrajectoryAndBodyE(h_fig1,shape,vXYZe,vEulerAngles, options);

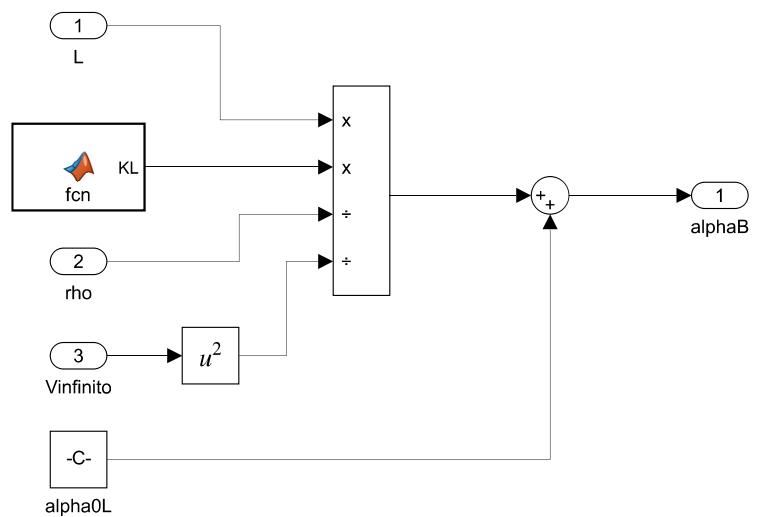
```

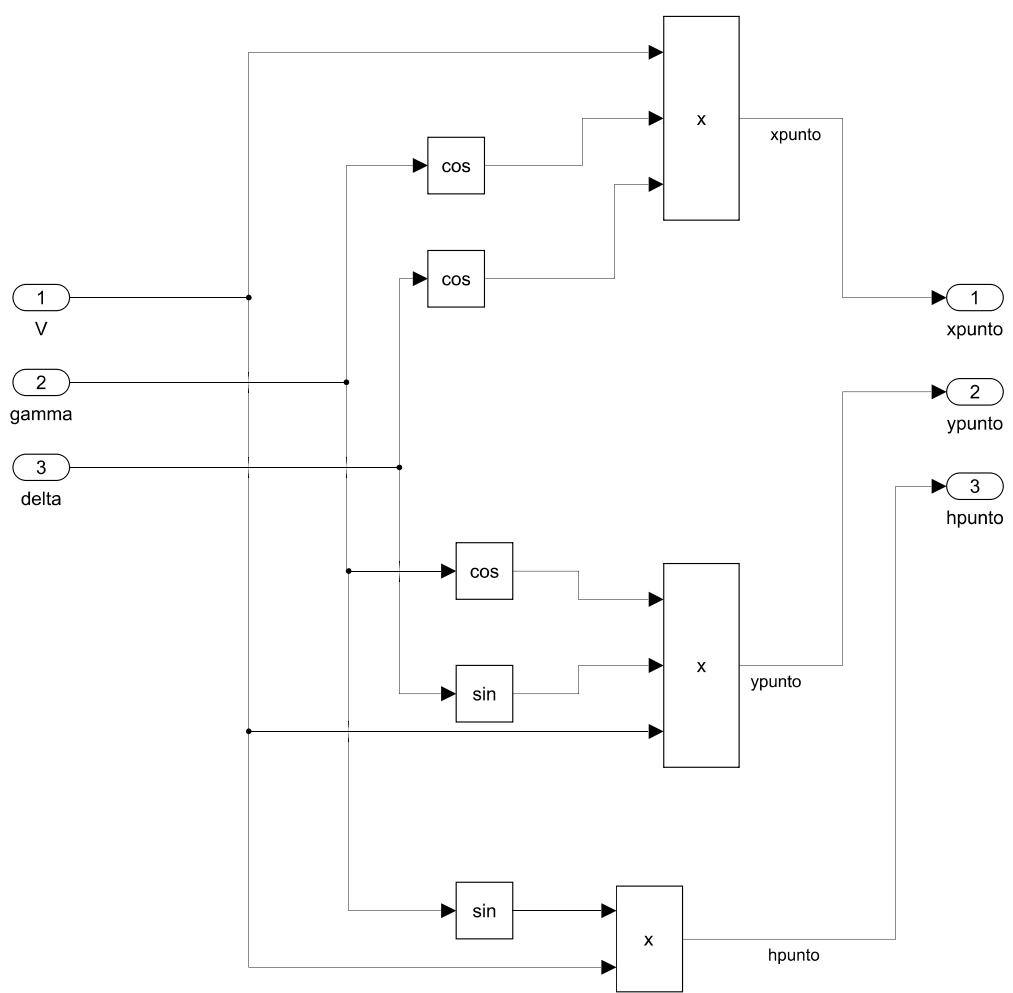
2.3 Modello Simulink

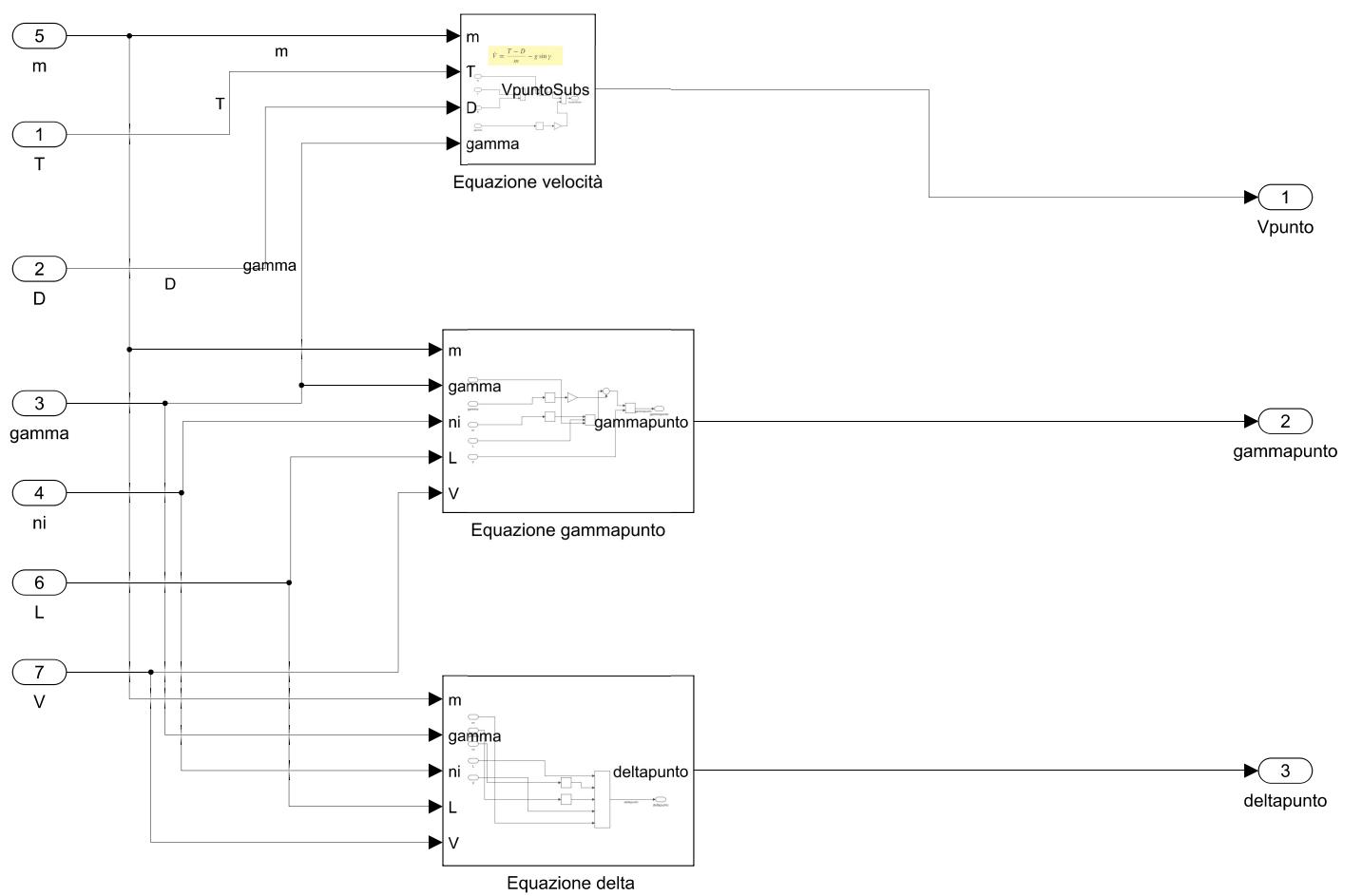


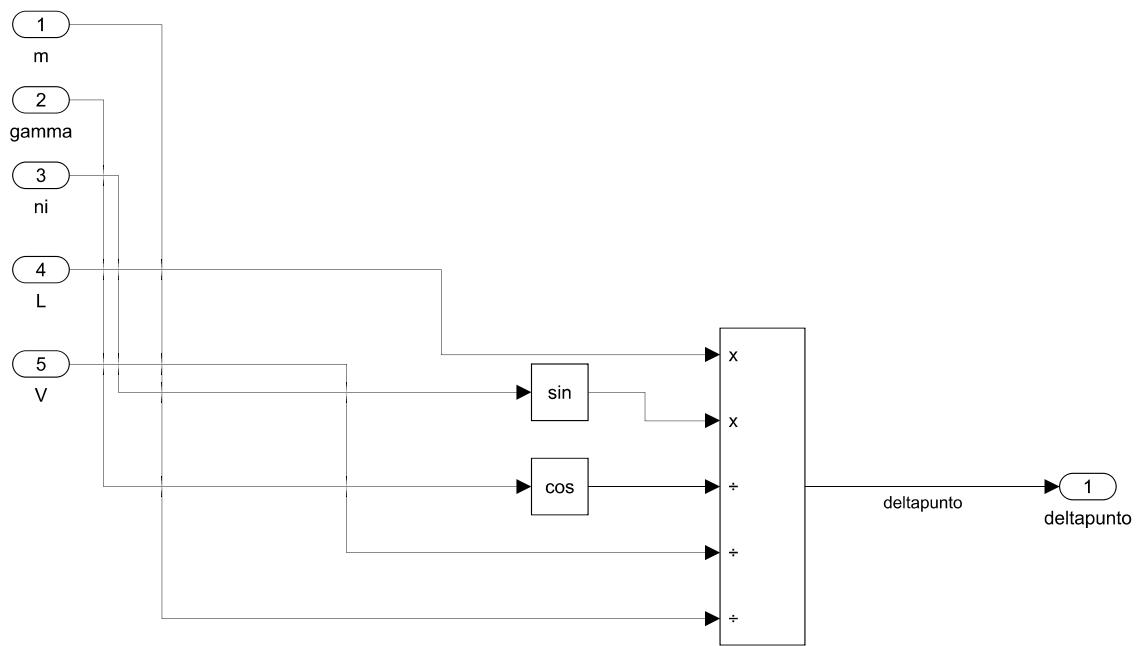


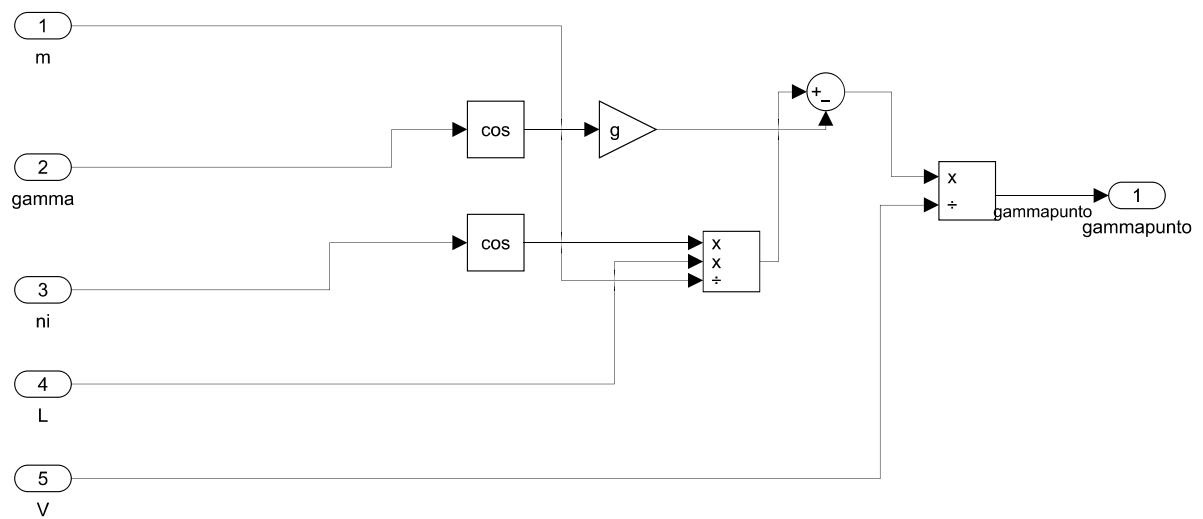




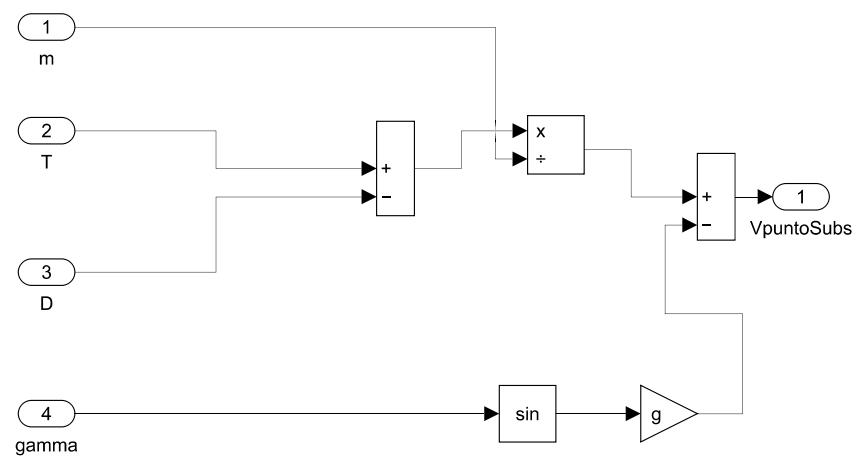




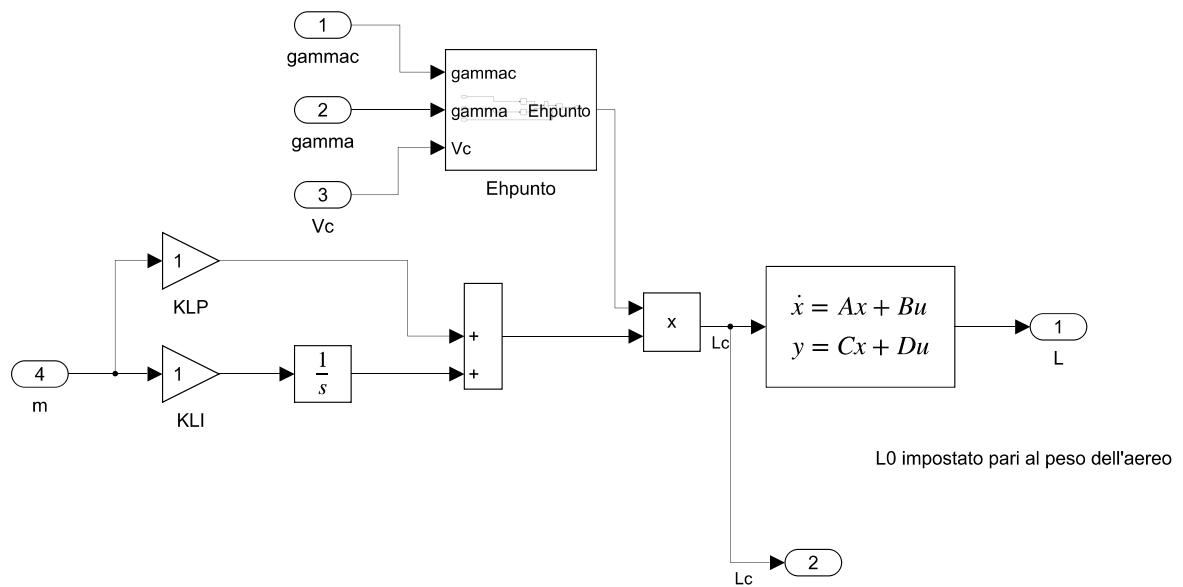


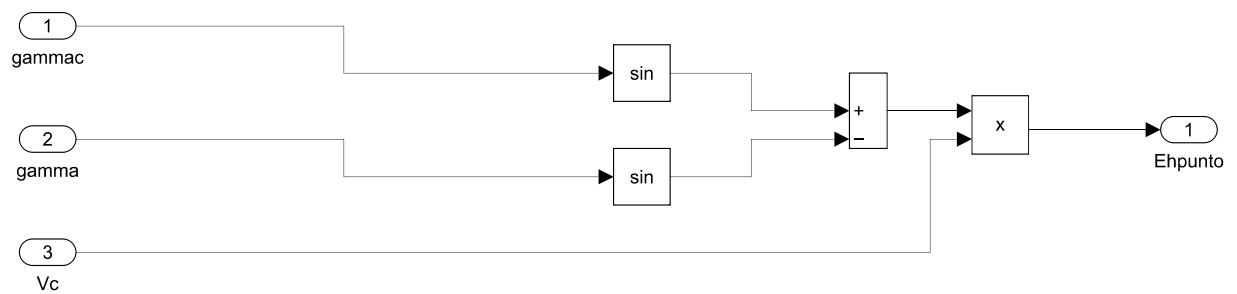


$$\dot{V} = \frac{T - D}{m} - g \sin \gamma$$



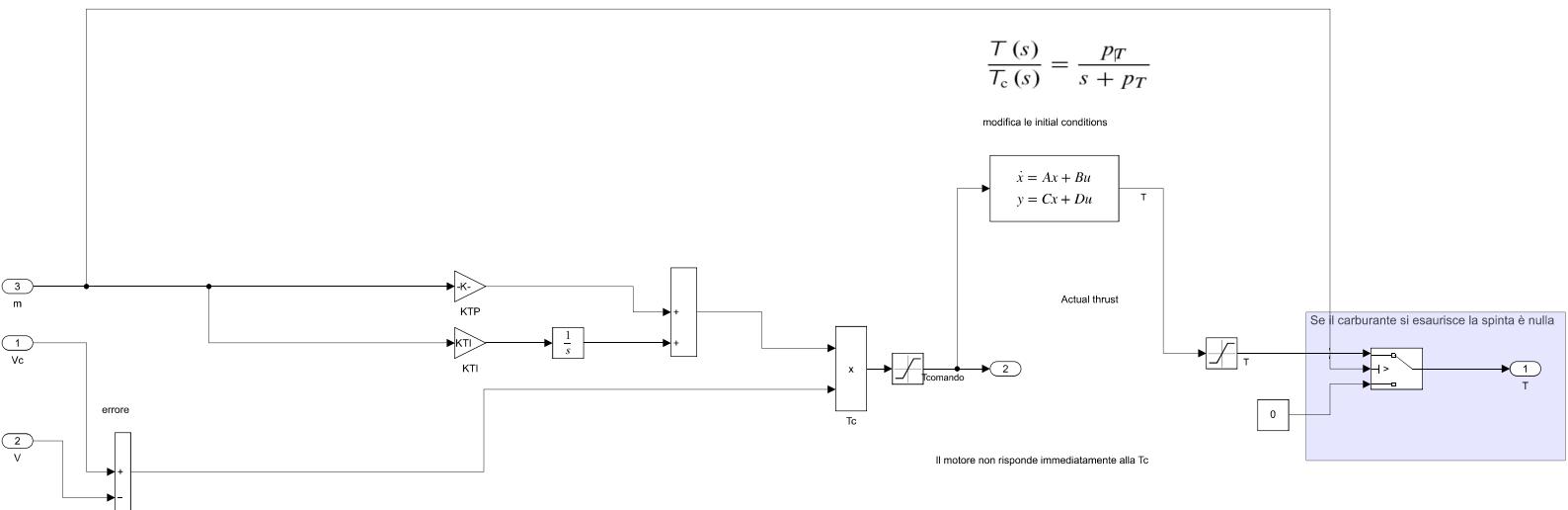
$$\frac{L_c(s)}{E_h(s)} = \frac{m K_{L_p} (s + K_{L_i}/K_{L_p})}{s}$$

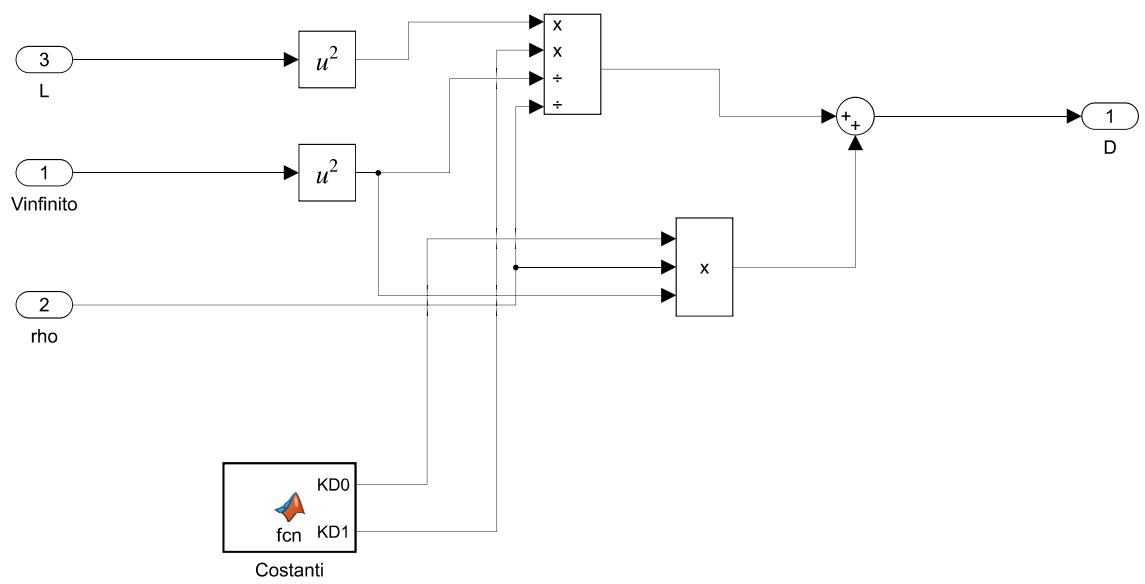




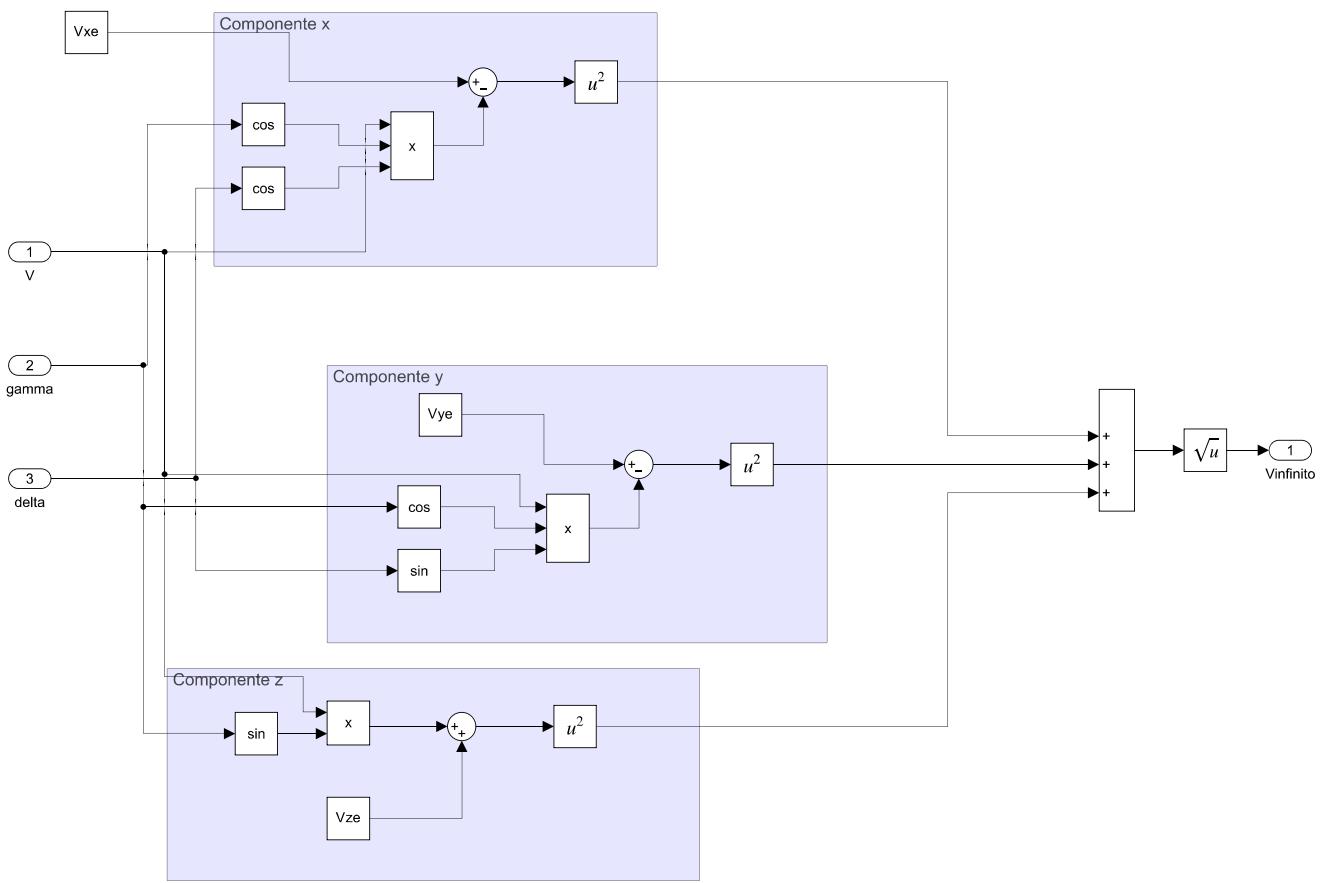
$$\frac{T(s)}{T_c(s)} = \frac{p_T}{s + p_T}$$

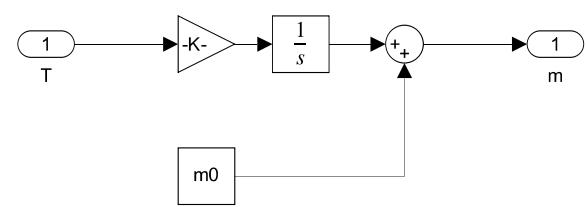
modifica le initial conditions





$$\begin{aligned}
 V_{\infty}^2 &= V_{x_{E,\infty}}^2 + V_{y_{E,\infty}}^2 + V_{z_{E,\infty}}^2 \\
 &= (V_{x_{E,w}} - V \cos \gamma \cos \delta)^2 + (V_{y_{E,w}} - V \cos \gamma \sin \delta)^2 + (V_{z_{E,w}} + V \sin \gamma)^2
 \end{aligned}$$





2.4 Risultati

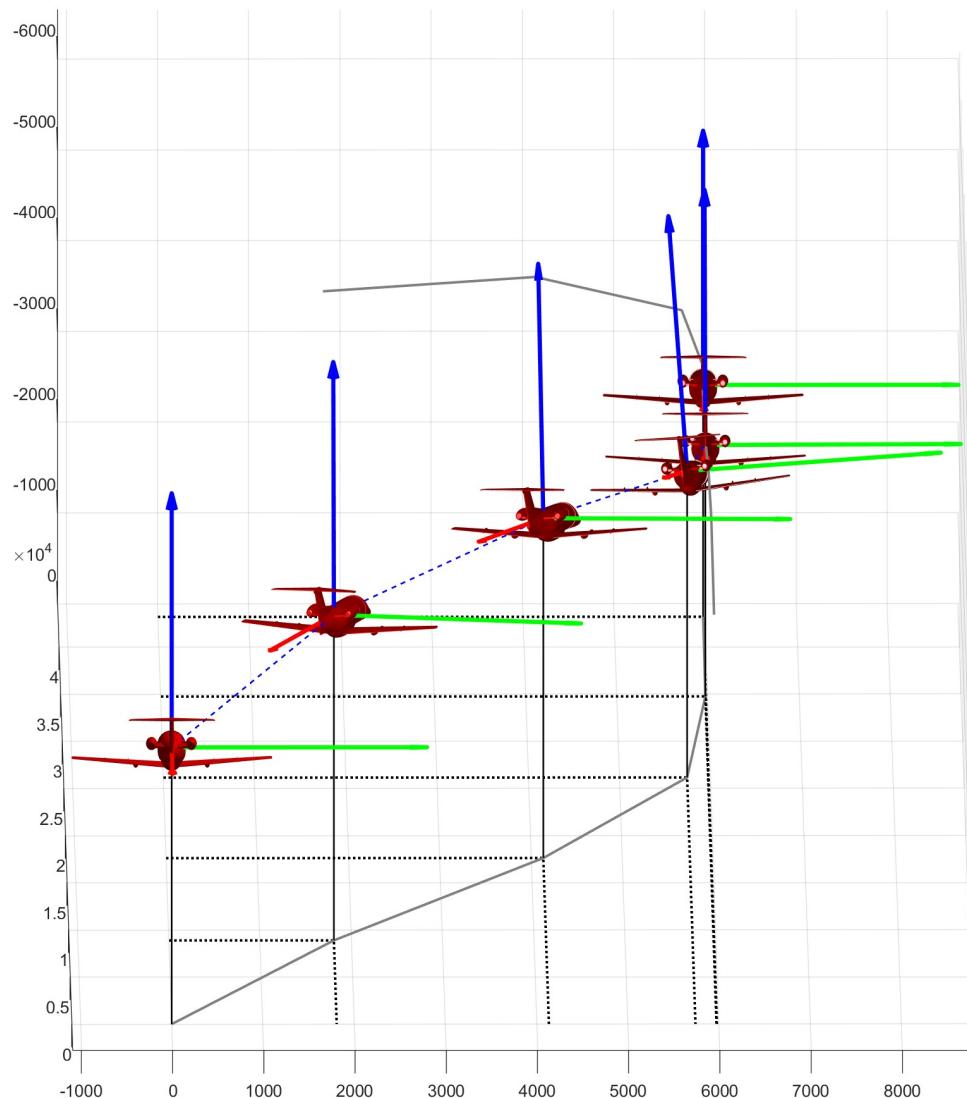


Figura 2.1 Traiettoria di volo

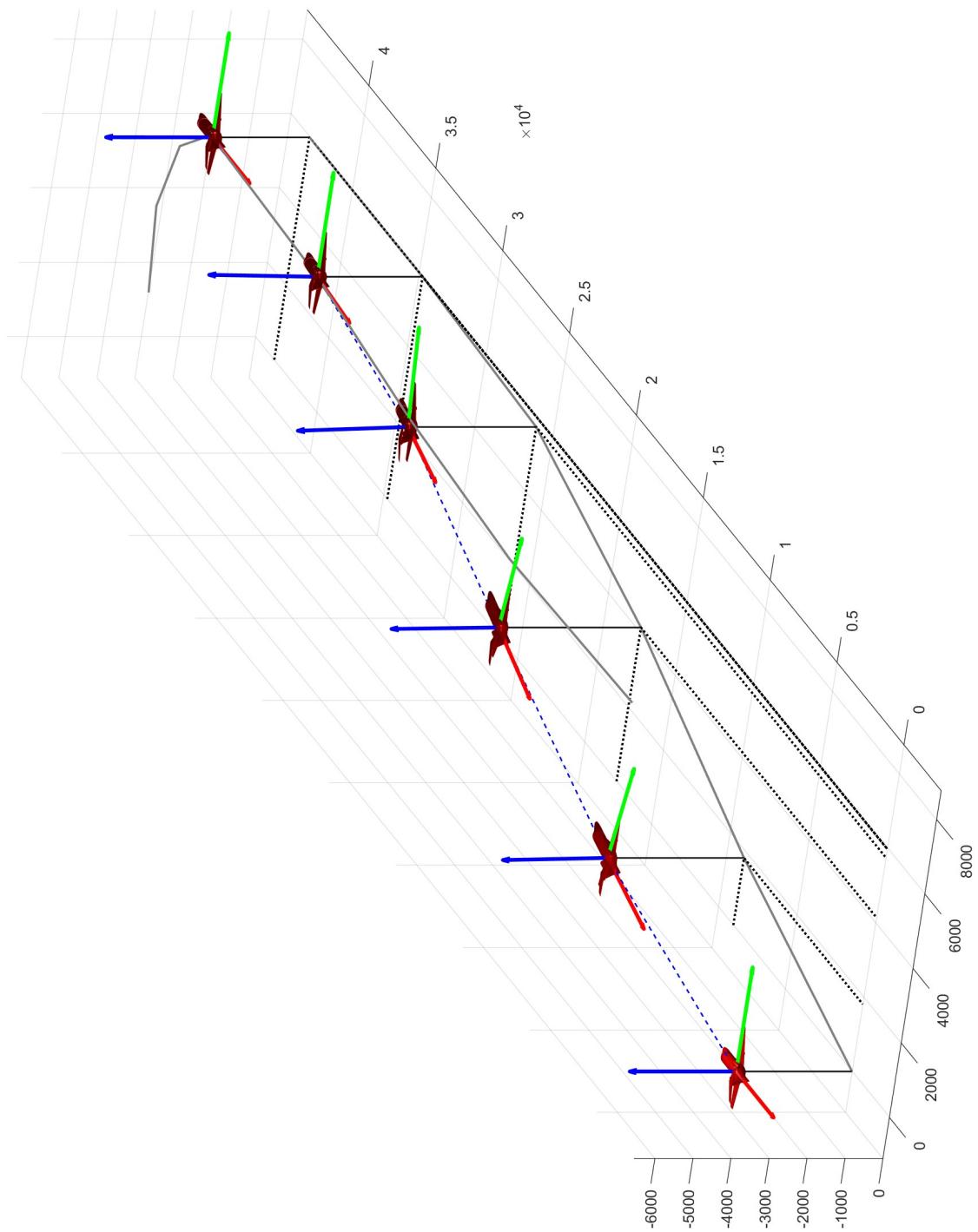


Figura 2.2 Traiettoria di volo

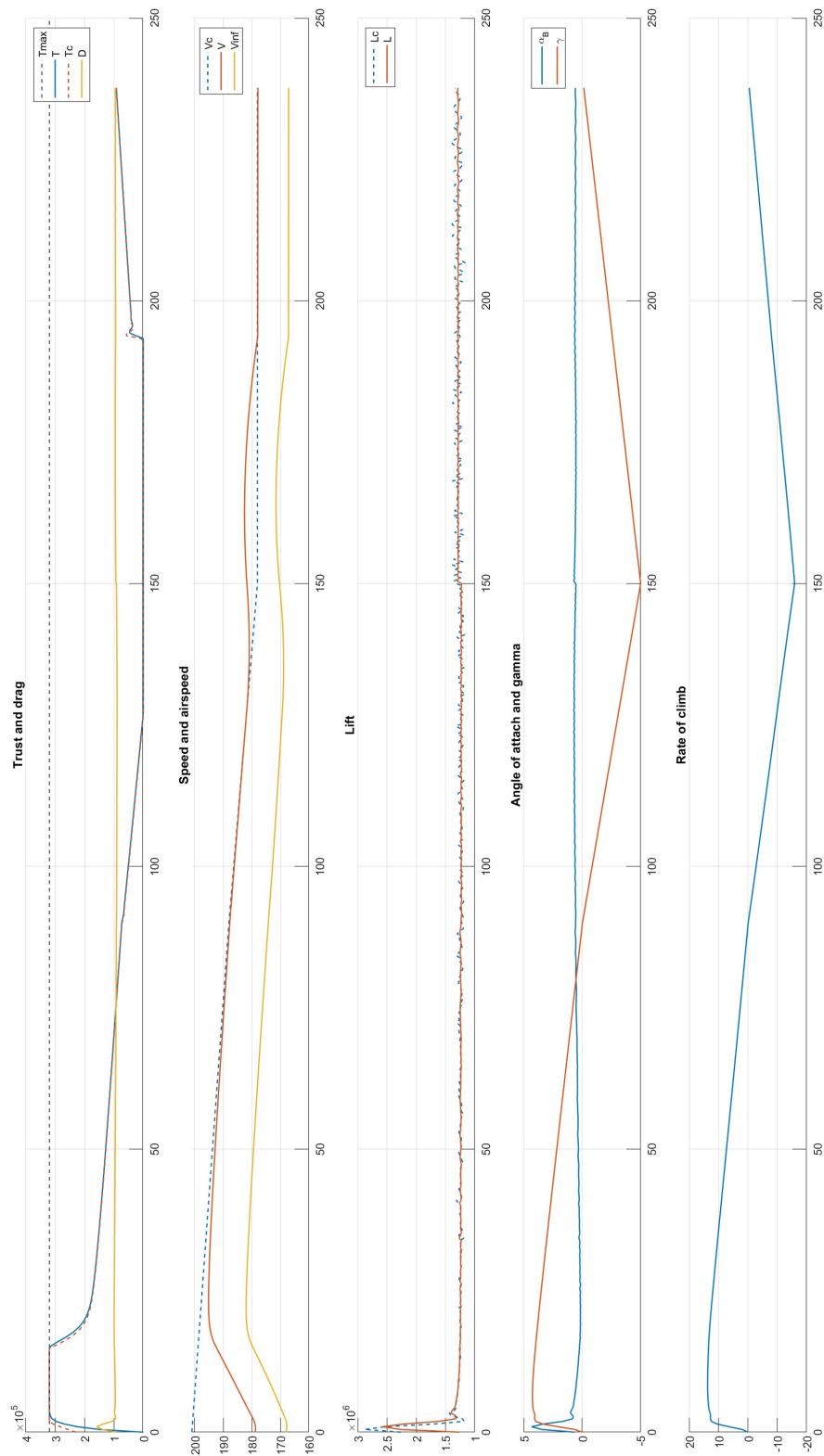


Figura 2.3 Variazione delle variabili nel tempo

Capitolo

3

Equazioni del moto

3.1 Introduzione

Lo scopo di questo capitolo è simulare la dinamica a sei gradi di libertà di un velivolo Cessna Citation I durante una manovra di virata utilizzando un modello aerodinamico creato mediante l'utilizzo del software digital Datcom+ e a alcune stime manuali. La creazione di tale modello è stata eseguita in maniera graduale dopo aver effettuato gli esercizi del Q7 di seguito riportati.

3.2 Esercizio 7.3 con HARV

Questo primo esercizio consiste nella sostituzione delle prime tre variabili di stato u, v, w con V, α, β , chiaramente questo implica la sostituzione delle equazioni (3.1) con il sistema (3.2) all'interno della funzione `eqLongDynamicStickFixed_6DoF_uvwqr_Euler`, che verrà rinominata come `eqLongDynamicStickFixed_6DoF_Vabpqr_Euler`.

$$\frac{W}{g} \left(\{\dot{V}_G\}_m + [\tilde{\Omega}]_m \right) [V_G]_m = \{F\}_m \quad (3.1)$$

$$\begin{cases} \dot{V} = \frac{1}{m} \left[-D \cos \beta + Y_A \sin \beta + X_T \cos \alpha \cos \beta + Y_T \sin \beta + \right. \\ \left. + Z_T \sin \alpha \cos \beta - mg(\cos \alpha \cos \beta \sin \theta - \sin \beta \sin \phi \cos \theta - \sin \alpha \cos \beta \cos \phi \cos \theta) \right] \\ \dot{\alpha} = \frac{1}{mV \cos \beta} \left[-L + Z_T \cos \alpha - X_T \sin \alpha + mg(\cos \alpha \cos \phi \cos \theta + \right. \\ \left. + \sin \alpha \sin \theta) \right] + q - \tan \beta(p \cos \alpha + r \sin \alpha) \\ \dot{\beta} = \frac{1}{mV} \left[D \sin \beta + Y_A \cos \beta - X_T \cos \alpha \sin \beta + Y_T \cos \beta - Z_T \sin \alpha \sin \beta + \right. \\ \left. + mg(\cos \alpha \sin \beta \sin \theta + \sin \phi \cos \beta \cos \theta - \sin \alpha \sin \beta \cos \phi \cos \theta) \right] + p \sin \alpha - r \cos \alpha \end{cases} \quad (3.2)$$

Seguono solo le sezioni modificate della funzione sopracitata:

Listing 3.1

```

1 function [dStatedt] = eqLongDynamicStickFixed_6DoF_Vabpqr_Euler(t,state)
2
3 global g...
4     delta_a delta_e delta_r delta_T...
5     myAC
6
7 % Assegnazione delle componenti del vettore di stato
8 V = state(1);
9 alpha_B = state(2);
10 beta_der = state(3);
11 p = state(4);
12 q = state(5);
13 r = state(6);
14 xEG = state(7);
15 yEG = state(8);
16 zEG = state(9);
17 phi = state(10);
18 theta = state(11);
19 psi = state(12);

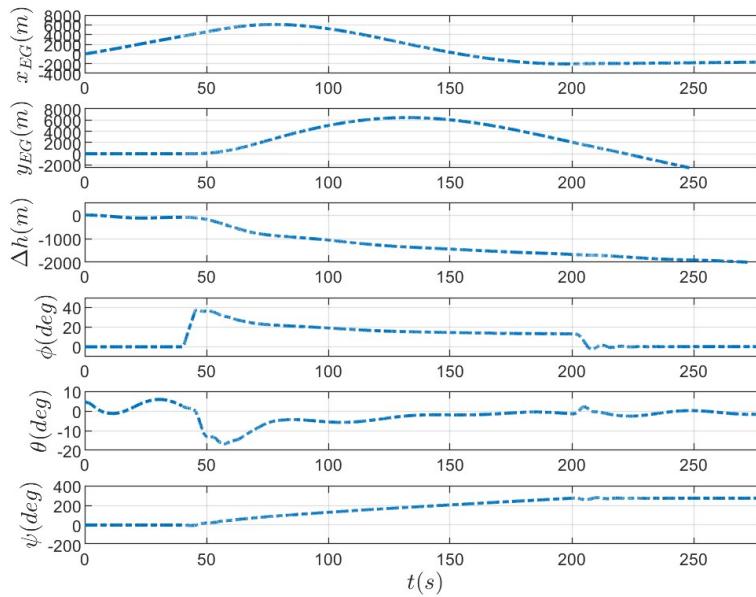
```

Listing 3.2

```

1
2 %% Costruzione della funzione integranda
3
4 %Calcolo i primi 3 termini, Vpunto, alphapunto e betapunto
5 Ca=cos(alpha_B);
6 Cb=cos(beta_der);
7 Sa=sin(alpha_B);
8 Sb=sin(beta_der);
9 St=sin(theta);
10 Ct=cos(theta);
11 Cp=cos(phi);
12 Sp=sin(phi);
13 Vpunto = (1/m) * (-D*Cb + Y_A*Sb + X_T * Ca * Cb + ...
14     + Y_T*Sb+Z_T*Sa*Cb-m*g*(Ca*Cb*St-Sb*Sp*Ct-Sa*Cb*Cp*Ct));
15 u=V*Cb*Ca;
16 v=V*Sb;
17 w=V*Cb*Sa;
18 alphapunto=(1/(m*V))*( -L+Z_T*Ca-X_T*Sa+m*g*(Ca*Cp*Ct+Sa*St)+...
19     +q*tan(beta_der)*(p*Ca+r*Sa));
20 betapunto=(1/(m*V))*(D*Sb+Y_A*Cb-X_T*Ca*Sb+Y_T*Cb-Z_T*Sa*Sb+...
21     +m*g*(Ca*Sb*St+Cb*Sp*Ct-Sa*Sb*Cp*Ct)+p*Sa-r*Ca);
22
23 % Vpunto=(Ca*Cb)/m * (X) + Ca*Cb*(r*v-q*w)+...
24 %     +Sb/m * (Y) + Sb * (p*w-r*u) +...
25 %     +(Sa*Cb)/m * (Z) + Sa*Cb*(q*u-p*v);
26
27 %Inserisco le prime tra variabili nel vettore di stato sostituendo al
28 % posto
29 %di omegatilde
30 upunto=Vpunto*Ca*Cb;
31 vpunto=Vpunto*Sb;
32 wpunto=Vpunto*Sa*Cb;
33 dVPB=[Vpunto; alphapunto; betapunto];

```

**Figura 3.1** Output del programma

```

34 dPQR=myAC.I_matrix\(-omegatilde*myAC.I_matrix*[p;q;r]+...
35     +[L_roll_A+L_roll_T;M_pitch_A+M_pitch_T;+N_yaw_A+N_yaw_T]);
36 dxyz=T_EB*[u;v;w];
37 dPTP=T_gimb*[p;q;r];
38 dStatedt = [dVPB;dPQR;dxyz;dPTP];

```

Dalla fig.(3.1) si può osservare che l'andamento delle coordinate e degli angoli di Euler non è variato nella manovra di approccio. Questa modifica al codice verrà quindi utilizzata per gli esercizi successivi.

3.3 Implementazione del modello aerodinamico del Citation I

In questa sezione verranno utilizzati i coefficienti aerodinamici del Citation I calcolati mediante Digital Datcom nella sezione apposita di questo documento.

E' stata mantenuta la struttura del file *eqLongDynamicStickFixed_6DoF_Vabpqr_Euler* mostrato nel precedente esercizio ma il modello dell'HARV all'interno delle funzioni dei coefficienti chiamate dalla funzione è stato eliminato e sostituito con procedure per calcolare quelle del Citation I a partire dai dati importati tramite la funzione *datcomimport*.

I coefficienti aerodinamici vengono importati nel workspace di matlab attraverso la funzione *datcomimport*, negli esercizi di questo capitolo è stato impostato un numero di Mach pari a $M = 0.25$ e altitudine $z = -457m(-1500ft)$. Rispetto al modello originale dell'HARV è stato anche introdotto l'effetto dei flaps. I comandi sono stati definiti come delle funzioni gradino nel signalEditor e sono stati importati nel workspace. La variabile *aeroDat* è dichiarata come global così che sia visibile dalle funzioni chiamate dallo script.

Listing 3.3

```

1 aircraftDataFileName = 'CITATION.txt';

```

```

2 myAC = DSVAircraft(aircraftDataFileName);
3 aeroDat=datcomimport("CitationM025.out");
4
5
6
7
8 % Costanti e condizioni iniziali
9 g = 9.81; %Accelerazione di gravit [m/s^2]
10 xEG_0 = 0; %[m]
11 yEG_0 = 0; %[m]
12 zEG_0 = -457; %Altitudine [m]
13 [air_Temp0,sound_speed0,air_pressure0,rho0] = atmosisa(-zEG_0);
14 phi0 = convang(0.000,'deg','rad'); %Angolo di bank
15 theta0 = convang(4.460,'deg','rad'); %Angolo di elevazione
16 psi0 = convang(0.000,'deg','rad'); %Angolo di heading
17 p0 = convangvel(0.000,'deg/s','rad/s'); %Velocit angolare di rollio
18 q0 = convangvel(0.000,'deg/s','rad/s'); %Velocit angolare di beccheggio
19 r0 = convangvel(0.000,'deg/s','rad/s'); %Velocit angolare di imbardata
20 M0 = 0.25; %Numero di Mach
21 V0 = M0*sound_speed0; %Velocit lineare del baricentro
22 alpha_B0 = convang(3,'deg','rad'); %Angolo di attacco valutato rispetto l
    'asse x Body
23 beta_der0 = convang(0.000,'deg','rad'); %Angolo di derapata
24 u0 = V0*cos(beta_der0)*cos(alpha_B0); %Componente della velocit lineare
    del baricentro lungo l'asse x Body
25 v0 = V0*sin(beta_der0); %Componente della velocit lineare del
    baricentro lungo l'asse y Body
26 w0 = V0*cos(beta_der0)*sin(alpha_B0); %Componente della velocit lineare
    del baricentro lungo l'asse z Body
27
28
29
30 %% Definizione delle leggi di comando
31 % Assegnazione delle leggi temporali dei comandi di volo
32 global delta_a delta_e delta_r delta_T delta_f
33
34 %Uso il signal editor per definire i comandi
35 flightScenario = load('Virata.mat');
36
37 delta_a = @(t) deg2rad(interp1(flightScenario.delta_a.Time,
    flightScenario.delta_a.Data,t,'previous','extrap'));
38 delta_e = @(t) deg2rad(interp1(flightScenario.delta_e.Time,
    flightScenario.delta_e.Data,t,'previous','extrap'));
39 delta_r = @(t) deg2rad(interp1(flightScenario.delta_r.Time,
    flightScenario.delta_r.Data,t,'previous','extrap'));
40 delta_f = @(t) deg2rad(interp1(flightScenario.delta_f.Time,
    flightScenario.delta_f.Data,t,'previous','extrap'));
41 delta_T = @(t) interp1(flightScenario.delta_t.Time,flightScenario.
    delta_t.Data,t,'previous','extrap');
42 t_fin = 110;% max([flightScenario.delta_a.Time(end),flightScenario.delta_e
    .Time(end),flightScenario.delta_r.Time(end), ...
    %flightScenario.delta_f.Time(end),flightScenario.delta_t.Time(end)]);
43 % Comandi di volo iniziali
44 delta_a0 = delta_a(0);
45 delta_e0 = delta_e(0);
46 delta_r0 = delta_r(0);

```

```

48 delta_f0 = delta_f(0);
49 delta_T0 = delta_T(0);
50
51 %Stato iniziale
52 state_0 = [v0,alpha_B0,beta_der0,p0,q0,r0,xEG_0,yEG_0,zEG_0,phi0,theta0,
53 psi0];
54 %% Eliminato
55 % delta_e_exc1 = convang(0,'deg','rad');
56 % delta_e_exc2 = convang(0,'deg','rad');
57 % delta_e = @(t) interp1([0, 20, 80, 180, t_fin],...
58 %                         [delta_e0, delta_e0, delta_e_exc1, delta_e_exc2,
59 %                          delta_e_exc2],t,'linear');
60 %
61 % delta_r_exc1 = convang(10,'deg','rad')*0;
62 % delta_r = @(t) interp1([0, 200, 203, 205, 208, t_fin],...
63 %                         [delta_r0, delta_r0, delta_r_exc1, delta_r_exc1,
64 %                          delta_r0, delta_r0],t,'linear');
65 %
66 % delta_T_exc1 = 0.43;
67 % delta_T = @(t) interp1([0, t_fin],[delta_T0 delta_T_exc1],t,'linear');
68 %
69
70
71 % Integrazione del sistema di equazioni differenziali
72 options = odeset('RelTol',1e-6,'AbsTol',1e-6*ones(1,12));
73 [vTime,mState] = ode45(@eqLongDynamicStickFixed_6DoF_Vabpqr_Euler,[0
74 t_fin],state_0,options);
75
76 %% Risultati in vettori
77
78 vVel_u = mState(:,1)*cos(mState(2))*cos(mState(3));
79 vVel_v = mState(:,2)*sin(mState(3));
80 vVel_w = mState(:,3)*sin(mState(2))*cos(mState(3));
81 vVel = mState(:,1);
82 vAlpha_B = rad2deg(mState(:,2));
83 vBeta = rad2deg(mState(:,3));
84 v_p = convangvel(mState(:,4),'rad/s','deg/s');
85 v_q = convangvel(mState(:,5),'rad/s','deg/s');
86 v_r = convangvel(mState(:,6),'rad/s','deg/s');
87 vXe = mState(:,7);
88 vYe = mState(:,8);
89 vZe = mState(:,9);
90 vPhi = convang(mState(:,10),'rad','deg');
91 vTheta = convang(mState(:,11),'rad','deg');
92 vPsi = convang(mState(:,12),'rad','deg');
93 vGamma = convang(asin(cos(vAlpha_B).*cos(vBeta).*sin(vTheta)-...
94 sin(vBeta).*sin(vPhi).*cos(vTheta)-...
95 sin(vAlpha_B).*cos(vBeta).*cos(vPhi).*cos(vTheta)), 'rad',
96 'deg');
96 mQuat = zeros(length(vTime),4);
97 for i = 1:length(vTime)
    mQuat(i,:) = angle2quat(mState(i,12),mState(i,11),mState(i,10), 'ZYX')

```

```

98 ;
99 end
100 %% Grafica

```

3.3.1 Funzioni matlab

Quasi tutte le funzioni sono state modificate estensivamente e per questo motivo verranno riportate interamente.

eqLongDynamicStickFixed_6DoF_Vabpqr_Euler.m

Listing 3.4

```

1 %Parametri cinematici
2 V = (u^2 + v^2 + w^2)^(1/2);
3 alpha_B = atan(w/u);
4 beta_der = asin(v/V);
5
6 %Propriet termodinamiche
7 rho = density(-zEG);
8
9 %% Definizione delle forze e dei momenti agenti sul velivolo
10 % Componenti in assi velivolo della forza risultante
11 C_L=CL(convang(alpha_B,'rad','deg'),...
12           convang(delta_e(t),'rad','deg'), convang(
13             delta_f(t),'rad','deg'));
14 L = 0.5*rho*V^2*myAC.S*C_L;
15 D = 0.5*rho*V^2*myAC.S*CD(convang(alpha_B,'rad','deg'), convang(delta_f(t),
16                           'rad','deg'));
17 Y_A = 0.5*rho*V^2*myAC.S*CY_A(C_L, ...
18           convang(beta_der,'rad','deg'),p);
19 X_T = delta_T(t)*myAC.T_max_SL*cos(myAC.mu_T);
20 %Come richiesto, si assume che la spinta propulsiva massima disponibile
21 %sia
22 %costante al variare della quota e della velocità e che sia pari al
23 %valore
24 %della spinta propulsiva massima disponibile al livello del mare.
25 Y_T = 0;
26 Z_T = delta_T(t)*myAC.T_max_SL*sin(myAC.mu_T);
27 X = X_T - D*cos(alpha_B) + L*sin(alpha_B) - myAC.W*sin(theta);
28 Y = Y_T + Y_A + myAC.W*sin(phi)*cos(theta);
29 Z = Z_T - D*sin(alpha_B) - L*cos(alpha_B) + myAC.W*cos(phi)*cos(theta);
30
31 % Componenti in assi velivolo del momento risultante
32 L_roll_A = 0.5*rho*V^2*myAC.S*myAC.b*croll(convang(alpha_B,'rad','deg')
33           ,...
34           convang(beta_der,'rad','deg'))
35           ,...
36           convang(delta_a(t),'rad','deg')
37           ),...
38           convang(delta_r(t),'rad','deg')
39           ),...

```

```

33                                     p , r );
34 L_roll_T = 0;
35 M_pitch_A = 0.5*rho*V^2*myAC.S*myAC.mac*Cpitch(convang(alpha_B, 'rad', 'deg
36 ', ...                                         convang(delta_e(t), 'rad', 'deg
37 deg'), ...                                         q);
38 C_M_T = (delta_T(t)*myAC.T_max_SL*myAC.e_T)/(0.5*rho*V^2*myAC.S*myAC.mac)
39 ;                                         ;
40 M_pitch_T = 0.5*rho*V^2*myAC.S*myAC.mac*C_M_T;
41 N_yaw_A = 0.5*rho*V^2*myAC.S*myAC.b*Cyaw(convang(alpha_B, 'rad', 'deg'), ...
42                                         C_L, ...                                         convang(beta_der, 'rad', 'deg')
43 , ...                                         p, ...
44                                         convang(delta_r(t), 'rad', 'deg')) ;
45 N_yaw_T = 0;
46
47 %Prime 3 equazioni
48 Ca=cos(alpha_B);
49 Sa=sin(alpha_B);
50 Cb=cos(beta_der);
51 Sb=sin(beta_der);
52 Ct=cos(theta);
53 St=sin(theta);
54 Cp=cos(phi);
55 Sp=sin(phi);
56 m=myAC.mass;
57 Vpunto=((Ca*Cb)/m)*X+Ca*Cb*(r*v-q*w)+Sb/m*Y+Sb*(p*w-r*u)+(Sa*Cb)/m*Z+(Sa*
58 Cb)*(q*u-p*v);
59 alphapunto=1/(m*V*Cb)*(-L_Z_T*Ca-X_T*Sa+m*g*(Ca*Cp*Ct+Sa*St))+q-tan(
60 beta_der)*(p*Ca+r*Sa);
61 betapunto=1/(m*V)*(D*Sb+Y_A*Cb-X_T*Ca*Sb+Y_T*Cb-Z_T*Sa*Sb+m*g*(Ca*Sb*St+
62 Cb*Sp*Ct-Sa*Sb*Cp*Ct))+p*Sa-r*Ca;
63
64 % upunto=Vpunto*Ca*Cb;
65 % vpunto=Vpunto*Sb;
66 % wpunto=Vpunto*Sa*Cb;
67
68 % Costruzione della funzione integranda
69 DVAB=[Vpunto;alphapunto;betapunto];
70 DPQR=myAC.I_matrix\[L_roll_A+L_roll_T;M_pitch_A+M_pitch_T;N_yaw_A+
71 N_yaw_T]+...
72 -omegatilde*myAC.I_matrix*[p;q;r]);
73 DXYZ=T_EB*[u;v;w];
74 DPTP=T_gimb*[p;q;r];
75
76 dStatedt=[DVAB;DPQR;DXYZ;DPTP];

```

CL**Listing 3.5**

```

1 function [C_L] = CL(alpha_B,delta_e,delta_f)
```

```

2 global aeroDat
3
4 % Tengo conto sia dei flaps che dell'elevatore prelevando i dati dai due
5 % case di Datcom, case 1 flaps, case 2 elevatore
6
7 %Prelevo il Cl dal modello completo (2)
8
9
10 alphaDat=aeroDat{1, 3}.alpha;
11 delta_eDat=aeroDat{1, 3}.delta; %Deflessione elevatore
12 delta_fDat=aeroDat{1, 1}.delta; %Deflessione flaps
13 clDat=aeroDat{1, 3}.cl; %Cl configurazione pulita
14 clDeltaeDat=aeroDat{1,3}.dcl_sym;%Variazione dovuta all'elevatore
15 clDeltafDat=aeroDat{1,1}.dcl_sym;%Variazione dovuta ai flaps
16
17 cl=interp1(alphaDat,clDat,alpha_B,'linear');
18 dClDeltae=interp1(delta_eDat,clDeltaeDat,delta_e,'linear');
19 dClflaps=interp1(delta_fDat,clDeltafDat,delta_f,'linear');
20
21 C_L=cl + dClDeltae + dClflaps;
22
23
24 end

```

CD**Listing 3.6**

```

1 function [C_D] = CD(alpha_B, delta_f)
2 global aeroDat
3
4
5 alphaDat=aeroDat{1, 3}.alpha;
6 delta_fDat=aeroDat{1, 1}.delta; %Deflessione flaps
7 cdDat=aeroDat{1, 3}.cd; %Cd configurazione pulita
8 cdDeltafDat=aeroDat{1,1}.dcdi_sym;%Variazione dovuta ai flaps
9
10 cd=interp1(alphaDat,cdDat,alpha_B,'linear');
11
12 [A,B]=meshgrid(alphaDat,delta_fDat);
13
14 deltaCdflaps=interp2(A,B, cdDeltafDat', alpha_B, delta_f); %
    Interpolazione con alpha_B e deltaf
15
16
17 C_D=cd + deltaCdflaps;
18
19 end

```

CY_A**Listing 3.7**

```

1 function [CY_A] = CY_A(CL,beta_der,p)
2 global aeroDat
3 %Effetto del delta_r non modellato
4

```

```
5  
6  
7 CY_A = aeroDat{1,3}.cyb(1)*beta_der + aeroDat{1,3}.cypcl*CL*p;  
8  
9 end
```

Cyaw

Listing 3.8

```
1 function [C_yaw] = Cyaw(alpha_B,~, beta_der,p, delta_r)  
2  
3 global aeroDat  
4  
5 pyaw=convangvel(p,'rad/s','deg/s');  
6 cnb=aeroDat{1,3}.cnb(1)*beta_der;  
7 cnp=interp1(aeroDat{1,3}.alpha,aeroDat{1,3}.cnp,alpha_B)*pyaw;  
8 cnr= interp1(aeroDat{1,3}.alpha,aeroDat{1,3}.cnr,alpha_B)*delta_r;  
9 C_yaw = cnb + cnp + cnr;  
10  
11 end
```

3.4 Utilizzo del nuovo modello per simulare una virata

Si è utilizzato il nuovo modello per simulare una manovra di virata effettuata tramite l'azionamento prima degli alettoni ed in seguito del timone e dell'elevatore per stabilizzare il velivolo durante la manovra. Seguono i valori delle variabili di stato e la rappresentazione della manovra.

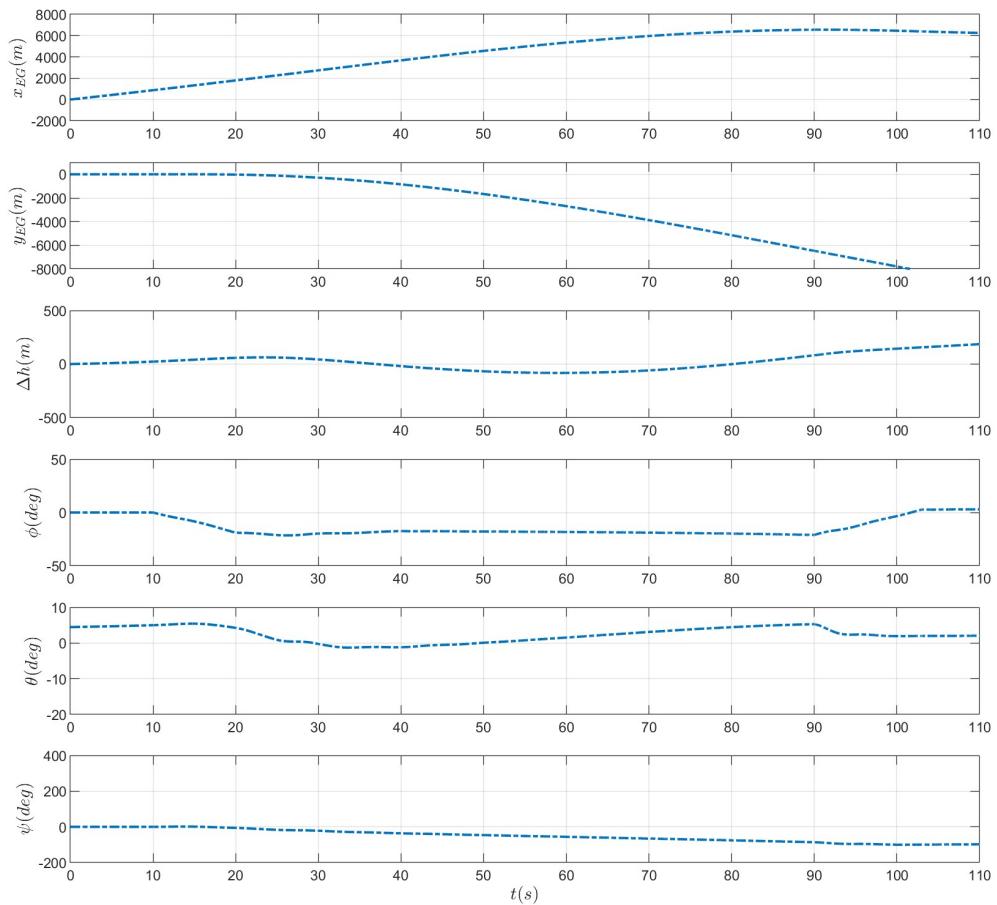


Figura 3.2 Coordinate ed angoli di Eulero

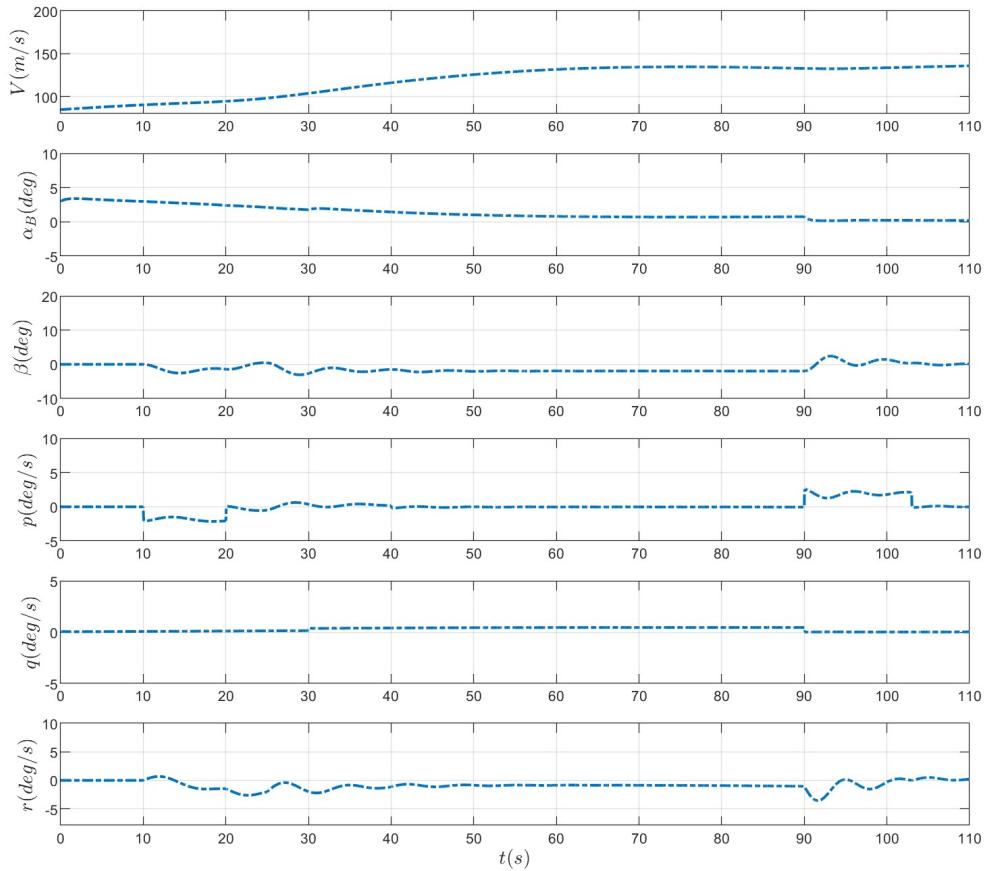


Figura 3.3 Parametri della velocità e velocità angolare del velivolo

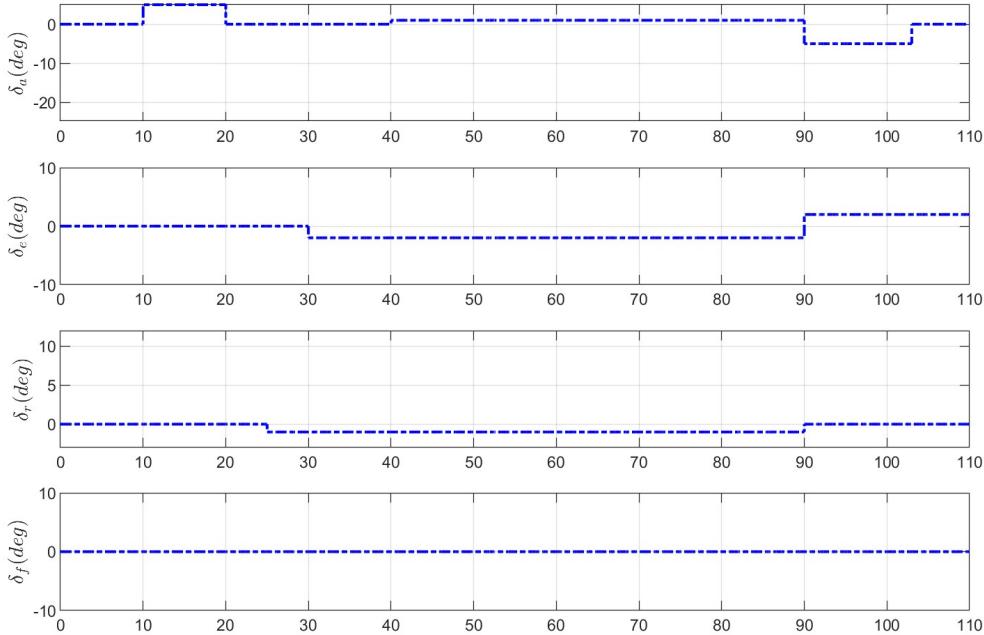


Figura 3.4 Comandi impartiti

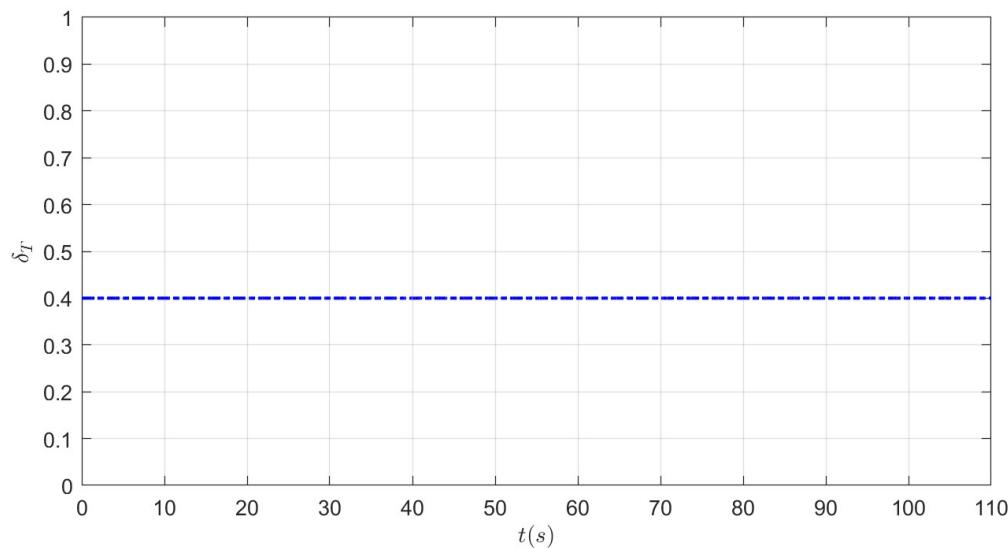


Figura 3.5 Manetta del motore

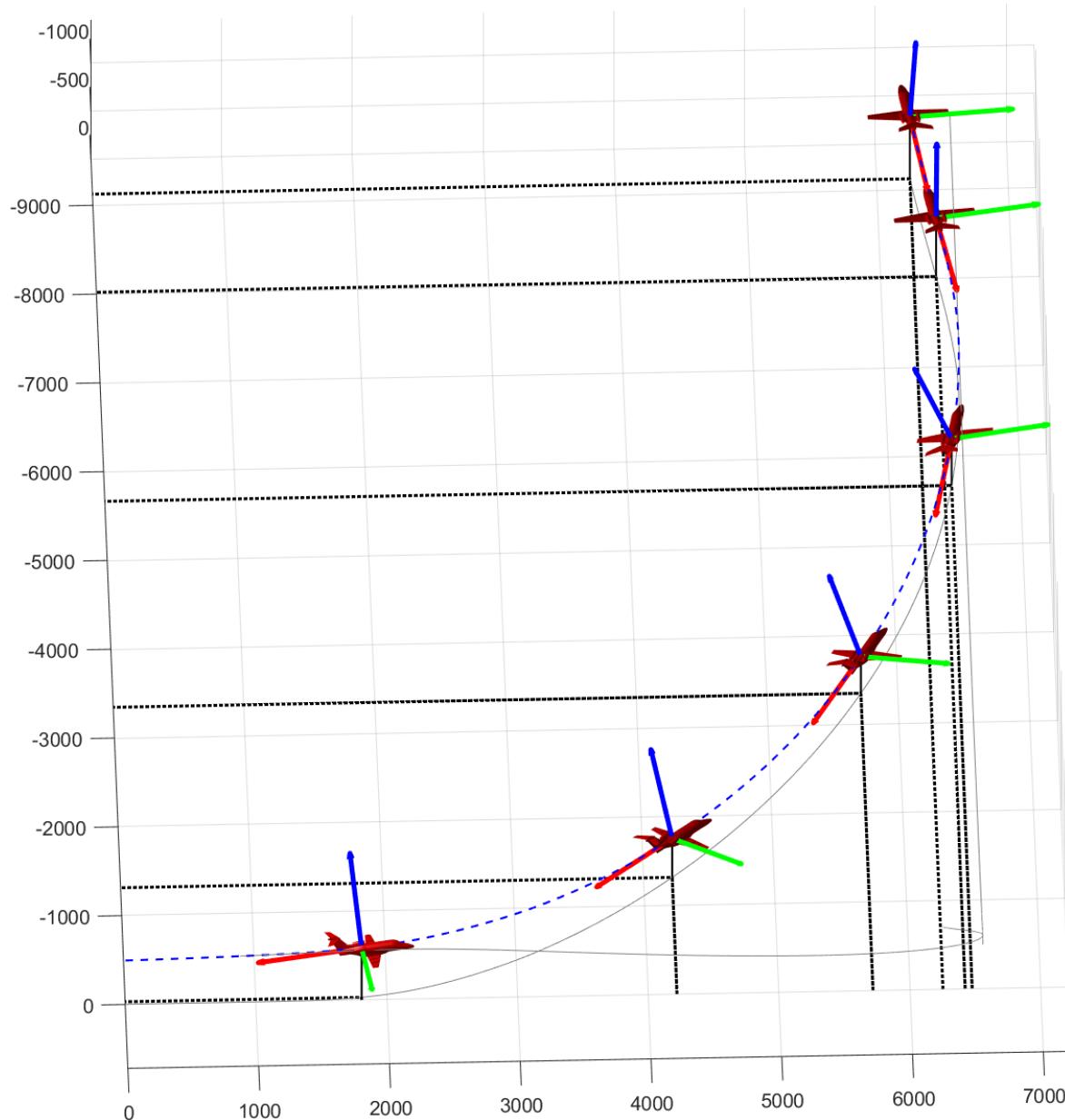


Figura 3.6 Visualizzazione della manovra, prima prospettiva

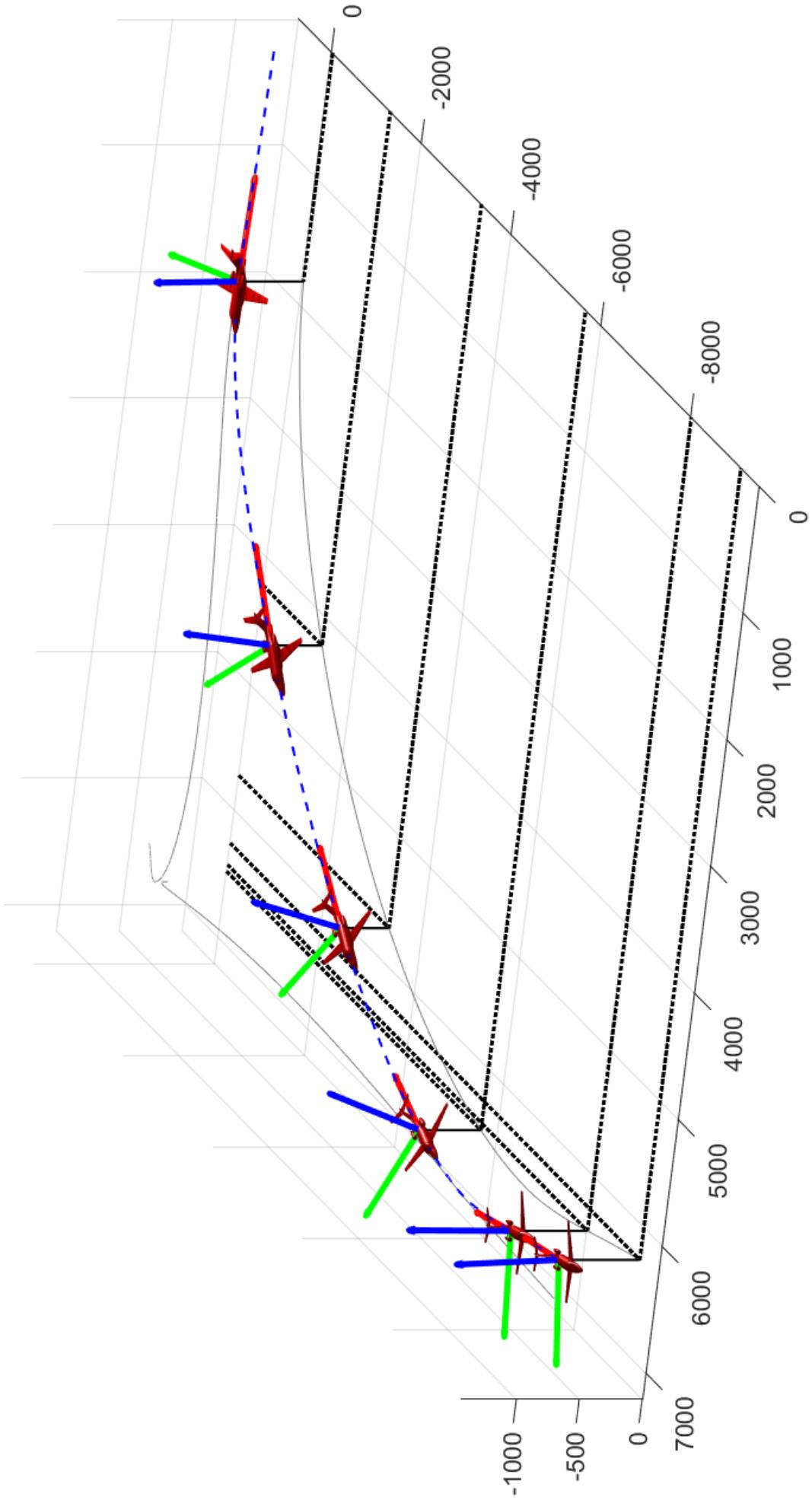


Figura 3.7 Visualizzazione della manovra, seconda prospettiva

3.5 Ricerca delle condizioni di trim per moto a 6DOF per il Citation I

In questo esercizio si ricercano le condizioni di trim per il velivolo scelto con parametri di volo indicati nelle tabelle 3.1 e 3.2. In seguito verrà effettuata una simulazione di 20 secondi di volo con questi parametri e il risultato sarà visualizzato.

V	84.8 m/s
p	0 deg/s
q	0 deg/s
r	0 deg/s

Tabella 3.1 Parametri le cui derivate vengono minimizzate nella funzione obiettivo

x_0	0 m
y_0	0 m
z_0	300 m
ϕ_0	10 deg
θ	5 deg
ψ	0

Tabella 3.2 Valori iniziali dei restanti parametri

Nel moto a 6 gradi di libertà in generale vale:

$$\left\{ \begin{array}{l} \dot{V} = f1(V, \alpha, \beta, p, q, r, \delta_e, \delta_a, \delta_r, \delta_T + \dots) \\ \dot{\alpha} = f1(V, \alpha, \beta, p, q, r, \delta_e, \delta_a, \delta_r, \delta_T + \dots) \\ \dot{\beta} = f1(V, \alpha, \beta, p, q, r, \delta_e, \delta_a, \delta_r, \delta_T + \dots) \\ \dot{p} = f1(V, \alpha, \beta, p, q, r, \delta_e, \delta_a, \delta_r, \delta_T + \dots) \\ \dot{q} = f1(V, \alpha, \beta, p, q, r, \delta_e, \delta_a, \delta_r, \delta_T + \dots) \\ \dot{r} = f1(V, \alpha, \beta, p, q, r, \delta_e, \delta_a, \delta_r, \delta_T + \dots) \end{array} \right. \quad (3.3)$$

Cerchiamo i valori di $\alpha, \beta, \delta_e, \delta_a, \delta_r, \delta_T$ che permettono di minimizzare il valore delle 6 equazioni della dinamica 3.3.

Nella funzione di costo vengono tenuti in considerazione solo la variazione di $V, \alpha, \beta, p, q, r$, abbiamo quindi ottenuto un sistema di 6 equazioni in 6 incognite da determinare.

Le variazioni degli angoli di Eulero sono indirettamente minimizzate anch'esse dato che se le velocità angolari p, q, r sono nulle anche la loro variazione lo sarà per le Gimbal equations.

Codice Matlab

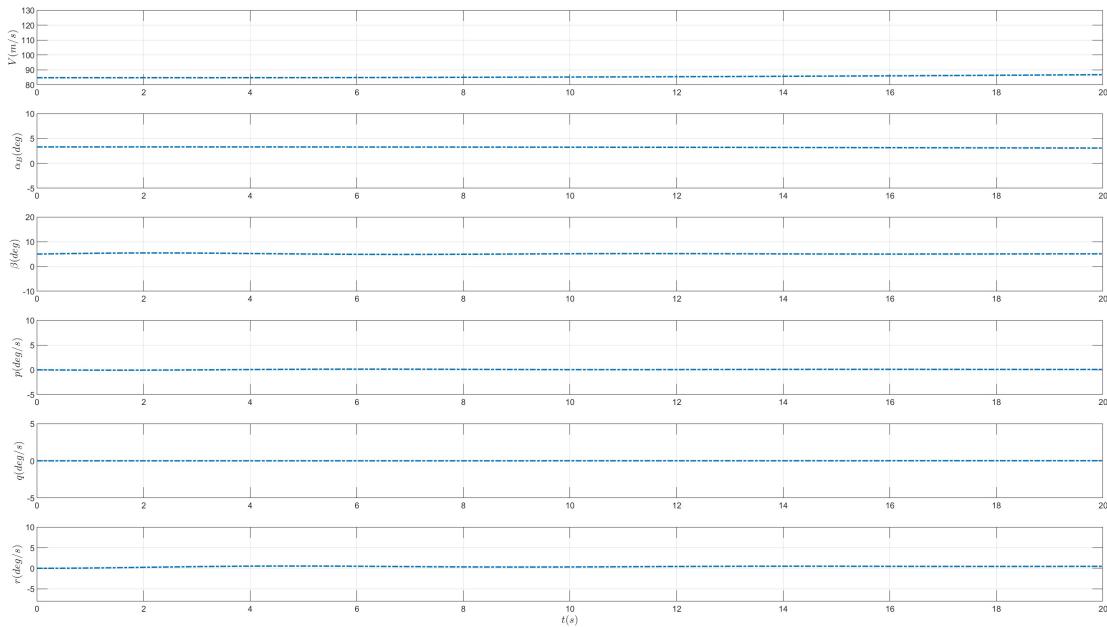
Listing 3.9 Funzione main

```

1 %% Ricerca delle condizioni di trim e verifica
2 clear all; clc; close all;
3 %Gamma nulla, velocità considerata orizzontale
4 disp('Moto del velivolo a 6 gradi di libertà');

```

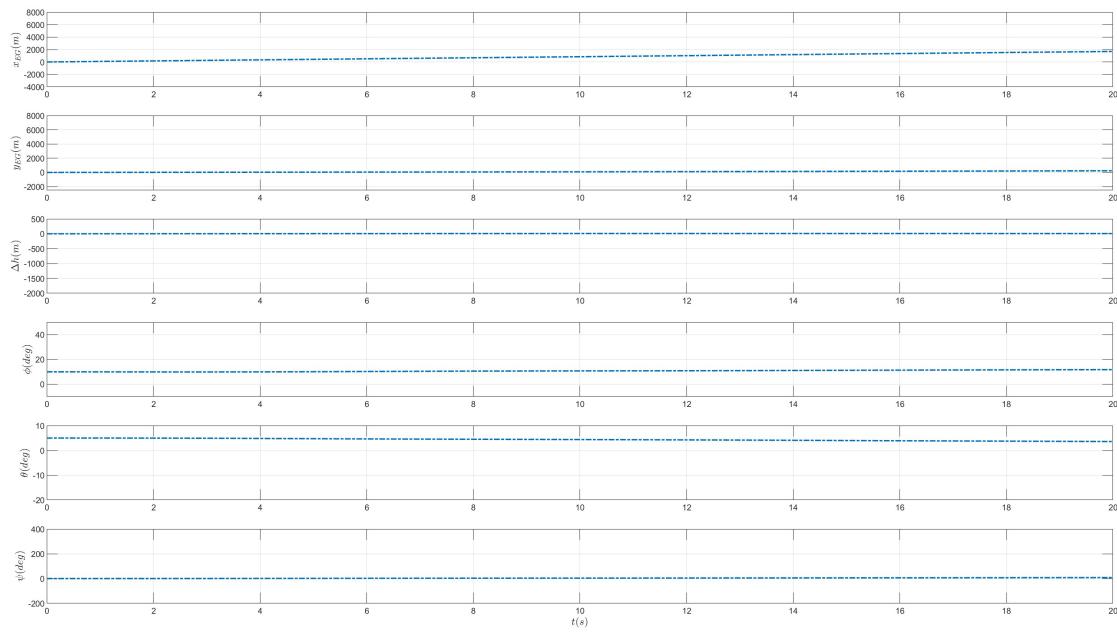
α_B	3.29 deg
β	5.00 deg
δ_e	0.43 deg
δ_a	-2.98 deg
δ_r	2.51 deg
δ_T	0.23

Tabella 3.3 Parametri trimmati**Figura 3.8** Variabili minimizzate

```

5  %% Dichiarazione delle variabili globali
6  global g phi0 theta0 psi0 p0 q0 r0 V0 myAC zEG_0 beta_der0 myAC aeroDat
7  % Definizione della classe DSVAircraft e dell'oggetto 'Velivolo'
8  aircraftDataFileName = 'CITATION.txt';
9  myAC = DSVAircraft(aircraftDataFileName);
10 aeroDat = datcomimport("CitationM025.out");
11 % Costanti e condizioni iniziali
12 g = 9.81; %Accelerazione di gravit [m/s^2]
13 xEG_0 = 0; %[m]
14 yEG_0 = 0; %[m]
15 zEG_0 = -300; %Altitudine [m]
16 [air_Temp0,sound_speed0,air_pressure0,rho0] = atmosisa(-zEG_0);
17 phi0 = convang(10.000,'deg','rad'); %Angolo di bank
18 theta0 = convang(5,'deg','rad'); %Angolo di elevazione
19 psi0 = convang(0.000,'deg','rad'); %Angolo di heading
20 p0 = convangvel(0.000,'deg/s','rad/s'); %Velocit angolare di rollio
21 q0 = convangvel(0.000,'deg/s','rad/s'); %Velocit angolare di beccheggio
22 r0 = convangvel(0.000,'deg/s','rad/s'); %Velocit angolare di imbardata
23 M0 = 0.25; %Numero di Mach
24 V0 = M0*sound_speed0; %Velocit lineare del baricentro
25 %alpha_B0 = convang(4.557754,'deg','rad'); %Angolo di attacco valutato
      rispetto l'asse x Body

```

**Figura 3.9** Coordinate e angoli di Eulero

```

26 beta_der0 = convang(0,'deg','rad'); %Angolo di derapata
27
28 % Ricerca dei valori di trim
29 %Vanno ricavati alphaB0,Deltaa0,Deltae0,Deltas0,DeltaT0.
30 %Importante: Flaps ignorati
31
32 % bounds
33 lb =[convang(-15,'deg','rad'), ... % minimum alpha_B
34 convang(-20,'deg','rad'), ... % minimum elevator deflection
35 convang( -10,'deg','rad'), ... % minimum aerelion incidence
36 convang( -10,'deg','rad'), ... % minimum rudder incidence
37 0.2 ... % minimum thrust fraction
38 convang(-5,'deg','rad')
39 ];
40 ub =[convang(10,'deg','rad'), ... % maximum alpha_B
41 convang(20,'deg','rad'), ... % maximum elevator deflection
42 convang( 10,'deg','rad'), ... % maximum aerelion incidence
43 convang( 10,'deg','rad'), ... % maximum rudder incidence
44 1 ... % maximum thrust fraction
45 convang(5,'deg','rad')
46 ];
47 options = optimset('tolfun',1e-9,'Algorithm','interior-point');
48 % I parametri da ricercare sono alpha_B, delta_e, delta_a, delta_r,
49 % delta_T
50 x0=[0,0,0,0,0,0];
51 [trim,j_val]=fmincon(@funzioneObiettivo6DOF, x0,[],[],[],[],lb,ub,
52 @myNonLinearConstraint, options);
53 display(["alpha_B: " + rad2deg(trim(1)), "Beta: " + rad2deg(trim(6)) ,
54 delta_e: " + rad2deg(trim(2)), "delta_a: " + rad2deg(trim(3)), "delta_r:
55 " + rad2deg(trim(4)), "delta_T: " + trim(5)]);

```

```

54
55 % Verifica delle condizioni di trim
56
57 % Assegnazione delle leggi temporali dei comandi di volo
58 global delta_a delta_e delta_r delta_T
59
60 alpha_B0=trim(1);
61
62 delta_e = trim(2);
63 delta_a = trim(3);
64
65 delta_r = trim(4);
66
67 delta_T = trim(5);
68
69 beta_der = trim(6);
70
71
72 %Verifico 20s
73 t_fin = 20;
74 state_0 = [V0,trim(1),beta_der,p0,q0,r0,xEG_0,yEG_0,zEG_0,phi0,theta0,
    psi0];
75
76
77 % Integrazione del sistema di equazioni differenziali
78 options = odeset('RelTol',1e-9,'AbsTol',1e-9*ones(1,12));
79 [vTime,mState] = ode45(@eqLongDynamicStickFixed_6DoF_Vabpqr_Euler,[0
    t_fin],state_0,options);
80
81
82 vVel_u = mState(:,1)*cos(mState(2))*cos(mState(3));

```

Listing 3.10 Funzione obiettivo

```

1 function [J_function] = funzioneObiettivo6DOF(x)
2
3 global g phi0 theta0 psi0 p0 q0 r0 V0 myAC zEG_0 %beta_der0
4
5 % Assegnazione delle componenti del vettore di stato
6
7 alpha_B = x(1);
8 delta_e=x(2);
9 delta_a=x(3);
10 delta_r=x(4);
11 delta_T=x(5);
12 beta_der = x(6);
13
14 V = V0;
15 %beta_der = beta_der0;
16 p = p0;
17 q = q0;
18 r = r0;
19 zEG = zEG_0;
20 phi = phi0;
21 theta = theta0;
22 psi = psi0;

```

```

23 u=V*cos(alpha_B)*cos(beta_der);
24 v=V*sin(beta_der);
25 w=V*cos(beta_der)*sin(alpha_B);
26 %% Quantit caratteristiche
27 % Equazioni generali del moto del velivolo
28 omegatilde = [0, -r, q;
29             r, 0, -p;
30             -q, p, 0];
31
32 % Equazioni cinematiche ausiliari
33 T_BE = angle2dcm(psi,theta,phi,'ZYX');
34 T_EB = T_BE';
35 T_gimb = [1, sin(phi)*sin(theta)/cos(theta), cos(phi)*sin(theta)/cos(theta);
36           0, cos(phi), -sin(phi);
37           0, sin(phi)/cos(theta), cos(phi)/cos(theta)];
38
39 %Parametri cinematici
40 V = (u^2 + v^2 + w^2)^(1/2);
41 alpha_B = atan(w/u);
42 beta_der = asin(v/V);
43
44 %Propriet termodinamiche
45 rho = density(-zEG);
46
47 %% Definizione delle forze e dei momenti agenti sul velivolo
48 % Componenti in assi velivolo della forza risultante
49 C_L=CL(convang(alpha_B,'rad','deg'),...
50         convang(delta_e,'rad','deg'));
51 L = 0.5*rho*V^2*myAC.S*C_L;
52 D = 0.5*rho*V^2*myAC.S*CD(convang(alpha_B,'rad','deg'));
53
54 Y_A = 0.5*rho*V^2*myAC.S*CY_A(C_L,...
55                               convang(beta_der,'rad','deg'),p);
56 X_T = delta_T*myAC.T_max_SL*cos(myAC.mu_T);
57 %Come richiesto, si assume che la spinta propulsiva massima disponibile
58 %sia
59 %costante al variare della quota e della velocit e che sia pari al
60 %valore
61 %della spinta propulsiva massima disponibile al livello del mare.
62 Y_T = 0;
63 Z_T = delta_T*myAC.T_max_SL*sin(myAC.mu_T);
64 X = X_T - D*cos(alpha_B) + L*sin(alpha_B) - myAC.W*sin(theta);
65 Y = Y_T + Y_A + myAC.W*sin(phi)*cos(theta);
66 Z = Z_T - D*sin(alpha_B) - L*cos(alpha_B) + myAC.W*cos(phi)*cos(theta);
67
68 % Componenti in assi velivolo del momento risultante
69 L_roll_A = 0.5*rho*V^2*myAC.S*myAC.b*Croll(convang(alpha_B,'rad','deg')
70 ,...
71             convang(beta_der,'rad','deg'))
72 ,...
73             convang(delta_a,'rad','deg'))
74 ,...
75             convang(delta_r,'rad','deg'))
76 ,...

```

```

71                               p , r );
72 L_roll_T = 0;
73 M_pitch_A = 0.5*rho*V^2*myAC.S*myAC.mac*Cpitch(convang(alpha_B, 'rad', 'deg
    '), ...
74                                         convang(delta_e, 'rad', 'deg
    '), ...
75                                         q);
76 C_M_T = (delta_T*myAC.T_max_SL*myAC.e_T)/(0.5*rho*V^2*myAC.S*myAC.mac);
77 M_pitch_T = 0.5*rho*V^2*myAC.S*myAC.mac*C_M_T;
78 N_yaw_A = 0.5*rho*V^2*myAC.S*myAC.b*Cyaw(convang(alpha_B, 'rad', 'deg'), ...
79                                         C_L, ...
80                                         convang(beta_der, 'rad', 'deg')
81                                         , ...
82                                         p, ...
83                                         convang(delta_r, 'rad', 'deg')));
84
85 %Prime 3 equazioni
86 Ca=cos(alpha_B);
87 Sa=sin(alpha_B);
88 Cb=cos(beta_der);
89 Sb=sin(beta_der);
90 Ct=cos(theta);
91 St=sin(theta);
92 Cp=cos(phi);
93 Sp=sin(phi);
94 m=myAC.mass;
95 Vpunto=((Ca*Cb)/m)*X+Ca*Cb*(r*v-q*w)+Sb/m*Y+Sb*(p*w-r*u)+(Sa*Cb)/m*Z+(Sa*
    Cb)*(q*u-p*v);
96 alphapunto=1/(m*V*Cb)*(-L+Z_T*Ca-X_T*Sa+m*g*(Ca*Cp*Ct+Sa*St))+q-tan(
    beta_der)*(p*Ca+r*Sa);
97 betapunto=1/(m*V)*(D*Sb+Y_A*Cb-X_T*Ca*Sb+Y_T*Cb-Z_T*Sa*Sb+m*g*(Ca*Sb*St+
    Cb*Sp*Ct-Sa*Sb*Cp*Ct))+p*Sa-r*Ca;
98
99 % upunto=Vpunto*Ca*Cb;
100 % vpunto=Vpunto*Sb;
101 % wpunto=Vpunto*Sa*Cb;
102
103 % Costruzione della funzione integranda
104
105 DVAB=[Vpunto;alphapunto;betapunto];
106 DPQR=myAC.I_matrix\[([L_roll_A+L_roll_T;M_pitch_A+M_pitch_T;N_yaw_A+
    N_yaw_T]+...
    -omegatilde*myAC.I_matrix*[p;q;r]);
107 % DXYZ=T_EB*[u;v;w];
108 % DPTP=T_gimb*[p;q;r];
109
110 J_function = sum([DVAB;DPQR].^2);

```

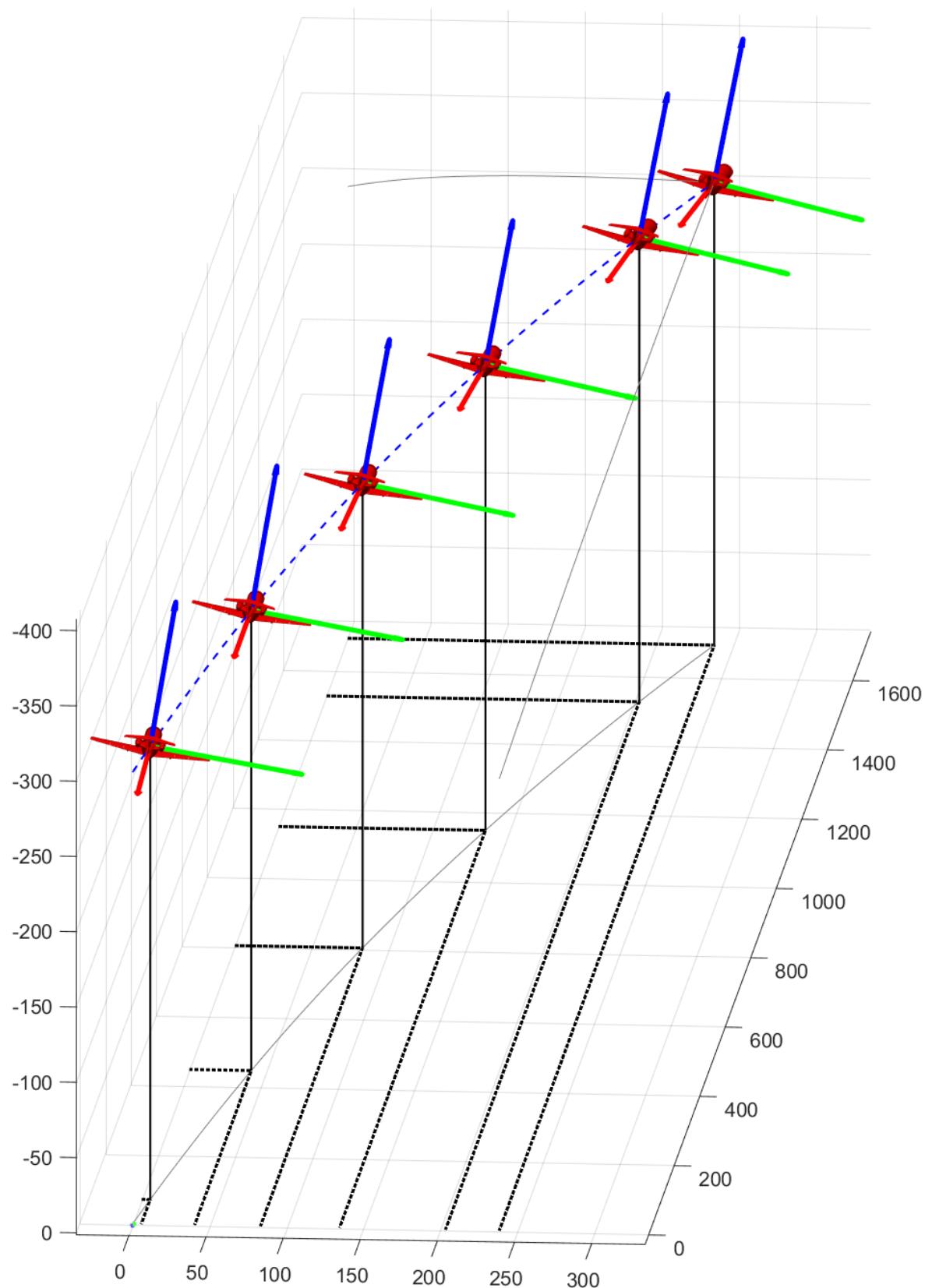


Figura 3.10 Visualizzazione di 20 secondi di volo

Capitolo

4

Superfici di governo

4.1 Volo longitudinal-simmetrico a comandi liberi

L'obiettivo di questo capitolo è esaminare come si comporta l'elevatore di coda del Citation I al variare della velocità di volo nel tempo. A tal scopo è stato sviluppato un modello aerodinamico a 3DOF per il Citation ed aggiunte due variabili di stato per modellare anche la dinamica dell'elevatore. In totale le variabili di stato sono: $V, \alpha_B, q, \theta, X, Z, \delta_e, \dot{(\delta_e)}$.

Lo stato dell'equilibratore, assunto per convenienza staticamente compensato, è governato dalle seguenti equazioni:

$$\ddot{\delta}_e = \frac{H_a + \cos \Lambda_e \cdot \dot{q} + H_c}{I_{ce}}$$

Dove $\Lambda_e = 3deg$ e I_c è stato stimato in prima approssimazione, considerando solo il peso della skin dell'elevatore essere circa $4kg \cdot m^2$. L'area dell'elevatore è stata prelevata dal modello datcom, allo stesso modo per la corda dell'elevatore.

4.1.1 Modello aerodinamico 3DOF

Listing 4.1

```
1      myAC
2
3  % Assegnazione delle componenti del vettore di stato
4  % deltae diventa variabile di stato, equilibratore supposto staticamente
5  % compensato
6
7  V = state(1);
8  alpha_B = state(2);
9  q = state(3);
10 xEG = state(4);
11 zEG = state(5);
12 theta = state(6);
13 u=V*cos(alpha_B);
14 w=V*sin(alpha_B);
```

```

15 delta_ePunto = state(7);
16 delta_e = state(8);
17
18 %% Quantit caratteristiche
19 % Equazioni generali del moto del velivolo
20 omegatilde = [0, 0, q;
21             0, 0, 0;
22             -q, 0, 0];
23
24 % Equazioni cinematiche ausiliari
25 T_BE = angle2dcm(0,theta,0,'ZYX');
26 T_EB = T_BE';
27 T_gimb = [1, 0, sin(theta)/cos(theta);
28            0, 1, 0;
29            0, 0, 1/cos(theta)];
30
31 %Propriet termodinamiche
32 rho = density(-zEG);
33
34 %% Definizione delle forze e dei momenti agenti sul velivolo
35 % Componenti in assi velivolo della forza risultante
36 C_L=CL(convang(alpha_B,'rad','deg'),...
37           convang(delta_e,'rad','deg'), convang(delta_f(t
38               ),'rad','deg'));
38 L = 0.5*rho*V^2*myAC.S*C_L;
39 D = 0.5*rho*V^2*myAC.S*CD(convang(alpha_B,'rad','deg'), convang(delta_f(t
40               ),'rad','deg'));
41 X_T = delta_T(t)*myAC.T_max_SL*cos(myAC.mu_T);
42 %Come richiesto, si assume che la spinta propulsiva massima disponibile
43 % sia
44 %costante al variare della quota e della velocit e che sia pari al
45 %valore
46 %della spinta propulsiva massima disponibile al livello del mare.
47 Z_T = delta_T(t)*myAC.T_max_SL*sin(myAC.mu_T);
48 X = X_T - D*cos(alpha_B) + L*sin(alpha_B) - myAC.W*sin(theta);
49 Z = Z_T - D*sin(alpha_B) - L*cos(alpha_B) + myAC.W*cos(theta);
50
51 % Componenti in assi velivolo del momento risultante
52 M_pitch_A = 0.5*rho*V^2*myAC.S*myAC.mac*Cpitch(convang(alpha_B,'rad','deg
53               '),...
54               convang(delta_e,'rad','deg
55               '),...
56               q);
57 C_M_T = (delta_T(t)*myAC.T_max_SL*myAC.e_T)/(0.5*rho*V^2*myAC.S*myAC.mac)
58 ;
59 M_pitch_T = 0.5*rho*V^2*myAC.S*myAC.mac*C_M_T;
60
61 %Prime 3 equazioni
62 Ca=cos(alpha_B);
63 Sa=sin(alpha_B);
64 Cb=cos(theta);
65 Sb=sin(theta);
66 Ct=cos(theta);
67 St=sin(theta);
68 Cp=cos(theta);

```

```

64 Sp=sin(θ);
65 m=myAC.mass;
66 Vpunto=((Ca)/m)*X+Ca*q*w+Sa/m*Z+Sa*q*u;
67 alphapunto=1/(m*V)*(-L+Z_T*Ca-X_T*Sa+m*g*(Ca*Ct+Sa*St))+q;
68
69 % Costruzione della funzione integranda
70
71 DVA=[Vpunto;alphapunto];
72 temp=omegatilde*myAC.I_matrix(2,2)*[0;q;0];
73 DQ=myAC.I_matrix(2,2)\(M_pitch_A+M_pitch_T-temp(2));
74 DXZ=T_EB*[u;0;w];DXZ=[DXZ(1);DXZ(3)];
75 DT=q;
76
77
78 % Dinamica dell'elevatore
79 %PHETE dal file di configurazione citation per datcom per ottenere
    LambdaeC
80 %CB dell'elevator per ottenere la corda media
81
82 LambdaeC = deg2rad(3);
83 Selev=1.17; %In m^2, calcolata a partire dai dati .dcm
84 Ihe = 4; %Stima rozza considerando l'elevatore un rettangolo
85 ce=convlength(0.84,'ft','m'); %Corda elevatore
86 Hce=0;
87 Hae=ce*Selev*0.5*rho*V^2*Chae(rad2deg(alpha_B),rad2deg(delta_e));
88
89
90
91
92 delta_eDuePunti = (cos(LambdaeC) * (DQ) + Hae + Hce)/Ihe;
93 delta_ePunto = delta_eDuePunti;
94
95
96 dStatedt=[DVA;DQ;DXZ;DT;delta_eDuePunti; delta_ePunto];
97
98 if t>=10
99     alpha_B
100 end

```

4.1.2 Coefficiente di momento aerodinamico elevatore

Listing 4.2

```

1 function [Cha_e] = Chae(alpha_B, delta_e)
2
3 global aeroDat
4
5 %Includo l'effetto del downwash
6 epsilon = interp1(aeroDat{1,3}.alpha, aeroDat{1,3}.eps, alpha_B);
7 alpha_Elevator = alpha_B - epsilon;
8
9 Cha_e = -aeroDat{1,3}.cha_sym(1)*alpha_Elevator - aeroDat{1,3}.chd_sym(1)
    *delta_e;
10
11
12 end

```

4.1.3 Risultati

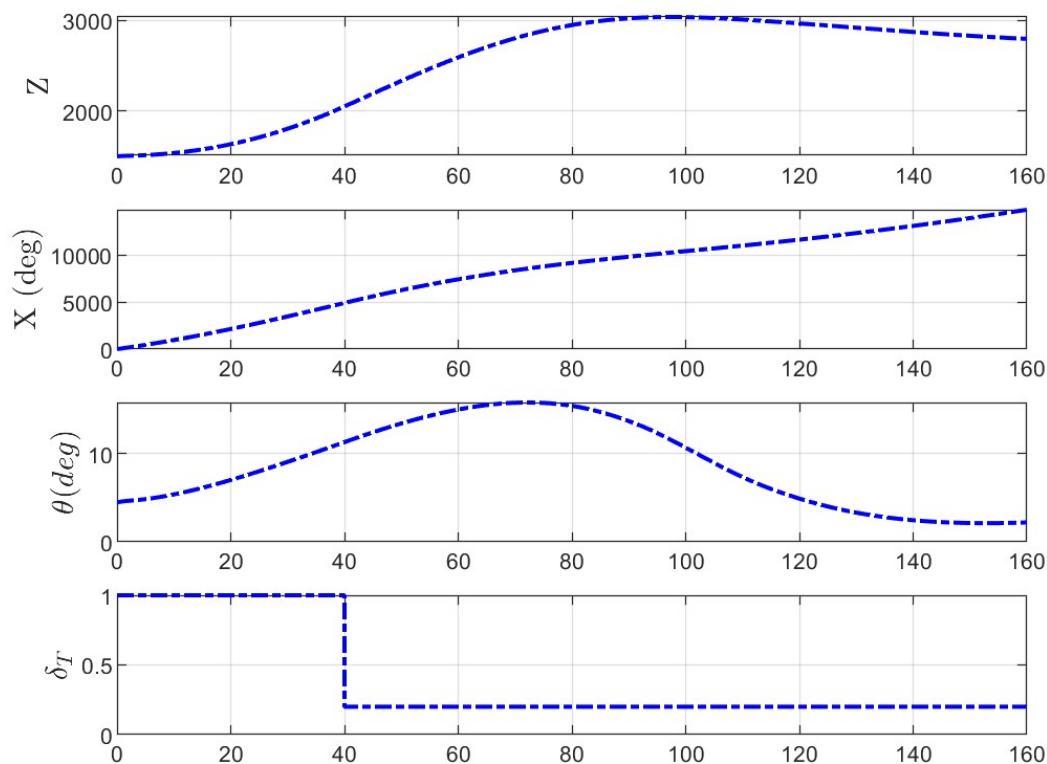


Figura 4.1 Parametri di volo

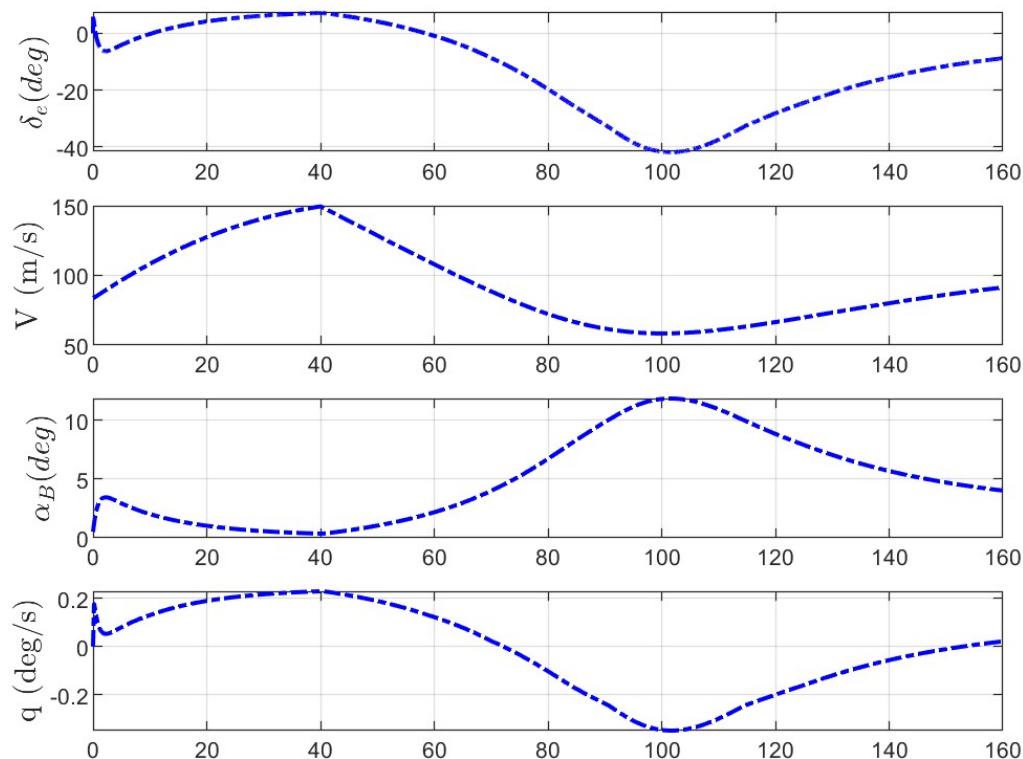


Figura 4.2 Parametri di volo

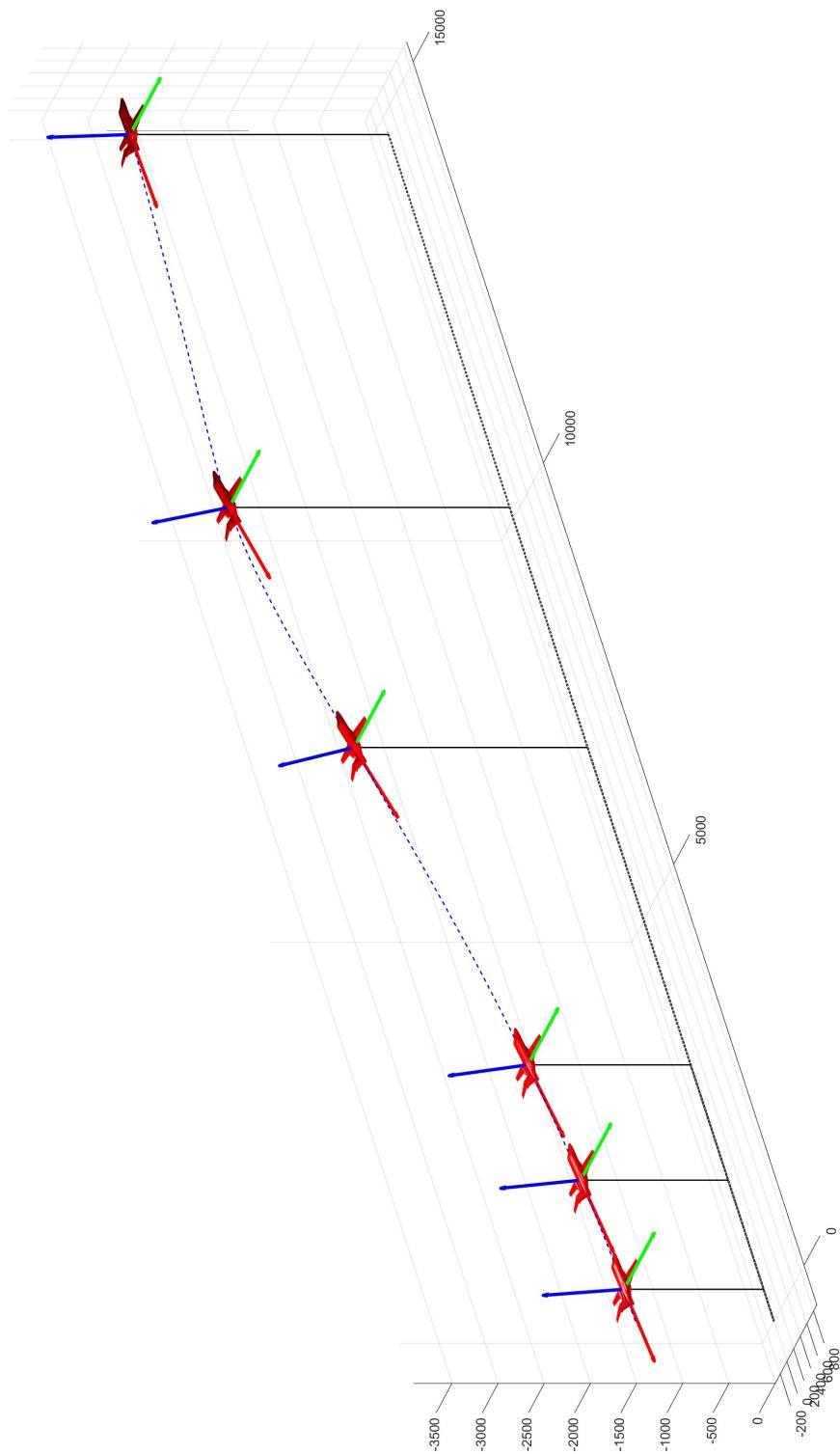


Figura 4.3 Traiettoria

4.2 Visualizzazione dell'accoppiamento tra dinamica del velivolo e dinamica dell'equilibratore

Come è noto anche se un equilibratore è staticamente compensato, a meno di utilizzare angoli di freccia di cerniera molto elevati, è presente un accoppiamento tra la dinamica del velivolo e la dinamica dell'equilibratore. Si è scelto di visualizzare la differenza tra le deflessioni dell'equilibratore tenendo conto dell'effetto di accoppiamento in una prima simulazione e poi eliminandolo in una seconda simulazione.

Le deflessioni degli alettoni e timone imposte sono le stesse e, a meno del piccolo effetto della deflessione del timone nell'arco del secondo simulato, la cinematica del velivolo è pressochè uguale.

4.2.1 Modello 6DOF

Listing 4.3

```

1 function [dStatedt] = eqLongDynamicFreeElevator_6DoF_Vabpqr_Euler(t,state
)
2
3 %%
4
5
6
7
8 global g...
9 delta_a delta_r delta_T delta_f ...
10 myAC
11
12 % Assegnazione delle componenti del vettore di stato
13 % deltae diventa variabile di stato, equilibratore supposto staticamente
14 % compensato
15
16 V = state(1);
17 alpha_B = state(2);
18 beta_der = state(3);
19 p = state(4);
20 q = state(5);
21 r = state(6);
22 xEG = state(7);
23 yEG = state(8);
24 zEG = state(9);
25 phi = state(10);
26 theta = state(11);
27 psi = state(12);
28 u=V*cos(alpha_B)*cos(beta_der);
29 v=V*sin(beta_der);
30 w=V*cos(beta_der)*sin(alpha_B);
31 delta_ePunto = state(13);
32 delta_e = state(14);
33
34 %% Quantit caratteristiche
35 % Equazioni generali del moto del velivolo

```

```

36 omegatilde = [0, -r, q;
37             r, 0, -p;
38             -q, p, 0];
39
40 % Equazioni cinematiche ausiliari
41 T_BE = angle2dcm(psi,theta,phi,'ZYX');
42 T_EB = T_BE';
43 T_gimb = [1, sin(phi)*sin(theta)/cos(theta), cos(phi)*sin(theta)/cos(theta);
44             0, cos(phi), -sin(phi);
45             0, sin(phi)/cos(theta), cos(phi)/cos(theta)
46 ];
47
48 %Parametri cinematici
49 V = (u^2 + v^2 + w^2)^(1/2);
50 alpha_B = atan(w/u);
51 beta_der = asin(v/V);
52
53 %Propriet termodinamiche
54 rho = density(-zEG);
55
56 %% Definizione delle forze e dei momenti agenti sul velivolo
57 % Componenti in assi velivolo della forza risultante
58 C_L=CL(convang(alpha_B,'rad','deg'),...
59         convang(delta_e,'rad','deg'), convang(delta_f(t
60         ),'rad','deg')));
61 L = 0.5*rho*V^2*myAC.S*C_L;
62 D = 0.5*rho*V^2*myAC.S*CD(convang(alpha_B,'rad','deg'), convang(delta_f(t
63         ),'rad','deg')));
64 Y_A = 0.5*rho*V^2*myAC.S*CY_A(C_L,...
65         convang(beta_der,'rad','deg'),p);
66 X_T = delta_T(t)*myAC.T_max_SL*cos(myAC.mu_T);
67 %Come richiesto, si assume che la spinta propulsiva massima disponibile
68 %sia
69 %costante al variare della quota e della velocit e che sia pari al
70 %valore
71 %della spinta propulsiva massima disponibile al livello del mare.
72 Y_T = 0;
73 Z_T = delta_T(t)*myAC.T_max_SL*sin(myAC.mu_T);
74 X = X_T - D*cos(alpha_B) + L*sin(alpha_B) - myAC.W*sin(theta);
75 Y = Y_T + Y_A + myAC.W*sin(phi)*cos(theta);
76 Z = Z_T - D*sin(alpha_B) - L*cos(alpha_B) + myAC.W*cos(phi)*cos(theta);
77
78 % Componenti in assi velivolo del momento risultante
79 L_roll_A = 0.5*rho*V^2*myAC.S*myAC.b*Croll(convang(alpha_B,'rad','deg')
80         ,...
81         convang(beta_der,'rad','deg'))
82         ,...
83         convang(delta_a(t),'rad','deg')
84         ),...
85         convang(delta_r(t),'rad','deg')
86         ),...
87         p,r);
88 L_roll_T = 0;
89 M_pitch_A = 0.5*rho*V^2*myAC.S*myAC.mac*Cpitch(convang(alpha_B,'rad','deg')

```

```

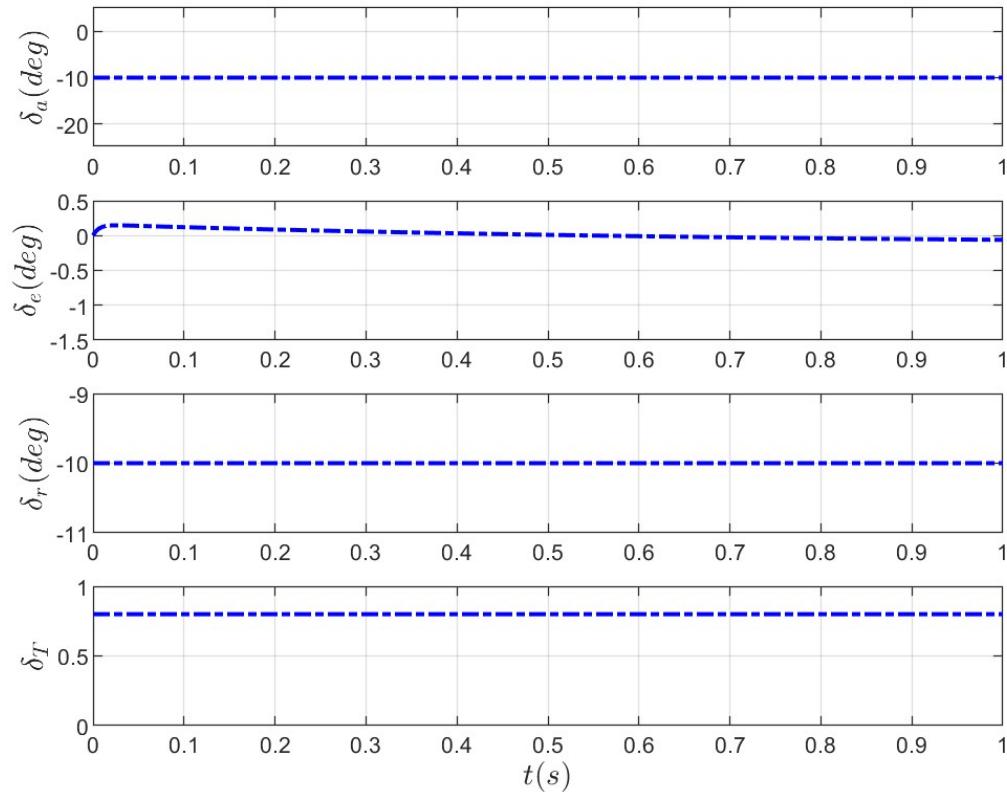
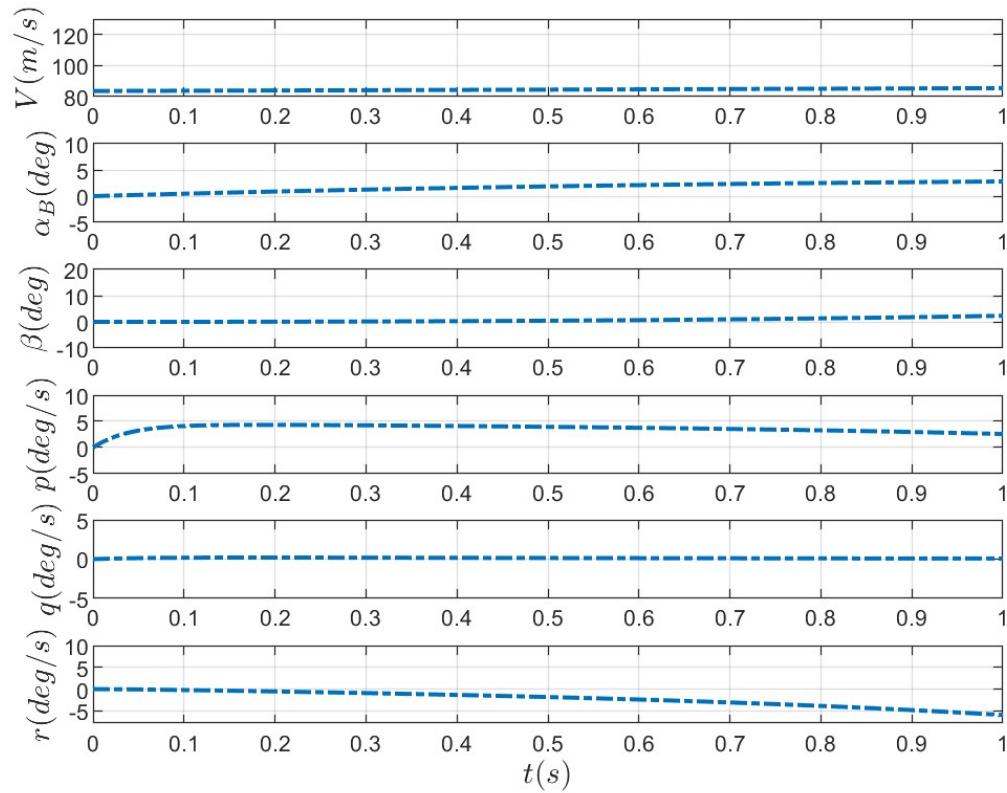
        ') , ...
82                                     convang(delta_e , 'rad' , 'deg
83                                     ') , ...
84 C_M_T = (delta_T(t)*myAC.T_max_SL*myAC.e_T)/(0.5*rho*V^2*myAC.S*myAC.mac)
85 ;
86 M_pitch_T = 0.5*rho*V^2*myAC.S*myAC.mac*C_M_T;
87 N_yaw_A = 0.5*rho*V^2*myAC.S*myAC.b*Cyaw(convang(alpha_B , 'rad' , 'deg') , ...
88                                         C_L , ...
89                                         convang(beta_der , 'rad' , 'deg')
90                                         , ...
91                                         p , ...
92                                         convang(delta_r(t) , 'rad' , 'deg'))
93 ;
94
95
96
97 %Prime 3 equazioni
98 Ca=cos(alpha_B);
99 Sa=sin(alpha_B);
100 Cb=cos(beta_der);
101 Sb=sin(beta_der);
102 Ct=cos(theta);
103 St=sin(theta);
104 Cp=cos(phi);
105 Sp=sin(phi);
106 m=myAC.mass;
107 Vpunto=((Ca*Cb)/m)*X+Ca*Cb*(r*v-q*w)+Sb/m*Y+Sb*(p*w-r*u)+(Sa*Cb)/m*Z+(Sa*
108     Cb)*(q*u-p*v);
109 alphapunto=1/(m*V*Cb)*(-L+Z_T*Ca-X_T*Sa+m*g*(Ca*Cp*Ct+Sa*St))+q-tan(
110     beta_der)*(p*Ca+r*Sa);
111 betapunto=1/(m*V)*(D*Sb+Y_A*Cb-X_T*Ca*Sb+Y_T*Cb-Z_T*Sa*Sb+m*g*(Ca*Sb*St+
112     Cb*Sp*Ct-Sa*Sb*Cp*Ct))+p*Sa-r*Ca;
113
114 % upunto=Vpunto*Ca*Cb;
115 % vpunto=Vpunto*Sb;
116 % wpunto=Vpunto*Sa*Cb;
117
118 % Costruzione della funzione integranda
119 DVAB=[Vpunto;alphapunto;betapunto];
120 DPQR=myAC.I_matrix\[([L_roll_A+L_roll_T;M_pitch_A+M_pitch_T;N_yaw_A+
121     N_yaw_T]+...
122     -omegatilde*myAC.I_matrix*[p;q;r]);
123 DXYZ=T_EB*[u;v;w];
124 DPTP=T_gimb*[p;q;r];
125
126 % Dinamica dell'elevatore
127 %PHETE dal file di configurazione citation per datcom per ottenere
128     Lambdaec
129 %CB dell'elevator per ottenere la corda media
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
878
879
880
881
882
883
884
885
886
887
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
917
918
919
919
920
921
922
923
924
925
926
927
927
928
929
929
930
931
932
933
934
935
936
937
937
938
939
939
940
941
942
943
944
945
945
946
947
947
948
949
949
950
951
952
953
954
955
956
956
957
958
958
959
959
960
961
962
963
964
965
965
966
966
967
967
968
968
969
969
970
971
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
981
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1580
1581
1581
1582
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1590
1591
1591
1592
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1600
1601
1601
1602
1602
1603
1603
1604
1604
1605
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1610
1611
1611
1612
1612
1613
1613
1614
1614
1615
1615
1616
1616
1617
1617
1618
1618
1619
1619
1620
1620
1621
1621
1622
1622
1623
1623
1624
1624
1625
1625
1626
1626
1627
1627
1628
1628
1629
1629
1630
1630
1631
1631
1632
1632
1633
1633
1634
1634
1635
1635
1636
1636
1637
1637
1638
1638
1639
1639
1640
1640
1641
1641
1642
1642
1643
1643
1644
1644
1645
1645
1646
1646
1647
1647
1648
1648
1649
1649
1650
1650
1651
1651
1652
1652
1653
1653
1654
1654
1655
1655
1656
```

```

128 LambdaeC = deg2rad(3);
129 Selev=1.17; %In m^2, calcolata a partire dai dati .dcm
130 Ihe = 4; %Stima rozza considerando l'elevatore un rettangolo
131 ce=convlength(0.84,'ft','m');
132 Hce=0;
133 Hae=ce*Selev*0.5*rho*V^2*Chae(rad2deg(alpha_B),rad2deg(delta_e));
134
135
136
137
138 delta_eDuePunti = (cos(LambdaeC) * (DPQR(2)-p*r) + Hae + Hce)/Ihe;
139 delta_ePunto = delta_eDuePunti;
140
141
142
143 dStatedt=[DVAB;DPQR;DXYZ;DPTP;delta_eDuePunti; delta_ePunto];
144
145
146 % dStatedt = ([-omegatilde, zeros(3);
147 % zeros(3), myAC.I_matrix\(-omegatilde*myAC.I_matrix)
148 % ;
149 % T_EB, zeros(3); zeros(3);
150 % zeros(3), T_gimb]*[u;
151 % v;
152 % w;
153 % p;
154 % q;
155 % r])+...
156 % [(g/myAC.W)*X;
157 % (g/myAC.W)*Y;
158 % (g/myAC.W)*Z;
159 % myAC.I_matrix\[L_roll_A + L_roll_T;
160 % M_pitch_A + M_pitch_T;
161 % N_yaw_A + N_yaw_T];
162 % zeros(6,1)];
163 %dstatedt[12x1] = Matrix[12x6]*Matrix[6x1] + Matrix[12x1]
164 %rad2deg(alpha_B)
165 t
166 end

```

4.2.2 Variabili di volo e risultati

**Figura 4.4** Deflessione superfici di controllo**Figura 4.5** Cinematica del velivolo

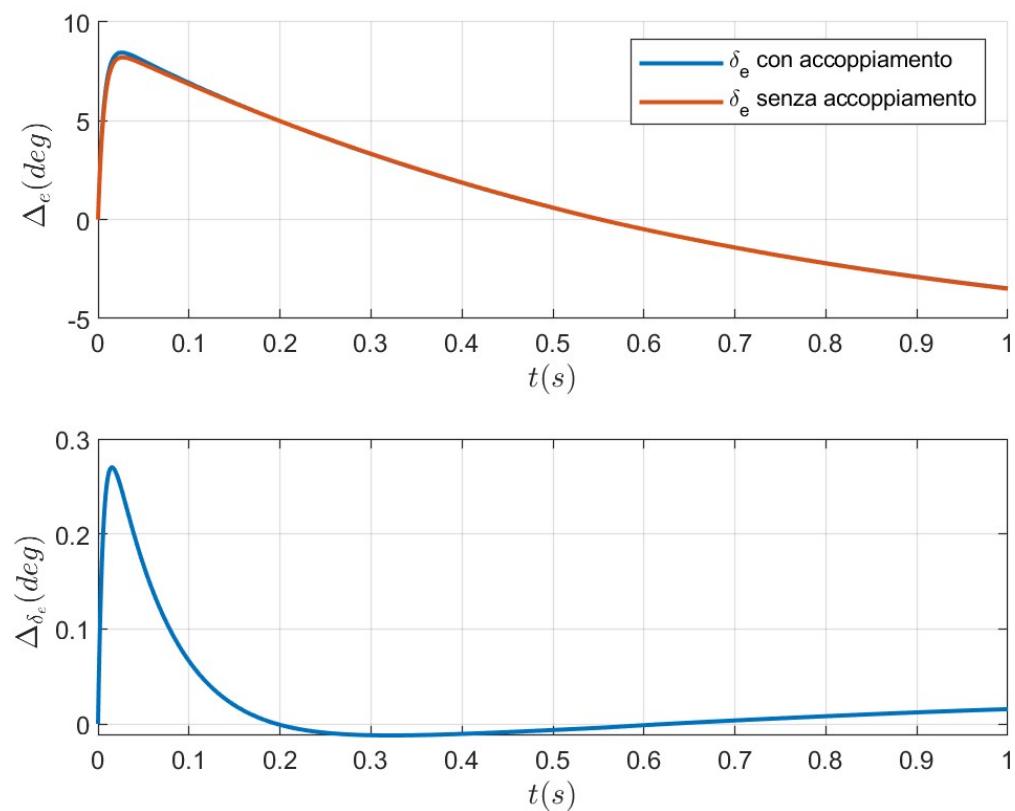


Figura 4.6 Deflessione elevatore con e senza accoppiamento

Capitolo

5

Piccole perturbazioni del moto di un velivolo

5.1 Risposta di un velivolo ad una raffica verticale

In questo capito si propone lo studio della dinamica linearizzata di un velivolo 747 in risposta all'interruzione di una raffica verticale con velocità variabile lungo l'apertura alare. La matrice della dinamica libera laterodirezionale è stata resa una funzione in modo da potere ottenerne i valori al variare della derivata del coefficiente di rollio rispetto alla velocità di rollio p . È stata implementata la matrice B per la risposta del sistema agli input dagli alettoni e timone. Il numero di Mach è stato tenuto fisso al valore 0.25. Per schematizzare l'effetto della raffica verticale questa è stata convertita in una velocità di rollio equivalente come indicato nella 5.1, w_{Tip} è la velocità vista dalla tip dell'ala, b è l'apertura alare del 747.

$$p = 2 \frac{w_{Tip}}{b} \quad (5.1)$$

La condizione iniziale è stata quindi posta pari a $[0, 0, p, 0]$ e si è analizzata la risposta al variare dello smorzamento al rollio dovuto alla velocità p stessa.

5.1.1 Codice Matlab

Listing 5.1

```
1 %% Qui inizia il mio lavoro
2
3
4 % Rendo Ald una funzione cos che possa variare i valori di Lp
5
6     A_ldf = @(Lp_1) [ ...
7     Nr_1, Nbeta_1, Np_1, 0; ...
8     Yr_1/U_0 - 1, Ybeta_1/U_0, Yp_1/U_0, g_0/U_0; ...
9     Lr_1, Lbeta_1, Lp_1, 0; ...
10    0, 0, 1, 0];
```

```

11
12
13
14
15 A_ld = A_ldf(Lp_1);
16
17 B_ld = [ ...
18 Ndeltaa_1, Ndeltar_1; ...
19 Ydeltaa_1/U_0, Ydeltar_1/U_0; ...
20 Ldeltaa_1, Ldeltar_1; ...
21 0, 0];
22
23 %% Caratteristiche del 747 per i moti puri
24 %Non interessano il mio esercizio
25 % Lateral-Directional eigenvalues/eigenvectors
26 % [Vld,Dld] = eig(A_ld); % right eigenvectors (columns) and eigenvalues
27 % Vld_Roll = Vld(:,1);
28 % Vld_Roll = Vld_Roll/Vld_Roll(4,1);
29 % Vld_Spiral = Vld(:,4);
30 % Vld_Spiral = Vld_Spiral/Vld_Spiral(4,1);
31 % lambda_DR = Dld(2,2);
32 % Vld_DR = Vld(:,2);
33 % Vld_DR = Vld_DR/Vld_DR(4,1);
34 % sigma_DR = real(lambda_DR);
35 % omega_DR = imag(lambda_DR);
36 % zeta_DR = zeta(sigma_DR,omega_DR);
37 % omega_n_DR = omega_n(sigma_DR,omega_DR);
38 % T_DR = Period(sigma_DR,omega_DR);
39 % t_half_DR = t_half(sigma_DR,omega_DR);
40 % N_half_DR = N_half(sigma_DR,omega_DR);
41
42 %% Risposta del 747 a deflessioni alterne degli alettoni
43 % Esercizio 16.8
44 %Fatto da me
45 %Costruisco la matrice del sistema
46
47 SistLatDir = sysLatDir(A_ld, B_ld);
48
49 x_init = zeros(4,1);
50
51 % Import delle leggi di comando
52 comandi = load('LeggiDiComando.mat');
53 delta_a = @(t) deg2rad(interp1(comandi.delta_a.Time,comandi.delta_a.Data,
      t,"previous","extrap"));
54 delta_r = @(t) deg2rad(interp1(comandi.delta_r.Time,comandi.delta_r.Data,
      t,"previous","extrap"));
55
56
57
58 % Visualizzazione impulso alettoni
59 Nt=300;
60 Times = linspace(0,100,Nt);
61 comandi = [delta_a(Times)', zeros(Nt,1)];
62 y_Alettoni = lsim(SistLatDir,comandi,Times,x_init);
63 %Impulso sul timone

```

```

65 comandi = [zeros(Nt,1), delta_r(Times)'];
66 y_Rudder = lsim(SistLatDir,comandi,Times,x_init);
67
68
69
70
71
72 % Risposta ad una raffica lineare
73 %Raffica sostenuta per t<0, la raffica si interrompe al tempo 0,
    valutiamo
74 %la stabilit dinamica del velivolo a questo fenomeno
75
76 wTip = 1; %velocit vista dall'ala destra
77 b=59.6; %Wingspan del 747
78 p=wTip^2/b;%Velocit di rotazione p che darebbe la stessa velocit del
    vento alla tip
79
80 x0Raffica=[0,0,p,0];
81 Times = 0:0.5:100;
82 y_raffica=initial(SistLatDir,x0Raffica, Times);
83
84 figure(1);
85 plot(Times, y_raffica,LineWidth=1); % Risposta dell'aereo
86 legend('r','\beta','p','\phi');
87
88 % Prova a vedere come varia all'aumentare in valore assoluto di Clp (Roll
    )
89 % Significa aumentare Lp
90
91 % 10% maggiore
92 A_ld1 = A_ldf(Lp_1*1.1);
93 s1=sysLatDir(A_ld1,B_ld);
94 y_raffica1=initial(s1,x0Raffica, Times);
95 % 20% maggiore
96 A_ld2 = A_ldf(Lp_1*1.2);
97 s2=sysLatDir(A_ld2,B_ld);
98 y_raffica2=initial(s2,x0Raffica, Times);
99 % 30% maggiore
100 A_ld3 = A_ldf(Lp_1*1.3);
101 s3=sysLatDir(A_ld3,B_ld);
102 y_raffica3=initial(s3,x0Raffica, Times);
103 % 40% maggiore
104 A_ld4 = A_ldf(Lp_1*1.4);
105 s4=sysLatDir(A_ld4,B_ld);
106 y_raffica4=initial(s4,x0Raffica, Times);
107 % 50% maggiore
108 A_ld5 = A_ldf(Lp_1*1.5);
109 s5=sysLatDir(A_ld5,B_ld);
110 y_raffica5=initial(s5,x0Raffica, Times);
111
112 %Autovalori
113 figure(10);
114 realValues=[real(eig(A_ld)),real(eig(A_ld1)),real(eig(A_ld2)),real(eig(
    A_ld3)),real(eig(A_ld4)),real(eig(A_ld5))];
115 imaginaryValues=[imag(eig(A_ld)),imag(eig(A_ld1)),imag(eig(A_ld2)),imag(
    eig(A_ld3)),imag(eig(A_ld4)),imag(eig(A_ld5))];

```

```

116 scatter(realValues,imaginaryValues,MarkerEdgeColor="flat",MarkerFaceColor
    ="flat");
117 legend("0%","10%","20%","30%","40%","50%");
118 grid on;
119 %Plotting dei risultati
120
121 %Valori di r
122 figure(2);
123 plot(Times, [y_raffica(:,1),y_raffica1(:,1),y_raffica2(:,1),y_raffica3
    (:,1),y_raffica4(:,1),y_raffica5(:,1)]);
124 legend('0%','10%','20%','30%','40%','50%');
125 title('Grafico di r');
126 %Valori di beta
127 figure(3);
128 plot(Times, [y_raffica(:,2),y_raffica1(:,2),y_raffica2(:,2),y_raffica3
    (:,2),y_raffica4(:,2),y_raffica5(:,2)]);
129 legend('0%','10%','20%','30%','40%','50%');
130 title('Grafico di \beta');
131 %Valori di p
132 figure(4);
133 plot(Times, [y_raffica(:,3),y_raffica1(:,3),y_raffica2(:,3),y_raffica3
    (:,3),y_raffica4(:,3),y_raffica5(:,3)]);
134 legend('0%','10%','20%','30%','40%','50%');
135 title('Grafico di p');
136 %Valori di phi
137 figure(5);
138 plot(Times, [y_raffica(:,4),y_raffica1(:,4),y_raffica2(:,4),y_raffica3
    (:,4),y_raffica4(:,4),y_raffica5(:,4)]);
139 legend('0%','10%','20%','30%','40%','50%');
140 title('Grafico di \phi');
141 % Controllo
142
143 %Implementazione di un controllo in retroazione sul valore di p che v
    ad
144 %impartire una deflessione degli alettoni
145 %PID Tuned con il control system design toolkit di simulink
146
147 sistemaConControllo=sim('SistemaPID.slx');
148 figure(6);
149 Evoluzione = sistemaConControllo.yout.getElement('systemEvolution');
150 Evoluzione = Evoluzione.Values;
151 stato = interp1(Evoluzione.Time, Evoluzione.Data, 0:0.1:Evoluzione.Time(
    end), 'linear');
152 plot(0:0.1:Evoluzione.Time(end), stato,LineWidth=1);
153 title('Dinamica del velivolo con controllo in retroazione');
154 legend('r','\beta','p','\phi');

155
156 figure(7)
157 Alettoni = sistemaConControllo.yout.getElement('deflessioneAlettoni');
158 Alettoni = Alettoni.Values;
159 defAlettoni = interp1(Alettoni.Time, rad2deg(Alettoni.Data), 0:0.1:
    Alettoni.Time(end), 'linear');
160 plot(0:0.1:Evoluzione.Time(end), defAlettoni,LineWidth=1);
161 title('Deflessione degli alettoni');

162
163

```

```
164 function SistLatDir = sysLatDir(A_ld , B_ld)
165
166     SistLatDir = ss( ...
167         A_ld, ... % A
168         B_ld, ... % B
169         eye(4,4), ... % C
170         zeros(4,2) ... % D
171     );
172
173 end
```

5.1.2 Risultati

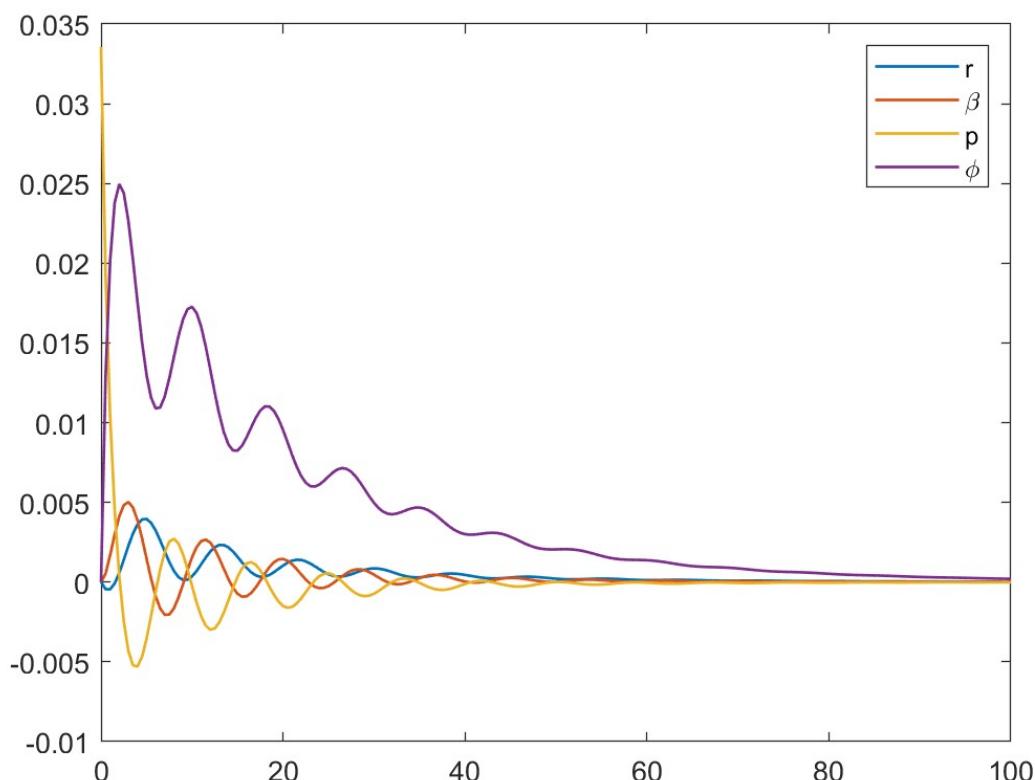


Figura 5.1 Risposta libera del velivolo alle condizioni iniziali

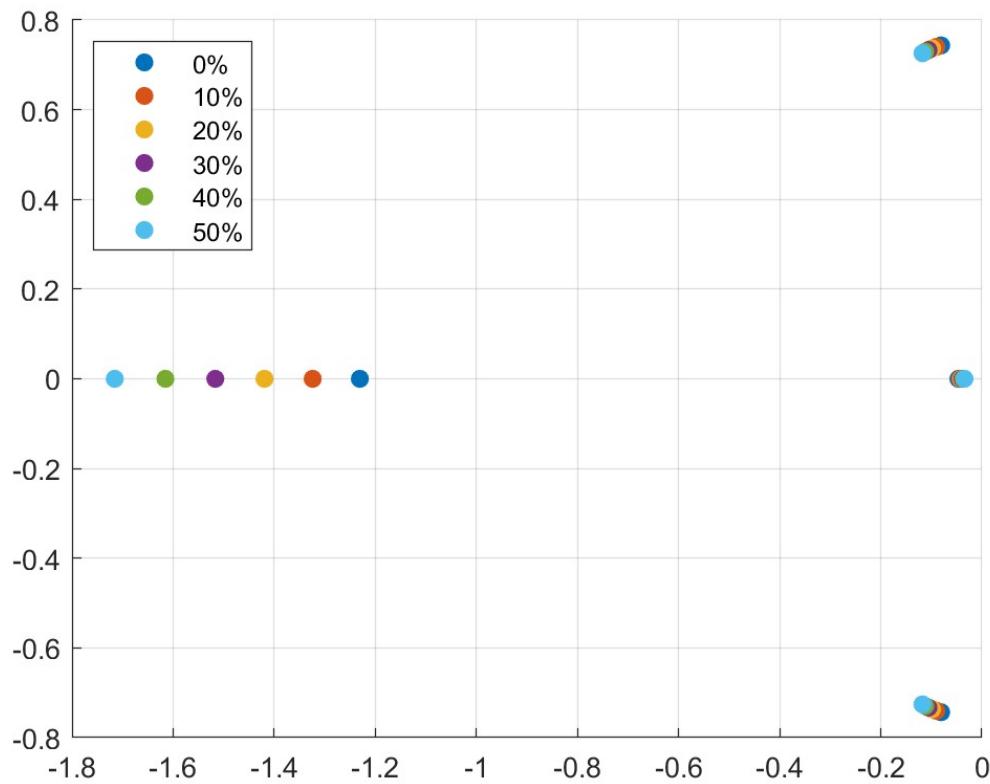


Figura 5.2 Variazione degli autovalori

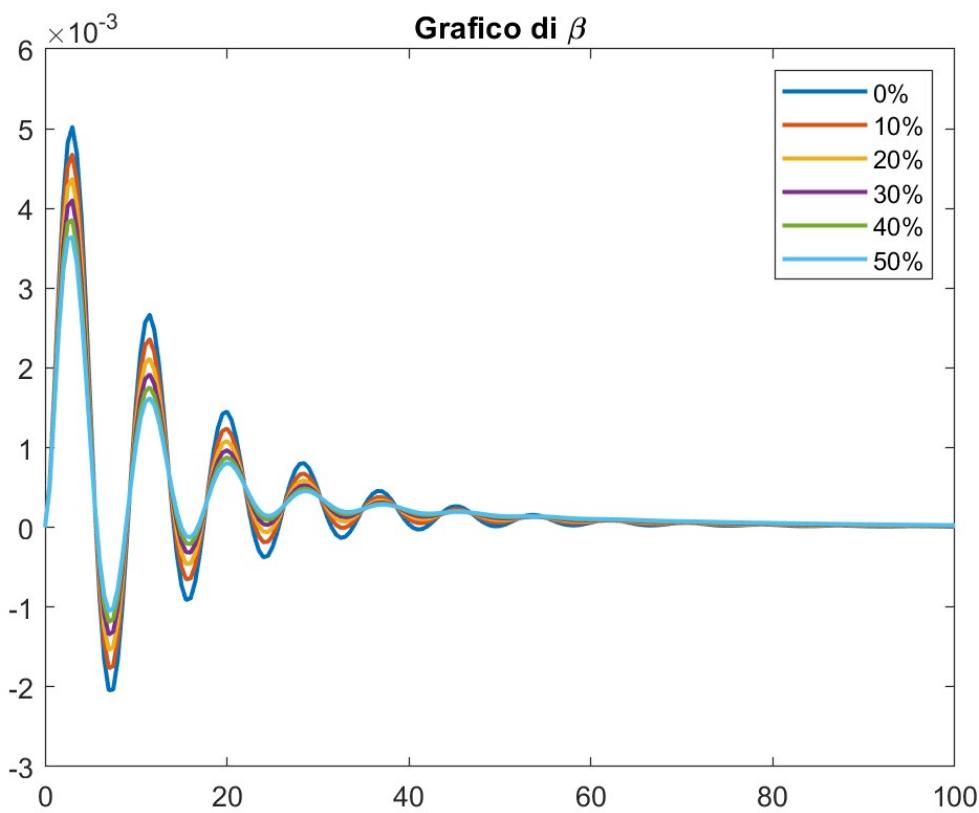


Figura 5.3 Variazione dell'angolo β

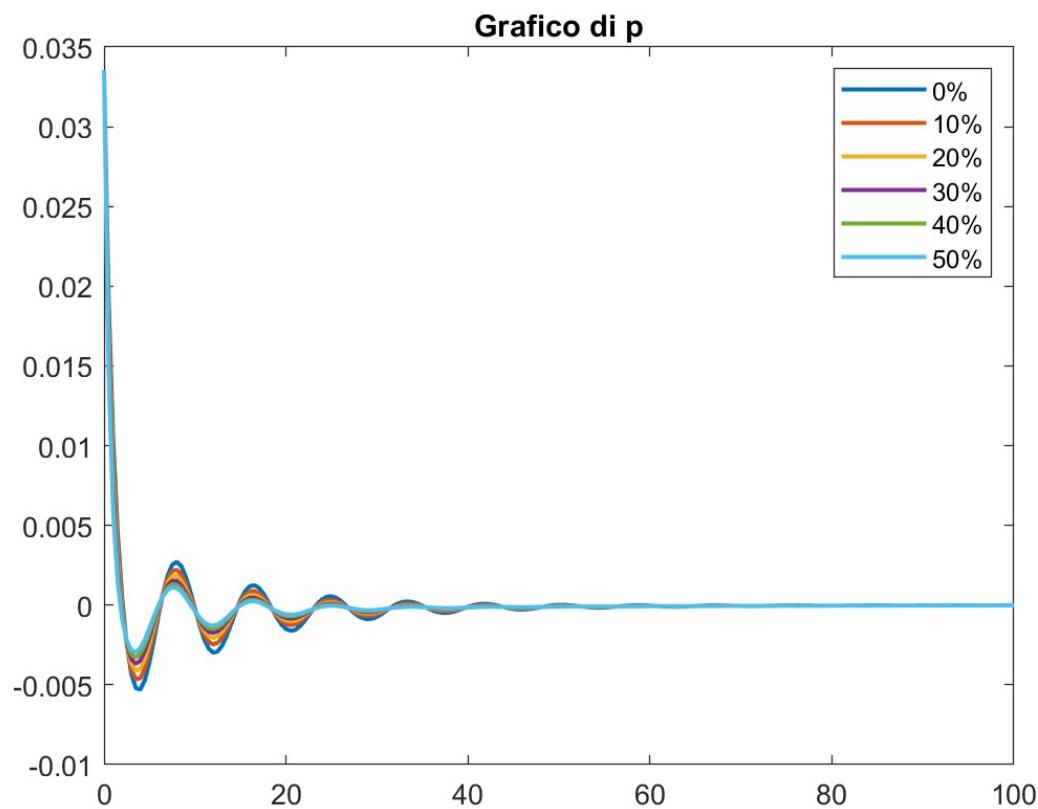


Figura 5.4 Variazione di p

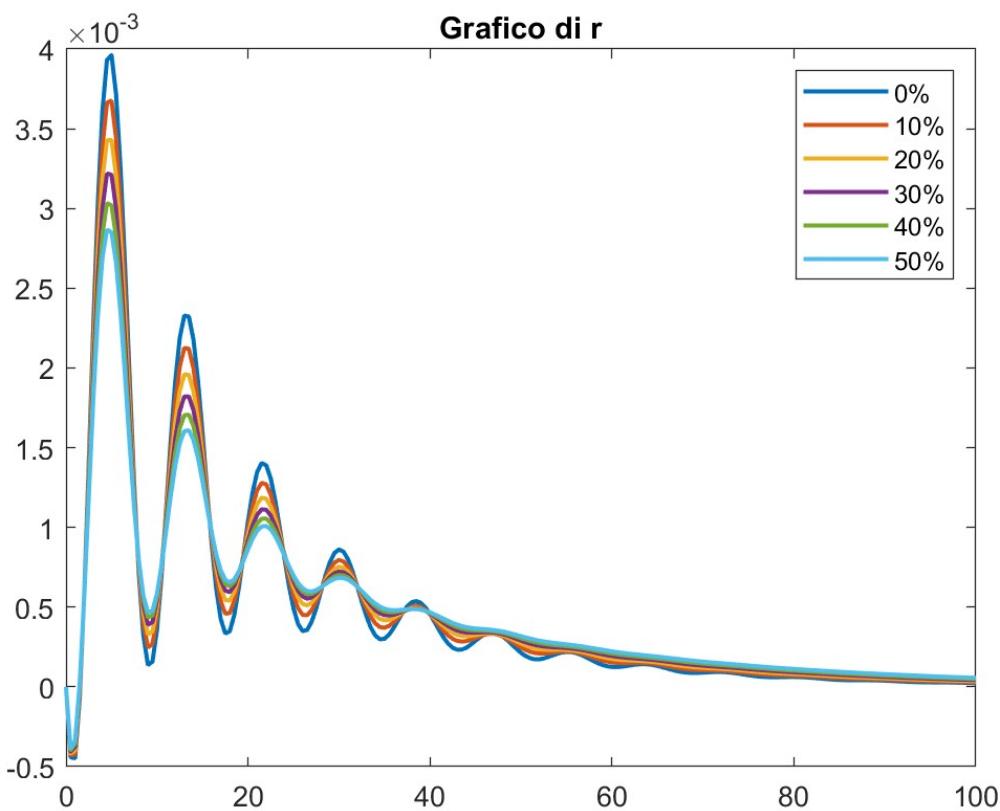


Figura 5.5 Variazione di r

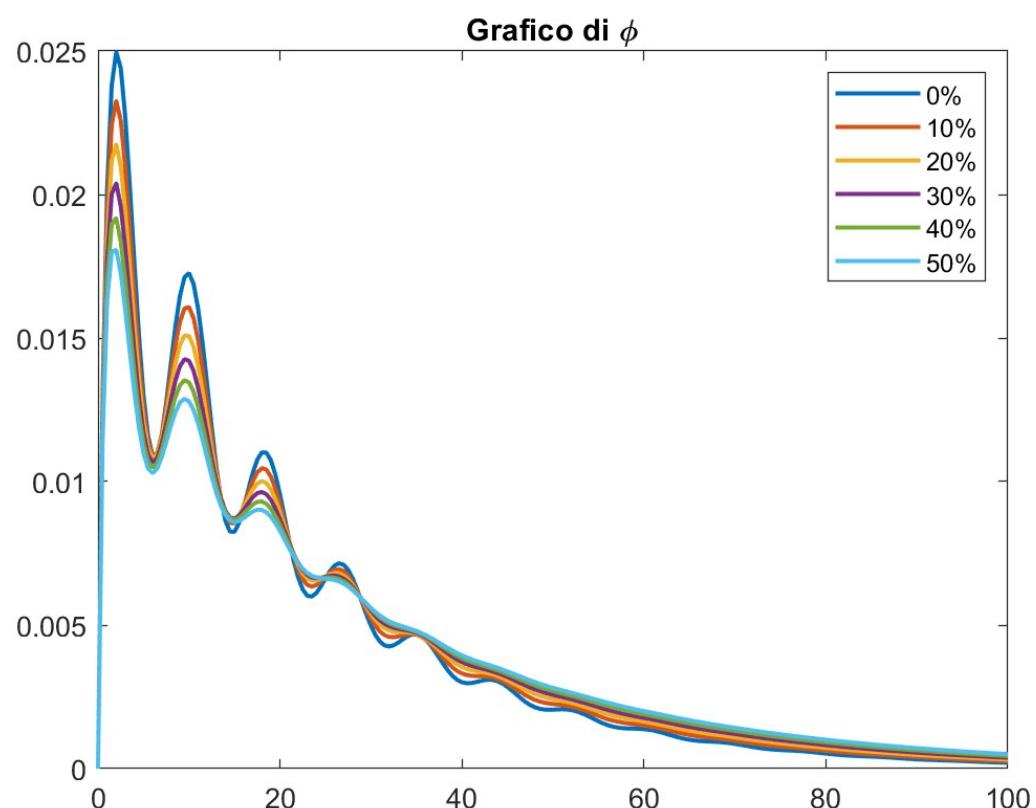


Figura 5.6 Variazione di ϕ

5.2 Implementazione di un controllo in retroazione

Questa sezione è un'ulteriore sviluppo di quanto fatto nella sezione (5.1), viene implementato un controllo in retroazione sulla velocità p che agisce sugli alettoni, per piccole perturbazioni questo sistema riesce a ridurre la durata e l'intensità delle oscillazioni.

Per ottenere la risposta libera desiderata è stato utilizzato il control system design toolbox di Simulink unitamente al tool PID Tuner.

5.2.1 Modello simulink e risultati

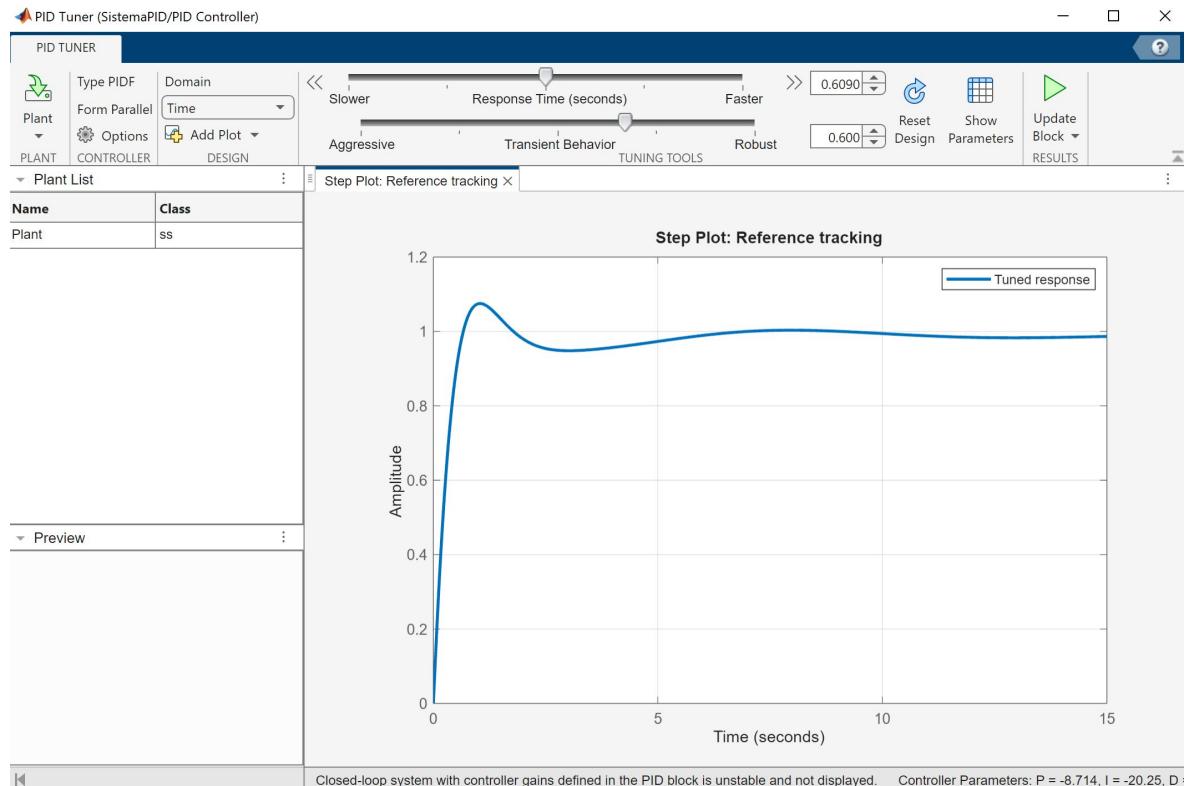
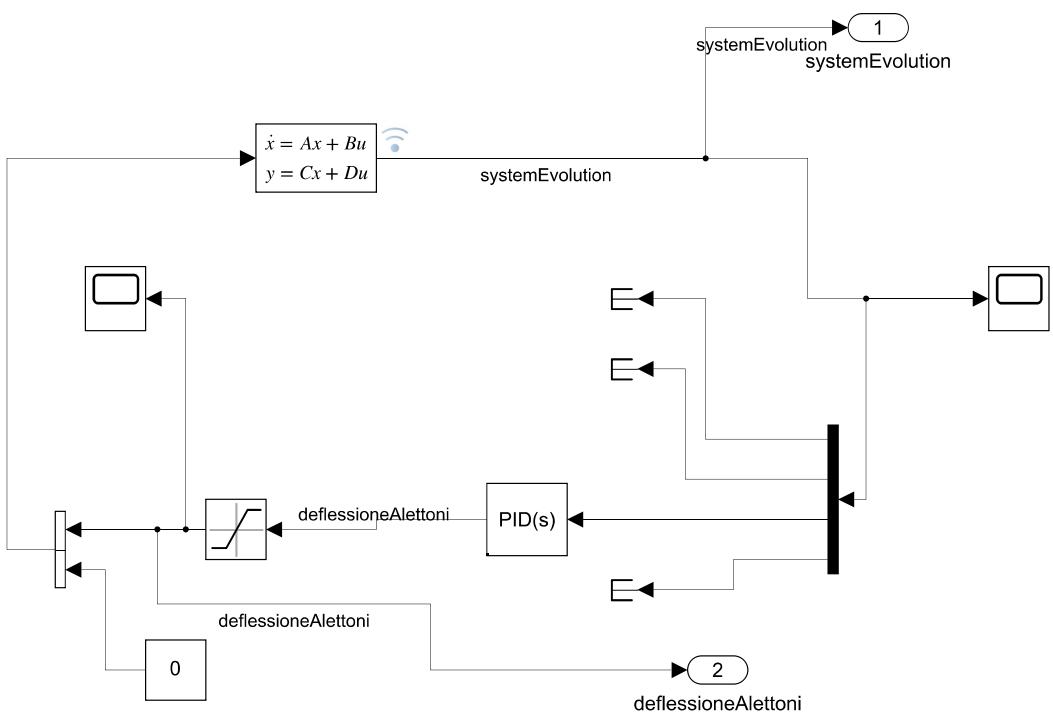


Figura 5.7 Schermata del tool PID Tuner



Deflessione rudder sempre nulla

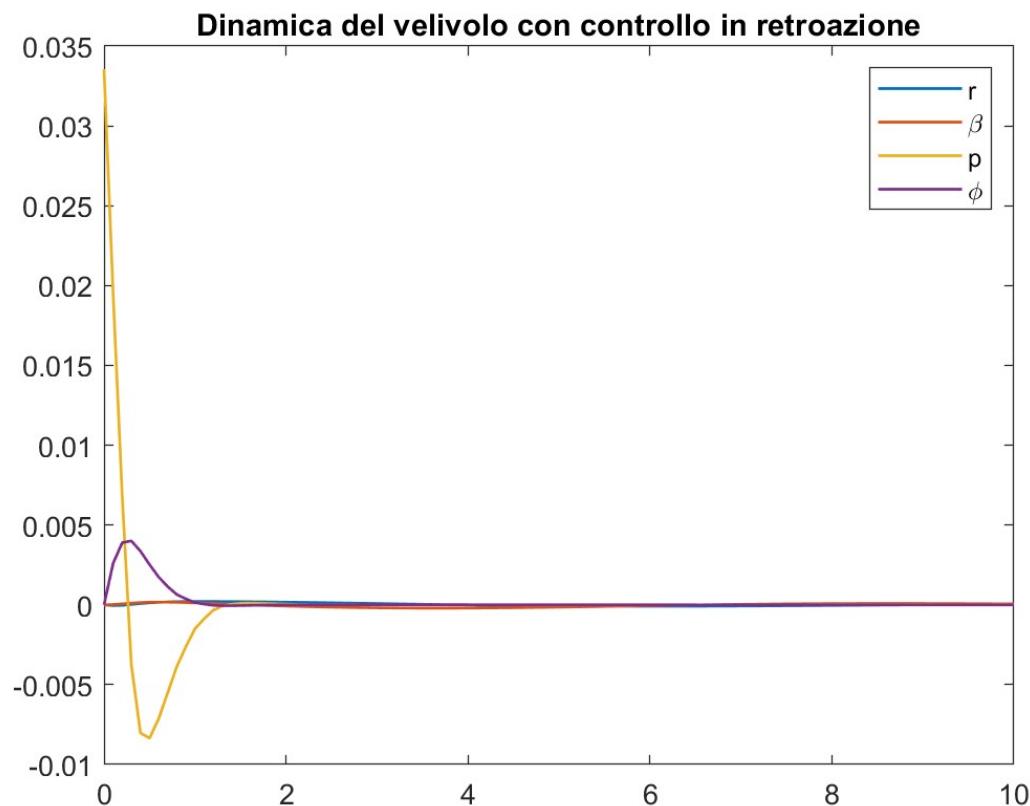


Figura 5.8 Dinamica del velivolo con controllo in retroazione

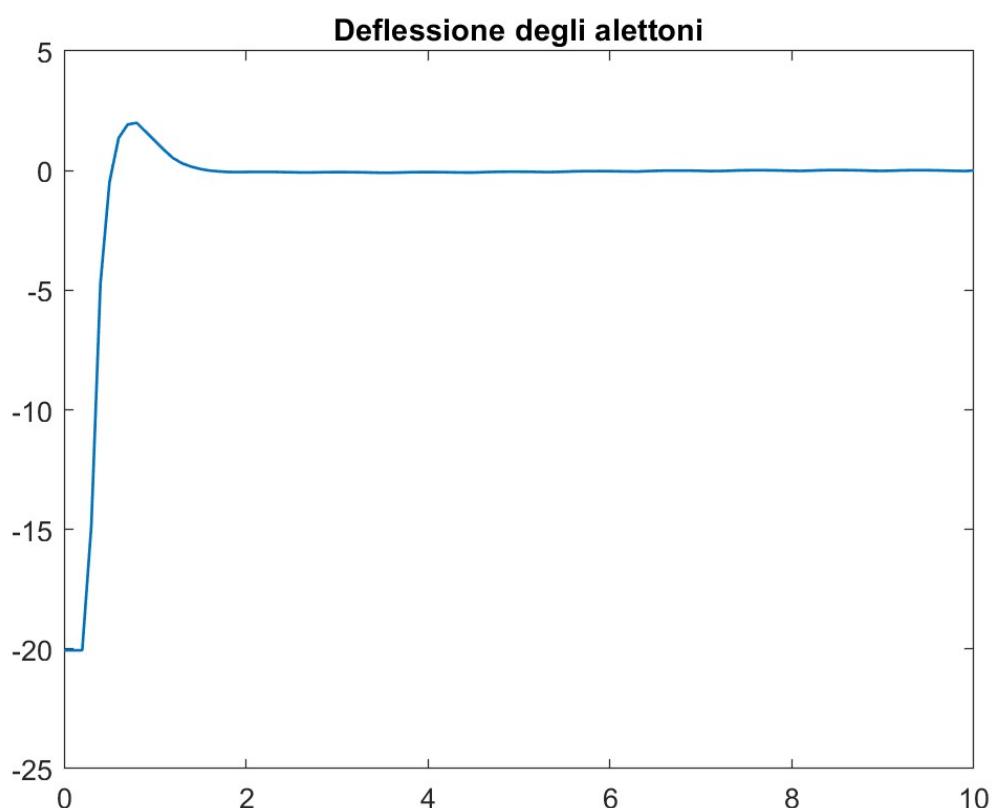


Figura 5.9 Risposta degli alettoni

Capitolo

6

Digital Datcom

Come si è visto, negli esercizi di questo capitolo è stato utilizzato il modello aerodinamico del citation I. I valori caratteristici sono stati utilizzati utilizzando il software Digital Datcom+. Come si può osservare nel paragrafo 6.0.1 sono state create 3 configurazioni:

1. Configurazione Body + Ali + Effetti propulsivi + flaps
2. Configurazione Body + Ali + Effetti propulsivi + alettoni
3. Configurazione completa + elevatore

Le configurazioni 1 e 2 vengono utilizzate esclusivamente per ottenere i valori delle potenze di controllo degli alettoni e dell'effetto dei flaps su CL e CD, per simulare il volo il resto dei valori sono prelevati dalla configurazione 3.

Sono stati creati 2 file di output con le stesse configurazioni ma a due numeri di mach diversi: 0.25 e 0.5 per visualizzare gli effetti della comprimibilità sulle caratteristiche aerodinamiche. L'altitudine è stata settata a circa 460 m.

6.0.1 File di configurazione per il Citation I

Listing 6.1

```
1  *
2  *   File : CITATION.dat
3  *
4  *   Purpose : This is an input file for the Digital Datcom program.
5  *
6  *   Author : Bill Galbraith
7  *           Holy Cows, Inc.
8  *           billg (at) holycows.net
9  *
10 *   Last update:
11 *       May 12, 2010 - Fixed vertical fin, removed ground and power
12 *                     effects.
13 *
```

```

14 ****
15 * List of Command Card
16 ****
17
18 DIM FT
19 DERIV DEG
20 DAMP
21 PART
22 * DUMP IOM
23
24
25
26 ****
27 * Flight Conditions *
28 ****
29
30
31 $FLTCON WT=7000.0, LOOP=2.0,
32      NMACH=1.0, MACH(1)=0.25,
33      NALT=1.0, ALT(1)=1500.0,
34      NALPHA=20.0,
35      ALSCHD(1)= -16.0, -8.0, -6.0, -4.0, -2.0, 0.0, 2.0, 4.0, 8.0,
36      9.0,
37      10.0, 12.0, 14.0, 16.0, 18.0, 19.0, 20.0, 21.0, 22.0, 24.0,
38      STMACH=0.6, TSMACH=1.4, TR=1.0$
39
40
41 ****
42 * Reference Parameters * pg 29
43 ****
44
45
46 $OPTINS SREF=320.8, CBARR=6.75, BLREF=51.7, ROUGFC=0.25E-3$
47
48
49
50 ****
51 * Group II      Synthesis Parameters * pg 33
52 ****
53
54 $SYNTHS XCG=21.9, ZCG=3.125,
55      XW=19.1, ZW=3.125, ALIW=2.5,
56      XH=39.2, ZH=7.75, ALIH=0.0,
57      XV=36.0, ZV=6.0,
58      XVF=28.0, ZVF=7.4,
59      SCALE=1.0, VERTUP=.TRUE.$
60
61
62 ****
63 * Body Configuration Parameters * pg 36
64 ****
65
66 $BODY NX=8.0,
67      X(1)=0.0,1.0,2.7,6.0,8.8,28.5,39.4,44.8,
68      R(1)=0.0,1.25,2.1,2.7,2.76,2.7,1.25,0.0,

```

```

69      ZU(1)=3.5,4.3,4.8,5.5,7.4,7.4,6.5,5.7,
70      ZL(1)=3.5,2.5,2.25,2.1,2.0,2.2,4.3,5.7,
71      BNOSE=1.0, BLN=8.8,
72      BTAIL=1.0, BLA=19.7,
73      ITYPE=1.0, METHOD=1.0$  

74  

75  

76  

77 ****  

78 *          Wing planform variables    pg 37-38  

79 ****  

80  

81  

82 $WGPNF  CHRDR=9.4,     CHRDTP=3.01,  

83      SSPN=25.85,   SSPNE=23.46,  

84      SAVSI=1.3,  

85      CHSTAT=0.25,  TWISTA=-3.0,  

86      DHDADI=3.6,  

87      TYPE=1.0$  

88  

89  

90 ****  

91 * Wing Sectional Characteristics Parameters * pg 39-40  

92 ****  

93  

94 NACA W 5 23014  

95  

96 $JETPWR  NENG SJ=2.0,   AIET LJ=2.0,      THST CJ=0.0,  

97      JIALOC=25.8,   JELLOC=4.33,   JEVLOC=5.625,  

98      JEALOC=33.3,   JINLTA=2.243,  

99      AMBTMP=59.7,   AMBstp=2116.8,  JERAD=0.755$  

100  

101 SAVE  

102  

103  

104 ****  

105 *          Symmetrical Flap Deflection parameters  

106 ****  

107  

108  

109 $SYMFLP  FTYP E=2.0,     NDELTA=9.0,  

110      DELTA(1)=0.0,5.0,10.0,15.0,20.0,25.0,30.0,35.0,40.0,  

111      PHETE=0.0522,  PHETEP=0.0391,  

112      CHRDFI=2.0,   CHRDFO=1.6,  

113      SPANFI=5.78,   SPANFO=15.3,  

114      CPRMEI(1)=8.1,8.1,8.2,8.2,8.3,8.3,8.3,8.4,8.4,  

115      CPRMEO(1)=3.7,3.7,3.8,3.8,3.9,3.9,3.9,4.0,4.0,  

116      NTYP E=1.0$  

117  

118 * At this point, we are going to terminate the case so that we can get  

119 * the flap effects. We can't save this data, as we are  

120 * also going to do control surfaces on the horizontal tail.  

121  

122 CASEID FLAPS: Citation II Model 550 Aircraft  

123 NEXT CASE
124

```

```

125
126
127
128 ****
129 *      Asymmetrical Control Deflection parameters : Ailerons
130 ****
131
132 *          slot-reflector, fraction of chord
133
134 $ASYFLP STYPE=4.0, NDELT=9.0,
135     DELTAL(1)=-32.0,-20.0,-10.0,-5.0,0.0,5.0,10.0,20.0,32.0,
136     DELTAR(1)=32.0,20.0,10.0,5.0,0.0,-5.0,-10.0,-20.0,-32.0,
137     SPANFI=15.2, SPANFO=24.0,
138     PHETE=0.05228,
139     CHRDFI=1.87, CHRDFO=1.2$  

140
141
142
143 *      Terminates the reading of input cards and begins execution of
144 *      the case. Case data are destroyed following execution of a case,
145 *      unless the SAVE card is present.
146 CASEID AILERONS: Citation II Model 550 Aircraft
147 SAVE
148 NEXT CASE
149
150
151
152
153 ****
154 *          Horizontal Tail Sectional Characteristics pg 39-40
155 ****
156 * Same build up as wing, if you'd like to use that instead.
157
158 NACA H 4 0010 ! Citation is 0010 at root, 0008 at tip
159
160
161 $HTPLNF CHRDR=4.99, CHRDTP=2.48,
162     SSPN=9.42, SSPNE=9.21,
163     SAVSI=5.32,
164     CHSTAT=0.25, TWISTA=0.0,
165     DHDADI=9.2,
166     TYPE=1.0$  

167
168
169
170 $VTPLNF CHRDTP=3.63, SSPNE=8.85, SSPN=9.42, CHRDR=8.3,
171     SAVSI=32.3, CHSTAT=0.25, TYPE=1.0$  

172
173
174 $VFPLNF CHRDR=11.8, CHRDTP=0.0, CHSTAT=0.0, DHDADO=0.0,
175     SAVSI=80.0, SSPN=2.3, SSPNE=2.1, TYPE=1.0$  

176
177
178
179
180

```

```

181 ****
182 *      Elevator Deflection parameters
183 ****
184
185
186 $SYMFLP FTYPE=1.0 ,
187     NDELTA=9.0,  DELTA(1)
188     =-20.0,-15.0,-10.0,-5.0,0.0,5.0,10.0,13.0,16.0,
189     PHETE=0.0522,  PHETEP=0.0523,
190     CHRDFI=1.94,  CHRDFO=1.03,
191     SPANFI=0.7,  SPANFO=9.21,
192     CB=0.84,          TC=0.3,          NTYPE=1.0$
193
194
195 ****
196 *      Vertical Tail Sectional Characteristics pg 39-40
197 ****
198 * Same build up as wing, if you'd like to use that instead.
199
200 NACA V 4 0012 ! Citation is 0012 at root, 0008 at tip
201
202
203
204 ****
205 *      Ventral Fin Sectional Characteristics pg 39-40
206 ****
207 * Same build up as wing, if you'd like to use that instead.
208
209 NACA F 4 0012 ! Guess it to be the same as vertical tail for Citation.
210
211
212
213 ****
214 *      Jet Power Effects parameters pg 51
215 ****
216
217
218
219
220 CASEID TOTAL: Citation II Model 550 Aircraft

```

6.0.2 Risultati

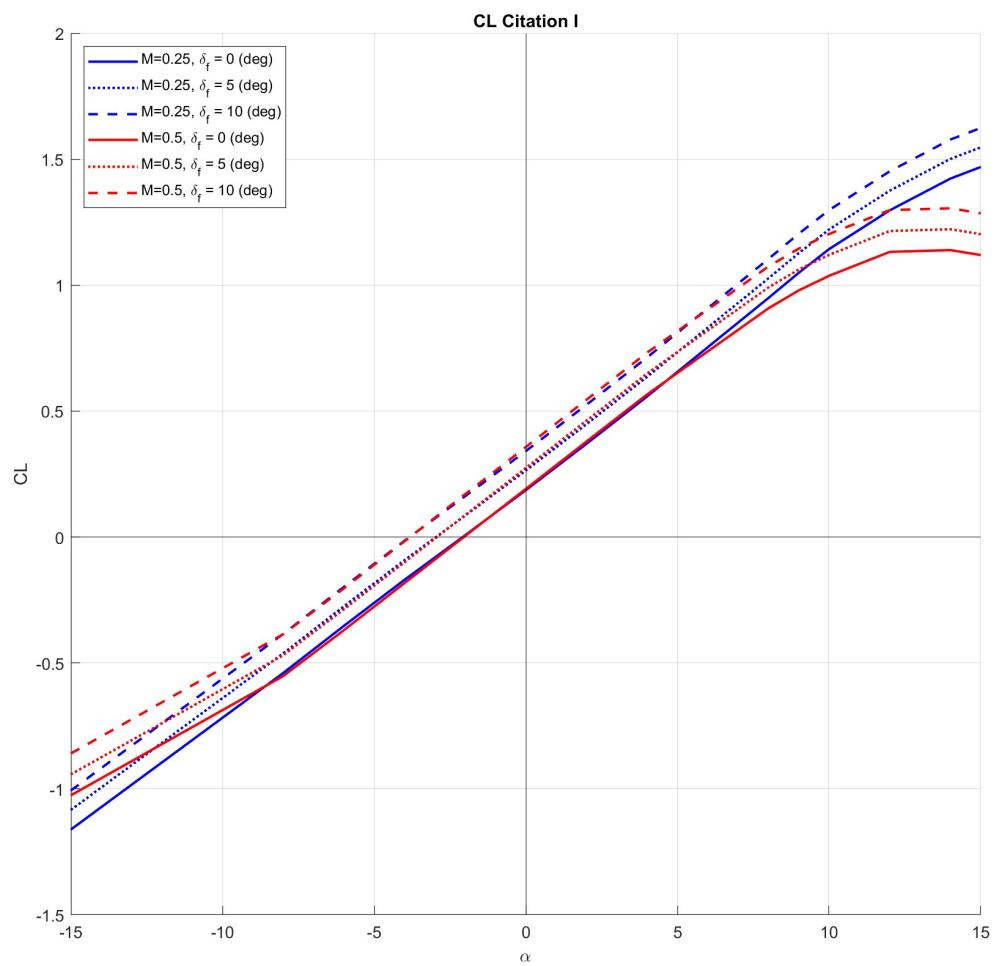


Figura 6.1 Curva del CL

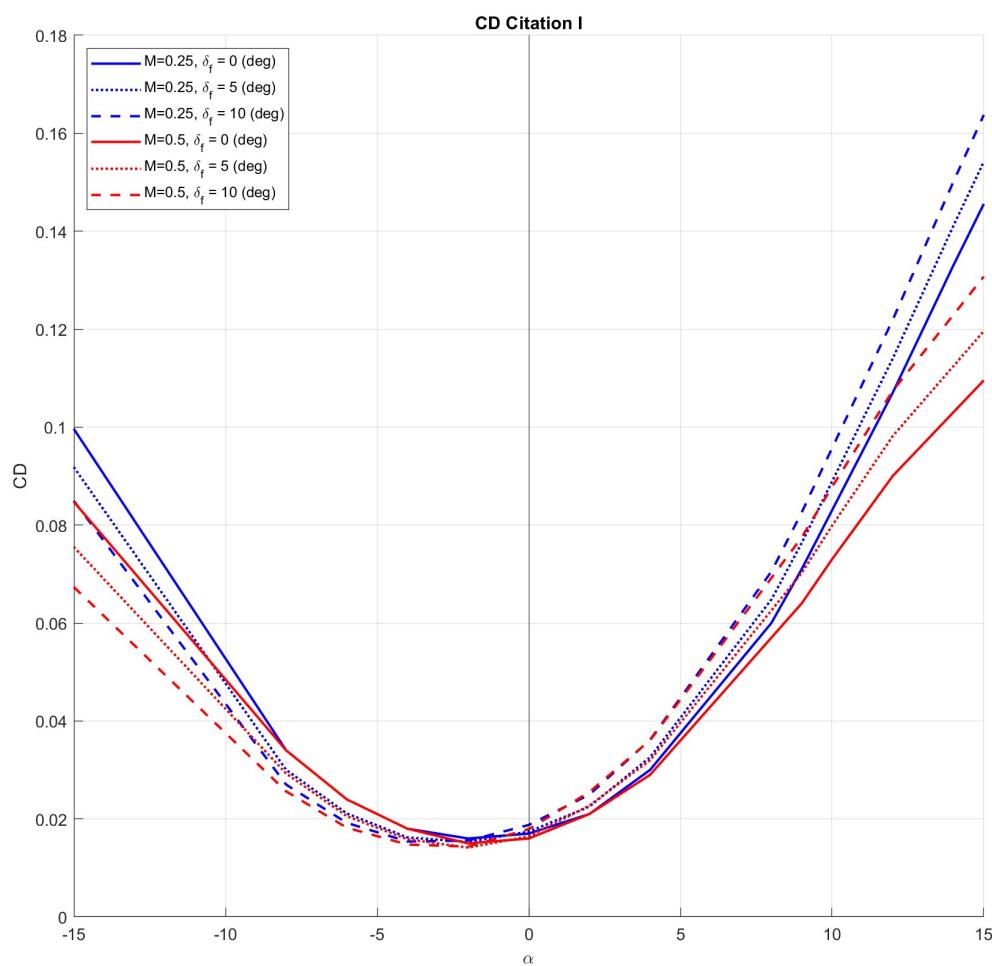


Figura 6.2 Curva del CD

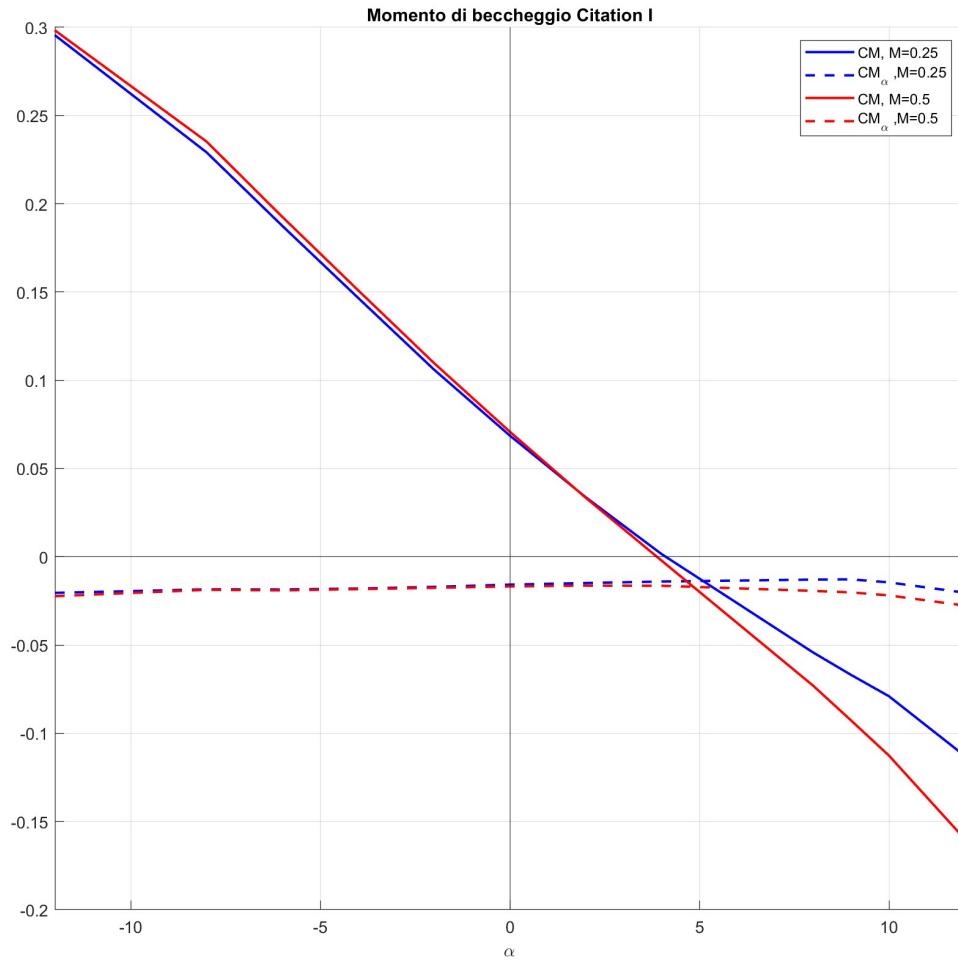


Figura 6.3 Curva del momento di beccheggio al variare di α_B

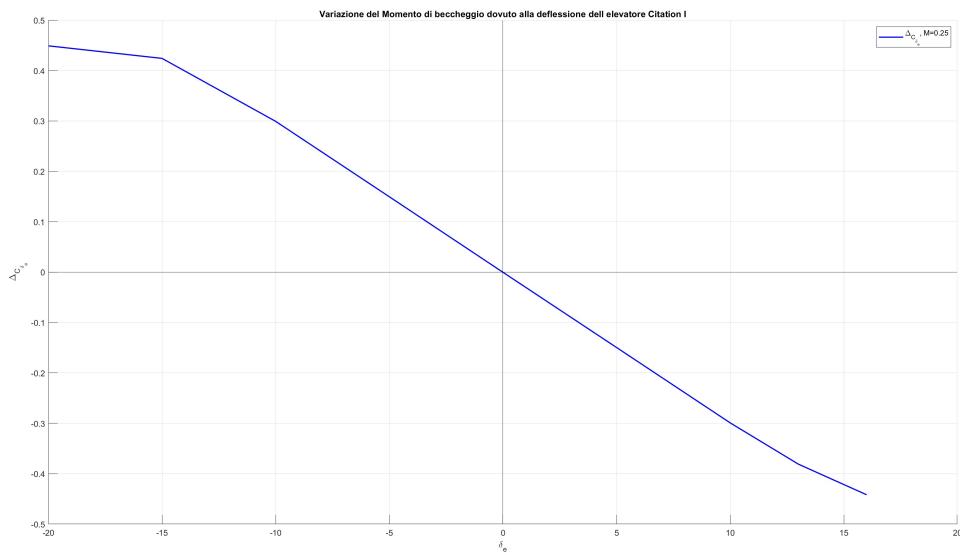


Figura 6.4 Variazione del momento di beccheggio dovuta all'elevatore

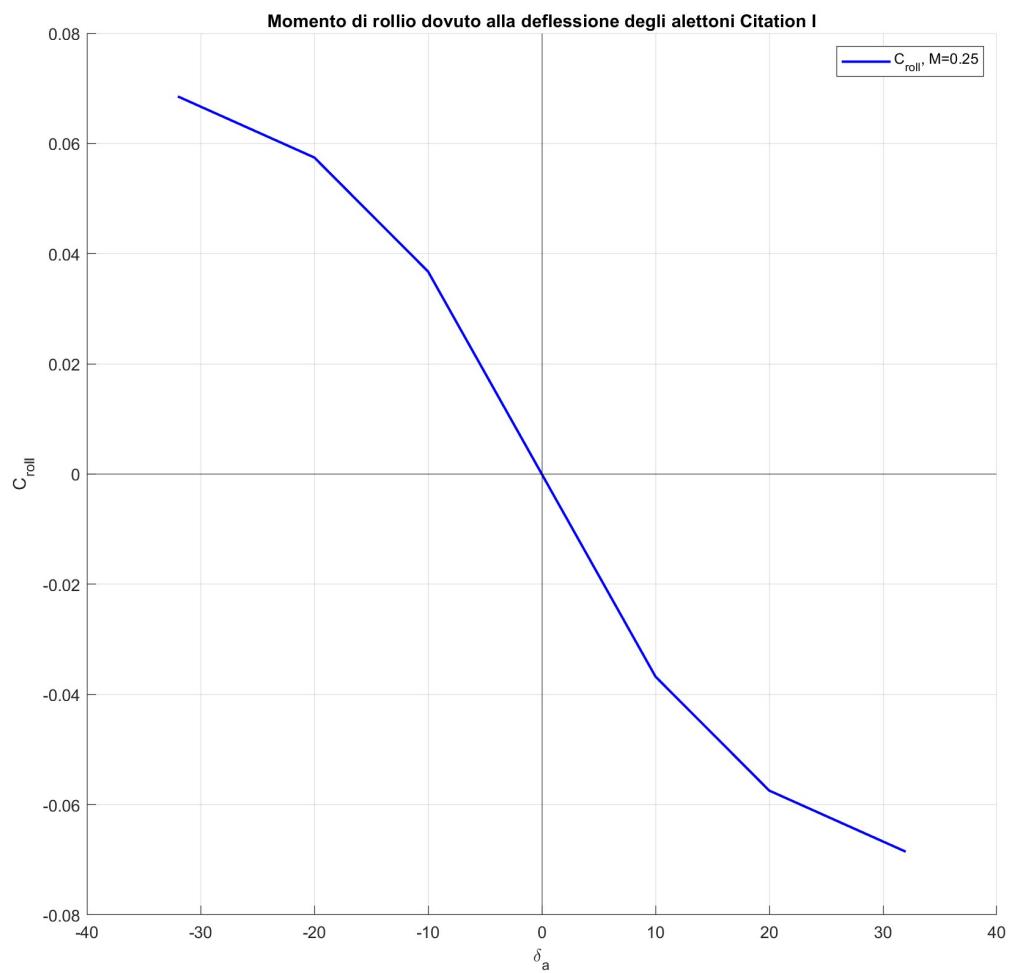


Figura 6.5 Momento di rollio dovuto alla deflessione degli alettoni

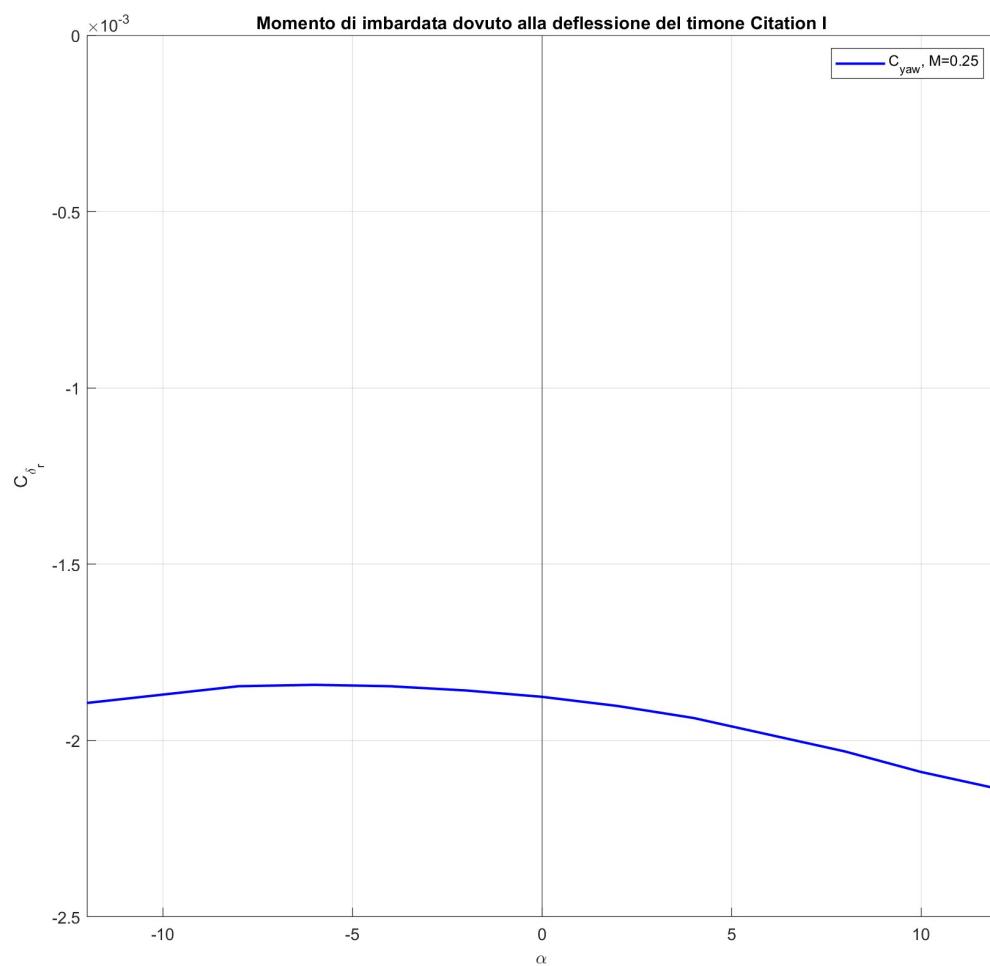


Figura 6.6 Momento di imbardata dovuto alla deflessione del timone

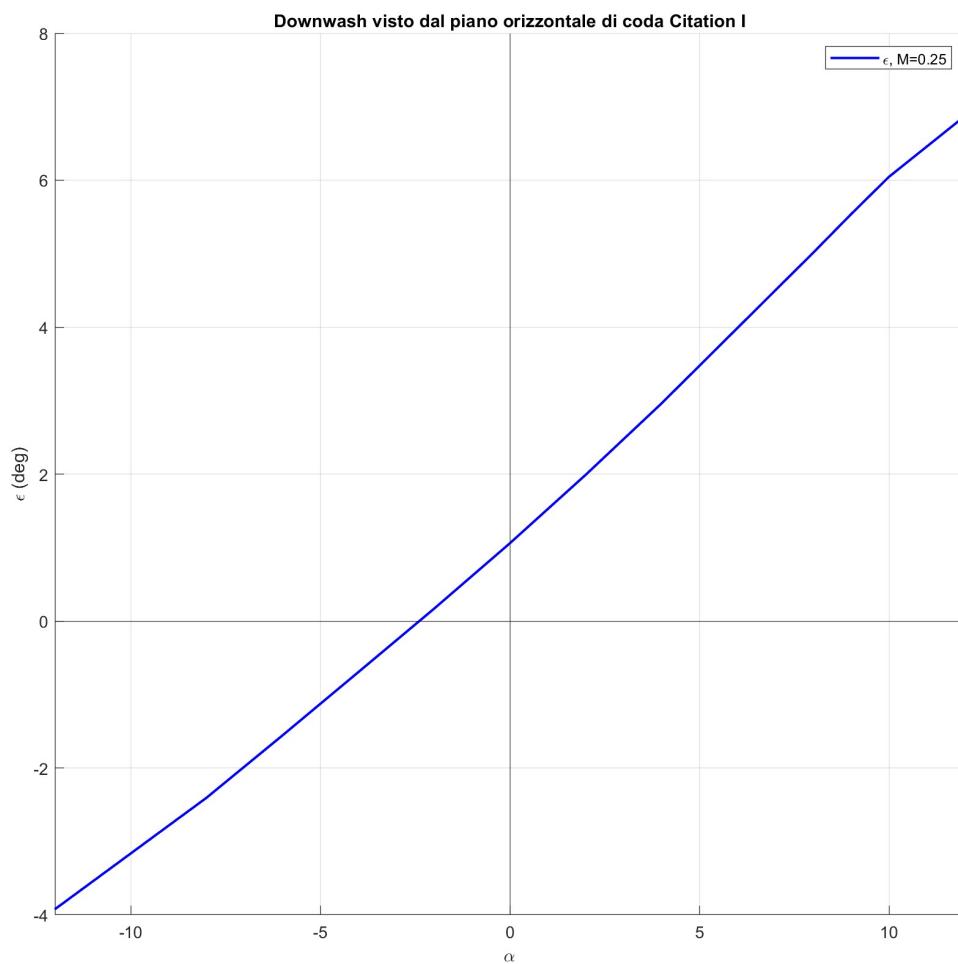


Figura 6.7 Downwash visto dal piano di coda