

Giuseppe Alferi

Esercitazione

di

Dinamica e simulazione di volo

Esercitazioni proposte per i quaderni

Aprile 2017

ver. 2017.a

Dichiarazione di Copyright

Copyright Giuseppe Alferi,
Università degli Studi di Napoli "Federico II".

(Legge italiana sul Copyright 22.04.1941 n. 633)

Indice

2	Esercitazione quaderno	15
2.1	Introduzione	15
2.2	Rappresentazione 3-D di un velivolo e della sua traiettoria - <i>Esercizio 2-1</i>	16
2.3	Componenti della forza peso in assi <i>Body</i> - <i>Esercizio 2-2</i>	17
2.4	Orientamento successivi del velivolo <i>AirBus A-320</i> per una manovra di virata equilibrata - <i>Esercizio 2.4</i>	18
2.5	Integrazione dell'equazioni della cinematica - <i>Esercizio 2-5</i>	18
2.6	Listati utilizzati	23
2.6.1	Listato 2.1	23
2.6.2	Listato 2.2	23
2.6.3	Listato 2.3	27
3	Quaternione dell'orientamento di un velivolo	31
3.1	Introduzione	31
3.2	Impostazione del problema	32
3.3	Il codice di calcolo	33
3.4	Manovra di Looping	38
3.4.1	Looping perfetto - <i>Esercizio 3.1</i>	39
3.4.2	Looping realistico - <i>Esercizio 3.1</i>	43
3.5	Manovra del <i>Tonneau</i>	46
3.5.1	Un primo esempio di <i>tonneau</i>	46
3.5.2	Un secondo esempio di <i>tonneau</i> - <i>Esercizio 3.2</i>	49
3.6	Manovra di <i>Eight</i> - <i>Esercizio 3.4</i>	52
3.7	Manovra di <i>Spirale</i> - <i>Esercizio 3.4</i>	55
3.8	Manovra di <i>Looping</i> eseguita in Simulink - <i>Esercizio 3.5</i>	58
3.9	Acquisizioni leggi temporali mediante sensori dello smartphone	65
7	Moto longitudinal-simmetrico a comandi bloccati	71
7.1	Introduzione	71
7.2	Impostazione del problema	73
7.3	Il modello aerodinamico e propulsivo	74
7.4	Ricerca delle condizioni di trim	75
7.5	Ricerca delle condizioni di trim - 3-Dof	78
7.5.1	Codice per la ricerca delle condizioni di trim - <i>Esercizio 7.6</i>	78
7.5.2	Nuova condizione di Trim per $\delta_s = -1deg$ - <i>Esercizio 7.7</i>	83

7.5.3	Verifica delle condizioni di trim - <i>Esercizio 7.8</i>	86
7.6	Simulazione legge <i>cabra-picchia</i> - <i>Esercizio 7.9</i>	94
7.7	Variazione della legge del δ_e - <i>Esercizio 7.10</i>	101
9	Modello aerodinamico di un velivolo completo con Digita DATCOM	107
9.1	Introduzione	107
9.2	Uso del programma	108
9.3	Formato del file di input	108
9.4	Un esempio di file di input	111
9.5	Caratteristiche aerodinamiche con Digital DATCOM - <i>Esercizio 9-1</i>	113
9.6	Elevatore e flap con <i>Digital DATCOM+</i> - <i>Esercizio 9-2</i>	121
9.7	Aerodinamica del velivolo <i>Tecnam P2006T</i> - <i>Esercizio 9.3</i>	126
9.8	Caratteristiche aerodinamiche del velivolo <i>ATR72</i> - <i>Esercizio 9.4</i>	139
11	Moto Longitudinal Simmetrico a Comandi Liberi	149
11.1	Introduzione	149
11.2	Impostazione del problema	150
11.3	Soluzioni moto a comandi liberi - <i>Esercizio 11-1</i>	155
11.3.1	Listato integrale per la risoluzione dell'esercizio 11-1	159
11.4	Angolo di flottaggio dell'equilibratore <i>Esercizio 11-2</i>	162
11.4.1	Listato integrale per la risoluzione dell'esercizio 11-2	168
11.5	Effetto del <i>trim tab</i> - <i>Esercizio 11-3</i>	172
11.6	Manovra <i>cabra-picchia</i> ed evoluzione a comandi liberi - <i>Esercizio 11-4</i>	176
16	Analisi delle piccole perturbazioni del volo longitudinal simmetrico	183
16.1	Introduzione	183
16.2	Impostazione del problema	184
16.3	Dinamica Longitudinale	184
16.3.1	Caratteristiche longitudinali di un Boeing 747 in Matlab - <i>Esercizio 16.1</i>	185
16.3.2	Caratteristiche longitudinali di un Boeing 747 - <i>Esercizio 16.2</i>	194
17	Evoluzione in un piano orizzontale	197
17.1	Introduzione	197
17.2	Virata corretta	198
17.2.1	Virata corretta per assegnata legge $C_L(t)$ - <i>Esercizi da 17.1 a 17.3</i>	198
18	Carichi sul piano orizzontale di coda	209
18.1	Introduzione	209
18.2	Dati del velivolo	210
18.3	Realizzazione del Diagramma di manovra	211
18.4	Carico aerodinamico di equilibrio	215
19	Data logging delle variabili stato	227
19.1	Introduzione	227
19.2	Decollo	229
19.3	Crociera 1	231
19.4	Crociera 2	233

19.5 Atterraggio 235

Elenco delle figure

2.1	Modello 3-D del velivolo <i>AirBus A-320</i> caricato in <i>Matlab</i>	16
2.2	Modello tridimensionale <i>AirBus A-320</i> posizionato ed orientato negli assi Terra.	17
2.3	Modello tridimensionale <i>AirBus A-320</i> in evidenza il vettore le componenti del peso.	18
2.4	Evoluzione di virata a sinistra, Esercizio 2.4, del velivolo <i>AirBus A-320</i> . . .	19
2.5	Componenti $p(t)$, $q(t)$ ed $r(t)$ della velocità angolare.	19
2.6	Componente u della velocità del baricentro.	20
2.7	Virata rapida a destra, risultato dell'integrazione dell'equazioni cinematiche dell' <i>Esercizio 2.5</i>	21
2.8	<i>Time history</i> degli angoli di Eulero.	22
2.9	<i>Time history</i> delle coordinate del baricentro in assi <i>Earth</i>	22
3.1	<i>Time history</i> dei parametri per la manovra di <i>Looping perfetto</i> per le leggi assegnate nel listato 3.3 a pagina 39	41
3.2	Rappresentazione della manovra di <i>Looping perfetto</i> per un orientamento iniziale definito dalle condizioni $\psi = \theta = \phi = 0$, e per le leggi di variazione delle componenti della velocità lineare e della velocità angolare riportate nel listato 3.3 a pagina 39.	42
3.3	<i>Time history</i> dei parametri per la manovra di <i>Looping</i> realistico e per le leggi assegnate nel listato 3.4 a pagina 43.	44
3.4	Rappresentazione della manovra di <i>looping</i> realistico per un orientamento iniziale definito dalle condizioni $\psi = \theta = \phi = 0$, e per le leggi di variazione delle componenti della velocità lineare e della velocità angolare rappresentate nel listato 3.4 a pagina 43.	45
3.5	<i>Time history</i> dei parametri per la manovra di <i>Tonneau</i> , per le leggi temporali assegnate nel listato 3.5 a pagina 46.	47
3.6	Rappresentazione della manovra di <i>tonneau</i> per un orientamento iniziale definito dalle condizioni $\psi = \theta = \phi = 0$ e per le leggi di variazione delle componenti della velocità lineare e della velocità angolare rappresentate nel listato 3.5 a pagina 46.	48
3.7	<i>Time history</i> dei paramentri per la manovra di <i>Tonneau</i> , per le leggi temporali assegnate nel listato 3.6 a pagina 49.	50

3.8	Rappresentazione della manovra di <i>tonneau</i> per un orientamento iniziale definito dalle condizioni $\psi = \theta = \phi = 0$ e per le leggi di variazione delle componenti della velocità lineare e della velocità angolare rappresentate nel listato 3.6 a pagina 49. Il velivolo non è in scala.	51
3.9	Time history dei parametri per la manovra di <i>Eight</i> , per le leggi temporali assegnate nel listato 3.7 a pagina 52 $\psi = \theta = \phi = 0$	53
3.10	Rappresentazione della manovra di <i>Eight</i> per un orientamento iniziale definito dalle condizioni $\psi = \theta = \phi = 0$, e per le leggi di variazione delle componenti della velocità lineare e della velocità angolare rappresentate nel listato 3.7 a pagina 52. Il velivolo non è in scala.	54
3.11	Storie temporali delle componenti del quaternion e degli angoli di Eulero corrispondenti alla manovra di <i>Eight</i> riportata nel listato 3.7 a pagina 52, per un orientamento iniziale definito dalle condizioni $\psi = \theta = \phi = 0$, Il velivolo non è in scala.	56
3.12	Rappresentazione della manovra di <i>Spirale</i> per un orientamento iniziale definito dalle condizioni $\psi = \theta = \phi = 0$, e per le leggi di variazione delle componenti della velocità lineare e della velocità angolare rappresentate nella figura 3.11 a pagina 56, Il velivolo non è in scala.	57
3.13	Aspetto del diagramma a blocchi del modello <i>Simulink</i> considerato.	59
3.14	Evoluzione di <i>Looping</i> ottenuta con il modello <i>Simulink</i>	60
3.15	Storie temporali effettive delle componenti di velocità angolari in assi vento, Blocco <i>Scope</i> , per la manovra di <i>Looping</i>	61
3.16	Storie temporali delle componenti del quaternion e degli angoli di Eulero, Blocco <i>Scope</i> , per la manovra di <i>Looping</i>	62
3.17	Storie temporali degli angoli d'Eulero, Blocco <i>Scope</i> , per la manovra di <i>Looping</i>	63
3.18	Storie temporali delle componenti delle coordinate del baricentro, Blocco <i>Scope</i> , per la manovra di <i>Looping</i>	64
3.19	Time histories acquisite mediante smartphone.	66
3.20	File.csv generato dallo smartphone	67
3.21	Time histories acquisite mediante smartphone fase costante	68
3.22	Traiettoria riportata mediante le coordinate di latitudine e longitudine	69
7.1	Volo equilibrato lungo una traiettoria inclinata di γ sull'orizzontale.	73
7.2	Valori di equilibrio al variare della velocità di volo iniziale, per $\delta_{s,0} = -1$ deg. I dati del velivolo sono quelli riportati nel listato 7.3 a pagina 81.	85
7.3	Storie temporali dei comandi invariate, a partire dalla condizione di trim riportata nel listato. I dati del velivolo sono quelli riportati nel listato 7.3 a pagina 81.	90
7.4	Storie temporali delle variabili di stato per leggi di comando identiche, a partire dalla condizione di trim riportata nel listato. Il velivolo è sempre lo stesso assegnato nel listato 7.3 a pagina 81. I $\Delta(*)$ rappresentati sono riferiti rispetto ai valori iniziali. Il baricentro è posto al 30% della corda.	91
7.5	Variazione di quota del baricentro, a partire dalla condizione di trim riportate nel listato	91
7.6	Leggi temporali dei comandi invariati, a partire dalla condizione di trim riportata nel listato	92

7.7	Storie temporali delle variabili di stato per leggi di comando identiche, a partire dalla condizione di trim riportata nel listato. Il velivolo è sempre lo stesso assegnato nel listato 7.3 a pagina 81. I $\Delta(*)$ rappresentati sono riferiti rispetto ai valori iniziali. Il baricentro è posto al 50 % della corda.	93
7.8	Storie temporali delle leggi di comando. La manovra <i>cabra-picchia</i> viene realizzata muovendo l'equilibratore δ_e come riportato. Le leggi di comando dello stabilizzatore e della spinta restano costanti a partire dalla condizione di trim riportata nel listato 7.9. Condizioni valutate per il velivolo assegnato nel listato 7.3 a pagina 81.	97
7.9	Storie temporali delle variabili di stato per la legge di comando "cabra-picchia" rappresentata in figura 7.8 a pagina 97, a partire dalla condizione di trim proposta nel listato 7.9 a pagina 94.	98
7.10	Storie temporali delle variabili di stato per la legge di comando "cabra-picchia" rappresentata in figura 7.8 a pagina 97, a partire dalla condizione di trim proposta nel listato 7.9 a pagina 94.	99
7.11	Andamento del fattore di carico per la legge "cabra-picchia" nel breve periodo, a partire dalla condizione di trim riportata nel listato e per il velivolo assegnato precedentemente.	100
7.12	Andamento dell'accelerazione angolare di beccheggio \dot{q} nei primi 10 secondi della manovra, a partire dalla condizione di trim riportata nel listato e per il velivolo assegnato precedentemente.	100
7.13	Confronto tra le due diverse leggi di comando per la deflessione dell'equilibratore δ_e	101
7.14	Confronto tra le storie temporali delle variabili di stato per due leggi di comando "cabra-picchia" rappresentate nella figura 7.13 a pagina 101, a partire dalla condizione di trim riportata nel listato 7.9 a pagina 94. Il velivolo è sempre lo stesso assegnato nel listato 7.3 a pagina 81. Sono rappresentate le due diverse leggi di comando nel Breve Periodo e le variazioni sono rispetto ai valori iniziali di trim 7.9 a pagina 94.	103
7.15	Confronto tra le storie temporali delle variabili di stato per due leggi di comando "cabra-picchia" rappresentate nella figura 7.13 a pagina 101, a partire dalla stessa condizione di trim riportata nel listato 7.9 a pagina 94. Il velivolo è quello assegnato nel listato 7.3 a pagina 81. Sono rappresentate le due diverse leggi di comando nel Lungo Periodo e le variazioni sono rispetto ai valori iniziali di trim 7.9 a pagina 94.	104
7.16	Due leggi di comando "cabra-picchia" a confronto. Si riportano nella stessa scala temporale le storie dei fattori di carico e dell'accelerazione angolare corrispondenti, a partire dalla condizione di trim riportata nel listato. Il velivolo è quello assegnato nel paragrafo precedente.	105
9.1	Viste Cessna Citation	114
9.2	Velivolo Cessna Citation II 500. Curve del coefficiente di portanza C_L , del coefficiente di resistenza C_D e della polare per le configurazioni <i>Wing</i> , <i>Wing-Body</i> , <i>Wing-Body-Vertical Tail</i> e velivolo completo.	115

9.3	Velivolo Cessna Citation II 500. Curve del coefficiente di momento C_M e dei coefficienti di forza assiale e normale C_A e C_N al variare dell'angolo d'attacco per le configurazioni <i>Wing</i> , <i>Wing-Body</i> , <i>Wing-Body-Vertical Tail</i> e velivolo completo.	116
9.4	Velivolo Cessna Citation II 500. Variazione delle derivate dei coefficiente aerodinamici di momento C_{La} , C_{Ma} e C_{LB} al variare dell'angolo d'attacco per le configurazioni <i>Wing</i> , <i>Wing-Body</i> , <i>Wing-Body-Vertical Tail</i> e velivolo completo.	117
9.5	Velivolo Cessna Citation II 500. Variazione delle derivate dei coefficienti aerodinamici di momento C_{Lp} , C_{Yp} e C_{Np} al variare dell'angolo d'attacco per le configurazioni <i>Wing</i> , <i>Wing-Body</i> , <i>Wing-Body-Vertical Tail</i> e velivolo completo.	118
9.6	Velivolo Cessna Citation II 500. Variazione delle derivate dei coefficienti aerodinamici di momento C_{Lr} e C_{Nr} al variare dell'angolo d'attacco per le configurazioni <i>Wing</i> , <i>Wing-Body</i> , <i>Wing-Body-Vertical Tail</i> e velivolo completo.	119
9.7	Velivolo Cessna Citation II 500. Variazione dell'angolo di downwash mesio ϵ_H , della sua derivata $(\frac{d\epsilon}{d\alpha})$ in funzione dell'angolo d'attacco per la configurazione del velivolo completo.	120
9.8	Andamento dei coefficienti C_L , C_D e C_M espressi per $\delta_e = 0$ deg e $\delta_f \in [0, 40]$	124
9.9	Andamento dei coefficienti C_L , C_D e C_M espressi per $\delta_f = 15$ deg e $\delta_e \in [-20, 16]$	125
9.10	Velivolo Tecnam P2006	126
9.11	Profilo della fusoliera e ingombri del velivolo Tecnam P2006T.	129
9.12	Vista laterale del velivolo Tecnam P2006T. Sono riportate le corde delle sezioni di radice dell'ala e del piano orizzontale di coda.	130
9.13	Vista Alto TECNAM P2006T.	130
9.14	Velivolo Tecnam P2006T. Curve del coefficiente di portanza C_L e del coefficiente di resistenza C_D al variare dell'angolo d'attacco e polare per le configurazioni <i>Wing</i> , <i>Wing-Body</i> , <i>Wing-Body-Vertical Tail</i> e velivolo completo.	131
9.15	Velivolo Tecnam P2006T. Curve del coefficiente di momento C_M e del coefficienti di forza assiale e normale C_A e C_N al variare dell'angolo d'attacco per le configurazioni <i>Wing</i> , <i>Wing-Body</i> , <i>Wing-Body-Vertical Tail</i> e velivolo completo.	132
9.16	Velivolo Tecnam P2006T. Curve del coefficiente di momento C_M e del coefficienti di forza assiale e normale C_A e C_N al variare dell'angolo d'attacco per le configurazioni <i>Wing</i> , <i>Wing-Body</i> , <i>Wing-Body-Vertical Tail</i> e velivolo completo.	133
9.17	Velivolo Tecnam P2006T. Variazione delle derivate dei coefficienti aerodinamici di momento C_{Lp} , C_{Yp} e C_{Np} in funzione dell'angolo d'attacco per le configurazioni <i>Wing</i> , <i>Wing-Body</i> , <i>Wing-Body-Vertical Tail</i> e velivolo completo.	134

9.18	Velivolo Tecnam P2006T. Variazione delle derivate dei coefficienti aerodinamici di momento C_{Lr} e C_{Nr} in funzione dell'angolo d'attacco per le configurazioni <i>Wing</i> , <i>Wing-Body</i> , <i>Wing-Body-Vertical Tail</i> e velivolo completo.	135
9.20	Velivolo Tecnam P2006T. Curve del coefficiente di portanza C_L , del coefficiente di resistenza C_D e del coefficiente di momento di beccheggio C_M al variare dell'angolo di deflessione dei flap per fissato valore dell'angolo di deflessione dell'elevatore parti a $\delta_e=0$ nel caso della configurazione del velivolo completo.	136
9.21	Velivolo Tecnam P2006T. Curve del coefficiente di portanza C_L , del coefficiente di resistenza C_D e del coefficiente di momento di beccheggio C_M al variare dell'angolo di deflessione dell'elevatore per fissato angolo di deflessione dei flap pari a $\delta_e=15$ deg nel caso della configurazione del velivolo completo.	137
9.19	Velivolo Tecnam P2006T. Variazione dell'angolo di downwash mesio ϵ_H , della sua derivata ($\frac{d\epsilon}{d\alpha}$) in funzione dell'angolo d'attacco per la configurazione del velivolo completo.	138
9.22	Velivolo ATR72 proposto nella rappresentazione Digital DATCOM	139
9.23	Velivolo ATR72. Curve del coefficiente di portanza C_L e del coefficiente di resistenza C_D al variare dell'angolo d'attacco e polare per le configurazioni <i>Wing</i> , <i>Wing-Body</i> , <i>Wing-Body-Vertical Tail</i> e velivolo completo. . .	140
9.24	Velivolo ATR72. Curve del coefficiente di momento C_M e del coefficienti di forza assiale e normale C_A e C_N al variare dell'angolo d'attacco per le configurazioni <i>Wing</i> , <i>Wing-Body</i> , <i>Wing-Body-Vertical Tail</i> e velivolo completo.	141
9.25	Velivolo ATR72. Curve del coefficiente di momento C_M e del coefficienti di forza assiale e normale C_A e C_N al variare dell'angolo d'attacco per le configurazioni <i>Wing</i> , <i>Wing-Body</i> , <i>Wing-Body-Vertical Tail</i> e velivolo completo.	142
9.26	Velivolo ATR72. Variazione delle derivate dei coefficienti aerodinamici di momento C_{Lp} , C_{Yp} e C_{Np} in funzione dell'angolo d'attacco per le configurazioni <i>Wing</i> , <i>Wing-Body</i> , <i>Wing-Body-Vertical Tail</i> e velivolo completo.	143
9.27	Velivolo ATR72. Variazione delle derivate dei coefficienti aerodinamici di momento C_{Lr} e C_{Nr} in funzione dell'angolo d'attacco per le configurazioni <i>Wing</i> , <i>Wing-Body</i> , <i>Wing-Body-Vertical Tail</i> e velivolo completo. . .	144
9.28	Velivolo ATR72. Variazione dell'angolo di downwash mesio ϵ_H , della sua derivata ($\frac{d\epsilon}{d\alpha}$) in funzione dell'angolo d'attacco per la configurazione del velivolo completo.	145
9.29	Velivolo ATR72. Curve del coefficiente di portanza C_L , del coefficiente di resistenza C_D e del coefficiente di momento di beccheggio C_M al variare dell'angolo di deflessione dei flap per fissato valore dell'angolo di deflessione dell'elevatore parti a $\delta_e=0$ nel caso della configurazione del velivolo completo.	146

9.30	Velivolo ATR72. Curve del coefficiente di portanza C_L , del coefficiente di resistenza C_D e del coefficiente di momento di beccheggio C_M al variare dell'angolo di deflessione dell'elevatore per fissato angolo di deflessione dei flap pari a $\delta_e=15$ deg nel caso della configurazione del velivolo completo.	147
11.1	Storie temporali delle variabili di stato per comandi bloccati fino all'istante $t_1 = 1s$ e comandi liberi nei successivi istanti di tempo. Le grandezze $\Delta(*)$ rappresentano gli scostamenti rispetto ai valori iniziali.	166
11.2	Storia temporale dell'angolo di deflessione dell'equilibratore, dei fattori di carico e dell'accelerazione angolare per comandi bloccati fino all'istante $t_1 = 1s$ e comandi liberi negli istanti di tempo successivi, a partire dalle condizioni di trime ricavate mediante la funzione <code>fmincon</code>	166
11.3	Storia temporale del momento di cerniera \mathcal{H}_e , per comandi bloccati fino all'istante $t_1 = 1s$ e comandi liberi nei successivi istanti di tempo, a partire dalle condizioni di trim ricavate mediante la funzione <code>fmincon</code> . . .	167
11.4	Storia temporale della variazione dell'angolo di deflessione dell'elevatore e della sua velocità, per comandi bloccati fino all'istante $t_1 = 1s$ e comandi liberi nei successivi istanti di tempo, a partire dalle condizioni di trim ricavate con la funzione <code>fmincon</code> e velivolo assegnato.	167
11.5	Storie temporale dell'angolo di deflessione dell'equilibratore, del δ_t e dell'accelerazione angolare, per comandi bloccati fino all'istante $t_1 = 1s$ e comandi liberi per i successivi istanti di tempo, a partire dalle condizioni di trim e per velivolo assegnato. Si nota come il δ_t che restituisce il valore della deflessione prossima a quella della condizione di trim è di ≈ 2.2 deg.	174
11.6	Storie temporali degli output dell'integrazione, effetto dell'aletta <i>trim tab</i> .	174
11.7	Storie temporali delle variabili di stato per la legge di deflessione del δ_t fornita al fine di ricercare la deflessione dell'equilibratore δ_e par alla deflessione fornita per le condizioni di trim $\delta_{e,0}$	175
11.8	Legge temporale impartita alla deflessione dell'equilibratore, aletta $\delta_t = 0$.	177
11.9	Storie temporali delle variabili di stato per comandi bloccati fino all'istante $t_1 = 1s$ e comandi liberi nei successivi istanti di tempo. Le grandezze $\Delta()$ rappresentano gli scostamenti rispetto ai valori iniziali, in particolare per il valore di δ_t che mi restituisce un valore asintotico del $\delta_e \approx \delta_{e,0}$	178
11.10	Grandezze d'interesse per l'assegnata legge cabra bicchia	179
11.11	Grandezze d'interesse per l'assegnata legge cabra bicchia, variazione del momento di Cerniera	180
11.12	Andamento Carichi	181
11.13	Variazione della quota a seguito della manovra cabra picchia	181
16.1	Risposta libera di un Boeing 747 a perturbazioni longitudinali della condizione iniziale. La risposta è stata ottenuta eccitando <i>solo</i> il modo di corto periodo.	192
16.2	Risposta libera di un Boeing 747 a perturbazioni longitudinali della condizione iniziale. La risposta è stata ottenuta eccitando <i>solo</i> il modo di lungo periodo.	193

16.3	Risposta libera di un Boeing 747 a perturbazioni longitudinali della condizione iniziale. La risposta è stata ottenuta eccitando <i>solo</i> il modo di corto periodo.	195
16.4	Risposta libera di un Boeing 747 a perturbazioni longitudinali della condizione iniziale. La risposta è stata ottenuta eccitando <i>solo</i> il modo di lungo periodo.	196
17.1	Legge di Sostentazione	199
17.2	Storie Temporali per la legge di sostentazione.	200
17.3	Storie Temporali per la legge di sostentazione.	201
17.4	Angoli Eulero per la manovra di virata	202
17.5	Angoli Eulero per la manovra di virata	203
18.1	Diagramma del coefficiente di momento del velivolo parziale rispetto al punto A_0 , bordo di attacco della corda media aerodinamica. Nel piano $(C_{M_{WB}}, C_L)$ si è tracciata l'isoclina passante per il punto a $C_L = 0$ per baricentro posto al 25% di \bar{c}	211
18.2	Diagramma di manovra del velivolo proposto per l'esercitazione	214
18.3	Determinazione grafica del coefficiente di momento del velivolo parziale $C_{M_{WB,CG}}$ rispetto al baricentro per un valore del coefficiente di portanza $C_L = 1.0$, nel caso di baricentro posto al 25% della corda media aerodinamica.	216
18.4	Curve del carico aerodinamico di equilibrio del piano di coda orizzontale ottenute al variare della velocità di volo per i punti sul bordo del diagramma di manovra. Il velivolo è un aereo da trasporto a breve raggio, i cui dati sono riportati in tabella 18.1 a pagina 210. Sono rappresentate le curve relative ai due diversi casi di baricentro posto al 25% e al 30% della corda media aerodinamica.	221
18.5	Curve del carico aerodinamico di equilibrio del piano di coda orizzontale ottenute al variare della velocità di volo equivalente per le condizioni di fattore di carico ± 1 . Il velivolo è un aereo da trasporto a breve raggio, i cui dati sono riportati in tabella 18.1 a pagina 210. Sono rappresentate le curve relative ai due diversi casi di baricentro posto al 25% e al 30% della corda media aerodinamica.	226
19.1	Posizionamento dello smartphone in fusoliera, in una zona a ridosso dell'ala	228
19.2	Time Histories delle variabili q_0, q_x, q_y e q_z	229
19.3	Time Histories delle variabili Yaw, Pitch, Roll e Speed.	229
19.4	Time Histories: Altitude.	230
19.5	Time Histories: Accelerazioni lungo x, y e z, m/s^2	230
19.6	Time Histories delle variabili q_0, q_x, q_y e q_z	231
19.7	Time Histories delle variabili Yaw, Pitch, Roll e Speed.	231
19.8	Time Histories: Altitude.	232
19.9	Time Histories: Accelerazioni lungo x, y e z, m/s^2	232
19.10	Time Histories delle variabili q_0, q_x, q_y e q_z	233
19.11	Time Histories delle variabili Yaw, Pitch, Roll e Speed.	233
19.12	Time Histories: Altitude.	234

19.13 Time Histories: Accelerazioni lungo x, y e z, m/s^2	234
19.14 Time Histories delle variabili q_0, q_x, q_y e q_z	235
19.15 Time Histories delle variabili Yaw, Pitch, Roll e Speed.	235
19.16 Time Histories: Altitude.	236
19.17 Time Histories: Accelerazioni lungo x, y e z, m/s^2	236
19.18 Punti in cui sono state fatte le acquisizioni dati.	237

Orientamento del velivolo e trasformazione di assi

Indice

2.1	Introduzione	15
2.2	Rappresentazione 3-D di un velivolo e della sua traiettoria - <i>Esercizio 2-1</i>	16
2.3	Componenti della forza peso in assi <i>Body</i> - <i>Esercizio 2-2</i>	17
2.4	Orientamento successivi del velivolo <i>AirBus A-320</i> per una manovra di virata equilibrata - <i>Esercizio 2.4</i>	18
2.5	Integrazione dell'equazioni della cinematica - <i>Esercizio 2-5</i>	18
2.6	Litati utilizzati	23
2.6.1	Listato 2.1	23
2.6.2	Listato 2.2	23
2.6.3	Listato 2.3	27

2.1 Introduzione

La funzione di questo quaderno è quella di preparare lo studente all'utilizzo di alcune funzioni implementate in Matlab che hanno l'obiettivo di orientare il posizionamento del velivolo in una terna di assi di riferimento. Le funzioni analizzate sono le seguenti:

- loadAircraftSTL.m
- loadAircraftMAT.m
- plotBodyE.m
- plotEarthAxes.m
- plotPoint3DHelperLines.m

- `plotTrajectoryAndBodyE.m`

Con l'uso di queste funzioni sono state rappresentate, in ambiente Matlab, le raffigurazioni del velivolo, degli assi *Body* (X_B, Y_B, Z_B), degli assi *NED* (X_E, Y_E, Z_E) e della traiettoria di una determinata manovra.

In fine, sono state integrate le *Gimbal Equations* attraverso la funzione `ode45` presente in MatLab.

2.2 Rappresentazione 3-D di un velivolo e della sua traiettoria - *Esercizio 2-1*

L'obiettivo di questa esercitazione è stato rappresentare il velivolo in 3-D. Si è scelto un *AirBus A-320*.

In particolare, si è considerato un modello geometrico d'estensione `.STL` (*Stereo Lithography interface format*), questo file contiene una o più tassellazioni superficiali costituita da elementi triangolari. Attraverso la funzione `loadAircraftSTL.m` si sono estrapolati tre vettori V, F e C che relativamente corrispondono al numero di vertici ed informazioni d'interconnessione dei vari triangoli che discretizzano il velivolo scelto.

Dopo la discretizzazione, si è salvato in un file `.mat` la geometria e attraverso la funzione `loadAircraftMAT.m` si è caricato il modello in Matlab in modo da evitare di utilizzare ogni volta la funzione utilizzata in precedenza. In fine si è ottenuto con un semplice `plot` in MatLab della Figura 2.1.

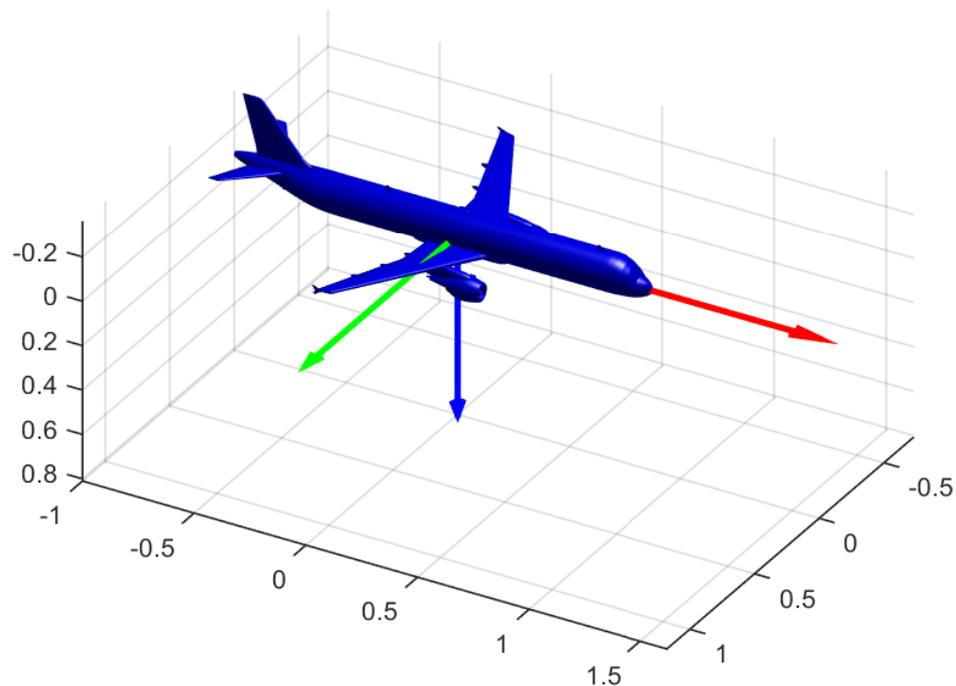


Figura 2.1 Modello 3-D del velivolo *AirBus A-320* caricato in *Matlab*.

Si è collocato il velivolo scelto nel sistema di riferimento Terra, con un possibile orientamento nel riferimento *Body* attraverso l'assegnazione degli angoli d'Eulero. Le funzio-

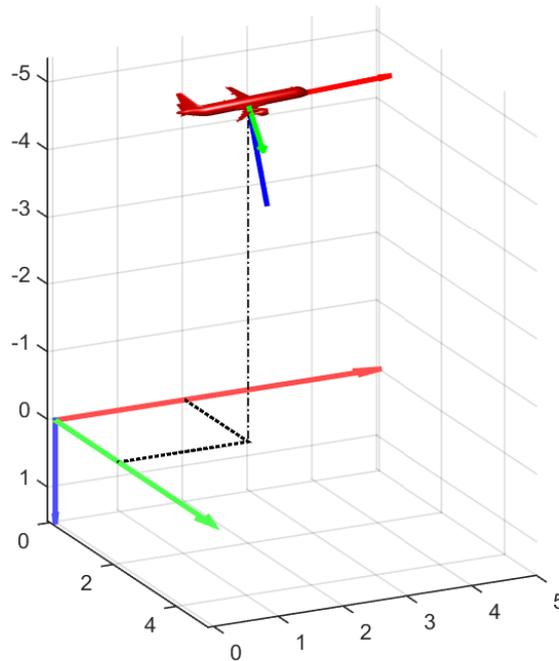


Figura 2.2 Modello tridimensionale *AirBus A-320* posizionato ed orientato negli assi Terra.

ni utilizzate sono `plotBodyE.m` e `plotEarthAxes.m`. In fine si è ottenuta la proiezione del baricentro sul piano X_E-Y_E mediante l'utilizzo della funzione `plotPoint3DHelperLines.m`, ottenendo così la Figura 2.2.

2.3 Componenti della forza peso in assi *Body* - Esercizio 2-2

Si sono calcolate le componenti di forza peso in assi velivolo del *AirBus A-320*. Considerando una massa di $m = 73900 \text{ Kg}$ ed un orientamento nello spazio secondo una terna $(\psi, \theta, \phi) = (20deg, 10deg, 0deg)$. Si osservi che, essendo ad ali livellate, la componente del peso $W_{y_b} = 0 \text{ N}$. Dal Listato1 di *MatLab*, allegato alla fine del capitolo, si ottengono le seguenti componenti del vettore W lungo gli assi *Body*, riportate di seguito.

$$W_{x_b} = -125\,890 \text{ N}, \quad W_{z_b} = 713\,950 \text{ N}$$

Tali grandezze sono state ottenute praticamente andando a costruire una matrice di trasformazione $[T_{BE}]$ ed il vettore $\{W_E\}$, dopo di che applicando la seguente relazione (1.2.1), si sono ottenuti le componenti del peso nel sistema di riferimento *Eart*. In particolare, la matrice di trasformazione si è ottenuta con la funzione `angle2dcm` implementata nel *Tool Box Aerospace* di Matlab (esplicitata negli appunti del docente Quaderino 2, Eq.2.29). In fine si è ottenuto la raffigurazione delle componenti peso per il *AirBus A-320*, riportate in Figura 2.3.

$$\{W_B\} = [T_{BE}]\{W_E\} \quad (2.1)$$

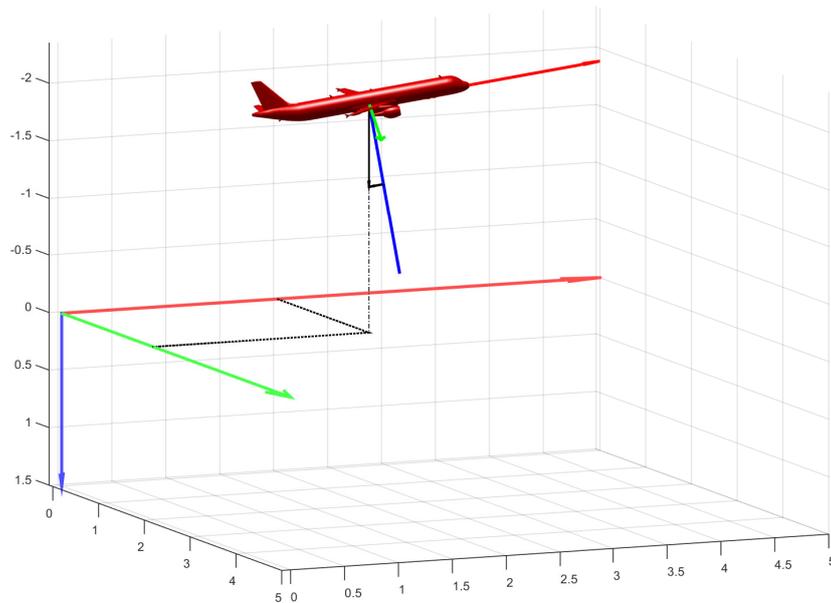


Figura 2.3 Modello tridimensionale *AirBus A-320* in evidenza il vettore le componenti del peso.

2.4 Orientamento successivi del velivolo *AirBus A-320* per una manovra di virata equilibrata - *Esercizio 2.4*

Si rappresenta una successione di posizioni e orientamenti nello spazio in cui il velivolo scelto passa da una configurazione ad assetto ad ali livellate e fusoliera orizzontale, ad un assetto di volo in virata a sinistra con ali inclinate di un angolo $\phi = -60$ deg. Si è scelto un raggio di virata pari a $R = 1500$ m, ad una quota di $h = 1000$ m. Per ottenere, l'orientamento successivo della suddetta manovra, si è caricato il Velivolo scelto, gli assi *Body* e gli assi *Earth*, come fatto nella precedente applicazione ed in fine, con l'ausilio della funzione `plotTrajectoryAndBodyE.m` si sono ottenute le posizioni successive del baricentro e dell'intero velivolo (in allegato consultare il Listato 2.2). Tutto è rappresentato con maggiore chiarezza nella figura 2.4 nella pagina successiva

2.5 Integrazione dell'equazioni della cinematica - *Esercizio 2-5*

Considerando le equazioni cinematiche (2.26 e 2.27 degli appunti forniti dal docente), sono assegnate le leggi di variazione delle componenti di velocità lineare $u(t)$, $v(t)$ e $w(t)$ ed angolare $p(t)$, $q(t)$ e $r(t)$, le condizioni iniziali in termini di posizione del baricentro ed i valori iniziali degli angoli di Eulero, si integrano numericamente le equazioni cinematiche. In particolare, la posizione iniziale del baricentro sarà quella riportate di seguito:

$$X_0=0.0 \text{ m}, Y_0=0.0 \text{ m}, Z_0=1000 \text{ m}$$

Le condizioni iniziali degli angoli di Eulero, sono:

$$\psi(0)=\theta(0)=\phi(0)=0$$

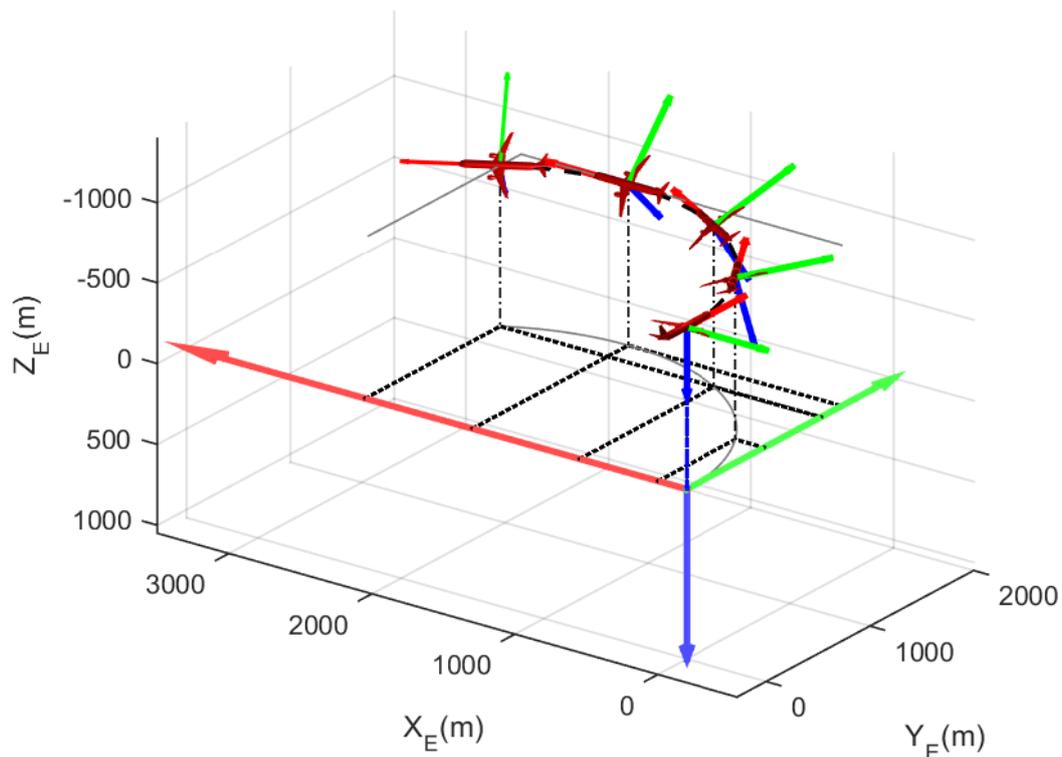


Figura 2.4 Evoluzione di virata a sinistra, Esercizio 2.4, del velivolo *AirBus A-320*.

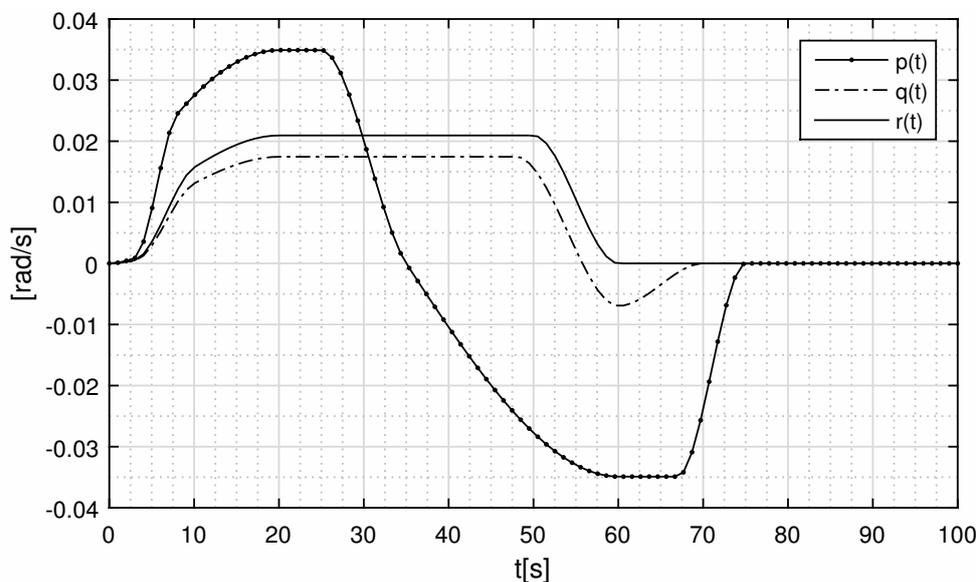


Figura 2.5 Componenti $p(t)$, $q(t)$ ed $r(t)$ della velocità angolare.

Per le velocità angolari si sono assegnate degli andamenti forniti dal docente, le quali verranno implementate in Matlab attraverso delle *funzioni anime* (aventi la seguente sintassi $\langle @(\tau) \rangle$, Listato 2.3). Tale scelta è stata fatta per poter avere tutto il codice che svolge l'Esercizio 2.5 in un unico *script*, così da facilitarne la leggibilità.

Nei grafici 2.5 e 2.6 nella pagina successiva è possibile apprezzare gli andamenti specifici assegnati per le velocità lineari ed angolari. Dopo aver impostato il problema si procede con l'integrazione numerica dell'equazione cinematica (2.27, *Quaderno 2*), riportata

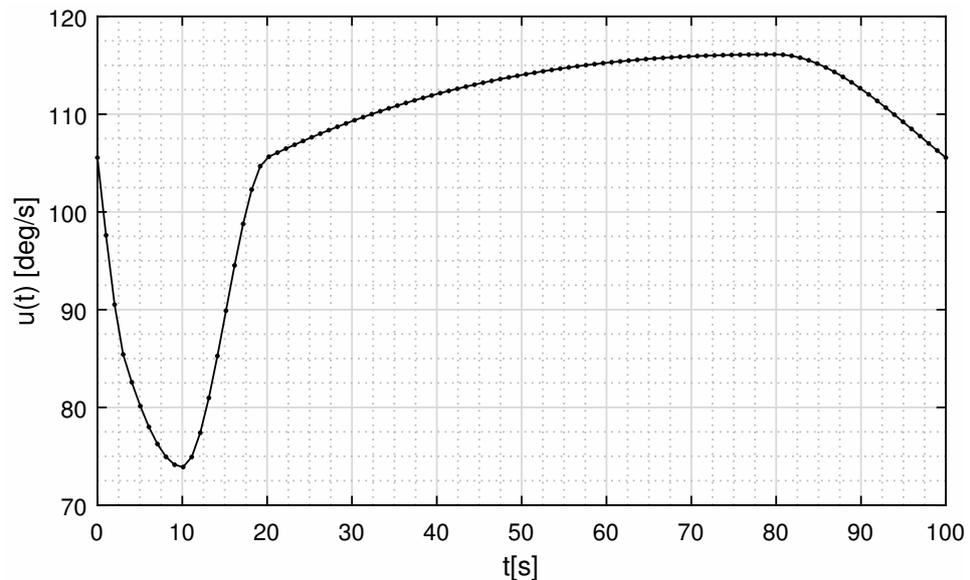


Figura 2.6 Componente u della velocità del baricentro.

di seguito per completezza:

$$\begin{pmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = \begin{bmatrix} 1 & \frac{\sin \phi \sin \theta}{\cos \theta} & \frac{\cos \phi \sin \theta}{\cos \theta} \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad (2.2)$$

Si integra numericamente la relazione 2.2 attraverso la funzione ode45 (Listato 2.4) di Matlab, che restituisce in *output* i vettori

- *vTime*, tempi d'integrazione utilizzati dalla ode45 nell'applicazione dei metodi di Runge-kutta.
- *vPhiThetaPsi*, storia temporale degli angoli d'Eulero ai tempi *vTime* (2.8 a pagina 22).

In seguito, conoscendo le storie temporali degli angoli d'Eulero, è possibile integrare numericamente anche l'equazione della cinematica (2.26), riportata di seguito per completezza.

$$\begin{pmatrix} \dot{X}_{E,G} \\ \dot{Y}_{E,G} \\ \dot{Z}_{E,G} \end{pmatrix} = \left[T_{EB}(\psi, \theta, \phi) \right]^T \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (2.3)$$

L'integrazione numericamente del sistema 2.3 mediante la funzione ode45 mi restituisce in *output* i seguenti vettori:

- *vTime2*, tempi d'integrazione utilizzati dalla ode45 nell'applicazione dei metodi di Runge-kutta.
- *vPose*, storia temporale della posizione del baricentro ai tempi *vTime2* (Figura 2.9).

In fine, si vuole raffigurare la manovra assegnata mediante le funzioni utilizzate in precedenza nell'applicazione dell'*Esercizio 2.4, 2.7* (`plotTrajectoryAndBodyE.m`).

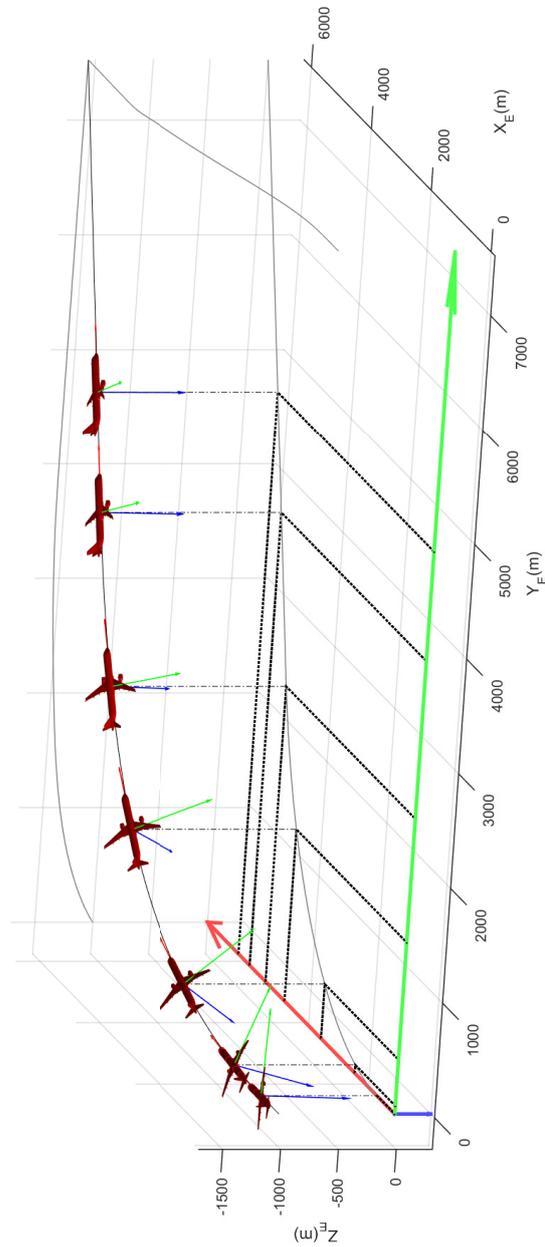


Figura 2.7 Virata rapida a destra, risultato dell'integrazione dell'equazioni cinematiche dell'*Esercizio 2.5*.

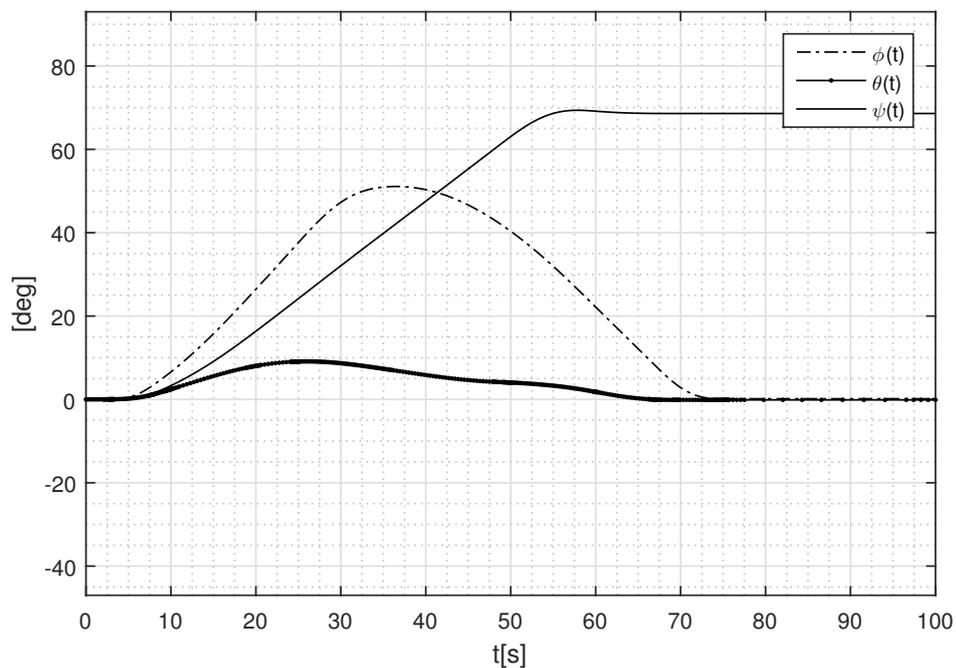


Figura 2.8 Time history degli angoli di Eulero.

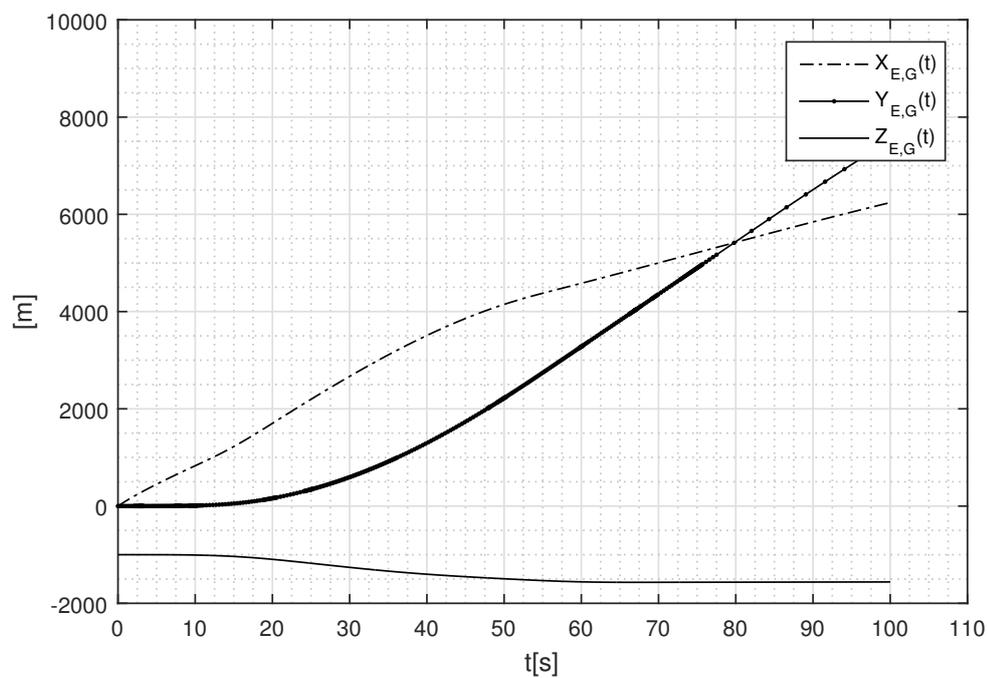


Figura 2.9 Time history delle coordinate del baricentro in assi Earth.

2.6 Litati utilizzati

2.6.1 Listato 2.1

```

%Dati relativi alla massa
m=73900;      %[Kg]
g=9.81;      %[m/s^2]
%Angoli di eulero assegnati
vAngoliEulero1=convang(vAngoliEulero,'rad','deg');
%visualizzo a video gli angoli assegnati
psi=vAngoliEulero(1);
theta=vAngoliEulero(2);
phi=vAngoliEulero(3);
%Costruisco la matrice di trasformazione assi terra assi body:
%Costruisco la matrice di trasformazione attraverso l'Aerospace ToolBox
%dalla funzione angle2dcm, la quale mi richiede in input gli angoli
%d'eulero e le rotazioni considerate;in tal caso si consideri una rotazione
%3-2-1:
Tbe=angle2dcm(psi,theta,phi,'ZYX');%dcm=matrici coseni direttori
vWE=[0;0;m*g];    %[N]
vWb=Tbe*vWE;      %Vettore peso riferimento body
%Rappresentazione di tutte le componenti peso lungo gli assi body:
hold on;
scala_peso=0.000001;
wscalato=m*g*scala_peso;

%Raffigurazione del vettore W nel riferimento Eart:
quiver3(vXYZe(1),vXYZe(2),vXYZe(3),...
    0,0,wscalato,...
    'AutoScale', 'off', 'Color',[0 0 0],'LineWidth',1.5);

%Componente del peso lungo Zb:

pWzb_b=scala_peso.*[0;0;vWb(3)];%forza peson nel riferimento body scalata
%valutata nell'esercizio precedente.

pWzb_e=vXYZe'+Tbe'*pWzb_b;

%Componente del peso lungo Xb:
vWxb_b=scala_peso.*[vWb(1);0;0];
vWxb_e=Tbe'*vWxb_b;

quiver3(pWzb_e(1),pWzb_e(2),pWzb_e(3),...
    vWxb_e(1),vWxb_e(2),vWxb_e(3),...
    'AutoScale', 'off', 'Color',[0 0 0],'LineWidth',2.0,...
    'MaxHeadSize',4.0);

```

2.6.2 Listato 2.2

```

%Esercizio 2.4
clc;clear all;close all;
h_fig1=figure(1);
grid on;
hold on;
light('Position',[1 1 -1200],'Style','local');%posizionamento della luce
%Impostazione dell'asse Ze, del sistema di riferimento inerziale, diretta
%verso il basso come da definizione:
set(gca,'xDir','reverse');
set(gca,'zDir','reverse');
%Carico il velivolo:
fattoreScala =280.0; %Impongo un fattore di scala per la visualizzazione
shape=loadAircraftMAT('aircraft.mat', fattoreScala);
%Definiamo la posizione del riferimento body rispetto all'Eart
%Condizione iniziale
vXYZe1=[0,0,-1000];%Definisco la posizione del baricentro rispetto gli assi

```

```

        %terra
vAngoliEulero1=convang([90,0,0], 'deg', 'rad');%Definisco il suo orientamento
        %dal punto di vista degli
        %angoli di Eulero.
theView=[420, 15]; %Punto di vista osservatore esterno

%Impostazioni assi body:
bodyAxesOptions.show=true; %Assi body visibili
bodyAxesOptions.magX=2.0*fattoreScala;
bodyAxesOptions.magY=2.0*fattoreScala;
bodyAxesOptions.magZ=1.5*fattoreScala;
bodyAxesOptions.lineWidth=1.5;
%Plotto tutti gli elementi creati in precedenza con la funzione del Toolbox
%Aerospace, plotBodyE:

plotBodyE(h_fig1,shape,vXYZe1,vAngoliEulero1,bodyAxesOptions,theView);

%Posiziono gli assi terra, per fare cio' utilizzo la funzione di matlab la
%quale opera avendo in input il valore massimo dell'asse x,y,z, centro
%degli assi ed in output mi restituisce sulla stessa figura utilizzata
%per il body, gli assi terra:

hold on;
xMax= max([abs(vXYZe1(1)),3500]);
yMax= max([abs(vXYZe1(2)),2000]);
zMax= 0.3*xMax;
vXYZ0=[0,0,0];
vExtent=[xMax,yMax,zMax];
plotEarthAxes(h_fig1, vXYZ0, vExtent);

%Raffigurazione della proiezione del baricentro sul piano Xe-Ye:
hold on;
options.lineColor=('k');
options.lineStyle=('-.');
options.lineWidth=0.5;
plotPoint3DHelperLines(h_fig1, vXYZe1,options);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Posizione 2 della manovra:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
hold on;
R=1500; %Definisco il raggio di virata.
t=convang(30, 'deg', 'rad');

vXYZe2=[R-R*cos(t),R*sin(t),-1000];%Definisco la posizione del baricentro rispetto
    gli assi
        %terra
vAngoliEulero2=convang([60,0,-15], 'deg', 'rad');%Definisco il suo orientamento
        %dal punto di vista degli
        %angoli di Eulero.
theView=[420, 15]; %Punto di vista osservatore esterno

%Impostazioni assi body:
bodyAxesOptions.show=true; %Assi body visibili
bodyAxesOptions.magX=2.0*fattoreScala;
bodyAxesOptions.magY=2.0*fattoreScala;
bodyAxesOptions.magZ=1.5*fattoreScala;
bodyAxesOptions.lineWidth=2.5;
%Plotto tutti gli elementi creati in precedenza con la funzione del Toolbox
%Aerospace, plotBodyE:

plotBodyE(h_fig1,shape,vXYZe2,vAngoliEulero2,bodyAxesOptions,theView);
%Disegno linea guida posizione2
hold on;
options.lineColor=('k');
options.lineStyle=('-.');

```

```

options.lineWidth=0.5;
plotPoint3DHelperLines(h_fig1, vXYZe2,options);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%posizione 3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t=convang(60,'deg','rad');

vXYZe3=[R-R*cos(t),R*sin(t),-1000];%Definisco la posizione del baricentro rispetto
    gli assi
    %terra
vAngoliEulero3=convang([30,0,-30],'deg','rad');%Definisco il suo orientamento
    %dal punto di vista degli
    %angoli di Eulero.
theView=[420, 15]; %Punto di vista osservatore esterno

%Impostazioni assi body:
bodyAxesOptions.show=true; %Assi body visibili
bodyAxesOptions.magX=2.0*fattoreScala;
bodyAxesOptions.magY=2.0*fattoreScala;
bodyAxesOptions.magZ=1.5*fattoreScala;
bodyAxesOptions.lineWidth=2.5;
%Plotto tutti gli elementi creati in precenza con la funzione del Toolbox
%Aerospace, plotBodyE:

plotBodyE(h_fig1,shape,vXYZe3,vAngoliEulero3,bodyAxesOptions,theView);
%Disegno linea guida posizione2
hold on;
options.lineColor='k';
options.lineStyle='-.';
options.lineWidth=0.5;
plotPoint3DHelperLines(h_fig1, vXYZe3,options);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Posizione 4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t=convang(90,'deg','rad');

vXYZe4=[R-R*cos(t),R*sin(t),-1000];%Definisco la posizione del baricentro rispetto
    gli assi
    %terra
vAngoliEulero4=convang([0,0,-45],'deg','rad');%Definisco il suo orientamento
    %dal punto di vista degli
    %angoli di Eulero.
theView=[420, 15]; %Punto di vista osservatore esterno

%Impostazioni assi body:
bodyAxesOptions.show=true; %Assi body visibili
bodyAxesOptions.magX=2.0*fattoreScala;
bodyAxesOptions.magY=2.0*fattoreScala;
bodyAxesOptions.magZ=1.5*fattoreScala;
bodyAxesOptions.lineWidth=2.5;
%Plotto tutti gli elementi creati in precenza con la funzione del Toolbox
%Aerospace, plotBodyE:

plotBodyE(h_fig1,shape,vXYZe4,vAngoliEulero4,bodyAxesOptions,theView);
%Disegno linea guida posizione2
hold on;
options.lineColor='k';
options.lineStyle='-.';
options.lineWidth=0.5;
plotPoint3DHelperLines(h_fig1, vXYZe4,options);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Posizione 5
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t=convang(120,'deg','rad');

vXYZe5=[R-R*cos(t),R*sin(t),-1000];%Definisco la posizione del baricentro rispetto

```

```

gli assi
    %terra
vAngoliEulero5=convang([-30,0,-45],'deg','rad');%Definisco il suo orientamento
    %dal punto di vista degli
    %angoli di Eulero.
theView=[420, 15]; %Punto di vista osservatore esterno

%Impostazioni assi body:
bodyAxesOptions.show=true; %Assi body visibili
bodyAxesOptions.magX=2.0*fattoreScala;
bodyAxesOptions.magY=2.0*fattoreScala;
bodyAxesOptions.magZ=1.5*fattoreScala;
bodyAxesOptions.lineWidth=1.5;
%Plotto tutti gli elementi creati in precenza con la funzione del Toolbox
%Aerospace, plotBodyE:
plotBodyE(h_fig1,shape,vXYZe5,vAngoliEulero5,bodyAxesOptions,theView);
%Disegno linea guida posizione5
hold on;
options.lineColor=('k');
options.lineStyle=('-.');
options.lineWidth=0.5;
plotPoint3DHelperLines(h_fig1, vXYZe5,options);
%Disegno la traiettoria con la funzione apposita fornita dal prof.:
%Individuo i vettori d'input del programma:
%1)mXYZe:
N=100;
inf=linspace(0,2/3*pi,N)';
x=(R-R*cos(inf));
y=R*sin(inf);
z=-1000*ones(N,1);
mXYZe=[x,y,z];
mXYZe=[vXYZe1;vXYZe2;vXYZe3;vXYZe4;vXYZe5];
%2)mEulerAngles
psi=linspace(pi/2,pi/6,N)';
theta=zeros(N,1);
phi=linspace(0,-60,N)';
mEulerAngles=[psi,theta,phi];
%3)options
options.theView=[105, 15]; %Punto di vista osservatore esterno
options.samples=1;
%---Impostazioni assi body:
options.bodyAxes.show=true; %Assi body visibili
options.bodyAxes.magX=2.0*fattoreScala;
options.bodyAxes.magY=2.0*fattoreScala;
options.bodyAxes.magZ=1.5*fattoreScala;
options.bodyAxes.lineWidth=2.5;
%Altre impostazioni:
options.helperLines.show='on';
options.helperLines.lineColor='b';
options.helperLines.lineWidth=1.5;
options.helperLines.lineStyle='-.';
options.trajjectory.show='on';
options.trajjectory.lineStyle='--';
options.trajjectory.lineColor='k';
options.trajjectory.lineWidth=1.5;
hold on;
plotTrajectoryAndBodyE(h_fig1,shape,mXYZe,mEulerAngles,options);

%Impostazioni grafiche:
xlabel('X_E(m)'); ylabel('Y_E(m)'); zlabel('Z_E(m)');

```

2.6.3 Listato 2.3

```

clear all; close all; clc;
%Assegnate le condizioni iniziali del problema(in termini di velocita' ed
%angoli di Eulero)si vanno ad integrare l'equazioni del moto in assi body.
%Durata della simulazione in secondi:
t_fin = 100.0; %[s]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Condizioni iniziali assegnate:
%1)Angoli d'Eulero:
psi0 = 0;
theta0 = 0;
phi0 = 0;
%2)Posizione baricentro
x0 = 0.0;          %[m]
y0 = 0.0;          %[m]
z0 = -1000;        %[m]
%Nota.1:-----
%Si considerano delle time history specifiche per i valori di velocita' del
%baricentro lungo la simulazione Vel_G=(u,v,w), per semplicita'
%considereremo la velocita nulla lungo y e z. Allo stesso modo verranno
%assegnate delle time history per le velocita' angolari degli angoli di
%Eulero-->(p,q,r)(Vell_rollio,Vell_imbardata,Vell_beccheggio).
%-----
%1)Time History velocita' angolari, lungo assi body:

p_max = convangvel(2.0,'deg/s','rad/s'); % rad/s
q_max = convangvel(1.0,'deg/s','rad/s'); % rad/s
r_max = convangvel(1.2,'deg/s','rad/s'); % rad/s

%Si costruiscono le storie temporali delle velocita' in funzione
%del tempo, attraverso la definizione di funzioni ''anonime'', nello scopo
%di andare ad ottenere un unico script per l'Esercizio 2.5.

vBreakPointsP(1,:) = [0, 0.030*t_fin, 0.08*t_fin, 0.20*t_fin, 0.25*t_fin, ...
    0.35*t_fin, 0.60*t_fin, 0.67*t_fin, 0.75*t_fin, t_fin];
vBreakPointsP(2,:)=[0, 0.025*p_max, 0.70*p_max, 1.00*p_max, 1.00*p_max, ...
    0, -1.00*p_max,-1.00*p_max, 0, 0];
p = @(t) ...%funzione anonima nella variabile t
    interp1( vBreakPointsP(1,:), vBreakPointsP(2,:), t, 'pchip' );

vBreakPointsQ(1,:) = [0, 0.030*t_fin, 0.10*t_fin, 0.20*t_fin, 0.48*t_fin, ...
    0.60*t_fin, 0.70*t_fin, t_fin];

vBreakPointsQ(2,:) = [0, 0.025*q_max, 0.75*q_max, 1.00*q_max, 1.00*q_max, ...
    -0.40*q_max, 0, 0];
q = @(t) ...%funzione anonima nella variabile t
    interp1( vBreakPointsQ(1,:), vBreakPointsQ(2,:), t, 'pchip');

vBreakPointsR(1,:) = [0, 0.030*t_fin, 0.10*t_fin, 0.20*t_fin, 0.50*t_fin, ...
    0.60*t_fin, t_fin];
vBreakPointsR(2,:) = [0, 0.025*r_max, 0.75*r_max, 1.00*r_max, 1.00*r_max, ...
    0, 0];

r = @(t) ...%funzione anonima nella variabile t
    interp1( vBreakPointsR(1,:), vBreakPointsR(2,:), t, 'pchip' );
%2)Time History velocita' baricentro, lungo assi body:
%Nota.2:-----
%Si consideri, per semplicita' solo la velocita' lungo l'asse x_b, mentre le
%velocita' lungo y_b e z_b si considerano nulle.
%-----

u0 = convvel(380.0,'km/h','m/s'); % m/s
v0 = convvel( 0.0,'km/h','m/s'); % m/s
w0 = convvel( 0.0,'km/h','m/s'); % m/s

```

```

vBreakPointsU(1,:) = [0, t_fin/30, t_fin/10, t_fin/5, 0.8*t_fin, t_fin];
vBreakPointsU(2,:) = [u0, 0.8*u0, 0.7*u0, u0, 1.1*u0, u0];

u = @(t) ...%funzione anonima nella variabile t
      interp1( vBreakPointsU(1,:), vBreakPointsU(2,:), t, 'pchip' );

v = @(t) 0;%Nota.2
w = @(t) 0;%Nota.2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Si definisce una funzione anonima dPhiThetaPsidt, che bisognerà passare
%alla funzione ode45 insieme ad un intervallo temporale e le condizioni
%iniziali per ottenere l'integrazione del sistema delle equazioni Gimbal:
%Implemento l'equazione 2.7-->GIMBAL EQUATIONS.
%Nota.3:-----
%La funzione anonima in questione e' funzione della finestra temporale
%ed il vettore x=(phi,theta,psi)
%-----

dPhiThetaPsidt = @(t, x) ... %Nota.3/Nota.4
    [1, sin(x(1)).*sin(x(2))./cos(x(2)), ...
      cos(x(1)).*sin(x(2))./cos(x(2)); ...
    0, cos(x(1)), -sin(x(1)); ...
    0, sin(x(1))./cos(x(2)), cos(x(1))./cos(x(2))...
    ]*[p(t); q(t); r(t)];

%Nota.4:-----
%La seguente funzione anonima ha nel suo interno le funzioni p(t),r(t)
%e q(t) che sono altrettanto anonime ed indicano la storia temporale delle
%componenti della velocità angolare degli angoli di Eulero.
%-----
%Integrazione dell'equazioni Gimbal con la funzione ode45:
%1)Impostazioni della funzione ode45:
options = odeset( ...
    'RelTol', 1e-9, ... %Nota.5
    'AbsTol', 1e-9*ones(1,3)... %Nota.5
);

%Nota.5:-----
%1)'RelTol'-->controlla la dimensione di ciascuna soluzione,
%contenendone la dimensione ad una grandezza impostata, in tal caso sarà
%considerato 1e-9;
%2)'AbsTol'-->Indica lo zero numerico, e' costituito da un vettore 3x1, perché
%ho che la mia soluzione sarà costituita da 3 vettori, ovvero le derivate
%rispetto al tempo dei tre angoli di eulero.
%-----
%Vettore delle condizioni iniziali:
vPhiThetaPsi0 = [phi0, theta0, psi0];

[vTime, vPhiThetaPsi] = ...
    ode45(dPhiThetaPsidt, [0 t_fin], vPhiThetaPsi0, options);
%Nota.6:-----
%La funzione ode45 si prende in entrata, la funzione definita in
%precedenza come funzione anonima, un intervallo di tempo, le condizioni
%iniziali ed in fine le opzioni definite in precedenza.
%-----
%Si procede con l'integrazione della posizione del baricentro, o
%meglio si ricaverà dagli angoli di Eulero valutati prima la posizione del
%baricentro, andando ad implementare l'equazione 2.29 del quaderno 2.
%1)Vado a valutare le velocità lungo x,y e z nei tempi vTime utilizzati per
%l'integrazione:
for i=1:numel(vTime)
    vU(i) = u(vTime(i)); % (anonymous functions evaluated)
    vV(i) = v(vTime(i));

```

```

    vW(i) = w(vTime(i));
end
%2)Il vettore vPhiThetaPsi e' un vettore costituito dalla time history
%degli angoli di Eulero ai tempi vTime.
%Andiamo a creare, come fatto per u(t)
%v(t) e w(t) funzioni anonime degli angoli di eulero che mi interpolano le
%i punti di controllo ottenuti dall'integrazione
%numERICA.vPhiThetaPsi(:,i)-->tutta la riga l'iesima colonna.

fPhi = @(t) ...
    interp1(vTime,vPhiThetaPsi(:,1),t);

fTheta = @(t) ...
    interp1(vTime,vPhiThetaPsi(:,2),t);

fPsi = @(t) ...
    interp1(vTime,vPhiThetaPsi(:,3),t);
%Si Integrano l'equazioni della navigazione 2.29, definendo una
%funzione anonima dPosEdt, la quale verra' data alla funzione ode45 che la
%integrera e ci restituira' la traiettorie del mio velivolo.
%Tale funzione mi permette di passare dalle velocita' nel
%riferimento Body u v e w alle velocita nel riferimento Eart, che non sono
%nient'altro che le velocita' del baricentro.
%Integrandole otteniamo la posizione temporale del baricentro
%nel riferimento eart.

dPosEdt = @(t,Pos) ...%Equazione 2.9 quaderno 2, funzione anonima
    transpose(angle2dcm(fPsi(t),fTheta(t),fPhi(t),'ZYX'))*[u(t);v(t);w(t)];

%Integrazione dell'equazione della navigazione:
options = odeset( ...
    'RelTol', 1e-3, ...
    'AbsTol', 1e-3*ones(3,1)...
    );

vPosE0 = [0;0;0];

[vTime2, vPosE] = ode45(dPosEdt, vTime, vPosE0, options);

vXe = x0 + vPosE(:,1);
vYe = y0 + vPosE(:,2);
vZe = z0 + vPosE(:,3);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%VISUALIZZAZIONE 3D Del velivolo e della sua manovra
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
h_fig1 = figure(1);
grid on;
hold on;

light('Position',[1 0 -4],'Style','local');%posizionamento della luce
%Impostazione dell'asse Ze, del sistema di rifermento inerziale, diretta
%verso il basso come da definizione:
set(gca,'XDir','reverse');
set(gca,'ZDir','reverse');
daspect([1 1 1]); %gli assi sono riportati in scale uguali 1 a 1
%Carico il velivolo:
shapeScaleFactor = 380.0;
shape = loadAircraftMAT('aircraft.mat',shapeScaleFactor);
%Mi costruisco i vettori mXYZe ed mEulerAngles, notiamo che le condizioni
%iniziali sono gia' considerate nei vettori valutati nell'integrazione.
mXYZe = [vPosE(:,1),vPosE(:,2),vPosE(:,3)+z0];
mEulerAngles = [vPhiThetaPsi(:,3),vPhiThetaPsi(:,2),vPhiThetaPsi(:,1)];

%Impostazioni immagine.-----
%1)Impostazioni generali:

```

```

options.samples = [50:60:numel(vTime)];
options.theView = [105 15];

%2)Impostazioni assi Body:

options.bodyAxes.show = true;
options.bodyAxes.magX = 1.5*shapeScaleFactor;
options.bodyAxes.magY = 2.0*shapeScaleFactor;
options.bodyAxes.magZ = 2.0*shapeScaleFactor;
options.bodyAxes.lineWidth = 0.5;

%3)Linea di traiettoria:

options.helperLines.show = true;
options.helperLines.lineStyle = '-.';
options.helperLines.lineColor = 'k';
options.helperLines.lineWidth = 0.2;

%4)Opzioni traiettoria:

options.trajectory.show = true;
options.trajectory.lineStyle = '-.';
options.trajectory.lineColor = 'k';
options.trajectory.lineWidth = 0.2;

plotTrajectoryAndBodyE(h_fig1, shape, mXYZe, mEulerAngles, options);

%Assi terra:
hold on;
hold on;
xMax= max([max(abs(mXYZe(:,1))),5]);
yMax= max([max(abs(mXYZe(:,2))),5]);
zMax= 0.05*xMax;
vXYZ0=[0,0,0];
vExtent=[xMax,yMax,zMax];
plotEarthAxes(h_fig1, vXYZ0, vExtent);
xlabel('X_E(m)');ylabel('Y_E(m)');zlabel('Z_E(m)');
hold off

```

Quaternione dell'orientamento di un velivolo

Indice

3.1	Introduzione	31
3.2	Impostazione del problema	32
3.3	Il codice di calcolo	33
3.4	Manovra di Looping	38
3.4.1	Looping perfetto - <i>Esercizio 3.1</i>	39
3.4.2	Looping realistico - <i>Esercizio 3.1</i>	43
3.5	Manovra del <i>Tonneau</i>	46
3.5.1	Un primo esempio di <i>tonneau</i>	46
3.5.2	Un secondo esempio di <i>tonneau</i> - <i>Esercizio 3.2</i>	49
3.6	Manovra di <i>Eight</i> - <i>Esercizio 3.4</i>	52
3.7	Manovra di <i>Spirale</i> - <i>Esercizio 3.4</i>	55
3.8	Manovra di <i>Looping</i> eseguita in Simulink - <i>Esercizio 3.5</i>	58
3.9	Acquisizioni leggi temporali mediante sensori dello smartphone	65

3.1 Introduzione

Obiettivo di questa esercitazione è la descrizione cinematica del moto di un aeromobile nel corso di una sua manovra generica.

Le due ipotesi fondamentali alla base del seguente studio sono quelle di velivolo considerato come corpo rigido e di Terra piatta e inerziale. La prima assunzione consente di descrivere lo stato del velivolo (posizione e assetto) in termini delle sei variabili $\{x_{\hat{E},G}, y_{\hat{E},G}, z_{\hat{E},G}, \phi, \theta, \psi\}^T$, ovvero le tre coordinate del baricentro nel sistema di riferimento assi Terra Γ_E e i tre angoli di Eulero, e delle loro derivate. Equivalentemente, l'evoluzione del velivolo si può descrivere per mezzo del vettore di stato $\{x_{\hat{E},G}, y_{\hat{E},G}, z_{\hat{E},G}, q_0, q_x, q_y, q_z\}^T$, nel quale agli angoli di Eulero si sono sostituite le quattro componenti del qua-

ternione dell'orientamento o parametri di Eulero-Rodrigues, che costituiscono una parametrizzazione dell'assetto priva di singolarità. La seconda assunzione permette di trattare la terna di degli assi Terra Γ_E come un sistema di riferimento fisso e inerziale.

Precindendo dall'insieme della azioni (inerziali, aerodinamiche e propulsive) agenti sul velivolo, si assegnano le specifiche leggi di variazione temporale della velocità lineare V e della velocità angolare Ω e si procede alla determinazione delle storie temporali delle variabili di stato. Si tratta, in definitiva, di risolvere un problema ai valori iniziali retto dalle equazioni della navigazione inerziale e dai parametri di Eulero. L'integrazione numerica viene operata per mezzo del software Matlab, sfruttando alcune delle funzioni di libreria offerte da questo ambiente di lavoro.

Dopo una breve esposizione del problema matematico e del procedimento risolutivo, nella seconda parte è riprodotta la cinematica di tre diverse manovre di volo. Per ciascuna di queste sono state graficate le storie temporali delle coordinate del baricentro, degli angoli di Eulero e delle componenti del quaternione dell'orientamento. Infine, si rappresentano la traiettoria del baricentro e gli orientamenti successivi assunti dall'aeromobile nell'arco dell'intera manovra.

3.2 Impostazione del problema

La formulazione che si è utilizzata è quella di *Eulero-Rodrigues*, la quale ci ha consentito di risolvere il problema dell'indeterminazione.

$$\begin{pmatrix} \dot{q}_0 \\ \dot{q}_x \\ \dot{q}_y \\ \dot{q}_z \end{pmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{pmatrix} q_0 \\ q_x \\ q_y \\ q_z \end{pmatrix} \quad (3.1)$$

dove $\{\omega\}_B = \{p, q, r\}^T$ è il vettore delle componenti della velocità angolare ω lungo gli assi del *del sistema di riferimento corpo (Body reference Frame, BRF)*. Dopo aver definito le leggi di variazione temporale $p(t), q(t), r(t)$, è possibile integrare (numericamente) il sistema di quattro equazioni differenziali ordinarie del I ordine (3.1) per ottenere le storie temporali delle componenti del quaternione $q_0(t), q_x(t), q_y(t), q_z(t)$ in un certo intervallo di tempo $[t_0, t_f]$. E' necessario, a tal fine, assegnare opportune condizioni iniziali, consistenti nei valori delle stesse componenti del quaternione all'istante t_0 . Per tutti gli esercizi svolti nel seguito si suppone un volo inizialmente ad ali livellate, fusoliera orizzontale e prua nella direzione nord, condizione nella quale tutti gli angoli di Eulero hanno valore nullo, mentre le componenti del quaternione valgono:

$$q_0(t_0) = 1, q_x(t_0) = q_y(t_0) = q_z(t_0) = 0. \quad (3.2)$$

Una volta determinate le storie temporali delle componenti del quaternione, le storie temporali degli angoli di Eulero si ricavano per mezzo della trasformazione:

$$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} = \begin{pmatrix} \text{atan2}\left[2(q_0 q_x + q_y q_z), (q_0^2 - q_x^2 - q_y^2 + q_z^2)\right] \\ \text{asin}\left[2(q_0 q_y - q_x q_z)\right] \\ \text{atan2}\left[2(q_0 q_z + q_x q_y), (q_0^2 + q_x^2 - q_y^2 - q_z^2)\right] \end{pmatrix} \quad (3.3)$$

Per determinare la traiettoria del velivolo bisogna integrare il sistema di equazioni della navigazione, che in termini degli angoli di Eulero, assume la seguente forma:

$$\begin{pmatrix} \dot{x}_{E,G} \\ \dot{y}_{E,G} \\ \dot{z}_{E,G} \end{pmatrix} = \begin{bmatrix} C_\theta C_\psi & S_\phi S_\theta C_\psi - C_\phi S_\psi & C_\phi S_\theta C_\psi + S_\phi S_\psi \\ C_\theta S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (3.4)$$

dove si sono usate le abbreviazioni per le funzioni trigonometriche $C_\xi = \cos(\xi)$ e $S_\xi = \sin(\xi)$. Anche in questo caso è necessario specificare le leggi di variazione temporale delle tre componenti della velocità lineare V lungo gli assi corpo, $u(t)$, $v(t)$, $w(t)$, e assegnare i valori iniziali delle tre incognite $x_{E,G}$, $y_{E,G}$, $z_{E,G}$. Per semplicità si supponrà nel seguito che l'origine del riferimento Γ_E coincida con la posizione del baricentro all'istante di tempo iniziale t_0 , ovvero:

$$x_{\hat{E},G}(t_0) = y_{\hat{E},G}(t_0) = z_{\hat{E},G}(t_0) = 0. \quad (3.5)$$

Siccome si è interessati soltanto alla descrizione cinematica dell'evoluzione del velivolo, tale assunzione non lede la generalità delle applicazioni.

3.3 Il codice di calcolo

Il problema ai valori iniziali delineato nella precedente sezione è stato risolto numericamente con l'ausilio di Matlab. In particolare, ai fini dell'integrazione numerica si è sfruttata la funzione `ode45` che implementa un algoritmo risolutivo del quarto e del quinto ordine. Si è imposta una tolleranza sull'errore numerico dell'ordine di $\varepsilon = 10^{-9}$.

Di seguito è riportato il codice di calcolo per la riproduzione della manovra di *looping realistico*, che sarà presentata nel capitolo 3.4.

Listato 3.1 Codice per una manovra di looping

```
%% Esercizio 3.1 Looping Perfetto

clc; clear all; close all;

%% Condizioni iniziali
psi0=0; theta0=0; phi0=0; %Angoli di Eulero in deg
psi0=convang(psi0,'deg','rad'); theta0=convang(theta0,'deg','rad');
psi0=convang(psi0,'deg','rad'); %Angoli di Eulero in rad
Q0=angle2quat(psi0,theta0,phi0); %Es. del Quaternione iniziale (Eq3.61)

x0=0; y0=0; z0=0; PosE0=[x0,y0,z0]; %Sulla posizione

ti=0; tf=2*pi; %Tempi di manovra
tt=linspace(ti,tf,1e3);

%% Leggi temporali delle componenti della velocità del Velivolo in BRF
p=@(t) 0; %Velocità angolare di rollio in rad/s

t_fin=2*pi;
qmax=1.00; %Velocità angolare in rad/s
q=@(t) interp1([0.05*t_fin 0.1*t_fin 0.2*t_fin 0.3*t_fin 0.4*t_fin 0.5*t_fin ...
0.6*t_fin 0.7*t_fin 0.8*t_fin 0.9*t_fin t_fin ],...
```

```

    [qmax qmax qmax qmax qmax qmax qmax qmax qmax qmax qmax],t,'pchip');
r=@(t) 0; %Velocità angolare di imbardata in rad/s
%% Leggi temporali delle componenti della velocità di traslazione di G inBRF
u0=100; v0=0; w0=0; %al Tempo t=0 in m/s
V0=sqrt(u0*u0+v0*v0+w0*w0); %modulo della velocità
u=@(t)interp1([0.05 0.1*t_fin 0.2*t_fin 0.3*t_fin 0.4*t_fin 0.5*t_fin...
0.6*t_fin 0.7*t_fin 0.8*t_fin 0.9*t_fin t_fin],[u0 u0 u0 ...
u0 u0 u0 u0 u0 u0 u0 u0],t,'pchip');
v=@(t) 0;
w=@(t) 0;
%% Funzione che implementa il secondo membro della gimbal equation (eq3.67)
dQuatdt = @(t,Q) 0.5.*[0,-p(t),-q(t),-r(t);
p(t), 0, r(t),-q(t);
q(t),-r(t), 0, p(t);
r(t), q(t),-p(t), 0]*Q;
options=odeset('RelTol',1e-9,'AbsTol',1e-9);%Opzioni di tolleranza numerica
[vtime, vquat] = ode45(dQuatdt, [0 t_fin], Q0, options); %Equazione ode45
%Runge-Kutta
%% Funzione che restituisce dal quaternione le velocità angolari
[vpsir,vthetar,vphir]=quat2angle(vquat); %Quaternione angoli in rad eq3.62
vpsi=convang(vpsir,'rad','deg'); %Converto gli angoli in deg
vtheta=convang(vthetar,'rad','deg');
vphi=convang(vphir,'rad','deg');

```

Nel listato sono sfruttate alcune funzioni particolari dell'Aerospace Toolbox:

- `angle2quat`: converte gli angoli di Eulero (espressi in radianti) nel quaternione dell'orientamento;
- `quat2angle`: opera la conversione inversa a quella della function `angle2quat`, ovvero trasforma le componenti del quaternione dell'orientamento nella terna di angoli di Eulero;
- `angle2dcm`: accetta in input i tre angoli di Eulero e restituisce la corrispondente $[\mathbf{T}]_{EB}$, matrice di trasformazione dal sistema di riferimento assi corpo Γ_B al sistema di riferimento assi terra Γ_E . Si tratta della matrice che compare nella euqazione 3.1 a pagina 32.

E' riportato di seguito il continuo del listato in cui si mostrano i comandi scelti per meglio rappresentare a video i risultati dell'integrazione numerica.

Listato 3.2 Codice per una manovra di looping realistico

```

%% Plots
h1=figure(1)
subplot(511) %Plotto gli angoli di Eulero
plot(vtime,vpsi,'-','LineWidth',2.0)
hold on
plot(vtime,vtheta,'--','LineWidth',2.0)
plot(vtime,vphi,'-.','LineWidth',2.0)
legend('\psi','\theta','\phi','Location','best')
set(legend,'color','none');
grid minor
xlabel('t(s)');
ylabel('(deg)')
title('Angoli di Eulero')

```

```

subplot 512                                %Stabilisco due immagini della stessa figura
plot(vtime,vquat(:,1),'-','LineWidth',2.0)
hold on
plot(vtime,vquat(:,2),'--','LineWidth',2.0)
plot(vtime,vquat(:,3),'-.','LineWidth',2.0)
plot(vtime,vquat(:,4),':','LineWidth',2.0)%Plotto le 4 componenti del quaternione
grid minor
legend('q_0','q_x','q_y','q_z','Location','best')
set(legend,'color','none');
xlabel('t(s)');
ylabel('');
title('Componenti del Quaternione')

subplot 513                                %Velocità angolari convertite in deg di p,q,r
plot(vtime, convangvel(p(vtime),'rad/s','deg/s'),'-','LineWidth',2.0)
hold on
plot(vtime, convangvel(q(vtime),'rad/s','deg/s'),'--','LineWidth',2.0)
plot(vtime, convangvel(r(vtime),'rad/s','deg/s'),'-.','LineWidth',2.0)
legend('p(t)','q(t)','r(t)','Location','best')
set(legend,'color','none');
xlabel('t (s)');
ylabel('(deg/s)')
title('Leggi temporali delle componenti della velocità del velivolo nel BRF')
grid minor;

subplot 514                                %Velocità del baricentro in m/s
plot(vtime, u(vtime),'-','LineWidth',2.0)
hold on
plot(vtime, v(vtime),'--','LineWidth',2.0)
plot(vtime, w(vtime),'-.','LineWidth',2.0)
legend('u(t)','v(t)','w(t)','Location','best')
set(legend,'color','none');
xlabel('t (s)');
ylabel('(m/s)');
title('Leggi temporali delle componenti della velocità di traslazione del
      baricentro')
grid minor;

%% Integrazione delle equazioni della Navigazione noto il quaternione
Quate = @(t) ... %Sto integrando le componenti del Quaternione nel tempo
    [interp1(vtime,vquat(:,1),t), ...
     interp1(vtime,vquat(:,2),t), ...
     interp1(vtime,vquat(:,3),t), ...
     interp1(vtime,vquat(:,4),t)];
T_BE = @(Q)quat2dcm(Q); % Matrice di trasformazione da Earth to body axes
      % RHS equazioni della navigazione (Eq3.25)
dPosEdt = @(t,PosE) transpose(quat2dcm(Quate(t)))*[u(t);v(t);w(t)];

%% Soluzioni delle equazioni della navigazione
options = odeset('RelTol',1e-9,'AbsTol',1e-9*ones(3,1));
[vtime, vPosE] = ode45(dPosEdt, vtime, PosE0, options);
N = length(vPosE);
vXe = vPosE(:,1); vYe = vPosE(:,2); vZe = z0 + vPosE(:,3);

subplot 515
plot(vtime,vPosE(:,1),'-','LineWidth',2.0)
hold on
plot(vtime,vPosE(:,2),'--','LineWidth',2.0)
plot(vtime,vPosE(:,3),'-.','LineWidth',2.0)
legend('x_EG(t)','y_EG(t)','z_EG(t)','Location','best')
set(legend,'color','none');
xlabel('t (s)'); ylabel('(m)')
title('Leggi temporali delle componenti della posizione del baricentro')
grid minor;
prop={'PaperType','PaperSize','PaperPosition','PaperPositionMode','PaperUnits','

```

```

Units'};
valori={'<custom>', [18,29],[0,0,18,29], 'manual', 'Inches', 'Inches'};
set(h1, prop, valori);
print(h1, '-deps', 'images\TuttiParametriLoopingPerfetto');
print(h1, '-deps', 'C:\Users\Giuseppe\Dropbox\DSV\__template_esercitazioni\quaderno3
\images\TuttiParametriLoopingPerfetto');

%% Impostazioni della scena per la visualizzazione 3D
h_fig3 = figure(4);
grid minor;
hold on;
light('Position', [1 0 -4], 'Style', 'local');
set(gca, 'XDir', 'reverse'); % Modifica di Ze
set(gca, 'ZDir', 'reverse');
daspect([1 1 1]);

%% Trasformazione dal quaternione agli angoli di Eulero
for kk=1:size(vquat,1)
    dcm = quat2dcm(vquat(kk,:));
    for ir=1:3
        for ic=1:3
            vDCM(kk,ir,ic) = dcm(ir,ic);
        end
    end
    [vpsi(kk), vtheta(kk), vphi(kk)] = quat2angle(vquat(kk,:));
end

%% Load aircraft shape
shapeScaleFactor = 25.0;

shape = loadAircraftMAT('Aircraft\aircraft_mig29.mat', shapeScaleFactor);

mXYZe = [vPosE(:,1),vPosE(:,2),vPosE(:,3)+z0];
mEul = [vpsi,vtheta,vphi];

%% Impostazioni per il plot
% Impostazioni generali
options.samples = [30,60,90,120:30:numel(vtime)]; %punti della traiettoria
options.theView = [105 15]; %che voglio visualizzare

% Impostazione assi body
options.bodyAxes.show = true;
options.bodyAxes.magX = 1.5*shapeScaleFactor;
options.bodyAxes.magY = 2.0*shapeScaleFactor;
options.bodyAxes.magZ = 2.0*shapeScaleFactor;
options.bodyAxes.lineWidth = 2.5;

% helper lines
options.helperLines.show = false;
options.helperLines.lineStyle = ':';
options.helperLines.lineColor = 'k';
options.helperLines.lineWidth = 1.5;

% Traietoria
options.trajectory.show = true;
options.trajectory.lineStyle = '--';
options.trajectory.lineColor = 'k';
options.trajectory.lineWidth = 1.0;

%% Stampa corpo e traiettoria
plotTrajectoryAndBodyE(h_fig3, shape, mXYZe, mEul, options);

%% Stampa assi Terra
hold on;
xMax = max([max(abs(mXYZe(:,1))),5]);
yMax = max([max(abs(mXYZe(:,2))),5]);

```

```
zMax = 0.05*xMax; % max([abs(max(vXYZe(1))),0.18*xMax]);
vXYZ0 = [0,0,0];
vExtent = [xMax,yMax,zMax];
plotEarthAxes(h_fig3, vXYZ0, vExtent);
xlabel('x_E (m)'); ylabel('y_E (m)'); zlabel('z_E (m)')
hold off
print(h_fig3, '-deps', 'images\immaginetraiettoria');
```

3.4 Manovra di Looping

Il *looping* è una manovra che si esegue cabrando l'aereo affinché esegue un giro completo a forma di anello (*loop*).

Un *looping* perfetto dovrebbe disegnare una traiettoria circolare perfetta che termina esattamente nella medesima condizione (in termini di posizione ma anche di orientamento del velivolo) in cui ha avuto inizio e durante la quale l'angolo fra il vettore velocità lineare V del velivolo (tangente alla traiettoria) e la prua, o meglio l'asse x_B del sistema di riferimento assi corpo Γ_B , rimane costante e uguale al valore iniziale.

Si tratta di una manovra durante la quale teoricamente il pilota interviene solo sui comandi longitudinali e l'evoluzione del velivolo è contenuta in un piano verticale. Le velocità angolari di rollio e imbardata sono nulle nel corso di tutta la manovra, come pure la componente di velocità lineare in direzione trasversale al piano della traiettoria.

Inoltre se le condizioni di assetto iniziali sono quelle espresse dalle 3.2 a pagina 32, che come già detto equivalgono alla nullità di tutti e tre gli angoli di Eulero, il piano verticale in cui si traccia il *loop* è proprio il piano di simmetria longitudinale del velivolo e allora risulterà in particolare $v(t) = 0, \forall t$.

Valendo le suddette ipotesi, bisognerà assegnare soltanto le funzioni temporali $u(t)$ e $q(t)$.

Di seguito vengono trattati i due casi di *looping perfetto* (o, si potrebbe dire, *stazionario*) e di *looping realistico*.

3.4.1 Looping perfetto - *Esercizio 3.1*

E' possibile riprodurre una manovra di *looping perfetto* assegnando le seguenti leggi temporali di velocità lineare e angolare:

$$u(t) = u_0 = \text{cost}, \quad q(t) = q_0 = \text{cost}$$

posto che siano:

$$v(t) = w(t) = 0, \quad p(t) = r(t) = 0$$

le quali sono state riportate all'interno del codice di calcolo proposto inizialmente:

Listato 3.3 Leggi temporali per la manovra di Looping Perfetto

```

%% Leggi temporali delle componenti della velocità del Velivolo in BRF
p=@(t) 0; %Velocità angolare di rollio in rad/s

t_fin=2*pi;
qmax=1.00; %Velocità angolare in rad/s
q=@(t) interp1([0.05*t_fin 0.1*t_fin 0.2*t_fin 0.3*t_fin 0.4*t_fin 0.5*t_fin ...
0.6*t_fin 0.7*t_fin 0.8*t_fin 0.9*t_fin t_fin ],...
[qmax qmax qmax qmax qmax qmax qmax qmax qmax qmax qmax],t,'pchip');

r=@(t) 0; %Velocità angolare di imbardata in rad/s

%% Leggi temporali delle componenti della velocità di traslazione di G inBRF
u0=100; v0=0; w0=0; %al Tempo t=0 in m/s
V0=sqrt(u0*u0+v0*v0+w0*w0); %modulo della velocità

u=@(t)interp1([0.05 0.1*t_fin 0.2*t_fin 0.3*t_fin 0.4*t_fin 0.5*t_fin...
0.6*t_fin 0.7*t_fin 0.8*t_fin 0.9*t_fin t_fin],[u0 u0 u0 ...
u0 u0 u0 u0 u0 u0 u0],t,'pchip');

v=@(t) 0;

w=@(t) 0;

```

Nella figura 3.1 a pagina 41 sono mostrati con il seguente ordine gli andamenti degli angoli di Eulero, le componenti del Quaternione dell'orientamento generatesi per tale manovra, le leggi temporali delle componenti della velocità del velivolo nel sistema di riferimento Body per $q_0 = 1,00$ rad/s, le leggi temporali delle componenti della velocità di traslazione del baricentro per $u_0 = 100$ m/s, e le leggi temporali delle componenti della posizione del baricentro.

Come si può notare le velocità q e u restano costanti durante tutto il tempo in cui viene eseguita la manovra.

La 3.2 a pagina 42 è una rappresentazione tridimensionale della stessa manovra.

Si noti che una manovra di *looping perfetto* equivale a un moto circolare uniforme con velocità tangenziale pari a u_0 e velocità angolare pari a q_0 , nel quale il velivolo mantiene invariato il proprio orientamento rispetto alla terna di assi vento (e quindi alla direzione del vettore velocità V). Il periodo, ovvero il tempo di manovra, è determinato esclusivamente dalla velocità angolare tramite la formula:

$$T = \frac{2\pi}{q_0} \quad (3.6)$$

e dunque in questo caso $T = 2\pi s = 6,28s$. Il raggio del *loop*, invece, dipende da entrambe le velocità, essendo queste legate dalla nota relazione $u_0 = Rq_0$. Nel caso

in esame $R = 100m$, che è proprio lo spostamento massimo del baricentro in direzione dell'asse $x_{\hat{E}}$ della terna Γ_E (mentre lo spostamento massimo lungo la verticale $z_{\hat{E}}$ è uguale al diametro della circonferenza).

In maniera altrettanto ovvia, l'angolo $\theta(t)$ inizialmente è uguale all'angolo che sottende l'arco di circonferenza descritto dal baricentro nel tempo $t - t_0$, dove $t_0 = 0$, e dunque cresce linearmente secondo la legge $\theta = q_0 t$; dopo aver raggiunto un valore pari a $\pi/2$ (condizione di *gimbal lock* diminuisce con un rateo di variazione in valore assoluto uguale ancora a q_0 fino al valore di $-\pi/2$ rad (nuova condizione di *gimbal lock*, per poi crescere nuovamente, ancora con legge lineare).

Gli angoli ϕ e ψ , invece, hanno un andamento temporale costante a tratti, presentando un salto di $\pm\pi$ in corrispondenza delle due condizioni di *gimbal lock*.

Si noti, inoltre, come le componenti del quaternione dell'orientamento variano con continuità, a conferma del fatto che la parametrizzazione dell'orientamento basata sui parametri di Eulero-Rodriguez non presenta singolarità.

Le componenti della velocità lineare lungo gli assi Terra x_E e z_E , hanno un andamento sinusoidale nel tempo, essendo le componenti lungo due direzioni ortogonali, di un vettore tangente punto per punto alla circonferenza descritta dal moto circolare uniforme del suo punto di applicazione. La loro espressione analitica, semplicissima da ricavare, è:

$$\dot{x}_{\hat{E},G} = u_0 \sin(q_0 t) \quad \dot{z}_{\hat{E},G} = u_0 \cos(q_0 t)$$

che rappresenta la particolarizzazione delle equazioni 3.1 a pagina 32, al caso in esame (tenendo conto che anche se in generale $\theta(t) \neq q_0 t$, vale $\cos(\theta(t)) = \cos(q_0 t) \forall t$).

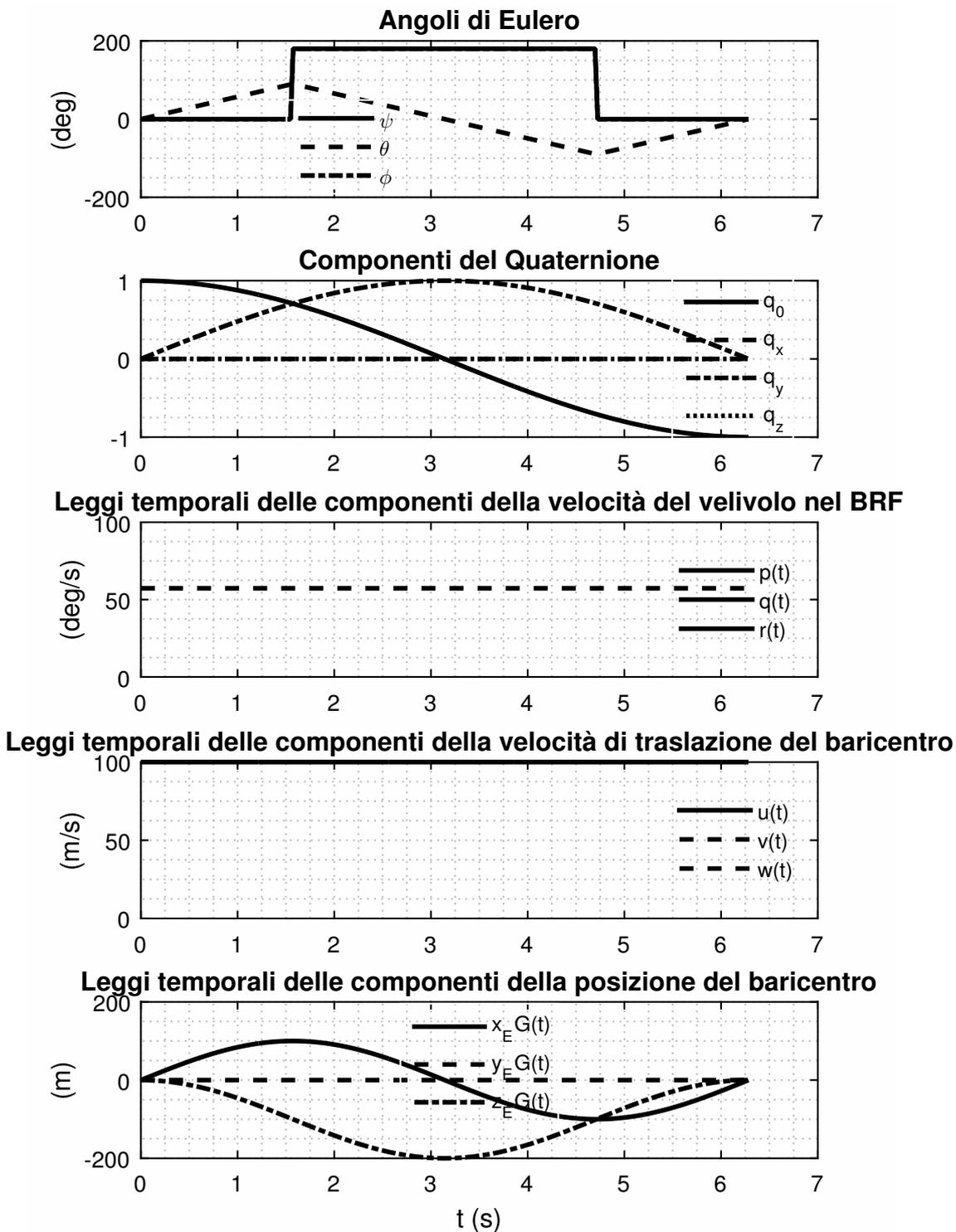


Figura 3.1 Time history dei parametri per la manovra di *Looping perfetto* per le leggi assegnate nel listato 3.3 a pagina 39

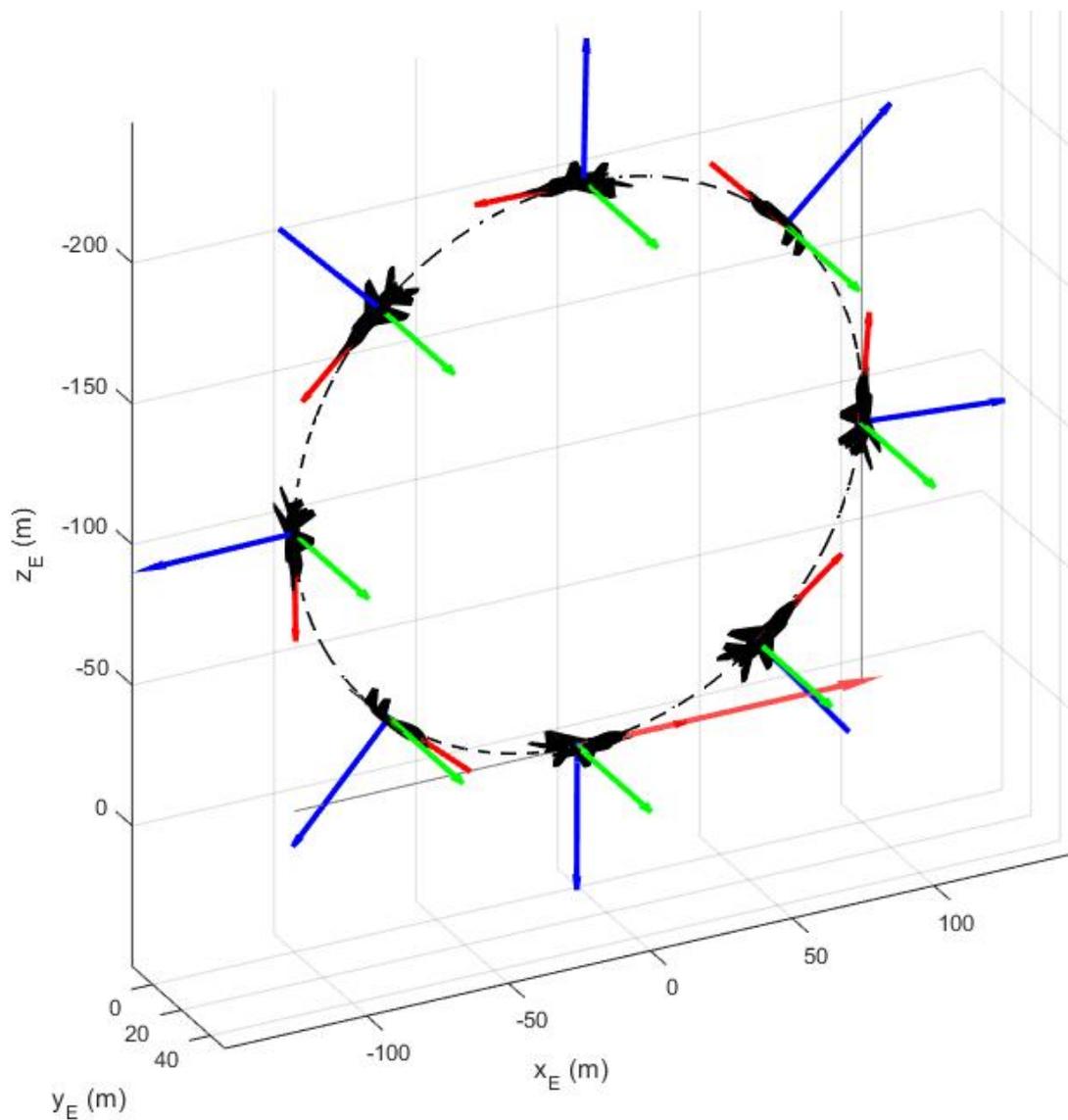


Figura 3.2 Rappresentazione della manovra di *Looping perfetto* per un orientamento iniziale definito dalle condizioni $\psi = \theta = \phi = 0$, e per le leggi di variazione delle componenti della velocità lineare e della velocità angolare riportate nel listato 3.3 a pagina 39.

3.4.2 Looping realistico - *Esercizio 3.1*

Una riproduzione più realistica della manovra di *looping* si può ottenere assegnando plausibili leggi temporali della componente u della velocità lineare e della componente q della velocità angolare e risolvendo nuovamente il problema retto dalle equazioni 3.1 e 3.4 con le condizioni iniziali date dalle 3.2 e 3.5.

Supponendo che inizialmente il moto del velivolo sia rettilineo uniforme, è verosimile immaginare che la componente $u(t)$, inizialmente pari a V_{cr} (velocità di crociera), decresca gradualmente con t , fino al raggiungimento di un valore costante V_m , mantenuto per tutta la manovra, mentre la velocità angolare di beccheggio $q(t)$, inizialmente nulla, cresca con t , fino al raggiungimento di un valore costante q_{max} , mantenuto per quasi tutta la manovra, per poi decrescere e tornare a 0 nella fase terminale.

Di seguito è riportato il listato 3.4 in cui sono state variate le leggi temporali per eseguire la manovra di un possibile Looping realistico.

Listato 3.4 Leggi temporali per la manovra di Looping realistico

```

%% Leggi temporali delle componenti della velocità del velivolo nel BRF
p=@(t) 0*sign(exp(t)); %Velocità angolare rad/s

t_fin=3*pi;
qmax=1.14;
q=@(t) interp1([0 0.1*t_fin 0.2*t_fin 0.3*t_fin 0.4*t_fin 0.5*t_fin ...
0.6*t_fin 0.7*t_fin 0.8*t_fin 0.9*t_fin t_fin],[0 0.25*qmax ...
0.50*qmax 0.75*qmax qmax qmax qmax 0.75*qmax 0.50*qmax 0.25*qmax 0],...
t,'pchip');

r=@(t) 0*sign(exp(t));

%% Leggi temporali delle componenti della velocità di traslazione di G
u0=250/3.6; v0=0; w0=0; %al Tempo t=0 in m/s
V0=sqrt(u0*u0+v0*v0+w0*w0); %modulo della velocità

u=@(t)interp1([0 0.1*t_fin 0.2*t_fin 0.3*t_fin 0.4*t_fin 0.5*t_fin...
0.6*t_fin 0.7*t_fin 0.8*t_fin 0.9*t_fin t_fin],[u0 0.9*u0 0.8*u0 ...
0.7*u0 0.6*u0 0.6*u0 0.6*u0 0.6*u0 0.6*u0 0.6*u0],t,'pchip');

v=@(t) 0*sign(exp(t));
w=@(t) 0*sign(exp(t));

```

Nella figura 3.3 sono mostrati con il seguente ordine gli andamenti degli angoli di Eulero, le componenti del Quaternione dell'orientamento generatesi per tale manovra, le leggi temporali delle componenti di velocità del velivolo nel sistema di riferimento Body, le leggi temporali delle componenti della velocità di traslazione del baricentro e le leggi temporali delle componenti della posizione del baricentro.

Sostanzialmente gli andamenti degli angoli di Eulero e delle coordinate del baricentro sono simili per il caso del *looping perfetto*, tranne nelle fasi iniziale e finale della manovra. Il tempo di osservazione in questo caso è più ampio e si osserva a valle della manovra una fase di moto rettilineo uniforme, come si evince anche dalla figura ??, che riporta una vista tridimensionale della traiettoria descritta dal velivolo.

Si noti, inoltre, che un effetto della decelerazione del velivolo nella prima fase della manovra è che lo spostamento nella direzione di avanzamento iniziale (ovvero quella

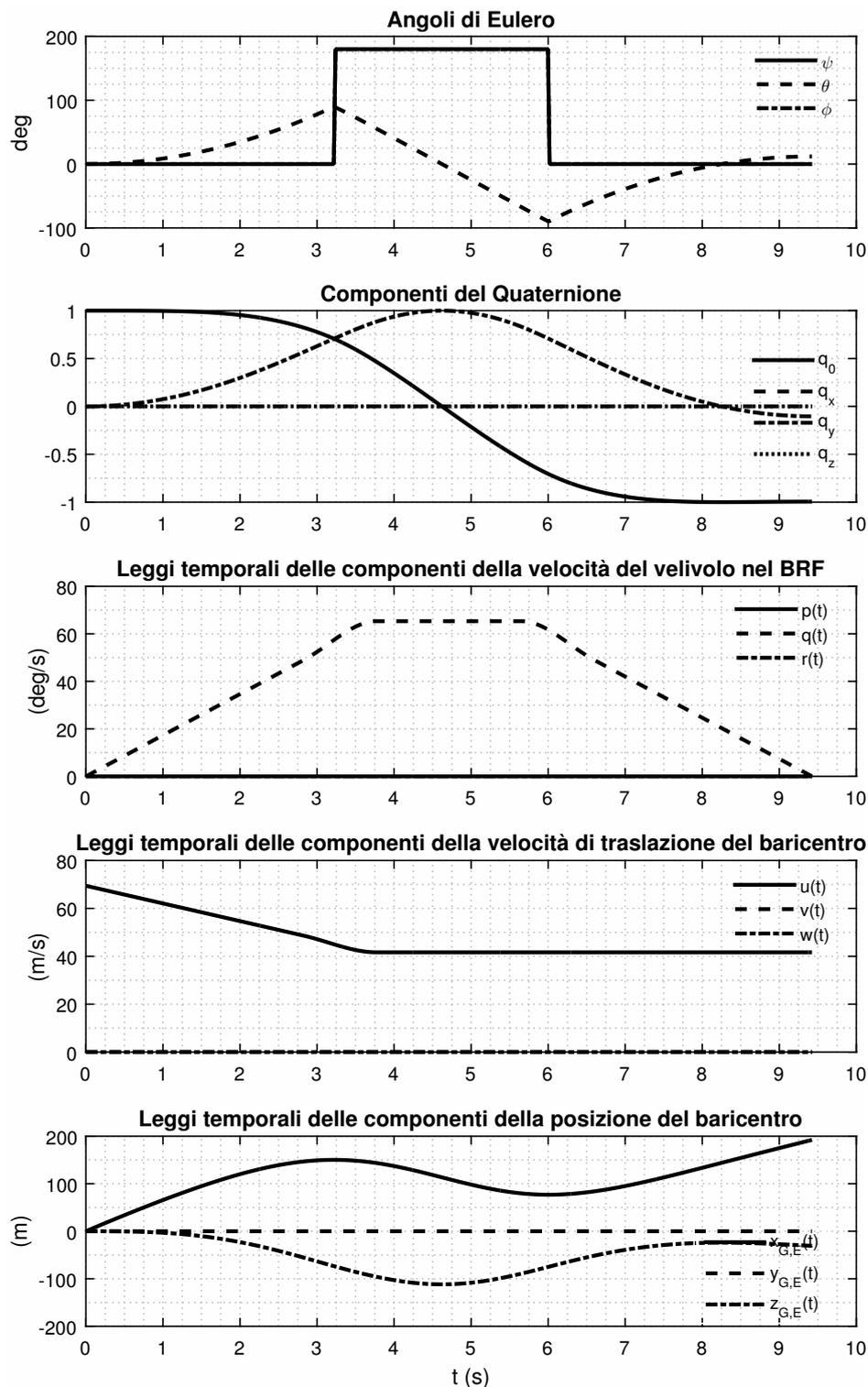


Figura 3.3 Time history dei parametri per la manovra di *Looping* realistico e per le leggi assegnate nel listato 3.4 nella pagina precedente.

dell'asse $x_{\hat{E}}$) durante l'arco di salita risulta ben maggiore dello spostamento (in valore assoluto) nell'arco di discesa.

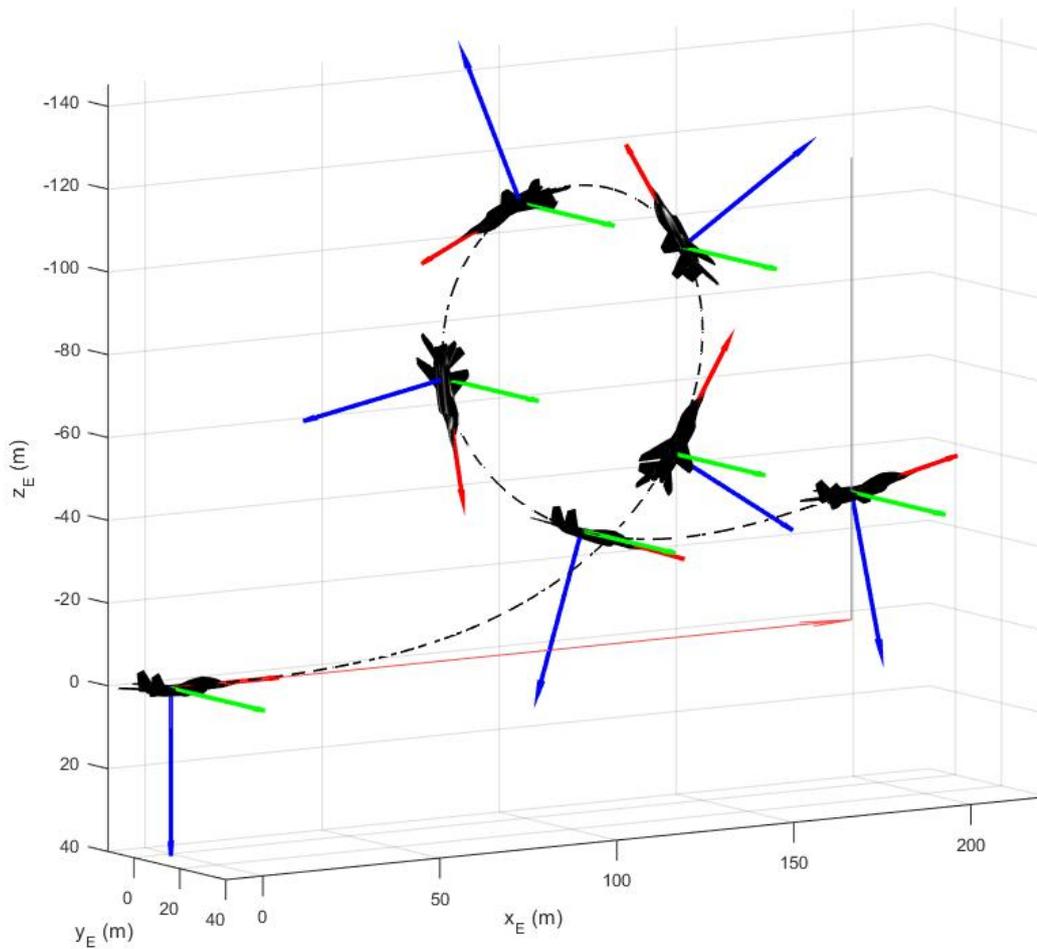


Figura 3.4 Rappresentazione della manovra di *looping* realistico per un orientamento iniziale definito dalle condizioni $\psi = \theta = \phi = 0$, e per le leggi di variazione delle componenti della velocità lineare e della velocità angolare rappresentate nel listato 3.4 a pagina 43.

3.5 Manovra del *Tonneau*

Il *Tonneau* è una manovra che porta il velivolo a ruotare di 360 deg attorno all'asse longitudinale di rollio. Esistono diverse varianti di questa manovra, a seconda che venga eseguita più o meno rapidamente e mantenendo orizzontale l'asse del velivolo oppure compiendo una volta più o meno ampia. In quest'ultimo caso si parla di *tonneau a botte*, in quanto il velivolo oltre a ruotare attorno all'asse longitudinale, descrive una sorta di traiettoria elicoidale che somiglia alla spirale sulla superficie interna di una botte. Se si ruota di soli 180deg si parla di *mezzo tonneau*.

3.5.1 Un primo esempio di *tonneau*

In questa sezione si rappresenta una prima manovra di *tonneau*. Questa è ottenuta ancora una volta assegnando come condizioni iniziali le 3.2 a pagina 32 e le 3.5 a pagina 33, ma assumendo un moto di rotazione del velivolo differente, per semplicità assumiamo le leggi seguenti:

$$q(t) = q_0 = 1,00rad/s, p(t) = p_0 = 0,50rad/s, r(t) = 0,00rad/s$$

e le seguenti velocità lineari

$$u(t) = u_0 = 250km/h, v(t) = w(t) = 0,00Km/h$$

le quali sono state implementate all'interno del codice di calcolo proposto nel seguente modo:

Listato 3.5 leggi temporali per la manovra di Tonneau

```
%% Leggi temporali delle componenti della velocità del velivolo in BRF
t_fin=7.0;

p_cost=0.5; %valori in rad/s
q_cost=1;
r_cost=0;

p=@(t) interp1([0, t_fin/30, t_fin/10, t_fin/5, 0.7*t_fin, 0.9*t_fin, ...
    t_fin],[p_cost, p_cost, p_cost, p_cost, p_cost, p_cost, p_cost],...
    t,'pchip'); %Velocità angolare di rollio in rad/s
q=@(t) interp1([0, t_fin/30, t_fin/10, t_fin/5, 0.8*t_fin, 0.9*t_fin, ...
    t_fin],[q_cost, q_cost, q_cost, q_cost, q_cost, q_cost],...
    t,'pchip'); %Velocità angolare di rollio in rad/s
r=@(t) 0*(t); %Velocità angolare di imbardata in rad/s

%% Leggi temporali delle componenti della velocità di G in BRF

u0=convvel(250.0, 'Km/h', 'm/s');
v0=convvel(0.0, 'Km/h', 'm/s');
w0=convvel(0.0, 'Km/h', 'm/s');

V0=sqrt(u0*u0+v0*v0+w0*w0); %modulo della velocità

u=@(t)interp1([0, t_fin/30, t_fin/10, t_fin/5, 0.8*t_fin,t_fin],...
    [u0, u0, u0, u0,u0, u0],...
    t,'pchip');

v=@(t) 0*(t);
w=@(t) 0*(t);
```

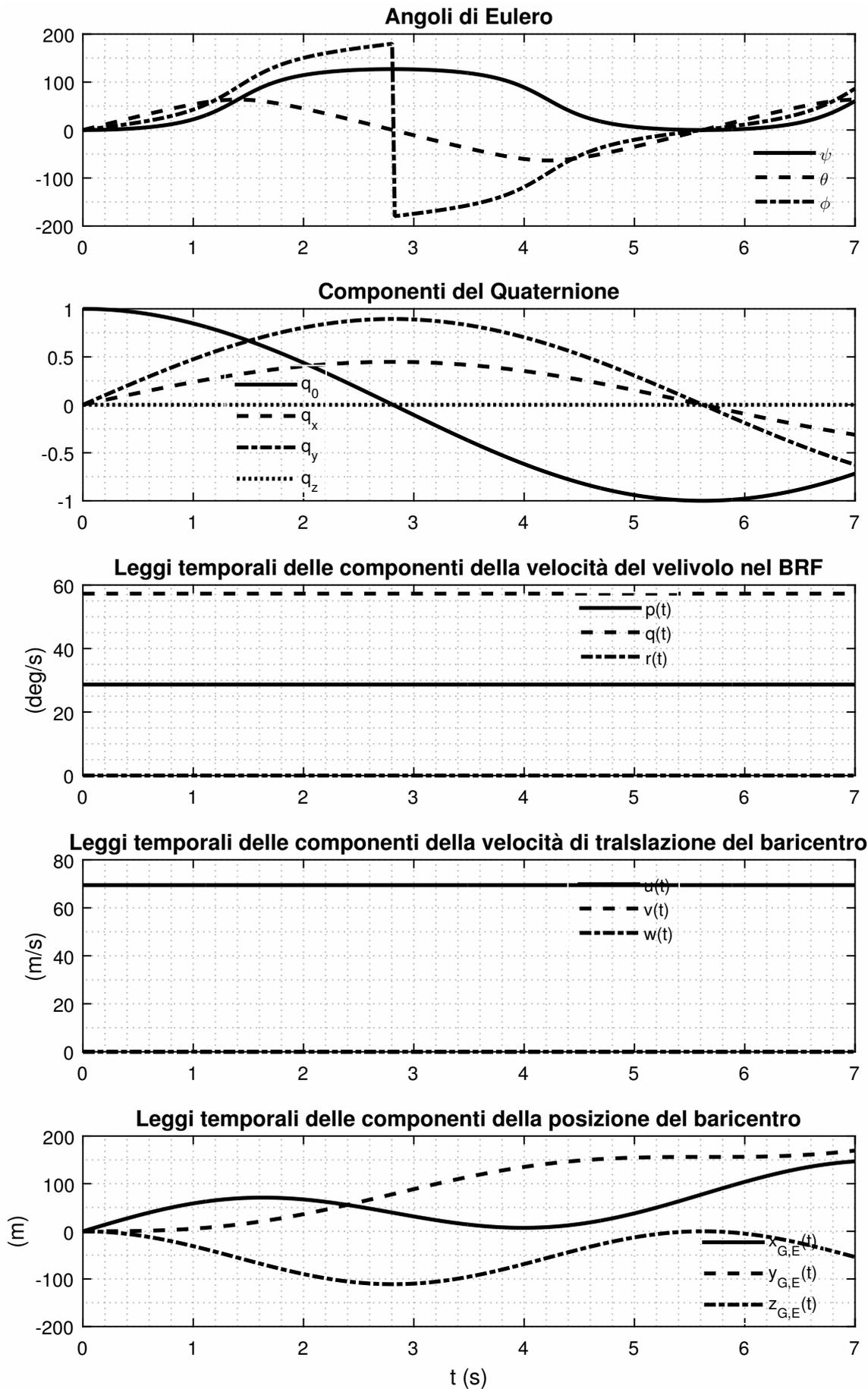


Figura 3.5 Time history dei parametri per la manovra di *Tonneau*, per le leggi temporali assegnate nel listato 3.5 a fronte.

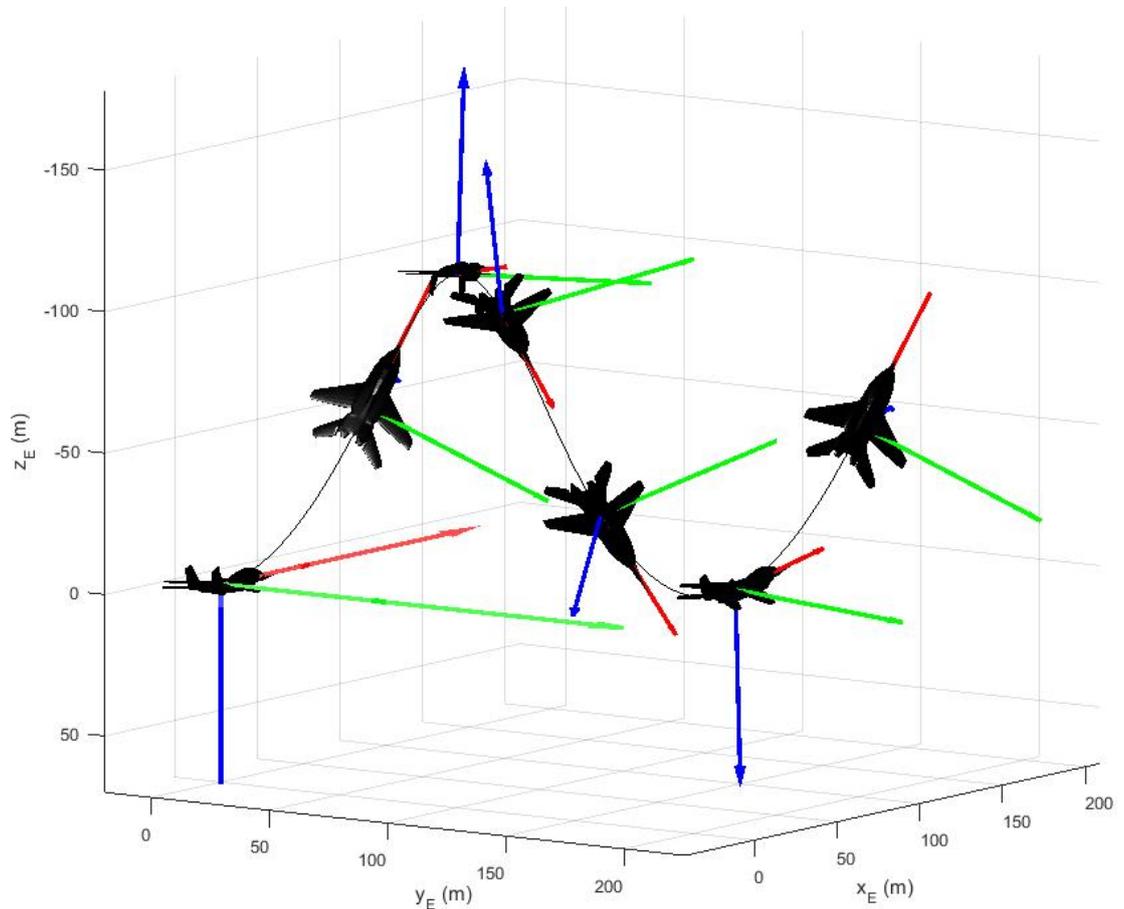


Figura 3.6 Rappresentazione della manovra di *tonneau* per un orientamento iniziale definito dalle condizioni $\psi = \theta = \phi = 0$ e per le leggi di variazione delle componenti della velocità lineare e della velocità angolare rappresentate nel listato 3.5 a pagina 46.

Il risultato dell'integrazione numerica è rappresentato nella figura 3.5 nella pagina precedente. In questo caso l'angolo di prua non subisce una discontinuità con salti poiché l'evoluzione è influenzata maggiormente dalla velocità angolare di rollio. Infatti, l'angolo ψ cresce nel tempo e raggiunge un massimo prima di raggiungere l'estremo superiore del suo intervallo di definizione. Successivamente, esso decresce e tende al valore nullo iniziale. D'altra parte, in questo moto a velocità angolare di rollio costante, come è possibile vedere in figura 3.5 nella pagina precedente, l'angolo ψ raggiunge il limite superiore del suo intervallo di definizione. Non appena l'angolo di *bank* tende a superare i 180 deg esso subisce una variazione a scatto di -180 deg che lo riporta in prossimità dell'estremo inferiore del suo intervallo di definizione.

3.5.2 Un secondo esempio di *tonneau* - *Esercizio 3.2*

Si ripropone l'esercizio precedente assegnando stavolta leggi delle componenti di velocità lineare e di velocità angolare non identicamente costanti. Ancora una volta:

$$r(t) = 0,00 \text{ rad/s}, v(t) = w(t) = 0,00 \text{ Km/h}$$

mentre si assume:

- la velocità angolare di roppil $p(t)$, inizialmente nulla e crescente con t , fino al raggiungimento di un valore costante p_{MAX} , mantenuto per quasi tutta la manovra; nella fase terminale dell'evoluzione si ipotizza un decadimento della $p(t)$ da p_{MAX} a 0;
- una analoga legge per la componente di velocità angolare di beccheggio $q(t)$, ma con q_{MAX} pari alla metà di p_{MAX} ;
- la $u(t)$ inizialmente pari a $u_0 = 380 \text{ Km/h}$, decresce con t fino a un valore minimo ($\approx 0,70u_0$) e successivamente crescente con t in modo da raggiungere u_0 a fine manovra.

Di seguito è riportato una parte dello script realizzato in Matlab in cui si mostra come sono state realizzate le leggi temporali proposte nella suddetta esercitazione:

Listato 3.6 Leggi temporali per la manovra di *Tonneau*

```
%% Leggi temporali delle componenti della velocità del velivolo in BRF
t_fin=105.0;

p_max=convangvel(4.0,'deg/s','rad/s');           %Ottengo valori in rad;
q_max=convangvel(2.0,'deg/s','rad/s');
r_max=0;

p=@(t) interp1([0, t_fin/30, t_fin/10, t_fin/5, 0.7*t_fin, 0.9*t_fin, ...
               t_fin],[0, p_max/40, p_max*3/4, p_max, p_max, 0, 0],...
               t,'pchip');                       %Velocità angolare di rollio in rad/s

q=@(t) interp1([0, t_fin/30, t_fin/10, t_fin/5, 0.8*t_fin, 0.9*t_fin, ...
               t_fin],[0, q_max/40, q_max*3/4, q_max, q_max, 0, 0],...
               t,'pchip');                       %Velocità angolare di rollio in rad/s

r=@(t) 0*(t);                                   %Velocità angolare di imbardata in rad/s

%% Leggi temporali delle componenti della velocità di G in BRF

u0=convvel(380.0, 'Km/h','m/s');
v0=convvel(0.0, 'Km/h','m/s');
w0=convvel(0.0, 'Km/h','m/s');

V0=sqrt(u0*u0+v0*v0+w0*w0);                     %modulo della velocità

u=@(t)interp1([0, t_fin/30, t_fin/10, t_fin/5, 0.8*t_fin,t_fin],...
              [u0, 0.8*u0, 0.7*u0, u0, 1.1*u0, u0],...
              t,'pchip');

v=@(t) 0*(t);
w=@(t) 0*(t);
```

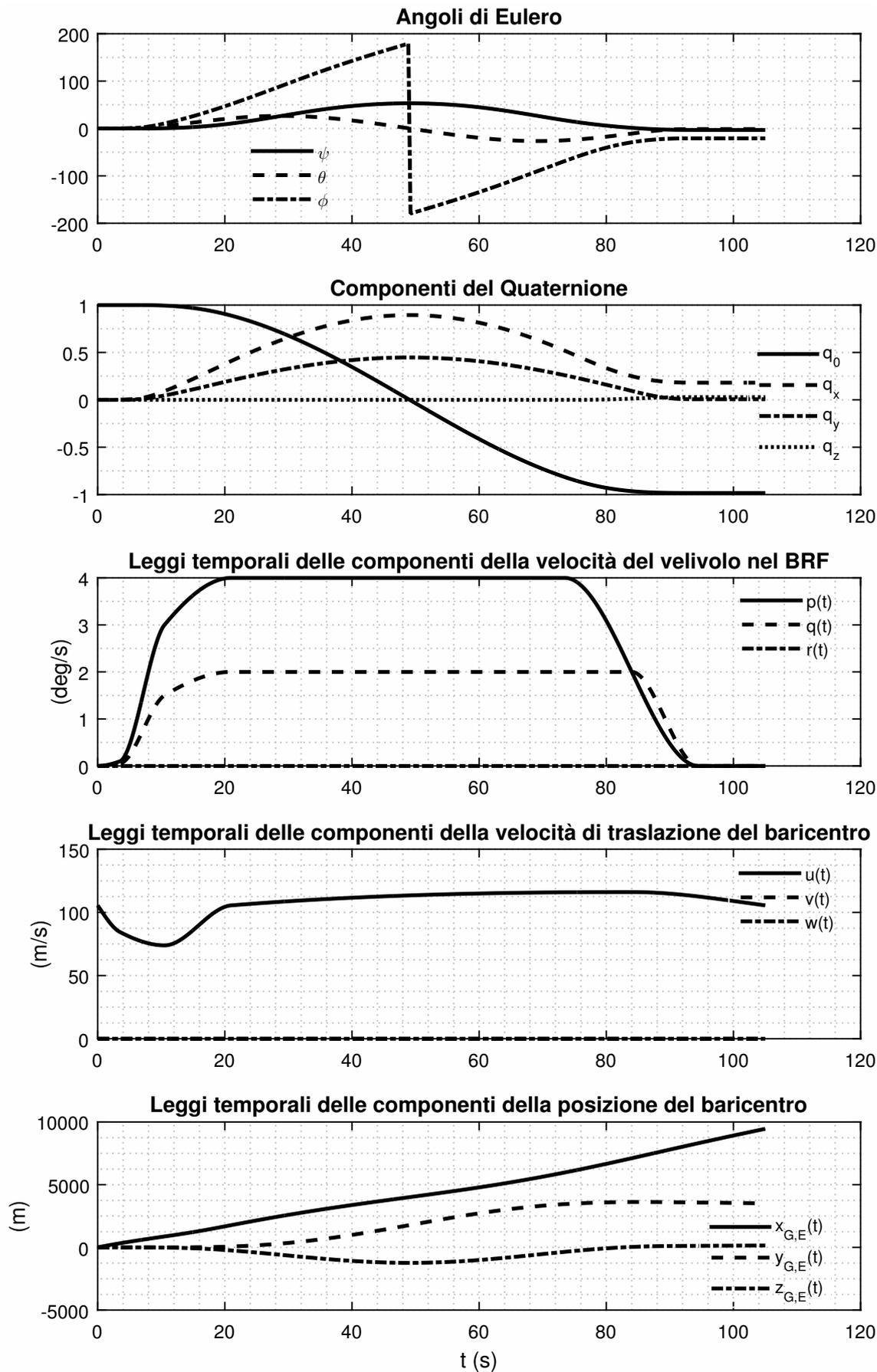


Figura 3.7 Time history dei parametri per la manovra di *Tonneau*, per le leggi temporali assegnate nel listato 3.6 nella pagina precedente.

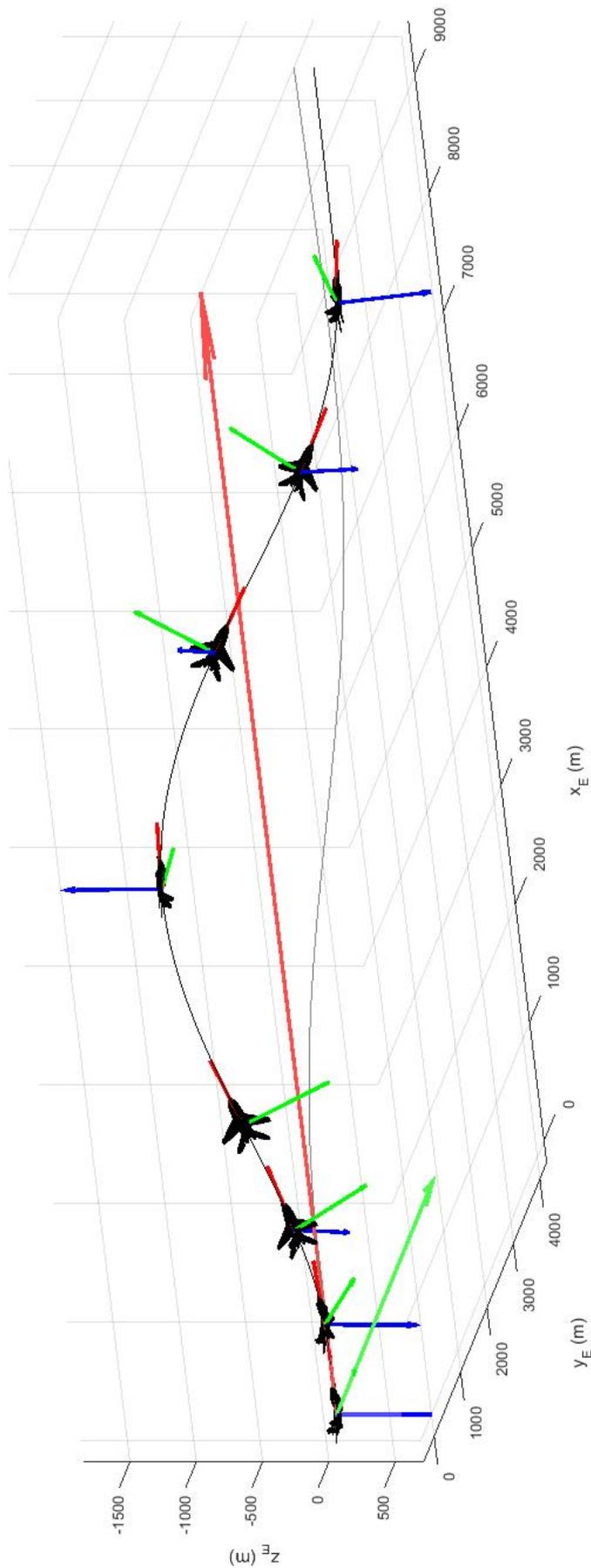


Figura 3.8 Rappresentazione della manovra di *tonneau* per un orientamento iniziale definito dalle condizioni $\psi = \theta = \phi = 0$ e per le leggi di variazione delle componenti della velocità lineare e della velocità angolare rappresentate nel listato 3.6 a pagina 49. Il velivolo non è in scala.

3.6 Manovra di *Eight* - Esercizio 3.4

Di seguito è proposta una possibile manovra di *eight*, sono riportate nel listato 3.7 le leggi temporali che ci hanno permesso di realizzare tale manovra. Come si evince dal listato non è stato riportata la parte di codice relativa al plot, perchè già presentata in precedenza e per questo ritenuta di scarso interesse.

Listato 3.7 Leggi temporali per la manovra di Eight

```
%% Funzioni Anonime

%Valori di riferimento per costruire la manovra dell'otto per p,q,r
p_max=convangvel(0.0,'deg/s','rad/s');           % rad/s
q_max=convangvel(2.0,'deg/s','rad/s');           % rad/s
r_max=convangvel(0.0,'deg/s','rad/s');           % rad/s

%Legge della p
vBreakPointsP(1,:)=[0,t_fin*0.2,t_fin*0.4,t_fin*0.6,0.8*t_fin,0.9*t_fin,...
    t_fin];
vBreakPointsP(2,:)=[0, 0, 0, 0,0,0, 0];
p=@(t)interp1(vBreakPointsP(1,:),vBreakPointsP(2,:),t,'pchip');

%legge della q
vBreakPointsQ(1,:)=[0,t_fin*0.1, t_fin*0.2,t_fin*0.3,t_fin*0.4,t_fin*0.5,...
    t_fin*0.6,t_fin*0.7,t_fin*0.8,t_fin*0.9,t_fin];
vBreakPointsQ(2,:)=[0,q_max*0.61,q_max*0.88,q_max,-q_max*0.5,-q_max,...
    -q_max,-q_max*0.82,q_max*0.77,0,0];
q=@(t)interp1(vBreakPointsQ(1,:),vBreakPointsQ(2,:),t,'pchip');

%legge della r
r = @(t) 0;

%Valori di riferimento per costruire la manovra dell'otto per u,v,w

u0 = convvel(380.0,'km/h','m/s'); % m/s
v0 = convvel( 0.0,'km/h','m/s'); % m/s
w0 = convvel( 0.0,'km/h','m/s'); % m/s

vBreakPointsU(1,:)=[0,t_fin/30,t_fin/10,t_fin/5,0.8*t_fin,t_fin];
vBreakPointsU(2,:)=[u0,0.8*u0,0.7*u0,u0,1.1*u0,u0];
u=@(t)interp1(vBreakPointsU(1,:),vBreakPointsU(2,:),t,'pchip');

% assume alpha = beta = 0
v=@(t)0;
w=@(t)0;
```

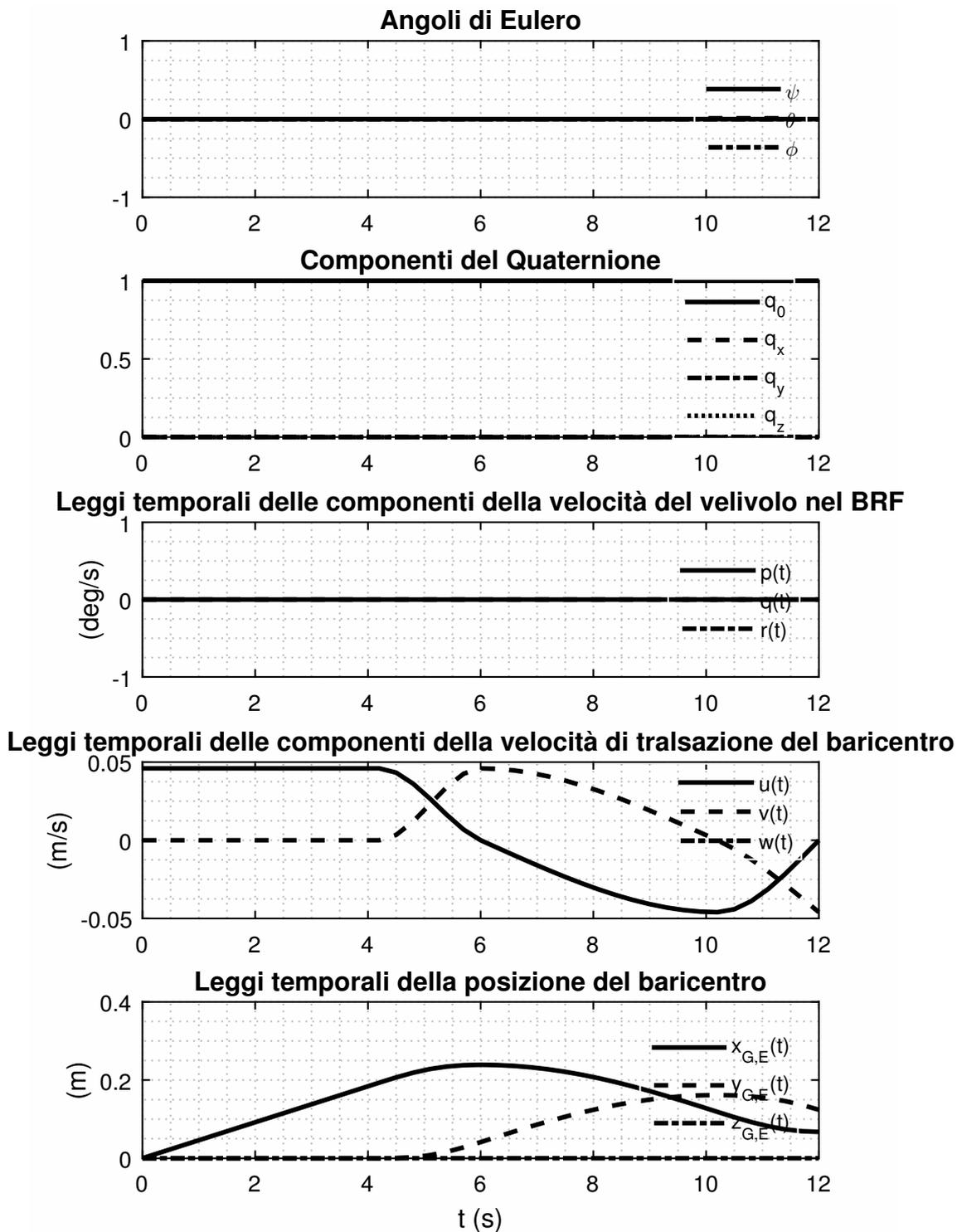


Figura 3.9 Time history dei parametri per la manovra di *Eight*, per le leggi temporali assegnate nel listato 3.7 a fronte $\psi = \theta = \phi = 0$.

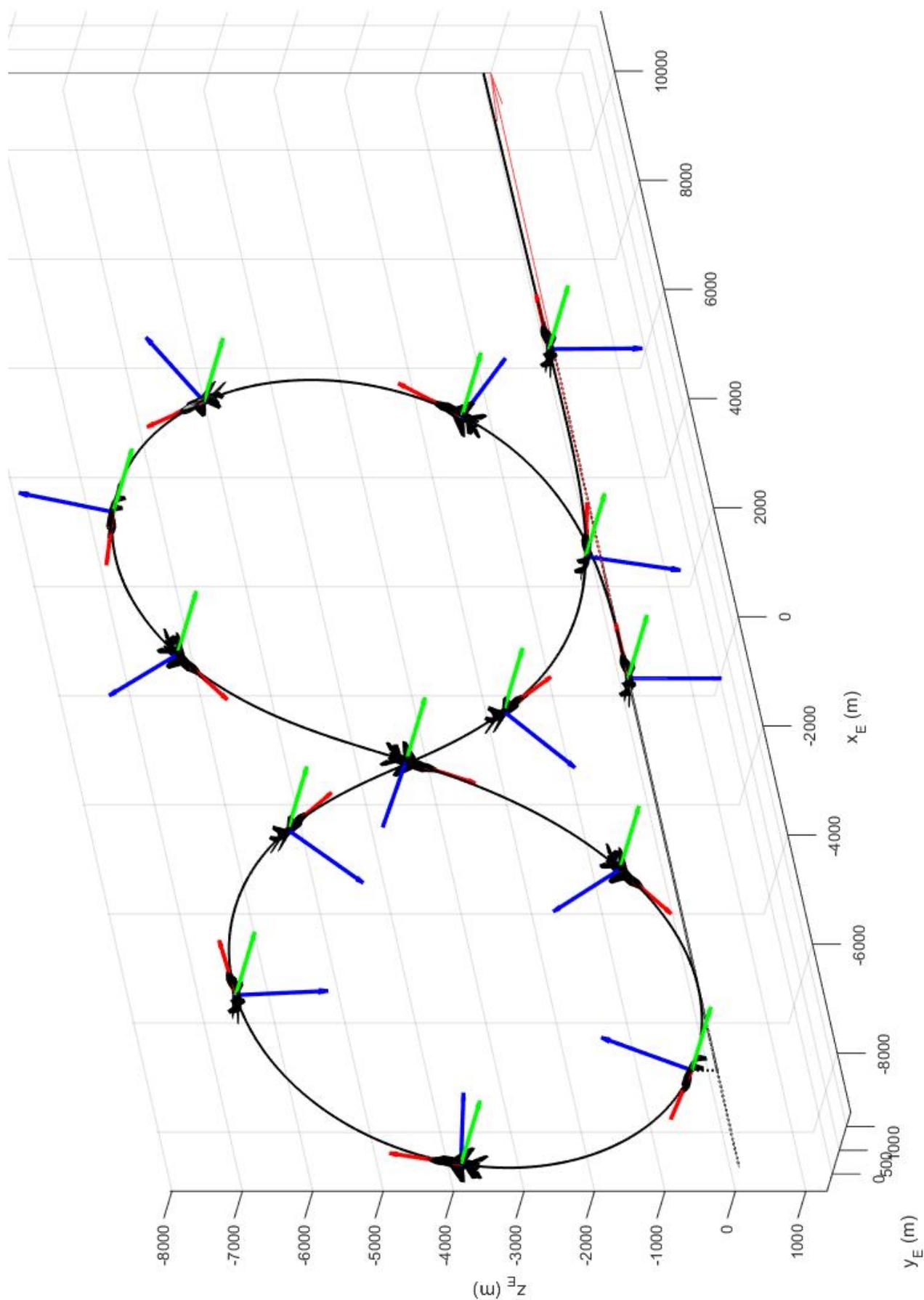


Figura 3.10 Rappresentazione della manovra di *Eight* per un orientamento iniziale definito dalle condizioni $\psi = \theta = \phi = 0$, e per le leggi di variazione delle componenti della velocità lineare e della velocità angolare rappresentate nel listato 3.7 a pagina 52. Il velivolo non è in scala.

3.7 Manovra di Spirale - Esercizio 3.4

Di seguito è proposta una possibile manovra di *spirale*, sono riportate nel listato 3.8 le leggi temporali che ci hanno permesso di realizzare tale manovra. Come si evince non è stata riportata la parte di codice relativa al plot, perché è stata già presentata precedentemente e quindi ritenuta di scarso interesse.

Listato 3.8 Leggi temporali per la manovra di Spirale

```
%% Funzioni anonime

p_max = convangvel(1.5,'deg/s','rad/s'); % rad/s
q_max = convangvel(2.5,'deg/s','rad/s'); % rad/s
r_max = convangvel(3.0,'deg/s','rad/s'); % rad/s

vBreakPointsP(1,:) = [0, t_fin*0.1, t_fin*0.2,t_fin*0.3,t_fin*0.4, t_fin*0.5,t_fin
    *0.6, t_fin*0.7, t_fin*0.8,t_fin*0.9, t_fin];
vBreakPointsP(2,:) = [p_max, p_max*0.5,p_max*0.5,p_max*0.5,p_max,p_max*0.8,0,0, 0,
    0,0];
p = @(t) interp1( vBreakPointsP(1,:), vBreakPointsP(2,:), t, 'pchip' );

vBreakPointsQ(1,:) = [0, t_fin*0.1, t_fin*0.2,t_fin*0.3,t_fin*0.4, t_fin*0.5,t_fin
    *0.6, t_fin*0.7, t_fin*0.8,t_fin*0.9, t_fin];
vBreakPointsQ(2,:) = [0, 0, q_max, q_max, q_max,q_max,q_max*0.7 ,0,0, 0,0];
q = @(t) interp1( vBreakPointsQ(1,:), vBreakPointsQ(2,:), t, 'pchip' );

vBreakPointsR(1,:) = [0, t_fin*0.1, t_fin*0.2,t_fin*0.3,t_fin*0.4, t_fin*0.5,t_fin
    *0.6, t_fin*0.7, t_fin*0.8,t_fin*0.9, t_fin];
vBreakPointsR(2,:) = [0,0,r_max,r_max,r_max, r_max,r_max,0, 0,0,0];
r = @(t) interp1( vBreakPointsR(1,:), vBreakPointsR(2,:), t, 'pchip' );

u0 = convvel(380.0,'km/h','m/s'); % m/s
v0 = convvel( 0.0,'km/h','m/s'); % m/s
w0 = convvel( 0.0,'km/h','m/s'); % m/s

vBreakPointsU(1,:) = [0, t_fin/30, t_fin/10, t_fin/5, 0.8*t_fin, t_fin];
vBreakPointsU(2,:) = [u0, 0.8*u0, 0.7*u0, u0, 1.1*u0, u0];

u = @(t)interp1( vBreakPointsU(1,:), vBreakPointsU(2,:), t, 'pchip' );

v = @(t) 0;
w = @(t) 0;
```

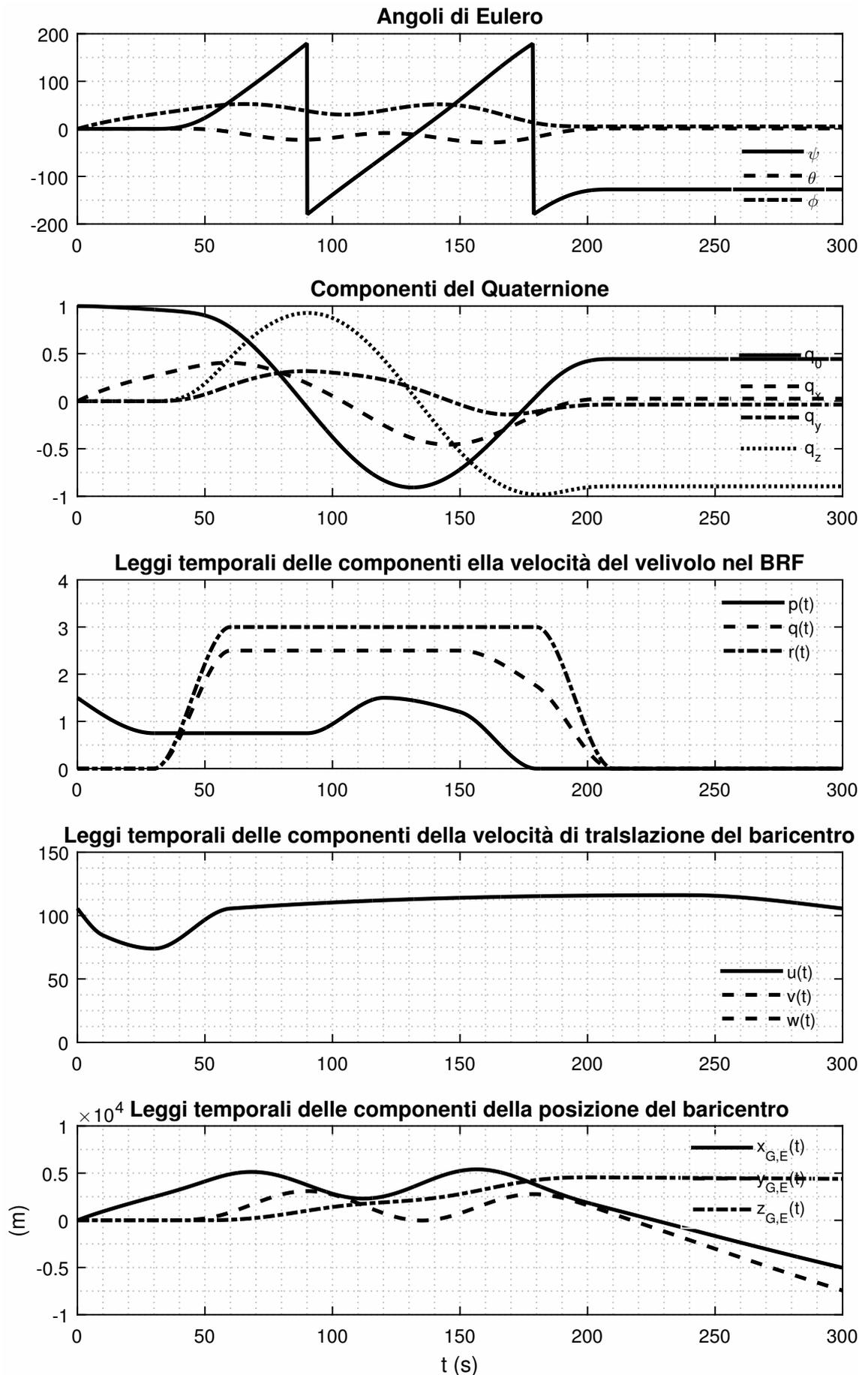


Figura 3.11 Storie temporali delle componenti del quaternione dell'orientamento e degli angoli di Eulero corrispondenti alla manovra di *Eight* riportata nel listato 3.7 a pagina 52, per un orientamento iniziale definito dalle condizioni $\psi = \theta = \phi = 0$, Il velivolo non è in scala.

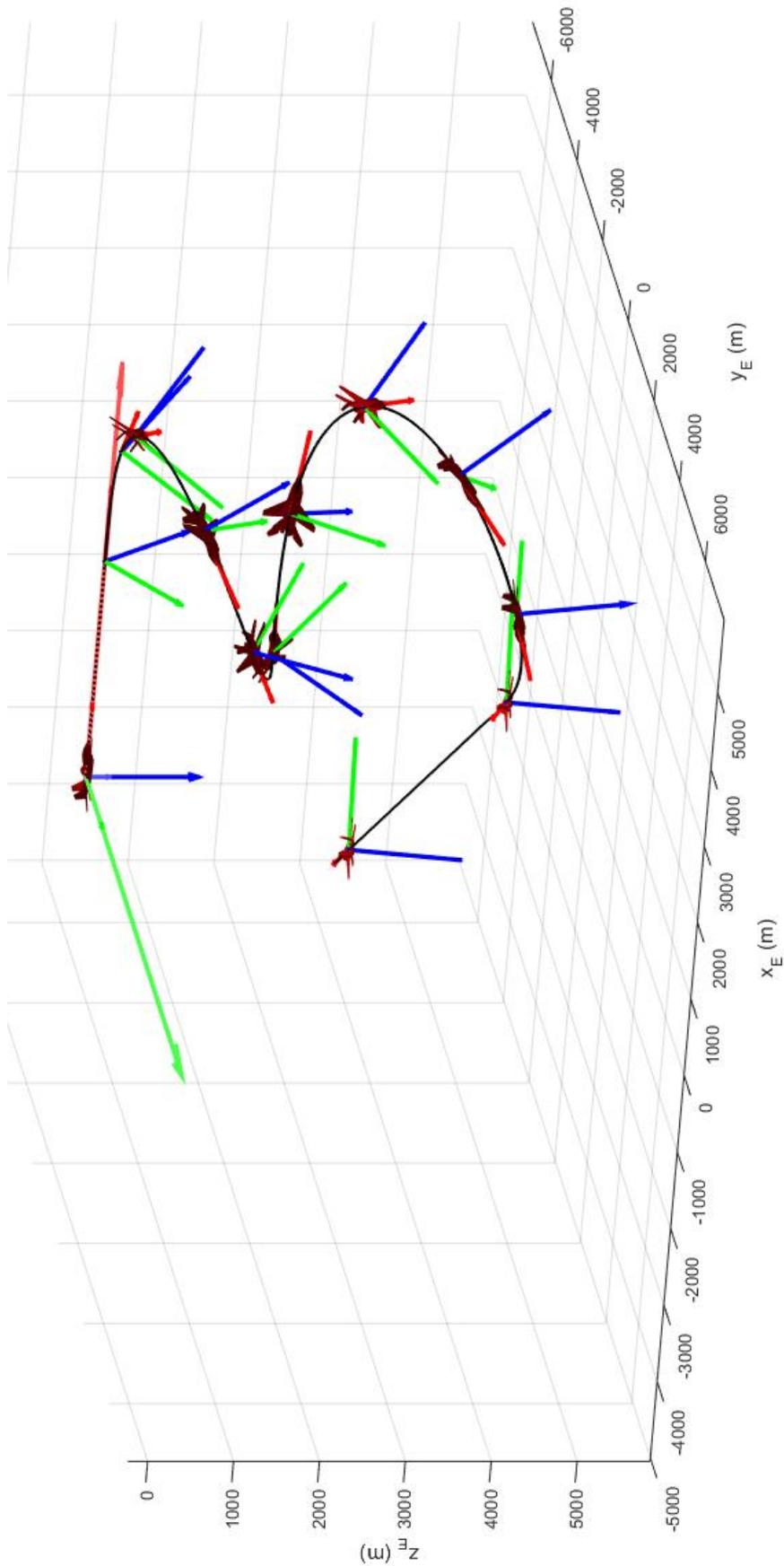


Figura 3.12 Rappresentazione della manovra di *Spirale* per un orientamento iniziale definito dalle condizioni $\psi = \theta = \phi = 0$, e per le leggi di variazionede delle componenti della velocità lineare e della velocità angolare rappresentate nella figura 3.11 nella pagina precedente, Il velivolo non è in scala.

3.8 Manovra di *Looping* eseguita in Simulink - *Esercizio 3.5*

Sono stati riproposti in Simulink alcuni esercizi svolti precedentemente. L'obiettivo di questa esercitazione è stato quello di imparare a risolvere i sistemi dinamici mediante la nuova interfaccia grafica di Matlab. Nella risoluzione di tali sistemi l'utente è agevolato grazie alla presenza di *toolboxes* pensati ed implementati appositamente per particolari classi di problemi. Infatti, nel campo aerospaziale ed in particolare nella dinamica del volo un *toolbox* molto importante è l'*Aerospace Blockset* per l'analisi e l'integrazione di sistemi dinamici che modellano il funzionamento di velivoli di vario genere di sistemi propulsivi. Per tale esercitazione si è scelto di studiare la manovra del *Looping* svolta precedentemente nell'esercizio 3.1. In tal caso si è considerato il diagramma a blocchi fornito dal docente nel Quaderno 3.

Per rendere eseguibile il diagramma a blocchi è stato necessario fornire degli input, che sono stati passati tramite uno script Matlab. Quindi prima di avviare la simulazione in Simulink si è reso necessario eseguire prima lo script in Matlab, si veda il listato 3.9, contenente valori iniziali dell'orientamento, dell'altitudine e della velocità.

Listato 3.9 Setup di Simulink della manovra di Looping

```
clear all; close all; clc;
%% initial position, near KSFO
altitude0_ft = 0.0;
altitude0_mt = convlength(altitude0_ft, 'ft', 'm');
latitude0_deg = 37.66553509;
latitude0_rad = convang(latitude0_deg, 'deg', 'rad');
longitude0_deg = -122.4808986;
longitude0_rad = convang(longitude0_deg, 'deg', 'rad');
heading0_deg = 0.0;
heading0_rad = convang(heading0_deg, 'deg', 'rad');
pitch0_deg = 0.;
pitch0_rad = convang(pitch0_deg, 'deg', 'rad');
roll0_deg = 0.0;
roll0_rad = convang(roll0_deg, 'deg', 'rad');

%% Initial quaternion components
q_max=6.28; %[deg/s]
psi0 = 0; % rad
theta0 = 0; % rad
phi0 = 0; %rad
quat0 = angle2quat(psi0, theta0, phi0);

%% Speed
speed_km_h = 250; %Km/h
speed_ms = convvel(speed_km_h, 'km/h', 'm/s');

vN0 = speed_ms*cos(heading0_rad);
vE0 = speed_ms*sin(heading0_rad);
```

Solo dopo aver fornito i dati di input è stato possibile avviare la simulazione con Simulink. L'output del sistema è visualizzabile all'interno del blocco *Scope*. Per fornire chiarezza nell'esposizione l'output contenuto nel blocco *scope* è stato riportato nelle figure 3.15 a pagina 61, 3.16 a pagina 62, e 3.21 a pagina 68.

Infine si è eseguito un ulteriore script di Matlab, presentato nel listato 3.10 a pagina 60 in cui sono stati definiti i parametri preferenziali per la visualizzazione della traiettoria

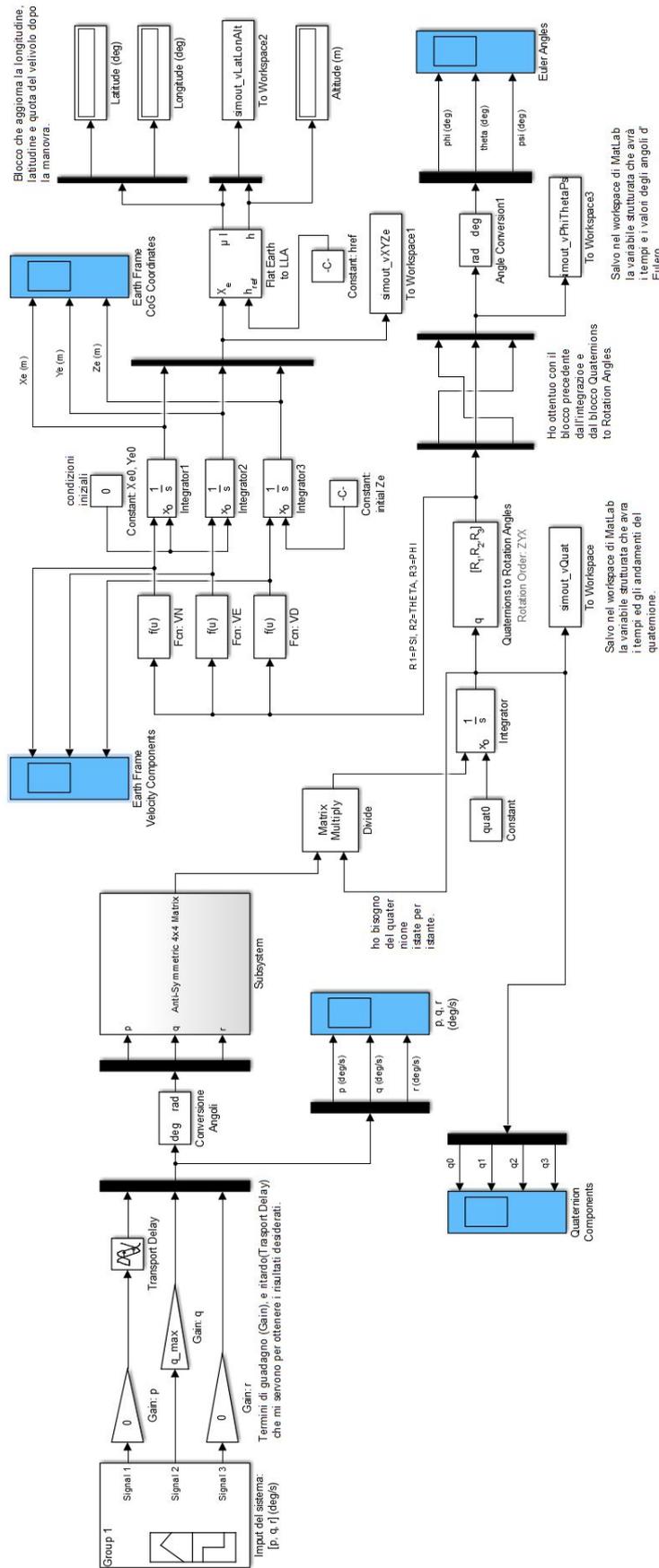


Figura 3.13 Aspetto del diagramma a blocchi del modello *Simulink* considerato.

della manovra che è stata riportata per completezza nella figura ?? a pagina ??.

Listato 3.10 Postprocess di Simulink della manovra di Looping

```
%postprocess_manovra_Looping
%=====
% run Simulink model: quaternion_and_trajectory_1.slx
%=====
% when Simulink model: quaternion_and_trajectory_1.slx
% is done, uncomment and plot 3D trajectory
% or
% load 'postprocess_animation.mat'
vQuat = simout_vQuat.Data;
vXe = simout_vXYZe.Data(:,1);
vYe = simout_vXYZe.Data(:,2);
vZe = simout_vXYZe.Data(:,3);
h_fig3D = figure(1);
theView = [105 15];
plotTrajectoryAndBody2(h_fig3D, ...
    vXe, vYe, vZe, vQuat, 0.006, ...
    [1,30,60,75,83,90,100,108], ...
    theView)
grid on;
```

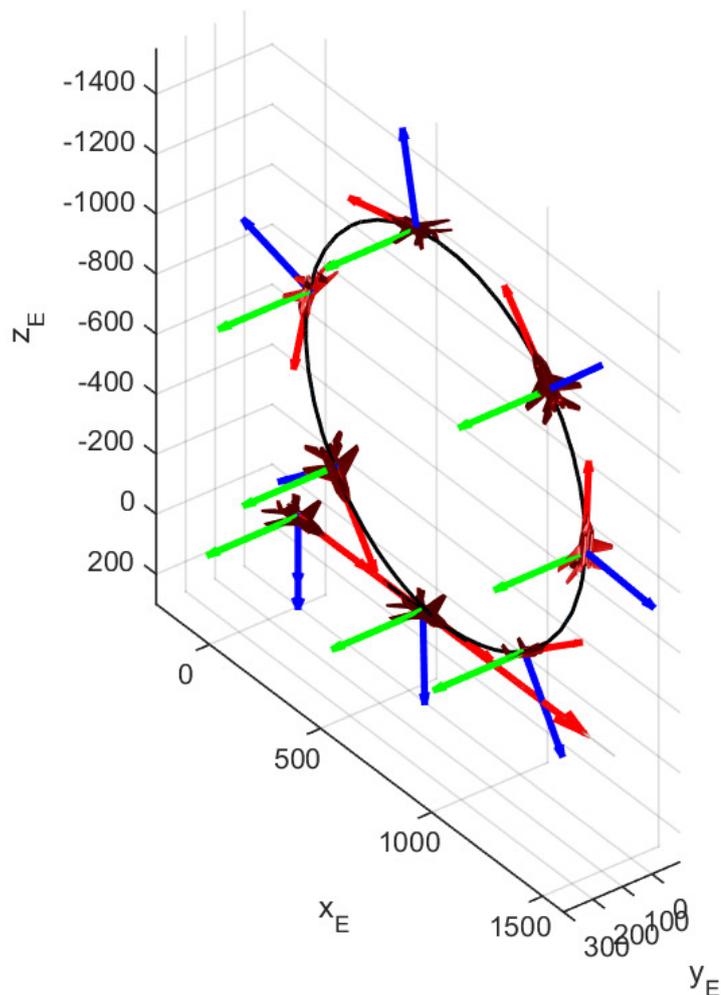


Figura 3.14 Evoluzione di *Looping* ottenuta con il modello *Simulink*

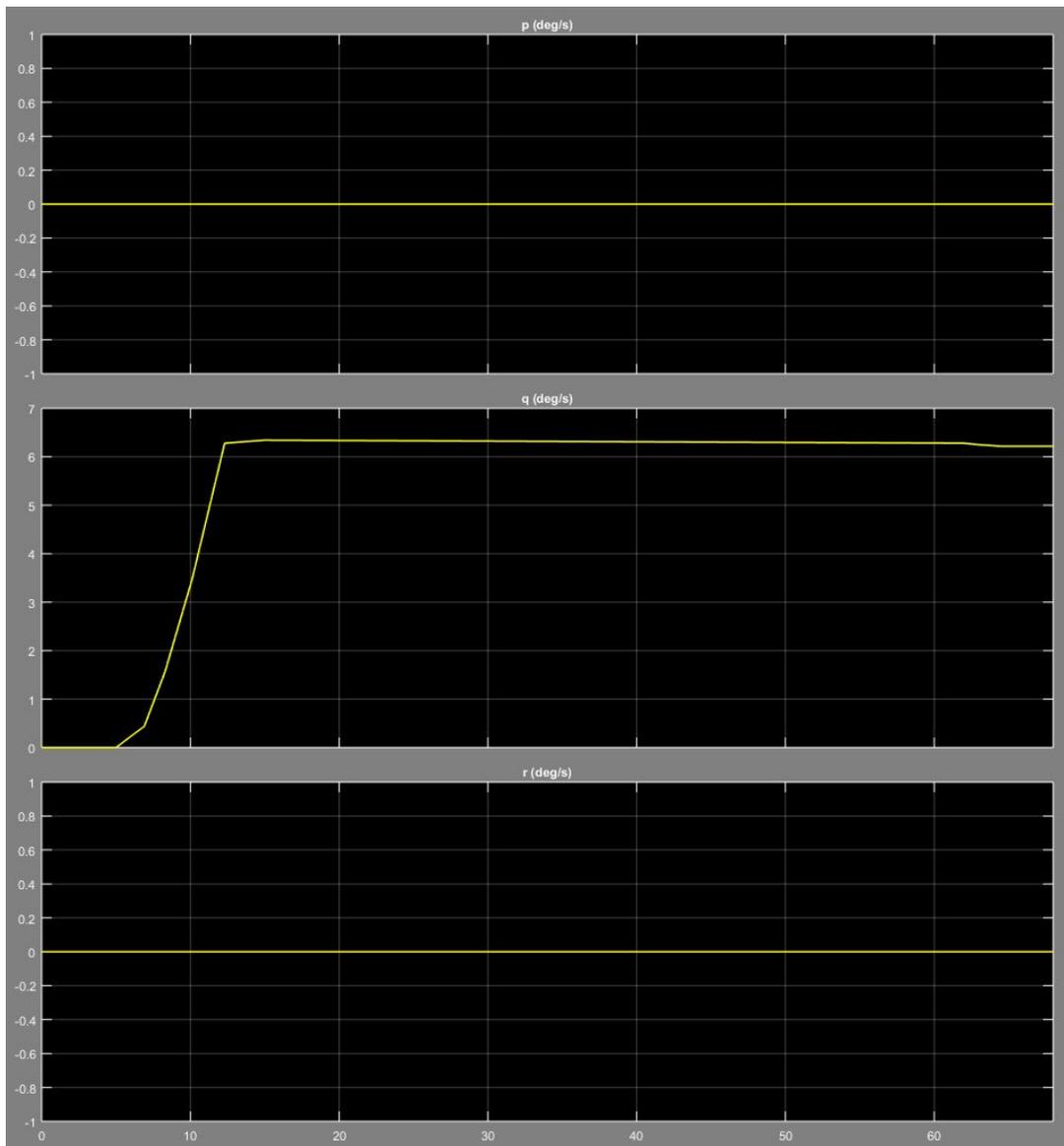


Figura 3.15 Storie temporali effettive delle componenti di velocità angolari in assi vento, Blocco *Scope*, per la manovra di *Looping*.

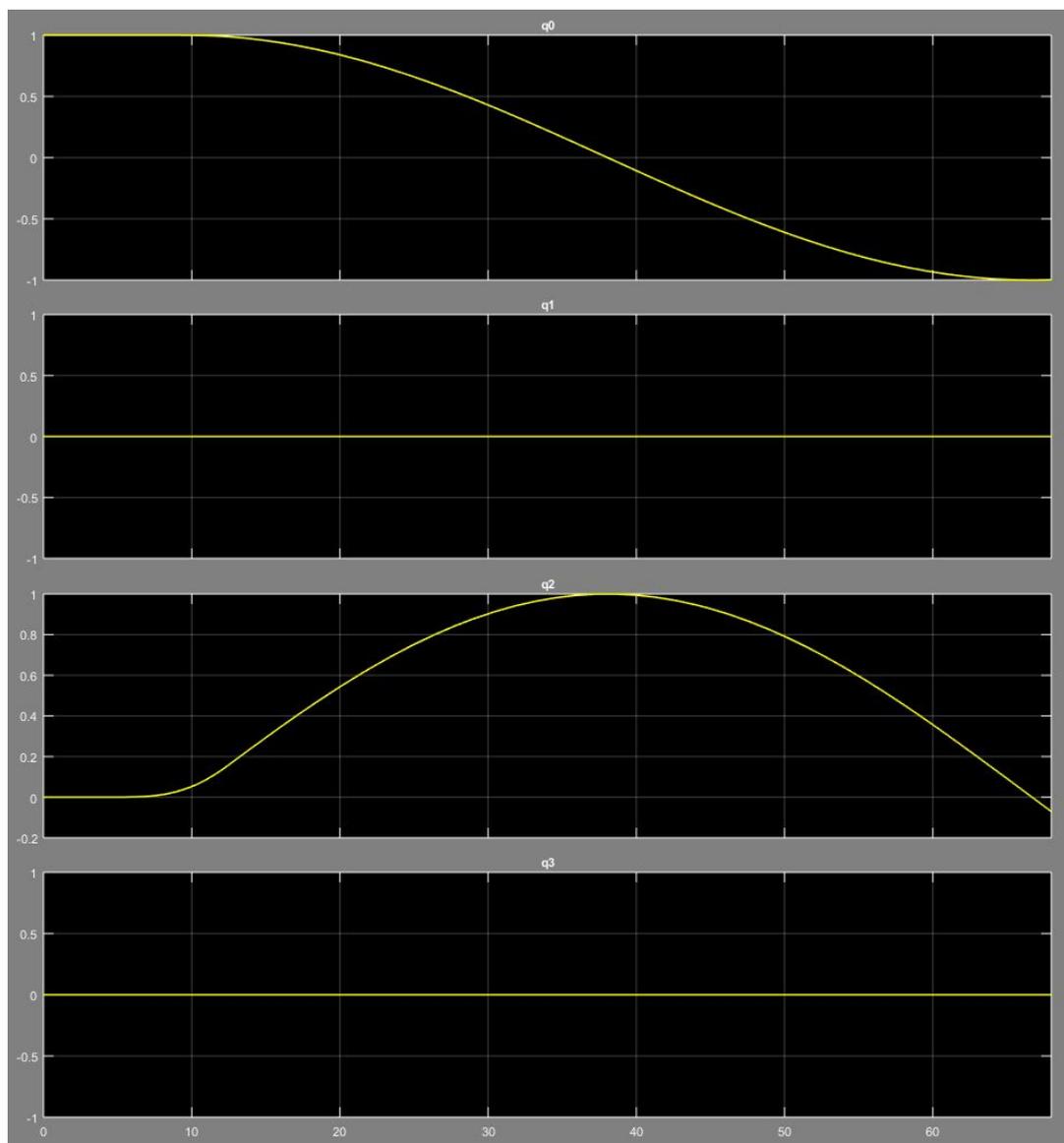


Figura 3.16 Storie temporali delle componenti del quaternione dell'orientamento, Blocco *Scope*, per la manovra di *Looping*

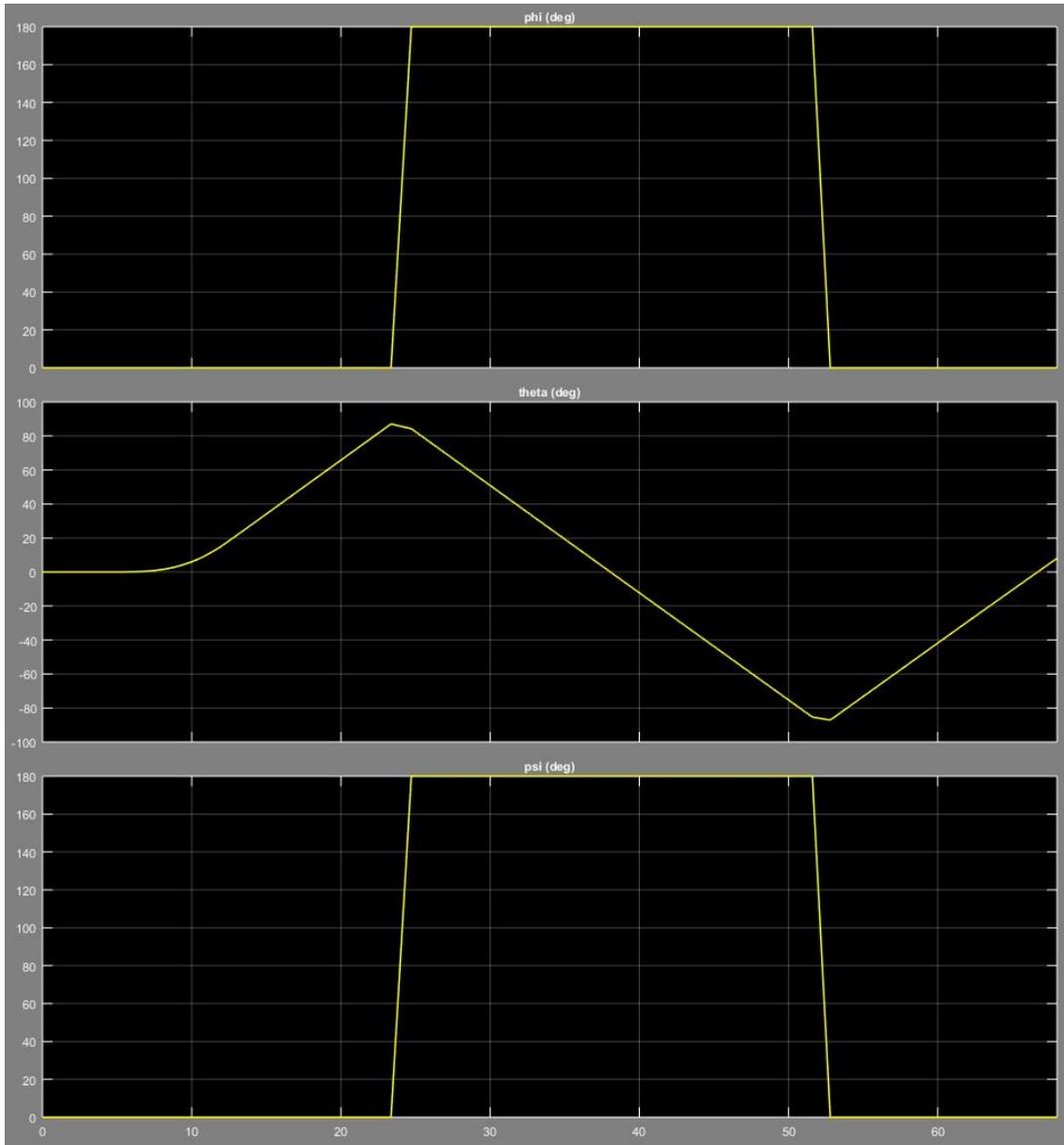


Figura 3.17 Storie temporali degli angoli d'Eulero, Blocco *Scope*, per la manovra di *Looping*.

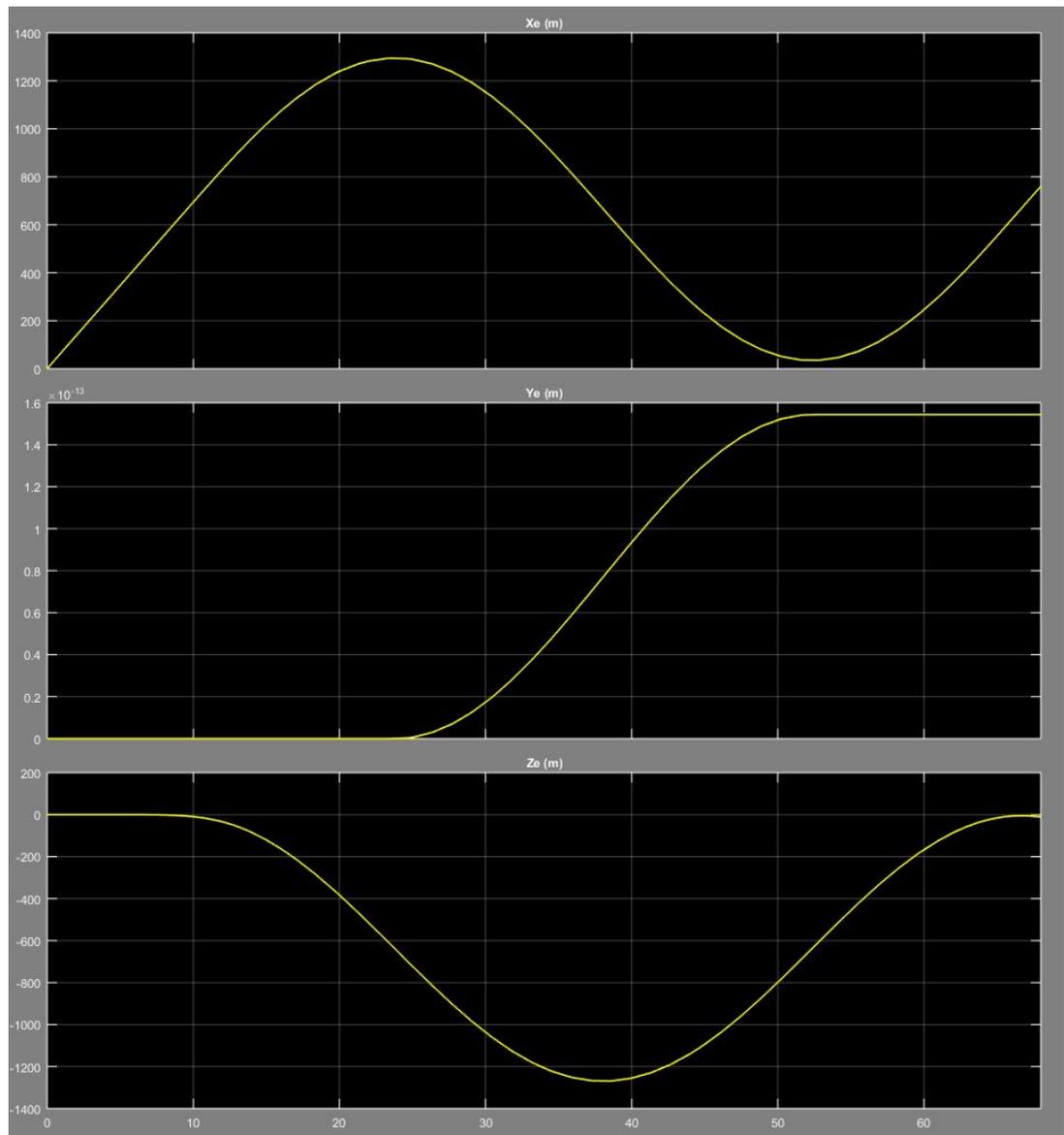


Figura 3.18 Storie temporali delle componenti delle coordinate del baricentro, Blocco *Scope*, per la manovra di *Looping*.

3.9 Acquisizioni leggi temporali mediante sensori dello smartphone

Con la seguente esercitazione si sono volute ricavare delle leggi temporali di u , v , w e di p , q , r mediante acquisizione dati provenienti da uno smartphone posto a registrare. Tale esercizio è stato svolto per meglio capire il funzionamento delle leggi temporali, infatti, prima di effettuare l'acquisizione dati, è stata pensata una plausibile traiettoria. A valle dell'acquisizione si è ricercato nelle leggi temporali prodotte quanto pensato prima della simulazione.

E' stata selezionata una strada rettilinea che puntava il più possibile verso Nord e che presentasse dei dossi abbastanza rilevanti. La prova è stata condotta partendo da una velocità nulla fino al raggiungimento di una velocità di circa $13m/s$, la quale è stata mantenuta costante per tutto il corso dell'acquisizione dati, mediante il navigatore cruise dell'autovettura.

Sono stati ricercati sul tracciato i dossi così da simulare delle forzanti al sistema che potessero andare a scaturire variazioni sulla dinamica di beccheggio.

Le leggi temporali sono state acquisite dallo smartphone mediante la generazione di file .csv, un esempio di tale file è riportato nella figura 3.20 a pagina 67.

Come si evince dal plot dell'immagine 3.20 a pagina 67 gli angoli ϕ , θ , ψ restano circa costanti durante tutto lo svolgimento della manovra, fatta eccezione della fase iniziale della manovra. Infatti, si evince una variazione dell'angolo di ϕ proprio nella fase iniziale della manovra, questo evento è dipeso dal fatto che nel momento in cui si è avviata la simulazione sullo smartphone, si è impegnata la corsia abbandonando l'area di sosta. Questa manovra ha quindi prodotto una variazione iniziale dell'angolo di pria.

Si evince come nelle fasi iniziali della manovra, mentre il veicolo raggiungeva la velocità di crociera impostata la velocità cresce per poi attestarsi intorno ad un valore costante e per poi tornare a decrescere nella fase finale della manovra.

Durante la fase di manovra si evincono tre picchi riportati dalle leggi temporali delle componenti del vettore velocità angolare istantanea, questi picchi sono stati voluti e sono da attribuire a 3 dossi ricercati durante la manovra. Si nota come la dinamica rotazionale, nell'incontrare i dossi, sia influenzata principalmente lungo l'asse di beccheggio e lungo l'asse di rollio.

La parte di script fornita dal docente è rimasta inalterata, si è provveduto solamnte ad ingolbare all'interno dello script le leggi del moto acquisite mediante lo smartphone. Di seguito è riportato il listato contenente il frammento di codice demandato al compito di input nello script.

Listato 3.11 Leggi temporali acquisite mediante smartphone

```
%Condizioni Iniziali
psi0=motionYawrad(1);
theta0=motionPitchrad(1);
phi0=motionRollrad(1);

Q0=angle2quat(psi0,theta0,phi0);           %sequenza(yaw,pitch,roll)

x0=0; y0=0; z0=0;                         %coordinate baricentro iniziali
PosE0=[x0,y0,z0];                         %coordinate baricentro iniziali

% ti=TimeLogging(1);                       %tempo acquisizione iniziali
```

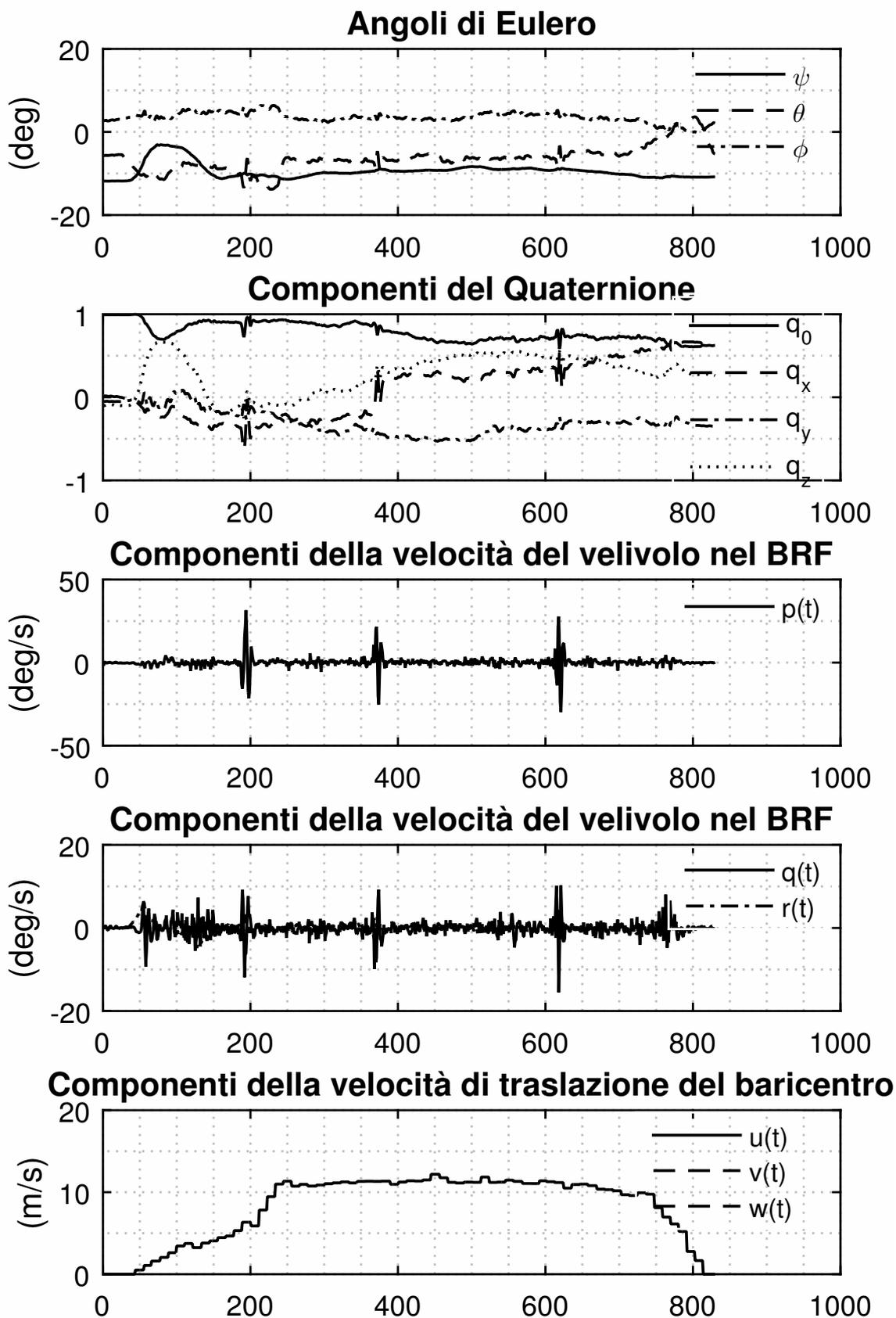


Figura 3.19 Time histories acquisite mediante smartphone.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	loggingTime(txt)	loggingSample(N)	identifierForVendor(txt)	locationTimestamp_since1970(s)	locationLatitude(WGS84)	locationLongitude(WGS84)	locationAltitude(m)	locationSpeed(m/s)	locationCourse(°)	locationSpeed(m/s)	locationCourse(°)	locationAltitude(m)	locationSpeed(m/s)	locationCourse(°)	locationAltitude(m)	locationSpeed(m/s)	locationCourse(°)
2	2017-10-28 07:55:09.297	+0200,1,F7B87B5B-E9C8-4CAC-BB51-3E4CBAC8154D,1509170108.999996,40.83214541900981,14.31224467234156,3.090886,0,135,3,5,-9999,1509170109.25715,-18.3271560668															
3	2017-10-28 07:55:09.407	+0200,2,F7B87B5B-E9C8-4CAC-BB51-3E4CBAC8154D,1509170108.999996,40.83214541900981,14.31224467234156,3.090886,0,135,3,5,-9999,1509170109.375142,-19.452022552															
4	2017-10-28 07:55:09.487	+0200,3,F7B87B5B-E9C8-4CAC-BB51-3E4CBAC8154D,1509170108.999996,40.83214541900981,14.31224467234156,3.090886,0,135,3,5,-9999,1509170109.375142,-19.452022552															
5	2017-10-28 07:55:09.567	+0200,4,F7B87B5B-E9C8-4CAC-BB51-3E4CBAC8154D,1509170108.999996,40.83214541900981,14.31224467234156,3.090886,0,135,3,5,-9999,1509170109.493046,-18.986553192															
6	2017-10-28 07:55:09.648	+0200,5,F7B87B5B-E9C8-4CAC-BB51-3E4CBAC8154D,1509170108.999996,40.83214541900981,14.31224467234156,3.090886,0,135,3,5,-9999,1509170109.612876,-18.986560821															
7	2017-10-28 07:55:09.737	+0200,6,F7B87B5B-E9C8-4CAC-BB51-3E4CBAC8154D,1509170108.999996,40.83214541900981,14.31224467234156,3.090886,0,135,3,5,-9999,1509170109.730245,-18.042701721															
8	2017-10-28 07:55:09.826	+0200,7,F7B87B5B-E9C8-4CAC-BB51-3E4CBAC8154D,1509170108.999996,40.83214541900981,14.31224467234156,3.090886,0,135,3,5,-9999,1509170109.730245,-18.042701721															

Figura 3.20 File.csv generato dallo smartphone

```
% tf=TimeLogging(end);           %tempo acquisizione finale
% tt=linspace(ti,tf,1e3);       %tempo totale

%% Leggi degli angoli di Eulero
p=@(t)interp1(TimeLogging,motionRotationRateXrads,t,'pchip');
q=@(t)interp1(TimeLogging,motionRotationRateYrads,t,'pchip');
r=@(t)interp1(TimeLogging,motionRotationRateZrads,t,'pchip');

%% Leggi temporali delle componenti della velocità di traslazione di G inBRF
u=@(t)interp1(TimeLogging,locationSpeedms,t,'pchip');
v=@(t)0;
w=@(t)0;
```

Per meglio visualizzare le leggi acquisite mediante lo smartphone si è provveduto ad eliminare le code del file csv, ovvero quelle contenenti la fase iniziale di accelerazione lineare del baricentro del veicolo e quella finale di decelerazione. Il tutto è stato fatto per visualizzare meglio la fase di traiettoria rettilinea uniforme.

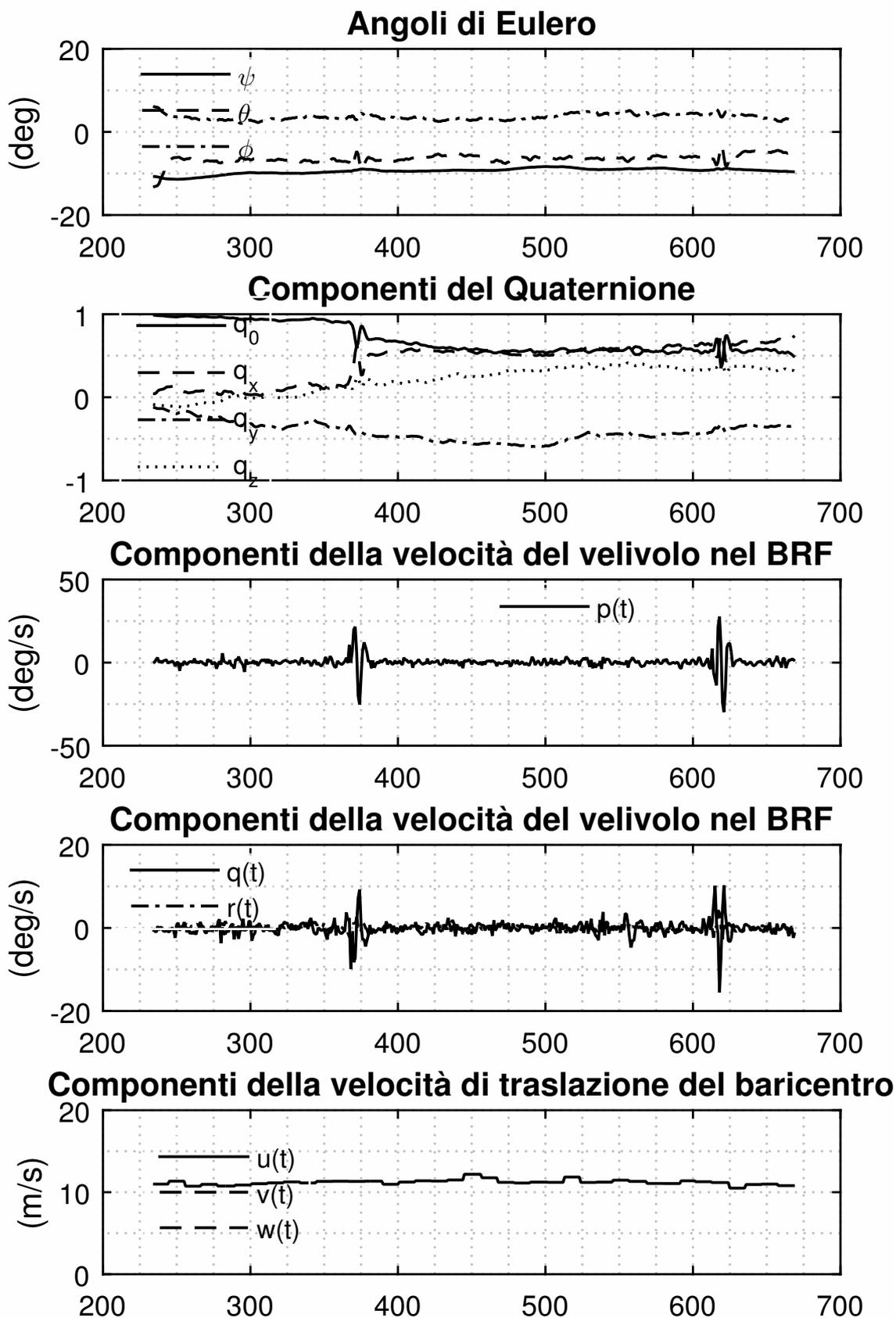


Figura 3.21 Time histories acquisite mediante smartphone fase costante

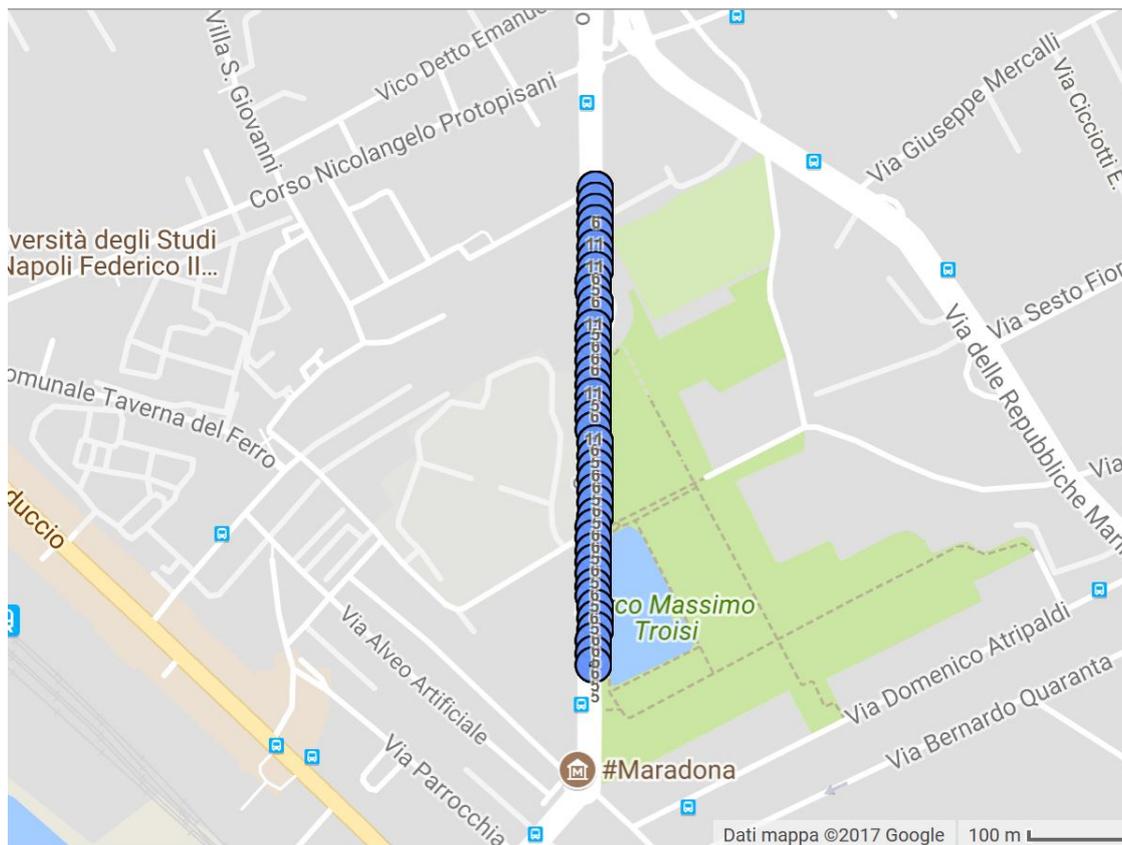


Figura 3.22 Traiettoria riportata mediante le coordinate di latitudine e longitudine

Moto longitudinal-simmetrico a comandi bloccati

Indice

7.1	Introduzione	71
7.2	Impostazione del problema	73
7.3	Il modello aerodinamico e propulsivo	74
7.4	Ricerca delle condizioni di trim	75
7.5	Ricerca delle condizioni di trim - 3-Dof	78
7.5.1	Codice per la ricerca delle condizioni di trim - <i>Esercizio 7.6</i>	78
7.5.2	Nuova condizione di Trim per $\delta_s = -1deg$ - <i>Esercizio 7.7</i>	83
7.5.3	Verifica delle condizioni di trim - <i>Esercizio 7.8</i>	86
7.6	Simulazione legge <i>cabra-picchia</i> - <i>Esercizio 7.9</i>	94
7.7	Variazione della legge del δ_c - <i>Esercizio 7.10</i>	101

7.1 Introduzione

Lo scopo di questa esercitazione è la rappresentazione del moto longitudinal-simmetrico a comandi bloccati di un velivolo rigido.

Nell'ipotesi di comandi bloccati, com'è noto, il pilota esercita delle forze sui comandi di volo che attraverso opportune catene cinematiche si traducono in momenti di cerniera sulle superfici di governo, detti *momenti di cerniera di comando*, i quali contrastano i momenti aerodinamici, dovuti all'interazione con la corrente, in modo da mantenere la superficie stessa in una desiderata posizione angolare. Siccome è impedita ogni escursione angolare, risulta che la configurazione del velivolo è congelata e non si devono contemplare ulteriori gradi di libertà ai sei di corpo rigido.

Adottare l'ipotesi di moto longitudinal-simmetrico ci consente di ridurre ulteriormente il numero di gradi di libertà che passano da sei a tre perché si suppone il velivolo simmetrico rispetto al piano longitudinale e la sua evoluzione contenuta interamente

in tale piano (si escludono rotazioni attorno ad assi nel piano di simmetria e traslazioni nella direzione ortogonale ad esso). Si parla allora di moto a tre gradi di libertà (*3-Degrees-of-freedom-Motion*, 3-DoF motion).

La determinazione del moto longitudinal-simmetrico a partire da assegnate condizioni iniziali, dunque, passa per la risoluzione di un sistema di tre equazioni differenziali, che esprimono l'equilibrio dinamico del sistema velivolo, considerato come corpo rigido, secondo i tre gradi di libertà che lo caratterizzano e che discendono direttamente dalle Equazioni Cardinali della Dinamica, (ECD).

L'approccio classico consiste nel proiettare le equazioni vettoriali lungo gli assi velivolo (*assi body*), ottenendo un sistema di equazioni scalari che, però da sole, non bastano per la determinazione del moto. Infatti, le ECD valgono soltanto per un sistema di riferimento inerziale e il sistema di riferimento solidale al velivolo Γ_B di certo non lo è essendo caratterizzato da moti vari della sua origine (baricentro del velivolo) e rotazioni dei suoi assi. E' indispensabile, quindi, conoscere la posizione e l'orientamento della terna Γ_B rispetto a un sistema di riferimento inerziale nel quale sono formulate le ECD (nominalmente la terna assi Terra Γ_E nella ipotesi di Terra piatta e inerziale). La posizione e l'orientamento della terna Γ_B coincidono con la posizione e l'assetto del velivolo e, dunque, sono esse stesse incognite del problema.

Bisogna, quindi, affiancare alle *equazioni dinamiche* altre equazioni di chiusura, che sono rappresentate dalle *relazioni cinematiche ausiliarie* (ovvero le *equazioni della navigazione inerziale* e le *gimbal equations*). Ciò significa che per risolvere il caso di moto longitudinal-simmetrico è necessario impostare e risolvere un sistema di 6 equazioni differenziali in sei incognite: 3 incognite dinamiche e 3 incognite cinematiche.

Un altro problema si manifesta nella modellazione delle forze e dei momenti agenti sul velivolo in funzione delle stesse incognite. Le azioni esterne sono di tre tipi: aerodinamiche, propulsive e inerziali. Le azioni inerziali si esprimono in maniera piuttosto semplice come funzioni delle incognite del moto. Per le azioni aerodinamiche e propulsive sarà necessario adottare opportuni modelli, specificando le cosiddette *leggi costruttive* delle medesime azioni.

Nel presente elaborato si propone un modello aerodinamico per il volo a 3-DoF e si procede alla determinazione del moto dopo aver assegnato opportune condizioni iniziali. Il primo problema che affronteremo sarà rappresentato da un sottoproblema del problema del moto longitudinal-simmetrico. Si procederà alla ricerca delle condizioni di equilibrio statico, dette *condizioni di trim*. Infatti le condizioni iniziali assegnate al problema $V_0, q_0, x_{E,G,0}, y_{E,G,0}, h_0, \gamma_0$ da sole non completano il vettore iniziale di stato, né assegnano il valore iniziale dei parametri di controllo.

La risoluzione di questo pre-problema consta, quindi, nell'individuare la combinazione di incognite $\alpha_B, \delta_{e,0}, \delta_{s,0}$ che annullino le accelerazioni $\dot{V}, \dot{\alpha}$ e \dot{q} ; volendo, invece, impostare un problema ancora più generale, si assegnerà anche una spinta T_0 corrispondente a quella massima, sottoposta ad un frazionamento legato alla regolazione della manetta: ciò comporterà l'introduzione di un ulteriore parametro di comando δ_T che sarà aggiunto alle incognite precedentemente presentate.

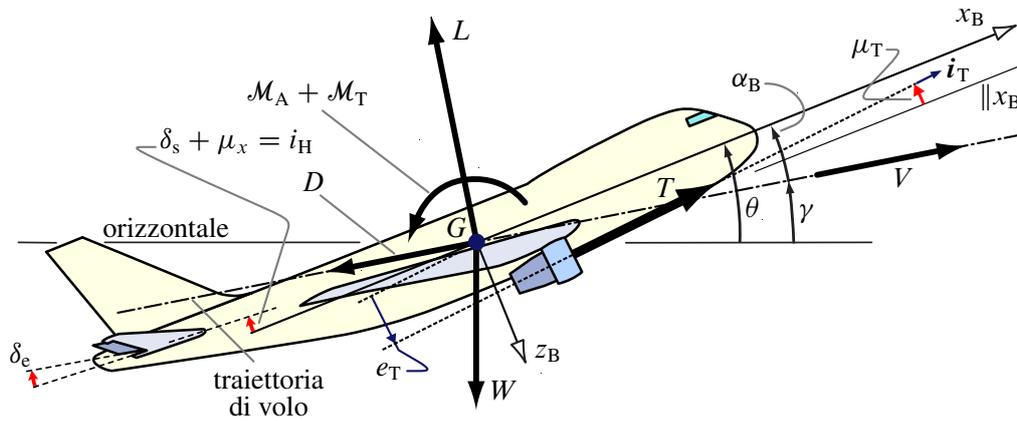


Figura 7.1 Volo equilibrato lungo una traiettoria inclinata di γ sull'orizzontale.

7.2 Impostazione del problema

Le equazioni dinamiche del moto longitudinale-simmetrico sono le equazioni di equilibrio secondo tre gradi di libertà che caratterizzano questo tipo di moto, ovvero le due equazioni di equilibrio alla traslazione lungo l'asse longitudinale e lungo l'asse trasversale e l'equilibrio alla rotazione attorno all'asse laterale (asse di beccheggio). Esse esprimono l'equilibrio tra le azioni esterne (aerodinamiche, propulsive e gravitazionali) e le azioni inerziali nel moto accelerato, rappresentate nella figura 7.1.

Assumendo le classiche adimensionalizzazioni della portanza e della resistenza:

$$L = \frac{1}{2}\rho V^2 S C_L \quad D = \frac{1}{2}\rho V^2 S C_D \quad (7.1)$$

e le seguenti adimensionalizzazioni per il momento aerodinamico di beccheggio e il momento dovuto alla spinta:

$$\mathcal{M}_A = \rho S \bar{c} C_{\mathcal{M},A} \quad \mathcal{M}_T = \rho S \bar{c} C_{\mathcal{M},T} \quad (7.2)$$

le equazioni espresse in assi aerodinamici assumono la seguente veste:

$$-T \cos(\alpha - \mu_x + \mu_T) - \rho S C_D - W \sin \gamma - \frac{W}{g} \dot{V} = 0 \quad (7.3a)$$

$$-T \sin(\alpha - \mu_x + \mu_T) - \rho S C_D - W \cos \gamma - \frac{W}{g} \dot{V} = 0 \quad (7.3b)$$

$$\rho S \bar{c} (C_{\mathcal{M},A} + C_{\mathcal{M},T}) - I_y (\ddot{\alpha} + \ddot{\gamma}) = 0 \quad (7.3c)$$

dove α è l'angolo d'attacco riferito alla retta di portanza nulla del velivolo parziale $\alpha_W = \alpha_B + \mu_x$, con μ_x angolo fra l'asse longitudinale e la retta di portanza nulla. Alle equazioni (7.3) vanno affiancate le 3 relazioni cinematiche ausiliarie:

$$\dot{x}_{E,G} = (V \cos \alpha_B) \cos \theta + (V \sin \alpha_B) \sin \theta = V \cos(\theta + \mu_x - \alpha) \quad (7.4a)$$

$$\dot{z}_{E,G} = -(V \cos \alpha_B) \sin \theta + (V \sin \alpha_B) \cos \theta = V \sin(\theta + \mu_x - \alpha) \quad (7.4b)$$

$$\dot{\theta} = q \quad (7.4c)$$

l'integrazione del sistema di sei equazioni differenziali costituito dall'insieme delle equazioni (7.3) e delle (7.4) permette di determinare l'evoluzione dinamica del velivolo.

Tuttavia, è necessario:

- esplicitare i coefficienti C_L , C_D , $C_{M,A}$ e $C_{M,T}$ in funzione delle incognite dinamiche assegnando un'opportuna legge costitutiva;
- assegnare il vettore delle condizioni iniziali;

La discussione di questi due punti è oggetto dei prossimi paragrafi.

7.3 Il modello aerodinamico e propulsivo

Il sistema di equazioni (7.3) - (7.4) non è in forma chiusa, dal momento che i coefficienti adimensionali di forza e momento sono funzioni delle variabili di stato, ed in particolare dell'angolo di attacco della corrente e della sua derivata, della velocità angolare di beccheggio ma anche del modulo della velocità lineare e della quota tramite il numero di Mach. L'integrazione delle equazioni richiede, dunque, la conoscenza della variazione funzionale di tali coefficienti con α , $\dot{\alpha}$, q e $M = \frac{V}{a}$ (a velocità del suono).

Il modo più semplice di risolvere il problema è assumere un modello di variazione lineare dei coefficienti aerodinamici. Tale assunzione è generalmente valida per le normali incidenze di volo, cadendo in difetto agli alti assetti. L'espressione linearizzata dei coefficienti è data da:

$$C_L = C_{L,\alpha}\alpha + (C_{L,\dot{\alpha}}\dot{\alpha} + C_{L,q}q)\frac{c}{2V} + C_{L,\delta_e}\delta_e + C_{L,\delta_s}\delta_s \quad (7.5a)$$

$$C_D = C_{D,0} + k(C_L)^m = C_{D,0} + (C_{L,\alpha}\alpha + C_{L,\delta_e}\delta_e + C_{L,\delta_s}\delta_s)^m \quad (7.5b)$$

$$C_M = C_{M,0} + C_{M,\alpha}\alpha + (C_{M,\dot{\alpha}}\dot{\alpha} + C_{M,q}q)\frac{\bar{c}}{2V} + C_{M,\delta_e}\delta_e + C_{M,\delta_s}\delta_s \quad (7.5c)$$

dove nella (7.5b) tipicamente $m = 2$ ed inoltre sono stati trascurati gli effetti instazionari e quelli legati alla rotazione del velivolo.

Con tali assunzioni e ponendo ancora:

$$T = \delta_T T_{MAX} \quad (7.6)$$

dove δ_T è il grado di ammissione della manetta (che varia teoricamente nell'intervallo $[0; 1]$), tenendo conto che:

$$\gamma = \theta + \mu_x - \alpha \quad (7.7)$$

il sistema di equazioni 7.3 e 7.4 nella pagina precedente diventa:

$$\dot{V} = g \left[\frac{\delta_T T_{MAX}}{W} \cos(\alpha_0 - \mu_x + \mu_T) - \sin(\theta - \mu_x + \alpha) \right]$$

$$-\frac{\rho V^2}{2(W/S)} \left[C_{D0} + k \left(C_{L\alpha} \alpha + C_{L\delta_e} \delta_e + C_{L\delta_s} \delta_s \right)^m \right] \quad (7.8a)$$

$$\dot{\alpha} = \frac{1}{1 + \frac{c/b}{4\mu} C_{L\dot{\alpha}}} \left[q \left(\frac{1 + \bar{c}/b}{4\mu} C_{Lq} \right) - \frac{\delta_T \Gamma_{MAX}}{W} \frac{g}{V} \sin(\alpha_0 - \mu_x + \mu_T) + \frac{g}{V} \cos(\alpha - \mu_x + \mu_T) - \frac{\rho V^2}{2(W/S)} \frac{g}{V} \left(C_{L\alpha} \alpha + C_{L\delta_e} \delta_e + C_{L\delta_s} \delta_s \right) \right] \quad (7.8b)$$

$$\dot{q} = \frac{V^2 \bar{c}/b}{K_y^2 2\mu} \left[C_{\mathcal{M},T} + C_{\mathcal{M}} + C_{\mathcal{M}\alpha} \alpha + C_{\mathcal{M}q} \frac{q_0 \bar{c}}{2V} + C_{\mathcal{M}\delta_e} \delta_e + C_{\mathcal{M}\delta_s} \delta_s \right] \quad (7.8c)$$

$$\dot{x}_{E,G} = V \cos(\theta + \mu_x - \alpha) \quad (7.8d)$$

$$\dot{z}_{E,G} = V \sin(\theta + \mu_x - \alpha) \quad (7.8e)$$

$$\dot{\theta} = q \quad (7.8f)$$

Tale sistema può essere posto in forma vettoriale come segue:

$$\begin{pmatrix} \dot{V} \\ \dot{\alpha} \\ \dot{q} \\ \dot{x}_{E,G} \\ \dot{z}_{E,G} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ -M_{32} f_2 + f_3 \\ f_4 \\ f_5 \\ f_6 \end{pmatrix} \quad (7.9)$$

dove le f_i ($i = 1, \dots, 6$) sono dati dai secondi membri delle equazioni 7.8 a fronte mentre M_{32} è dato da:

$$M_{23} = -\frac{1}{4\mu} \frac{\bar{c}^2}{k_y^2} C_{\mathcal{M}\dot{\alpha}} \dot{\alpha} \frac{V}{b} \quad (7.10)$$

7.4 Ricerca delle condizioni di trim

Per formulare un problema ai valori iniziali per le equazioni del moto longitudinal-simmetrico 7.8 [nella pagina precedente](#), è necessario affiancare ad esse il valore delle variabili di stato all'istante iniziale $t_0 = 0$, ovvero specificare il vettore $\mathbf{x}(0)$:

$$x(0) = \begin{pmatrix} V(0) \\ \alpha(0) \\ q(0) \\ x_{E,G}(0) \\ z_{E,G}(0) \\ \theta(0) = \gamma(0) + \alpha_B(0) \end{pmatrix} = \begin{pmatrix} V_0 \\ \alpha_0 \\ q_0 \\ x_{E,G,0} \\ -h_0 \\ \theta_0 = \gamma_0 + \alpha_{B,0} \end{pmatrix} \quad (7.11)$$

oltre che il vettore dei parametri di controllo $\mathbf{u}(0)$ ancora all'istante iniziale:

$$\mathbf{u}(0) = \begin{Bmatrix} \delta_e(0) \\ \delta_s(0) \\ \delta_T(0) \end{Bmatrix} = \begin{Bmatrix} \delta_{e,0} \\ \delta_{s,0} \\ \delta_{T,0} \end{Bmatrix} \quad (7.12)$$

Nell'ipotesi di comandi bloccati inoltre saranno assegnate le leggi $\delta_e(t)$, $\delta_s(t)$ e $\delta_T(t)$ che dipendono dall'azione che il pilota esercita sui comandi.

Le condizioni iniziali possono essere assegnate in maniera arbitraria, ma particolarmente interessante risulta il caso in cui si analizza l'evoluzione dinamica del velivolo da una condizione iniziale di volo equilibrato (detta anche condizione di *trim*), nella quale $q_0 = 0$ e risultano costanti velocità e angolo di rampa (o equivalentemente l'angolo di elevazione). Si tratta di una condizione di moto traslatorio uniforme con un'assegnata pendenza $\gamma_0 = 0$ della traiettoria del baricentro sul piano orizzontale.

Qualora siano assegnati V_0 , $z_{E,G}(0)$ e $\gamma(0)$ restano da determinare l'angolo d'attacco α_0 e i valori dei parametri di controllo all'istante iniziale $\delta_{e,0}$, $\delta_{s,0}$ e $\delta_{T,0}$. In questo consiste il problema della ricerca delle condizioni di trim.

I valori iniziali α_0 , $\delta_{e,0}$, $\delta_{s,0}$ e $\delta_{T,0}$ devono soddisfare le equazioni dinamiche per il moto stazionario, che non sono altro che le equazioni 7.13a, 7.13b e 7.13c nelle quali si impone l'annullarsi delle derivate delle variabili di stato dinamiche \dot{V} , $\dot{\alpha}$ e \dot{q} . Si tratta quindi di equazioni algebriche che, adottando il modello di velivolo presentato nel paragrafo 7.3 a pagina 74, assumono la seguente forma:

$$0 = \frac{\delta_{T,0} T_0}{W} \cos(\alpha_0 - \mu_x + \mu_T) - \sin \gamma_0 - \frac{\rho V_0^2}{2(W/S)} \left[C_{D_0} + k \left(C_{L_\alpha} \alpha_0 + C_{L_{\delta_e}} \delta_{e,0} + C_{L_{\delta_s}} \delta_{s,0} \right)^m \right] \quad (7.13a)$$

$$0 = q_0 \left(1 - \frac{\bar{c}/b}{4\mu} C_{L_q} \right) - \frac{\delta_{T,0} T_0}{W} \frac{g}{V_0} \sin(\alpha_0 - \mu_x + \mu_T) + \frac{g}{V_0} \cos \gamma_0 - \frac{\rho V_0^2}{2(W/S)} \frac{g}{V_0} \left(C_{L_\alpha} \alpha_0 + C_{L_{\delta_e}} \delta_{e,0} + C_{L_{\delta_s}} \delta_{s,0} \right) \quad (7.13b)$$

$$0 = C_{\mathcal{M},T} + C_{\mathcal{M}_0} + C_{\mathcal{M}_\alpha} \alpha_0 + C_{\mathcal{M}_q} \frac{q_0 \bar{c}}{2V_0} + C_{\mathcal{M}_{\delta_e}} \delta_{e,0} + C_{\mathcal{M}_{\delta_s}} \delta_{s,0} \quad (7.13c)$$

Le 7.13 sono valide se si trascurano gli effetti delle variazioni di quota, cioè assegnando un valore costante di ρ_0 alla densità e μ_0 alla densità relativa del velivolo. Per q_0 le 7.13 descrivono un volo traslatorio equilibrato lungo una traiettoria rettilinea. Se una delle incognite α_0 , $\delta_{e,0}$, $\delta_{s,0}$ o $\delta_{T,0}$ fosse nota, allora il sistema ammetterebbe una ed una sola soluzione, perchè si avrebbero 3 equazioni in 3 incognite; lasciando, invece, indeterminati le 4 incognite otteniamo un problema con ∞^1 soluzioni.

Ecco quindi che per un caso generale si palesa la necessità di risolvere un problema di estremo relativo, e precisamente è necessario ricercare il minimo della funzione di costo:

$$J\left(V_0, \alpha_{B,0}, q_0, z_{E,G,0}, \gamma_0 - \alpha_{B,0}, \delta_{e,0}, \delta_{s,0}, \delta_{T,0}\right) = f_1^2 + f_2^2 + f_3^2 \quad (7.14)$$

Tale funzione J definita dalla 7.14 è detta *funzione di costo* del problema del *trim*. Essa è sempre non negativa e, in particolare, si annulla per quella combinazione di parametri che annulla ciascuna delle equazioni di equilibrio 7.13.

7.5 Ricerca delle condizioni di trim - 3-Dof

7.5.1 Codice per la ricerca delle condizioni di trim - *Esercizio 7.6*

Si riporta un possibile script di un codice di calcolo eseguibile in Matlab utile per la risoluzione del problema della ricerca della condizione di trim per il volo equilibrato per un moto a 3-DoF. I dati del velivolo con cui sono stati eseguiti i calcoli sono riportati nel listato 7.3 a pagina 81.

Listato 7.1 Codice per la ricerca della condizione di Trim

```
clear all; clc; close all;

%% Ricerca condizioni di TRIM per un moto a 3 dof a comandi bloccati
disp(['Aircraft motion, 3DoF.']);
disp(['Solving the problem of trim at given altitude and flight speed.']);
%% Dichiarazioni
global ...
g ... % accelerazione di gravità
z_0 V_0 q_0 gamma_0 ... % condizioni iniziali
rho_0 ... % densità aria alla quota stabilita
myAC ... % oggetto aircraft

%% Nome file Aircraft
aircraftDataFileName = 'DSV_Aircraft_data2010.txt';

%% Oggetto Aircraft
myAC = DSVAircraft(aircraftDataFileName);

if (myAC.err == -1)
    disp(['... terminating.']);
else
    disp(['File ', aircraftDataFileName, ' letto correttamente.']);

    %% Condizioni iniziali
    z_0 = -4000.; % altitudine s.l. (m)
    V_0 = 257.; % velocità di volo
    q_0 = 0.; % velocità di beccheggio (rad/s)
    gamma_0 = convang(0, 'deg', 'rad'); % gamma (rad)

    %% Modello ISA per la costruzione della funzione densità
    [air_Temp_0, sound_speed_0, air_pressure_0, rho_0] = atmosisa(-z_0);

    %% Accelerazione di gravità
    g = 9.81; % (m/s^2)

    %% condizioni iniziali per il vettore di progetto
    x0 = [ ...
        0.; ... % alpha_0
        0.; ... % delta_e_0
        0.; ... % delta_s_0
        0.5 ... % delta_T_0
    ];

    %% Costruzione della funzione di minimizzazione

    % confini
    lb = [convang(-15, 'deg', 'rad'), ... % minimo alpha
        convang(-20, 'deg', 'rad'), ... % minima deflessione elevatore
        convang(-5, 'deg', 'rad'), ... % minima incidenza stabilizzatore
        0.2 ... % minimo rateo di spinta
    ];
    ub = [convang( 15, 'deg', 'rad'), ... % massimo alpha
        convang( 13, 'deg', 'rad'), ... % massima deflessione elevatore
```

```

convang( 2,'deg','rad'), ... % massima incidenza stabilizzatore
1.0          ... % massimo rateo di spinta
];

options = optimset( ...
'tolfun',1e-9, ...           % tolleranza numerica
'Algorithm','interior-point' ... % tipo di algoritmo
);
[x,fval] = ...
fmincon(@costLongEquilibriumStaticStickFixed,...%funzione obiettivo
x0, ... %vettore iniziale
[], ... %vincoli lineari
[], ... %vincoli lineari
[], ... %vincoli lineari
[], ... %vincoli lineari
lb,ub, ... %condizioni contor.
@myNonLinearConstraint, ... %vincoli non linear
options);
%listato interrotto...

```

Tutto lo script è stato realizzato preparando e fornendo in input i parametri necessari per una corretta esecuzione della funzione di libreria Matlab *fmincon*.

Con la chiamata $x_0 = [...]$ si è impostato il vettore iniziale di progetto, *design point*, che viene inizializzato con valori di tentativo e che costituirà la nostra variabile vettoriale della funzione scalare **J** riportata nell'equazione 7.14 a pagina 77.

Siccome la nostra variabile x_i può variare soltanto in un intervallo reale finito, sono state imposti dei vincoli che rappresentano aspetti fisici del problema, come la massima e minima deflessione dell'equilibratore, la massima e minima deflessione dello stabilizzatore, massimo e minimo angolo di attacco e massimo e minimo rateo di spinta.

Con la chiamata *options* è stata imposta la tolleranza numerica da rispettare 10^{-9} , che sta a indicare che la ricerca del minimo condizionato sarà sospesa qualora si registrerà un errore numerico inferiore a quello dichiarato.

A valle del processo di ricerca la funzione *fmincon* produrrà un vettore $[x, fval]$ contenente i valori delle variabili tali per cui sarà stata raggiunta la condizione della ricerca del minimo condizionato per la funzione scalare **J**.

Di seguito sono riportate le istruzioni per la stampa a video del risultato per la condizione di trim:

```

%...continuo listato
alpha_0      = x(1); %valori ottenuti
alpha_0_deg  = convang(alpha_0,'rad','deg');
theta_0      = gamma_0 + alpha_0 + myAC.mu_x;
theta_0_deg  = convang(theta_0,'rad','deg');
delta_e_0    = x(2);
delta_e_0_deg = convang(delta_e_0,'rad','deg');
delta_s_0    = x(3);
delta_s_0_deg = convang(x(3),'rad','deg');
delta_T_0    = x(4);

x0 = [ ...
x(1); ... % alpha_0
x(2); ... % delta_e_0
x(3); ... % delta_s_0
x(4) ... % delta_T_0
];

disp('');
disp('Condizione di trim:');
disp(['Velocità V_0= ',num2str(V_0),' m/s']);
disp(['Angolo d''attacco alpha_0 = ',num2str(alpha_0_deg),'deg']);

```

```

disp(['Elevatore delta_e_0 = ', num2str(delta_e_0_deg), 'deg']);
disp(['Stabilizzatore delta_s_0 = ', num2str(delta_s_0_deg), 'deg']);
disp(['Manetta delta_T_0 = ', num2str(delta_T_0)]);
end

```

Si evince dal listato 7.1 che per il funzionamento della funzione `fmincon` sono state richiamate due funzioni esterne, lo si riconosce dalla chiamata della funzione `@`. `fmincon` restituisce a valle dell'analisi il valore assunto da ogni variabile di stato per rispettare la condizione di trim.

- **costLongEquilibriumStaticStickFixed**: implementa i secondi membri del sistema di equazioni 7.13 a pagina 76 fornendo il minimo di una funzione scalare f , detta anche *funzione obiettivo*. Nella condizione di trim siamo alla ricerca di quella combinazione di variabili di stato e di comando che mi annulla la f . Tale funzione accetta in input il vettore delle variabili di design e restituisce f . Il seguente listato riporta un estratto rappresentate il *core* del codice di calcolo della seguente funzione:

```

function f = costLongEquilibriumStaticStickFixed(x)
...dichiarazioni globali
%% Equation of motion right-hand-sides, for steady flight

F1 = (delta_T*myAC.T/myAC.W)*cos(alpha - myAC.mu_x + myAC.mu_T) ...
    -sin(gamma_0) ...
    -((rho_0*V_0^2)/(2*(myAC.W/myAC.S))) ...
    *(myAC.CD_0 + myAC.K*((myAC.CL_alpha*alpha ...
        + myAC.CL_delta_e*delta_e ...
        + myAC.CL_delta_s*delta_s)^myAC.m));

F2 = ( 1. - myAC.CL_q*(myAC.mac/myAC.b)/(4*mu_rel) )*q_0 ...
    -(g/V_0)*(delta_T*myAC.T/myAC.W)*sin(alpha - myAC.mu_x + myAC.mu_T)
    ...
    +(g/V_0)*cos(gamma_0) ...
    -(g/V_0)*((rho_0*V_0^2)/(2*(myAC.W/myAC.S))) ...
    *( myAC.CL_alpha*alpha + myAC.CL_delta_e*delta_e ...
        + myAC.CL_delta_s*delta_s);

F3 = myAC.Cm_0 + myAC.Cm_alpha*alpha ...
    + myAC.Cm_delta_s*delta_s ...
    + myAC.Cm_delta_e*delta_e ...
    + (myAC.mac/(2*V_0))*myAC.Cm_q*q_0 ...
    + myAC.Cm_T_0 + myAC.Cm_T_alpha*alpha;

%% Build the cost function
f = F1*F1 + F2*F2 + F3*F3;
end

```

- **myNonLinearConstraint**: implementa eventuali vincoli non lineari (uguaglianze e/o disuguaglianze) nella ricerca del minimo. Nel caso in esame tale funzione non serve, ma è necessario crearla per rispettare gli input richiesti dalla funzione `fmincon`

```

function [c,ceq] = myNonLinearConstraint(x)
c = ... % Compute nonlinear inequalities at x.
    [];
ceq = ... % Compute nonlinear equalities at x.
    [];
end

```

I valori frutto dell'analisi della ricerca di trim sono stati stampati a display nella *command window*:

Listato 7.2 Condizioni di trim

```
Condizione di trim:
Velocità V_0= 257 m/s
Angolo d'attacco alpha_0= 2.0283deg
Elevatore delta_e_0= -4.223deg
Stabilizzatore delta_s_0= -0.10259deg
Manetta delta_T_0= 0.4082
```

Le specifiche del velivolo sono state fornite in input al codice di calcolo mediante una classe pensata per la gestione dei dati di un velivolo nell'ambito delle simulazioni di volo.

Tale versione risulta un po' più sofisticata di una semplice struttura dati. Essa è stata battezzata **DSVAircraft** ed è da implementarsi in un file di nome **DSVAircraft.m**. Il listato 7.3 contiene i dati forniti in input al solutore:

Listato 7.3 File per inserimento dati velivolo in Matlab

```
-----
Dinamica e Simulazione di Volo / Coiro, De Marco A.A. 2010-2011
-----

A/C MAIN GEOMETRIC DATA
17.0 Wing planform area, S, (m^2)
6.90 Wing span, b, (m)
2.60 Mean aerodynamic chord, mac, (m)

MASS AND INERTIA
6000 Mass, mass (kg)
2.20 Inertia gyration radius around pitch axis, k_y, (m)
0.00 Wing-Body zero-lift line angle wrt x-body axis, mu_x, (rad)
0.300 Normalized CG pos. wrt mac l.e., Xcg_adim

AERODYNAMICS
A/C NEUTRAL POINT
0.450 Normalized CG pos. wrt mac l.e., Xn_adim
POLAR
0.0580 CD0 in the parabolic polar: CD=CD0+K*CL^m
0.3500 K in the parabolic polar
2.00 m in the parabolic polar
AERODYNAMIC DERIVATIVES
4.1800 CL_alpha (1/rad)
0.2870 CL_delta_e (1/rad)
0.5220 CL_delta_s (1/rad)
2.270 CL_alpha_dot
4.720 CL_q
-0.0150 Cm(CL=0)
-0.5070 Cm_delta_e (1/rad)
-0.9230 Cm_delta_s (1/rad)
-4.000 Cm_alpha_dot
-8.34 Cm_q
0.000 Cm_delta_e_dot
-0.375 Cm_alpha

ELEVATOR DATA
GEOMETRY
1.1590 Reference surface, S_e (m^2)
0.5235988 Hinge axis sweep angle, Lambda_e (rad)
-3.300 Coord. along x-body of elev.-fixed ref. frame origin, x_C_e (m)
0.420 Elevator mean aerod. chord, mac_e (m)
MASS/INERTIA
50.00 Elevator mass, mass_e (kg)
```

```

0.0000 Offset of elev. CG wrt elev. mac (m)
0.090 Elevator inertia gyration radius around hinge, k_e, (m)
AERODYNAMICS
0.0000 Ch_e_0
-0.1300 Ch_e_alpha (1/rad)
-0.1100 Ch_e_delta_s (1/rad)
-0.3000 Ch_e_delta_e (1/rad)
-0.5000 Ch_e_delta_e_dot
-1.0000 Ch_e_q
-0.5 Ch_e_alpha_dot
0 eps_0
0 deps/dalpha

ELEVATOR COMMAND CHARACTERISTICS
TYPE
reversible
0.3000 Stick sensitivity ratio, Rs_e (push/pull)
0.40000 Gearing ratio, i.e. stick displ./delta_e, Rg_e (m/rad)
ELEVATOR ANGULAR RANGE
0.5235988 max angle, push-over-wise, delta_e_max (rad)
-0.5235988 min angle, pull-up-wise, delta_e_min (rad)

PROPULSION DATA
7530 Thrust, T (kg-force)
0.0000 Cm_T_0
0.0000 Cm_T_alpha
0.0000 Setting angle wrt x-body axis, mu_T (rad)
0.0000 offset (with sign) of T-vector wrt cg, e_T (m)

AERODYNAMIC/STRUCTURAL LIMITATIONS
0.85 Max CL, CL_max
-0.75 Min CL, CL_min
7.80 Max load factor, n_max
-4.50 Min load factor, n_min

STICK-FORCE LIMITATIONS
75.000 Max push force (kg-force)
-75.000 Max pull force (kg-force)
-----

```

7.5.2 Nuova condizione di Trim per $\delta_s = -1deg$ - *Esercizio 7.7*

Nel paragrafo 7.5.1 (*esercizio 7.6* dal testo) sono state ricavate le condizioni di *trim* senza assegnare specifiche richieste per il vettore di input, si è lasciato che fosse la funzione `fmincon` a ricavare i valori necessari per la condizione desiderata di volo equilibrato.

Per questo esercizio si è assegnato, invece, un particolare valore per la deflessione dello stabilizzatore ponendolo pari a -1 deg e da qui sono state ricavate le condizioni di volo equilibrato.

Per raggiungere l'obiettivo, è stato assegnato un valore fisso ad una delle componenti del vettore nell'argomento di ricerca del minimo di f sfruttando il modo in cui è congegnata la funzione `fmincon`.

Il listato per la risoluzione di questo problema è uguale al listato 7.1 a pagina 78 fino alla parte in cui si tratta la %% Costruzione della funzione di minimizzazione da quella parte segue il listato 7.4 e a seguire gli altri proposti.

Listato 7.4 Condizione di Trim per δ_s vincolato

```

%...continuo listato
%% Costruzione della funzione di minimizzazione

% Aeq, in Aeq*x=beq vincoli lineari
Aeq      = zeros(4,4);
beq      = zeros(4,1);
Aeq(3,3) = 1; % pick delta_s
delta_s_0 = convang(-1.0,'deg','rad');
beq(3,1) = delta_s_0; % keep delta_s fixed
% listato interrotto...

```

Volendo vincolare il parametro di input $\delta_{s,0}$, ricercando quindi una condizione di trim nella quale lo stabilizzatore ha un certo angolo fissato, bisogna ridefinire la matrice A_{eq} , imponendo che alla terza riga e terza colonna il valore non sia più zero, ma unitario, quindi è stato posto $[A_{eq}(3,3)] = 1$; e $b_{eq}(3,1)$ pari al valore di $\delta_{s,0}$ che vogliamo imporre.

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} \alpha \\ \delta_e \\ \delta_s \\ \delta_T \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \delta_{s,0} = -1 \\ 0 \end{pmatrix} \quad (7.15)$$

```

%...continuo listato
% confini
lb =[convang(-15,'deg','rad'), ... % minimo alpha
    convang(-20,'deg','rad'), ... % minima deflessione elevatore
    convang(-5,'deg','rad'), ... % minima incidenza stabilatore
    0.2 ... % minimo rateo di spinta
];
ub =[convang( 15,'deg','rad'), ... % massimo alpha
    convang( 13,'deg','rad'), ... % massima deflessione elevatore
    convang( 5,'deg','rad'), ... % massima incidenza stabilatore
    1.0 ... % massimo rateo di spinta
];

options = optimset( ...
    'tolfun',1e-6, ... % tolleranza numerica
    'Algorithm','interior-point' ... % tipo di algoritmo
);

```

```

V = linspace(220,380,14);

for i=1:14

    V_0 = V(i);

    [xi,fval] = ...
        fmincon(@costLongEquilibriumStaticStickFixed, ...
            xi0, ...
            [], ...
            [], ...
            Aeq, ... % Aeq, Aeq*x=beq
            beq, ... % beq
            lb,ub, ...
            @myNonLinearConstraint, ...
            options);

    alpha_0(i) = xi(1);
    alpha_0_deg(i) = convang(alpha_0(i), 'rad', 'deg');
    theta_0(i) = gamma_0 + alpha_0(i) + myAC.mu_x;
    theta_0_deg(i) = convang(theta_0(i), 'rad', 'deg');
    delta_e_0(i) = xi(2);
    delta_e_0_deg(i) = convang(delta_e_0(i), 'rad', 'deg');
    delta_s_0(i) = xi(3);
    delta_s_0_deg(i) = convang(xi(3), 'rad', 'deg');
    delta_T_0(i) = xi(4);

    xi0 = [ ...
        xi(1); ... % alpha_0
        xi(2); ... % delta_e_0
        xi(3); ... % delta_s_0
        xi(4) ... % delta_T_0
    ];

end

figure(1)
subplot(3,1,1)
plot(V,alpha_0_deg, '-b', 'LineWidth',1.5, 'markersize',5);
%xlabel('V_T_A_S', 'fontsize',12)
ylabel('\alpha_{0} deg', 'fontsize',14)
grid on;

subplot(3,1,2)
plot(V,delta_e_0_deg, '-b', 'LineWidth',1.5, 'markersize',5);
%xlabel('V_T_A_S', 'fontsize',12)
ylabel('\delta_{e,0} deg', 'fontsize',14)
grid on;

subplot(3,1,3)
plot(V,delta_T_0, '-b', 'LineWidth',1.5, 'markersize',5);
xlabel('V_T_A_S', 'fontsize',12)
ylabel('\delta_{T,0} deg', 'fontsize',14)
grid on;

end

```

Per questo esercizio è stata fatta esplicita richiesta di calcolare i valori di equilibrio al variare della velocità di volo iniziale e per il valore della deflessione dello stabilizzatore bloccato a -1 deg.

Si è dovuto quindi ricercare le condizioni di trim al variare della velocità. Si vede che è stato definito un vettore velocità che andava da 220m/s fino a 380m/s usando un passo d'indagine ogni 14m/s .

Nella figura 7.2 nella pagina successiva sono riportati i valori rappresentativi delle

condizioni di equilibrio per α_0 riferente al subplot (3, 1, 1), $\delta_{e,0}$ riferente al subplot (3, 1, 2) e $\delta_{T,0}$ riferente al subplot (3, 1, 3) i valori sono stati ricavati per $\delta_{s,0} = -1$ deg, per quota fissata ad $h_0 = 4000$ m, al variare del parametro di V_0 .

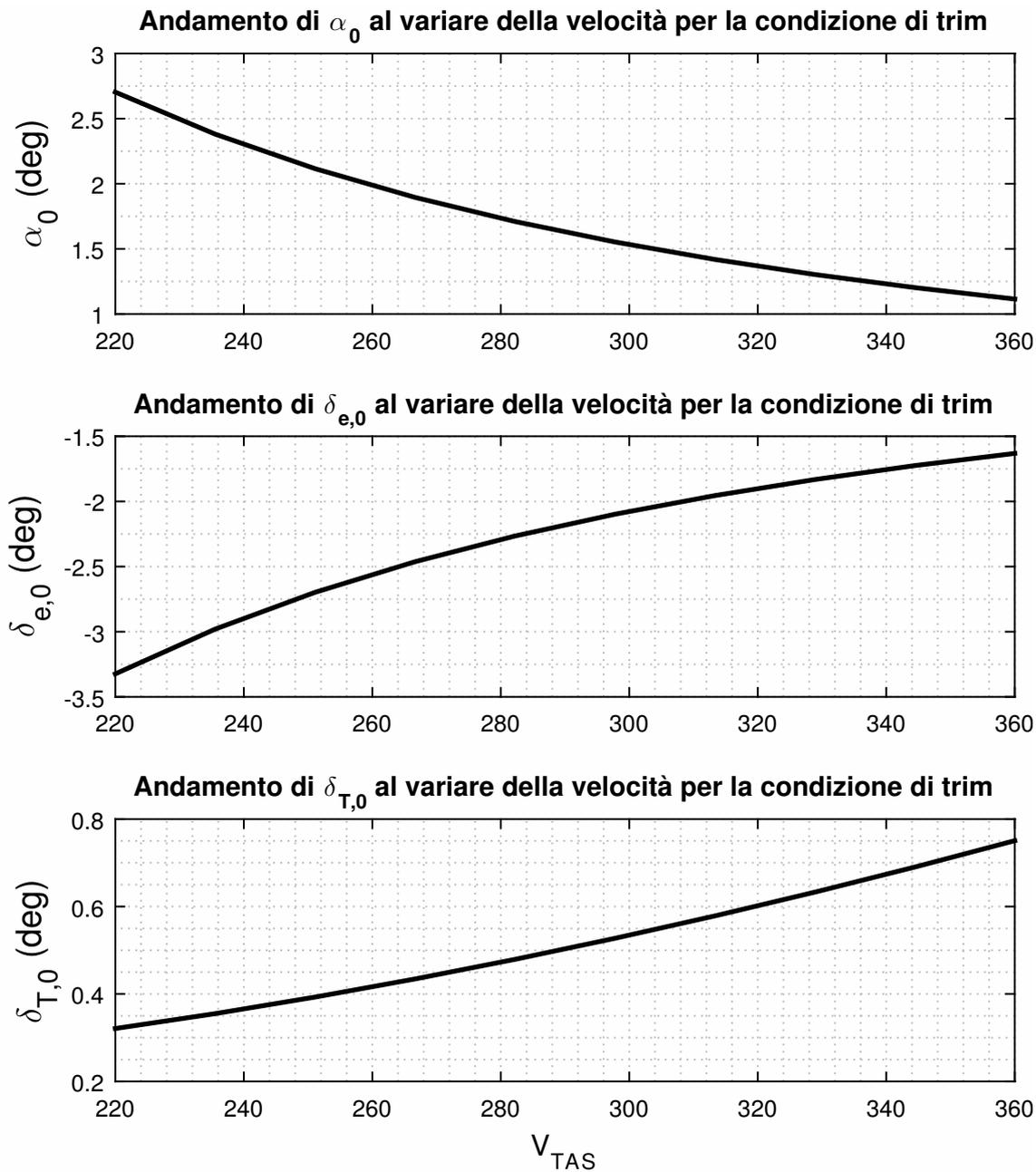


Figura 7.2 Valori di equilibrio al variare della velocità di volo iniziale, per $\delta_{s,0} = -1$ deg. I dati del velivolo sono quelli riportati nel listato 7.3 a pagina 81.

7.5.3 Verifica delle condizioni di trim - *Esercizio 7.8*

Per risolvere questo esercizio è stato, anche questa volta, implementato il primo codice di calcolo proposto nel listato 7.1 a pagina 78, il nuovo blocco del listato, da aggiungere al listato 7.1, muove i suoi passi dalla parte in cui viene presentata la %% Costruzione della funzione di minimizzazione.

E' stato richiesto di sviluppare il codice per una legge di comando longitudinale e della manetta assegnate. Il velivolo considerato è sempre quello proposto nel listato 7.3 a pagina 81. Sono stati rimossi i vincoli assegnati nell'esercizio precedente per la deflessione dell'equilibratore.

```
%% Costruzione della funzione di minimizzazione - RICERCA CONDIZIONE DI TRIM

% Aeq, in Aeq*x=beq vincoli lineari - dichiarati, ma non utilizzati.
Aeq      = zeros(4,4);
beq      = zeros(4,1);
Aeq(3,3) = 1;                               % blocco delta_s
Aeq(2,2) = 0;                               % libero l'equilib.
delta_s_0 = convang(-1, 'deg', 'rad');
delta_e_0 = convang(0., 'deg', 'rad');
beq(2,1)  = delta_e_0;
beq(3,1)  = delta_s_0;

% confini
lb = [convang(-15., 'deg', 'rad'), ... % minimo alpha
      convang(-20., 'deg', 'rad'), ... % minima deflessione elevatore
      convang(-5., 'deg', 'rad'), ...  % minima deflessione stabilizzatore
      0.2 ... % minimo rateo di spinta
      ];
ub = [convang( 15., 'deg', 'rad'), ... % massimo alpha
      convang( 13., 'deg', 'rad'), ... % massima deflessione elevatore
      convang(  5., 'deg', 'rad'), ... % massima incidenza stabilizzatore
      1.0 ... % massimo rateo di spinta
      ];

options = optimset( ...
    'tolfun',1e-9, ... % tolleranza numerica
    'Algorithm','interior-point' ... % tipo di algoritmo
);

[xi,fval] = ...
    fmincon(@costLongEquilibriumStaticStickFixed, ...
        xi0, ...
        [], ...
        [], ...
        [], ...
        lb,ub, ...
        @myNonLinearConstraint, ...
        options);

alpha_0      = xi(1);
alpha_0_deg  = convang(alpha_0, 'rad', 'deg');
theta_0      = gamma_0 + alpha_0 - myAC.mu_x;
theta_0_deg  = convang(theta_0, 'rad', 'deg');
delta_e_0    = xi(2);
delta_e_0_deg = convang(delta_e_0, 'rad', 'deg');
delta_s_0    = xi(3);
delta_s_0_deg = convang(xi(3), 'rad', 'deg');
delta_T_0    = xi(4);
```

```

xi0 = [ ...
    xi(1); ... % alpha_0
    xi(2); ... % delta_e_0
    xi(3); ... % delta_s_0
    xi(4) ... % delta_T_0
];

%Visualizzazione Monitor valori di TRIM
disp('')
disp('Condizione di trim:')
disp(['Velocità V_0= ',num2str(V_0),' m/s'])
disp(['Angolo d''attacco alpha_0= ',num2str(alpha_0_deg),'deg'])
disp(['Elevatore delta_e_0= ',num2str(delta_e_0_deg),'deg'])
disp(['Stabilizzatore delta_s_0= ',num2str(delta_s_0_deg),'deg'])
disp(['Manetta delta_T_0= ',num2str(delta_T_0)])

%listato interrotto...

```

A partire dalle condizioni di trim sono state riportate le storie temporali risolvendo le equazioni del moto a 3-DoF per un tempo $t = 10s$. Si è reso quindi necessario definire una nuova funzione nella quale fossero riscritte le 6 equazioni **eqLongDynamicStickFixed** e che fosse capace di poter passare i valori alla funzione di libreria ODE45 la quale integra un sistema di equazioni differenziali ordinarie a partire da condizioni iniziali note, sfruttando il metodo di integrazione al passo di Runge-Kutta.

Abbiamo quindi dovuto stabilire delle nuove condizioni iniziali differenti dagli esercizi precedenti y_0 , poichè raggiunte le condizioni di trim, devo vedere per condizioni come variano le altre variabili di interesse.

```

%...continuo listato
%% STUDIO STORIE TEMPORALI A COMANDI INVARIATI(condizioni di volo equilibrato)

% Ora devo imporre delle condizioni iniziali differenti dagli esercizi precedenti
% poiché, raggiunte le condizioni di trim, devo vedere in queste condizioni
% come variano le altre variabili di interesse.

%Condizioni Iniziali
y0 = [ ...
    V_0; ...
    alpha_0; ...
    q_0; ...
    0.0; ... % x_G_E_0
    z_0; ...
    theta_0 ...
];

t_fin = 10.; %tempo finale

[T, Y] = ode45(@eqLongDynamicStickFixed, [0 t_fin], y0);

% derived vars
v_gamma = Y(:,6) + myAC.mu_x - Y(:,2);

for i = 1:t_fin

delta_T(i) = delta_T_0; %Costruisco le leggi temporali per
    %i comandi invariati come richiesto nella traccia
delta_E(i) = delta_e_0_deg;

delta_S(i) = delta_s_0_deg;

end

```

```

ti = linspace(0,t_fin,t_fin);

%Plot Storia Temporale dei comandi invariati ...
end

```

La parte di codice inerente al plot è stata omessa perchè ritenuta di scarso interesse. Per completezza è stata riportata la funzione proposta dal docente.

- **eqLongDynamicStickFixed** tale funzione implementa i secondi membri delle equazioni del moto longitudinal-simmetrico, 7.8 a pagina 74. Accetta in input la variabile indipendente tempo t e il vettore di stato corrispondente al medesimo tempo e restituisce il vettore al secondo membro dell'equazione 7.9 a pagina 75, ovvero il vettore delle derivate delle variabili di stato. Va passata, insieme al vettore dei tempi e al vettore delle condizioni iniziali, alla funzione `ode45`, la quale risolve numericamente il sistema di equazioni differenziali.

Listato 7.5 eqLongDynamicStickFixed

```

function f = eqLongDynamicStickFixed(t,y)

%% Declaring global variables
global ...
    g ... % gravity acceleration
    z_0 V_0 q_0 gamma_0 ... % initial conditions
    rho_0 ... % air density at z_0
    myAC ... % the aircraft object, populated outside this
    func.
    delta_T_0 ...
    delta_e_0...
    delta_s_0

%% Aircraft relative density
mu_rel = (myAC.W/g)/(rho_0*myAC.S*myAC.b);

%% Variabili di input costanti nell'integrazione
delta_T = delta_T_0;
delta_e = delta_e_0;
delta_s = delta_s_0;

%Variabili di stato da integrare
V = y(1);
alpha = y(2);
q = y(3);
xEG = y(4);
zEG = y(5);
theta = y(6);

% flight-path angle
gamma = theta + myAC.mu_x - alpha;

%% Equazione del moto LONGITUDINAL-SIMMETRICO ,7.61 del sistema completo a 3
dof

F1 = g*( ...
    (delta_T*myAC.T/myAC.W)*cos(alpha - myAC.mu_x + myAC.mu_T) ...
    -sin(gamma) ...
    -((rho_0*V^2)/(2*(myAC.W/myAC.S))) ...
    *(myAC.CD_0 + myAC.K*((myAC.CL_alpha*alpha ...
    + myAC.CL_delta_e*delta_e ...
    + myAC.CL_delta_s*delta_s)^myAC.m)) ...
);

```

```

F2 = ( ...
    ( 1. - myAC.CL_q*(myAC.mac/myAC.b)/(4*mu_rel) ) * q ...
    - (delta_T*myAC.T/myAC.W)*(g/V)*sin(alpha - myAC.mu_x + myAC.mu_T) ...
    + (g/V)*cos(gamma) ...
    - (g/V)*((rho_0*V^2)/(2*(myAC.W/myAC.S))) ...
    *( ...
        myAC.CL_alpha*alpha ...
        + myAC.CL_delta_e*delta_e ...
        + myAC.CL_delta_s*delta_s) ...
    ) / ( 1. + myAC.CL_alpha_dot*(myAC.mac/myAC.b)/(4*mu_rel) );

F3 = ((V^2/myAC.k_y^2)*(myAC.mac/myAC.b)/(2*mu_rel)) ...
    *( ... % Aerodynamics
        myAC.Cm_0 + myAC.Cm_alpha*alpha ...
        + myAC.Cm_delta_s*delta_s ...
        + myAC.Cm_delta_e*delta_e ...
        + (myAC.mac/(2*V))*((myAC.Cm_q*q)) ...
        ... % Thrust
        + (myAC.Cm_T_0 + myAC.Cm_T_alpha*alpha)*delta_T ...
    );

F4 = V*cos(gamma) ;
F5 = -V*sin(gamma) ;
F6 = q ;

f = [F1;F2;F3;F4;F5;F6];

```

Risultati ricerca delle condizioni di trim e volo con leggi di comando identiche

Le condizioni di trim, ricavate per mezzo del codice, sono riportate di seguito:

Listato 7.6 Risultati condizione di Trim per velivolo stabile

```

Condizione di trim:
Velocità V_0= 257 m/s
Angolo d'attacco alpha_0= 2.0283 deg
Elevatore delta_e_0= -4.223 deg
Stabilizzatore delta_s_0= -0.10259 deg
Manetta delta_T_0= 0.4082

```

Le leggi di comando visibili in figura 7.3 nella pagina seguente sono state tenute invariate nel tempo e sono state fornite realizzando 3 vettori mantenuti sempre costanti e di dimensione pari al tempo di simulazione imposto.

Le storie temporali delle variabili di stato per queste leggi di comando sono state rappresentate nella figura 7.4 e 7.5 a pagina 91.

Le oscillazioni delle variabili di stato rispetto ai corrispondenti valori iniziali sono dovute in questo caso all'errore numerico, come dimostra il fatto che esse sono infinitesimi di ordine non inferiore all'ordine dello schema integrativo utilizzato (schema di Runge-Kutta del quarto e del quinto ordine, implementati dalla funzione `ode45`). Se il processo di integrazione fosse esatto, tutte le variabili di stato dovrebbero risultare costanti e le variazioni $\Delta(\ast)$ nulle, tranne per la coordinata baricentrale $x_{E,G}$ che, essendo il moto rettilineo uniforme, deve crescere in maniera lineare nel tempo (tale andamento è apprezzabile nella figura 7.5 a pagina 91).

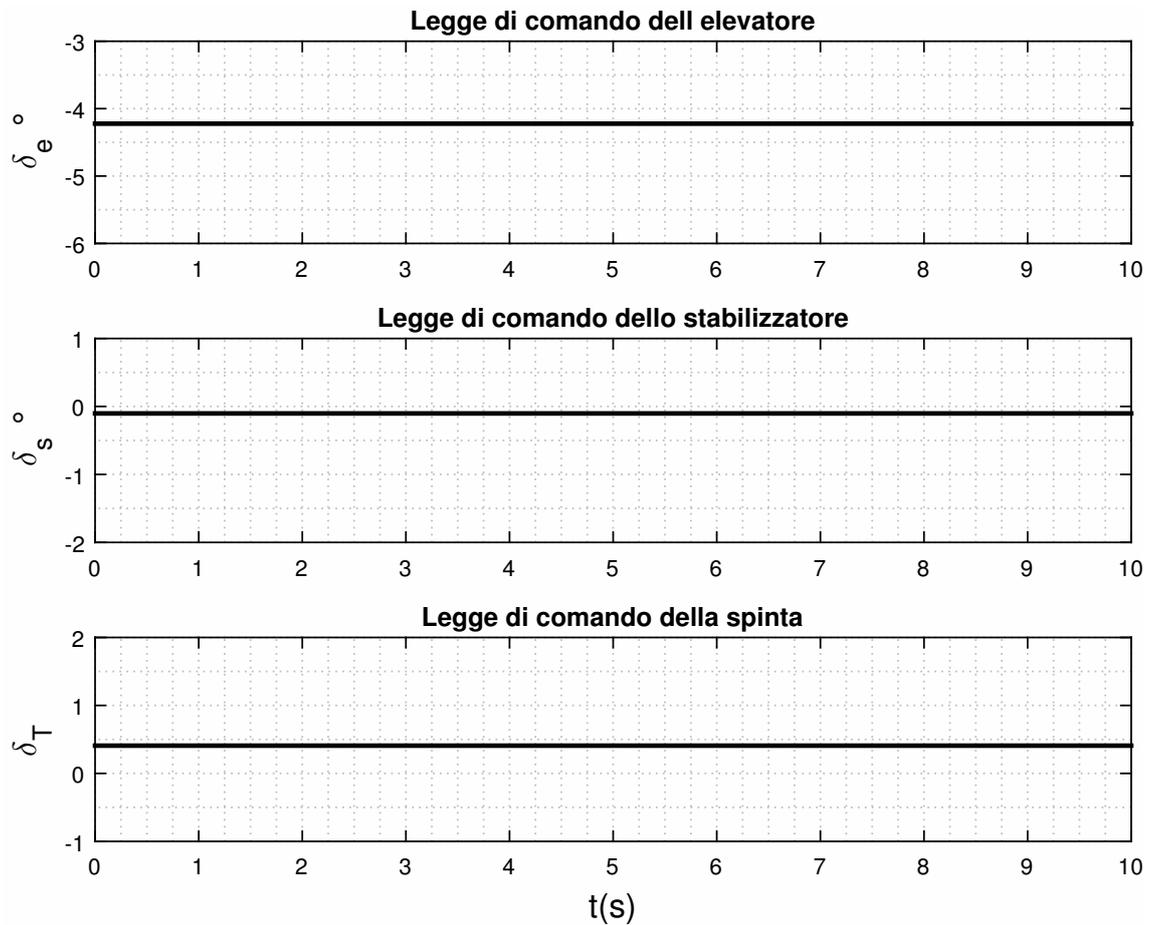


Figura 7.3 Storie temporali dei comandi invariate, a partire dalla condizione di trim riportata nel listato. I dati del velivolo sono quelli riportati nel listato 7.3 a pagina 81.

I risultati proposti nelle figure 7.4 e 7.5 nella pagina successiva hanno questi andamenti, perchè per questo caso il margine di stabilità statica MSS del velivolo è negativo e pari a $MSS = X_G - X_N = -0.13$. In altre parole, essendo il velivolo stabile, il fatto che le variazioni rispetto al valore iniziale non superino l'ordine dell'errore numerico è una prova della corretta determinazione delle condizioni di trim.

Andamento delle variabili di stato per la variazione del baricentro

A tal proposito può essere interessante valutare gli effetti che lo spostamento del baricentro provoca sugli scostamenti delle variabili di stato dai valori corrispondenti alle condizioni di trim.

Si considera perciò un secondo caso in cui la distanza adimensionale del baricentro dal bordo d'attacco della corda media aerodinamica è pari a $\bar{X}_G = 0,50$ e la derivata aerodinamica C_{M_α} risulta modificata secondo la relazione:

$$C_{M_\alpha} = -C_{L_\alpha}(\bar{X}_N - \bar{X}_G) \quad (7.16)$$

a parità di posizione del punto neutro \bar{X}_N e di gradiente della retta di portanza C_{L_α} .

Chiaramente, questa variazione delle caratteristiche aerodinamiche provocherà una variazione anche sulle condizioni di trim, che saranno per questo ricalcolate rispetto al caso precedente. Il listato per la risoluzione di questo problema è stato lasciato inalterato, si è provveduto a modificare il file **DSVAircraft.txt**, 7.3 a pagina 81, contenente i para-

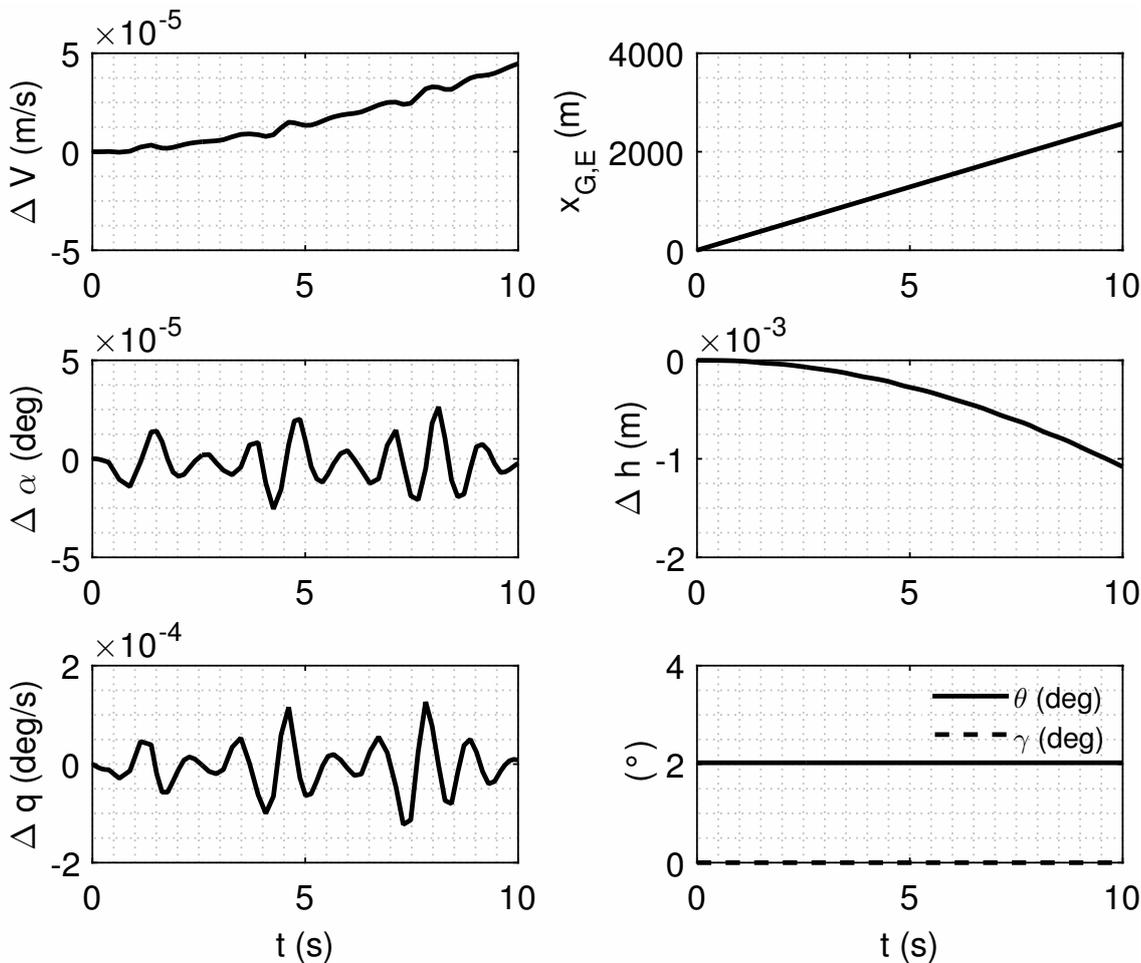


Figura 7.4 Storie temporali delle variabili di stato per leggi di comando identiche, a partire dalla condizione di trim riportata nel listato. Il velivolo è sempre lo stesso assegnato nel listato 7.3 a pagina 81. I Δ (*) rappresentati sono riferiti rispetto ai valori iniziali. Il baricentro è posto al 30% della corda.

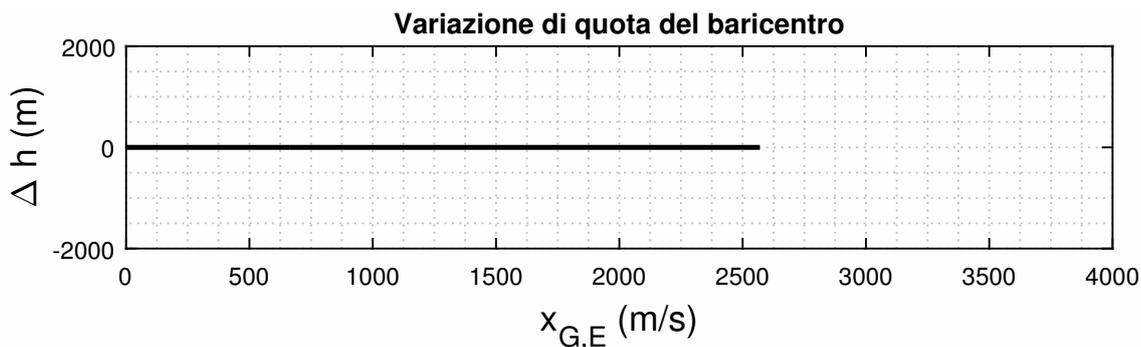


Figura 7.5 Variazione di quota del baricentro, a partire dalla condizione di trim riportate nel listato

metri completi per caratterizzare il velivolo, modificando la disposizione percentuale del baricentro come riportato le listato 7.7.

Listato 7.7 Variazione della posizione del baricentro x_G

```

MASS AND INERTIA
6000   Mass, mass (kg)
2.20   Inertia gyration radius around pitch axis, k_y, (m)
0.00   Wing-Body zero-lift line angle wrt x-body axis, mu_x, (rad)
0.500  Normalized CG pos. wrt mac l.e., Xcg_adim

```

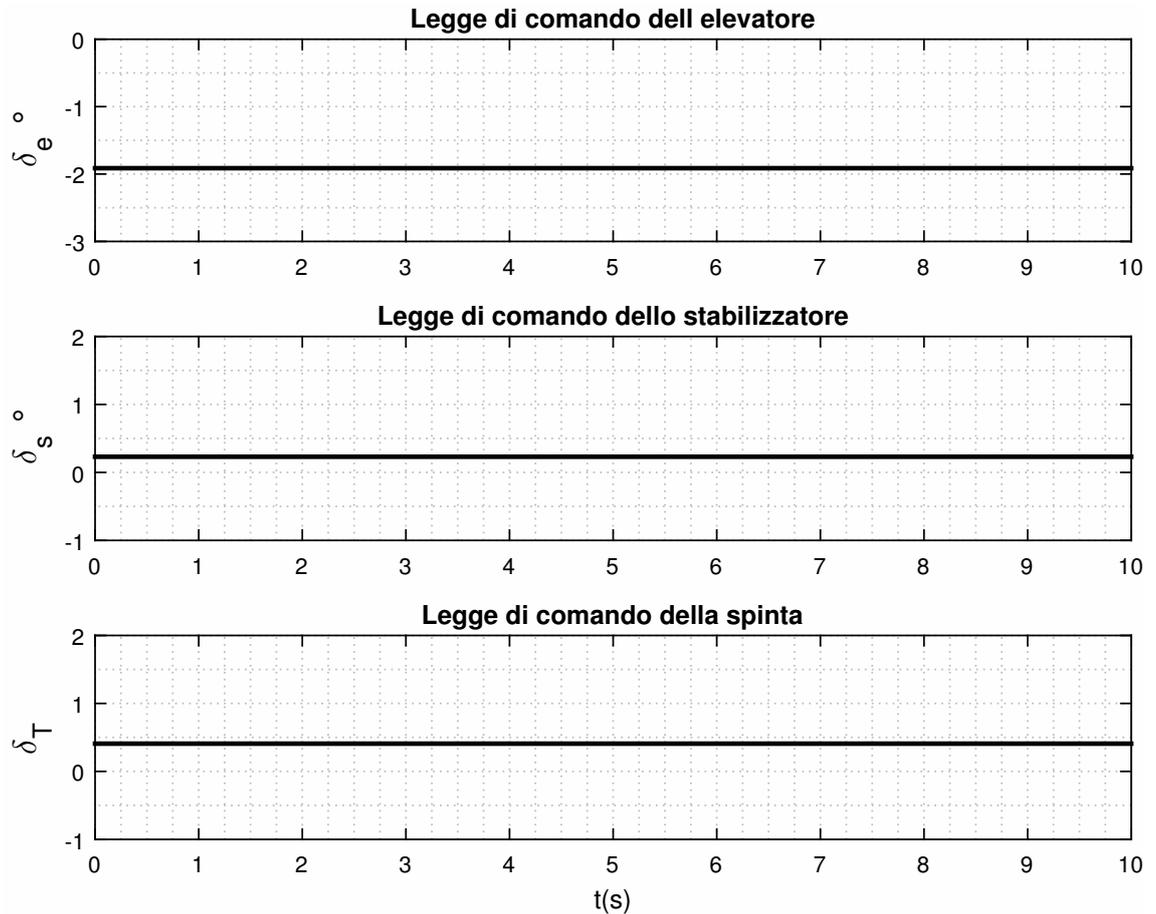


Figura 7.6 Leggi temporali dei comandi invariati, a partire dalla condizione di trim riportata nel listato

Sono state presentate nel seguente listato le nuove condizioni di trim raggiunte con la modifica della posizione del baricentro.

Listato 7.8 Condizione di Trim per $\bar{x}_G = 0.50$

```

Condizione di trim:
Velocità V_0= 257 m/s
Angolo d'attacco alpha_0= 1.8297 deg
Elevatore delta_e_0= -1.9133 deg
Stabilizzatore delta_s_0= 0.23075 deg
Manetta delta_T_0= 0.40827

```

Nelle figure 7.6 e 7.7 nella pagina successiva sono riportati gli andamenti delle leggi temporali impartite ai comandi e delle storie temporali dell'evoluzione del velivolo. In questo caso, essendo il velivolo instabile le variazioni delle grandezze del moto dai valori iniziali non sono più contenute entro i limiti dell'errore numerico; al contrario sono di ordine di grandezza unitario.

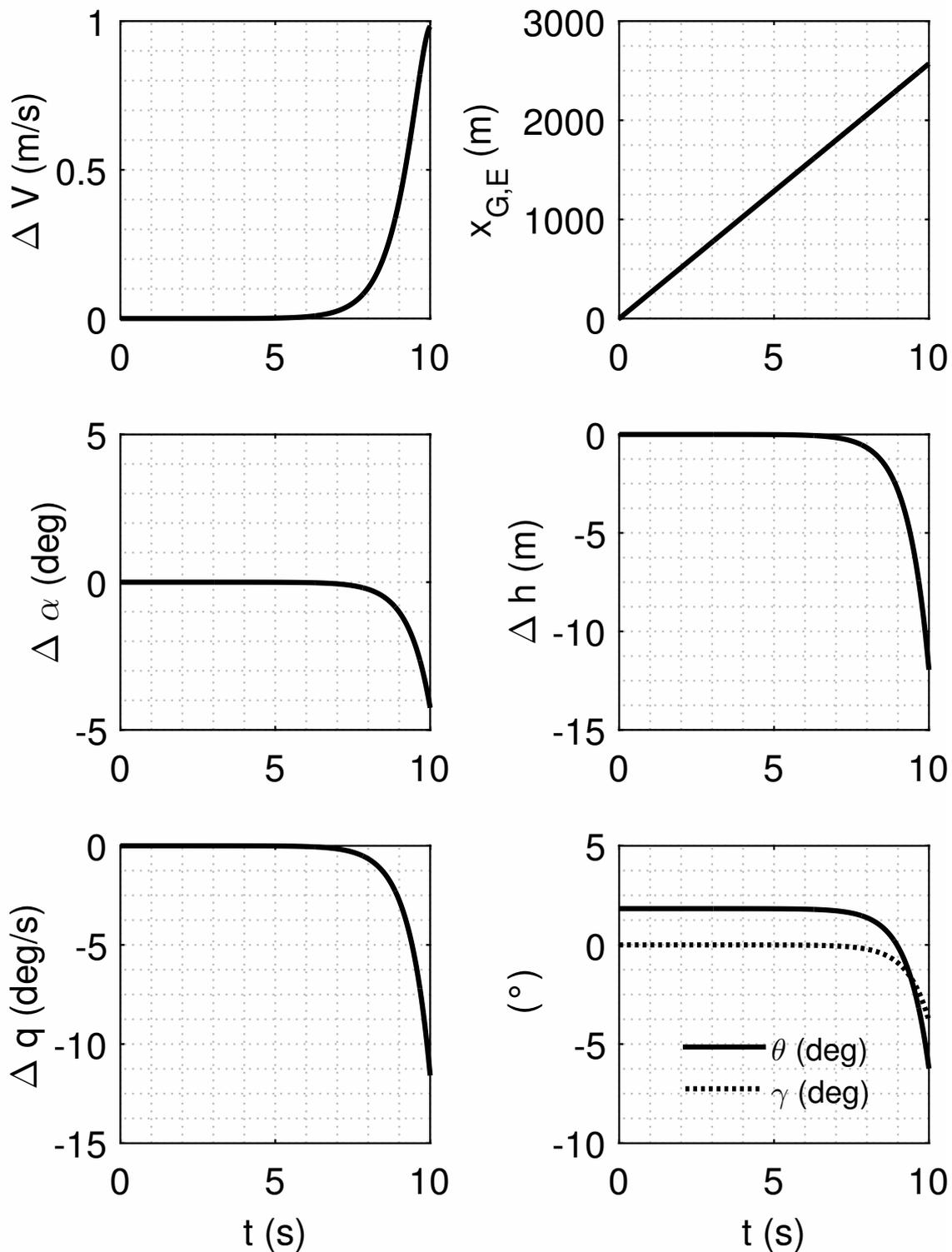


Figura 7.7 Storie temporali delle variabili di stato per leggi di comando identiche, a partire dalla condizione di trim riportata nel listato. Il velivolo è sempre lo stesso assegnato nel listato 7.3 a pagina 81. I $\Delta(*)$ rappresentati sono riferiti rispetto ai valori iniziali. Il baricentro è posto al 50 % della corda.

7.6 Simulazione legge *cabra-picchia* - Esercizio 7.9

Per questo esercizio è stata proposta una legge particolare del comando longitudinale denominata "cabra-picchia" o (*elevator pulse*). Si procede, quindi, all'integrazione delle equazioni del moto a 3-DoF considerando la legge del δ_e proposta in figura 7.8 a pagina 97.

Si immagina che per un tempo $t_1 = 1s$ il pilota mantenga fissa l'escursione δ_e , pari al valore di equilibrio di trim $\delta_{e,0}$ riportato nel listato 7.9.

Listato 7.9 Condizione di Trim per manovra cabra-picchia

```
Condizione di trim:
Velocità V_0= 257 m/s
Angolo d'attacco alpha_0= 2.0283 deg
Elevatore delta_e_0= -4.223 deg
Stabilizzatore delta_s_0= -0.10259 deg
Manetta delta_T_0= 0.4082
```

A partire dall'istante t_1 fino all'istante $t_2 = 2,50s$ si applica una legge $\delta_e(t)$ che incrementa linearmente l'escursione dell'elevatore dal valore $\delta_{e,0}$ al valore $\delta_{e,0} + \Delta\delta_e$, con $\Delta\delta_e = -3deg$. A partire dall'istante t_2 fino all'istante $t_3 = 4s$ si applica una legge $\delta_e(t)$ che riporta l'escursione dell'elevatore al valore iniziale $\delta_{e,0}$. Per il resto della manovra, ovvero per $t_3 \leq t \leq t_f$, si suppone che il pilota mantenga bloccato il comando longitudinale al valore riportato nel listato 7.9 per la condizione di trim. Nel corso di tutta la manovra, gli altri comandi restano invariati nella condizione di trim. Tutte le leggi di comando sono mostrate nella figura 7.8 a pagina 97.

Dover assegnare una legge dell'equilibratore, implica che nel codice venga definita una funzione interpolante di δ_e da passare a `ode45`. Ciò richiede che ad ogni istante t , nel processo d'integrazione, si avrà un valore del δ_e corrispondente al valore che nell'istante t ho per la legge definita.

Il frammento di codice di calcolo che ci ha consentito di effettuare la simulazione è presentato nel listato 7.10 questi è da intendersi come continuo del listato 7.1 a pagina 78, infatti la risoluzione di tale codice muove i suoi passi dalla conoscenza della condizione di trim.

Listato 7.10 Codice legge temporale δ_e

```
%% CABRATA-PICCHIATA

%costruzione legge del tab
t_0 = 0.0;           % initial time
t_1 = 1.0;           % first command law breakpoint   (s)
%grad_delta_e = 0.5267; % time rate of delta_e       (rad/s)
t_2cp = 2.5;
t_3cp = 4;
t_2 = 1.5;           % 2nd breakpoint                (s)
t_3 = 2.0;           % 3rd breakpoint                (s)
t_f = 10.0;          % total observation time        (s)

freq_Hz = 220.;      % frequency of state propagation, 1/delta t

n0      = floor((t_1 - t_0)*freq_Hz);
n1      = floor((t_2 - t_1)*freq_Hz);
n2      = floor((t_3 - t_2)*freq_Hz);
n3      = floor((t_f - t_3)*freq_Hz);
v_time  = [linspace(t_0,t_1,n0),...
           linspace(t_1,t_2,n1),...]
```

```

        linspace(t_2,t_3,n2),...
        linspace(t_3,t_f,n3)...
    ]';
v_time = unique(v_time);
n_times = length(v_time);
%Bisogna definire un vettore tempo e un vettore con i valori da
%interpolare, in una matrice nsteps*2:
%Nella prima riga abbiamo il vettore dei punti tempo, nella seconda i
%valori dei comandi in quegli istanti di tempo.
%L'interpolazione é fatta nel sottoprogramma chiamato
%longEquilibriumDynamicStickFixed, qui si defiscono solo i vettori
%percio devono essere dichiarati globali.

%Costruzione per punti delle curve di comando
vt_delta_T = [v_time, ones(n_times,1)*delta_T_0];
vt_delta_e = [v_time, ones(n_times,1)*x(2)];
vt_delta_s = [v_time, ones(n_times,1)*x(3)];

%CABRAPICCHIA
delta_e_increment_deg = -3.0;
grad_delta_e = ...
    convang(delta_e_increment_deg, 'deg', 'rad')/(t_2cp - t_1);

for ke=1:length(v_time)
    if ( (vt_delta_e(ke,1)>=t_1) && (vt_delta_e(ke,1)<t_2cp))
        vt_delta_e(ke,2) = ...
            vt_delta_e(ke-1,2) ...
            + (grad_delta_e/freq_Hz);
    elseif ( (vt_delta_e(ke,1)>=t_2cp) && (vt_delta_e(ke,1)<t_3cp))
        vt_delta_e(ke,2) = ...
            vt_delta_e(ke-1,2) ...
            - (grad_delta_e/freq_Hz);
    end
end

vt_delta_e1 = vt_delta_e;

%COMANDI BLOCCATI SIMULAZIONE CABRA PICCHIA
y0 = [ ...
    V_0; ...
    alpha_0; ...
    q_0; ...
    0.0; ... % x_G_E_0
    z_0; ...
    theta_0 ...
];

%Integrazione
[T1,Y1] = ode45(@eqLongDynamicStickFixed, [0 t_f] ,y0');

%% POLT RISULTATI...

```

La figura 7.9 a pagina 98 riporta le oscillazioni delle variabili di stato nei primi 10s di manovra, mentre la figura 7.10 a pagina 99 mostra la storia temporale delle variabili di stato per un lungo periodo pari a $t_f = 120s$.

Si possono notare oscillazioni significative dell'angolo d'attacco e della velocità angolare di beccheggio nel breve periodo, mentre le variabili che mostrano significative oscillazioni nel lungo periodo sono la velocità, la quota e l'angolo di elevazione.

Nella figura 7.11 a pagina 100 sono rappresentate le storie temporali dei fattori di carico f_{x_A} e f_{z_A} .

Nella figura 7.12 a pagina 100 è riportata l'accelerazione angolare di beccheggio \dot{q} nei primi 10 s della manovra. Si osserva che:

- la \dot{q} in fase di cabrata tende inizialmente ad aumentare a causa della variazione positiva del momento di beccheggio di controllo legata alla diminuzione dell'escursione dell'equilibratore, che raggiunge un massimo e poi diminuisce. All'inizio della fase di picchata, subisce una diminuzione fino a raggiungere un valore minimo, dopodichè aumenta nuovamente e al termine della manovra (nella fase di risposta libera del velivolo) oscilla attorno allo 0 smorzandosi rapidamente;
- il fattore di carico f_{z_A} inizialmente pari a 1 (volo non accelerato) rimane costante per un brevissimo istante temporale, cresce fino ad un valore massimo circa pari a 3, raggiunto in corrispondenza del minimo dell'accelerazione angolare di beccheggio, poi diminuisce per oscillare attorno al valore unitario nella fase di risposta libera;
- il fattore di carico longitudinale f_{x_A} presenta lo stesso andamento di f_{z_A} ma con valori molto minori (circa 1/10).

Nella figura 7.12 a pagina 100 è riportata l'accelerazione angolare di beccheggio \dot{q} nei primi 10 s della manovra. Si osserva che:

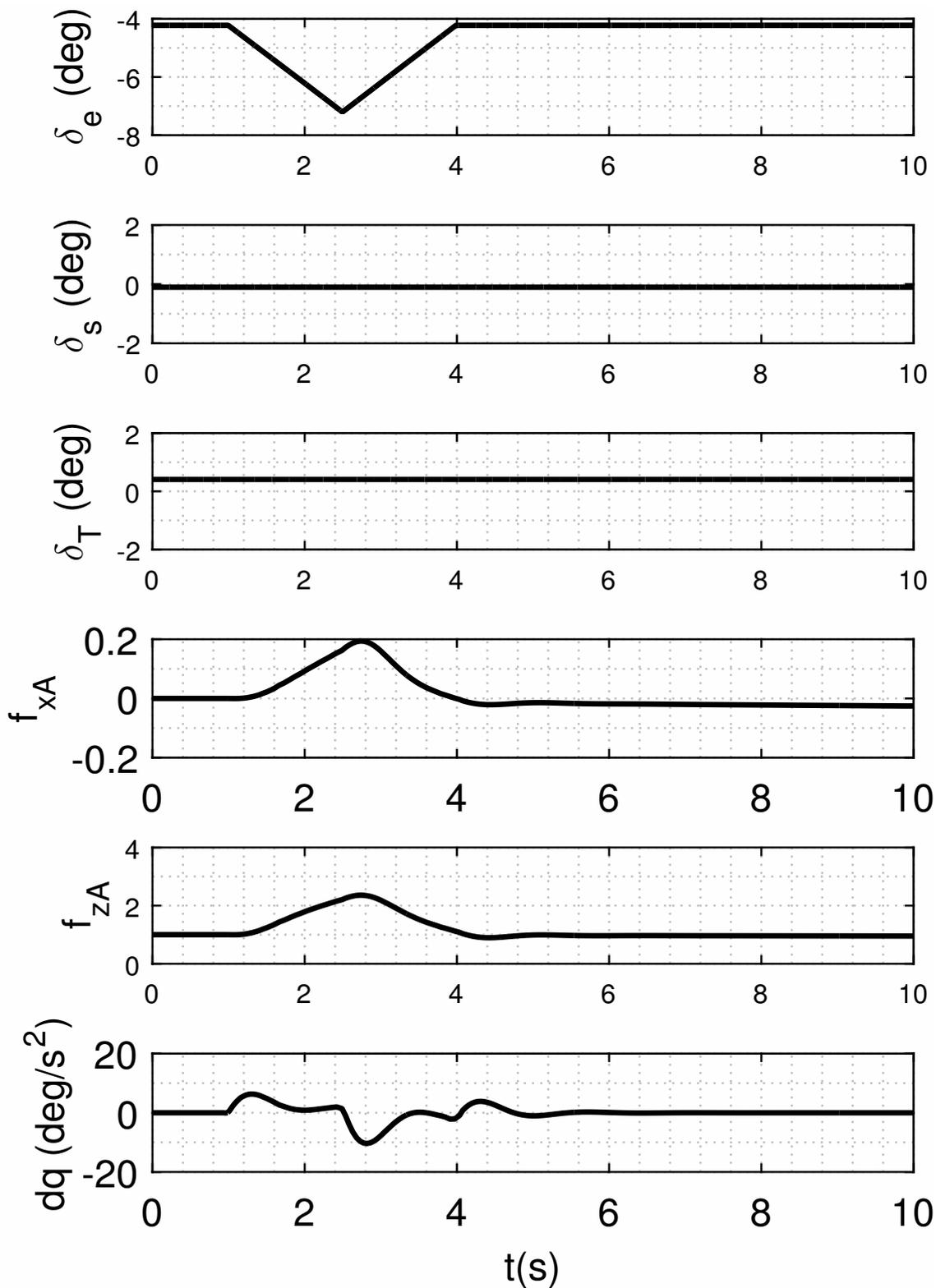


Figura 7.8 Storie temporali delle leggi di comando. La manovra *cabra-picchia* viene realizzata muovendo l'equilibratore δ_e come riportato. Le leggi di comando dello stabilizzatore e della spinta restano costanti a partire dalla condizione di trim riportata nel listato 7.9. Condizioni valutate per il velivolo assegnato nel listato 7.3 a pagina 81.

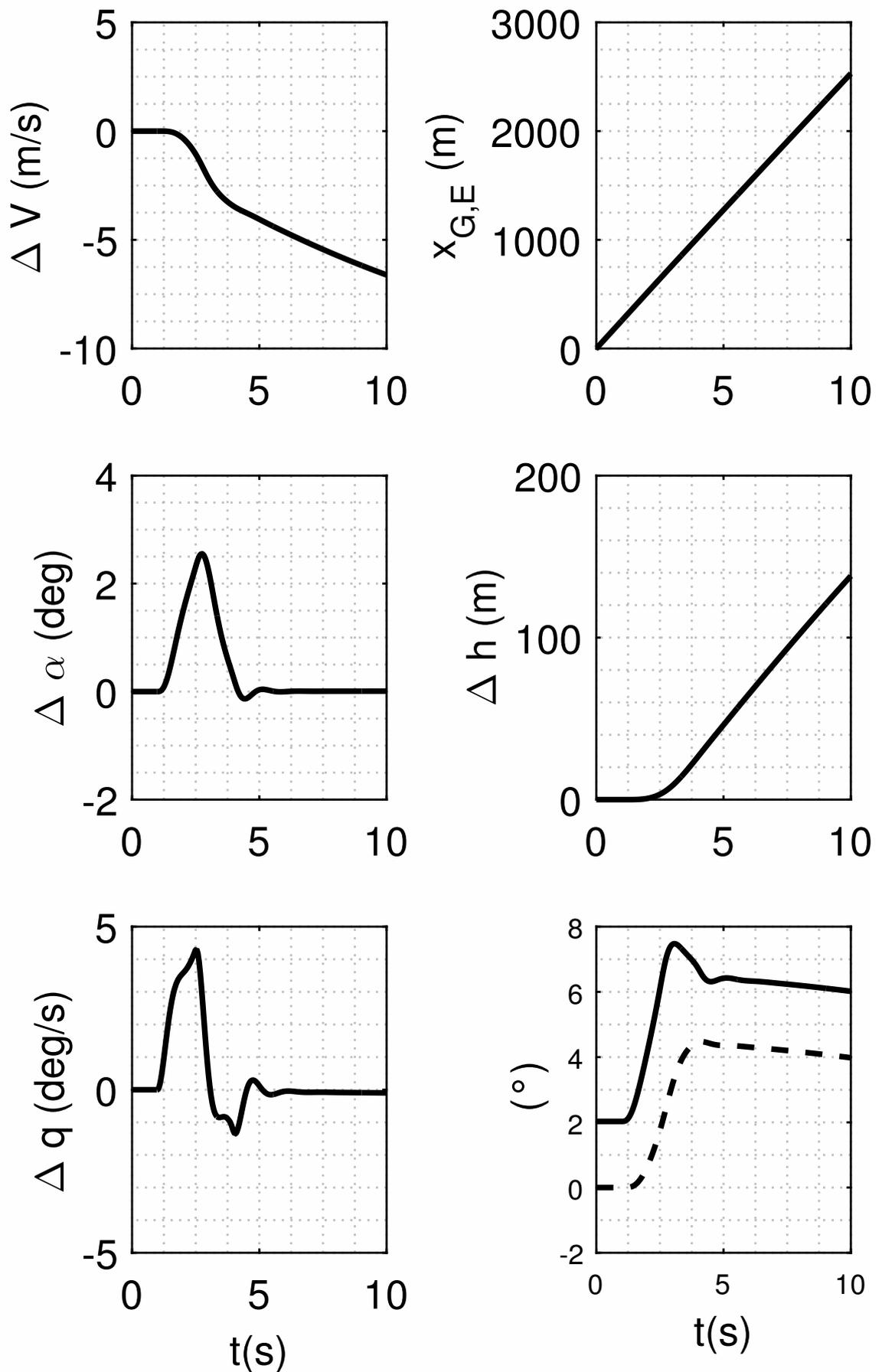


Figura 7.9 Storie temporali delle variabili di stato per la legge di comando "cabra-picchia" rappresentata in figura 7.8 nella pagina precedente, a partire dalla condizione di trim proposta nel listato 7.9 a pagina 94.

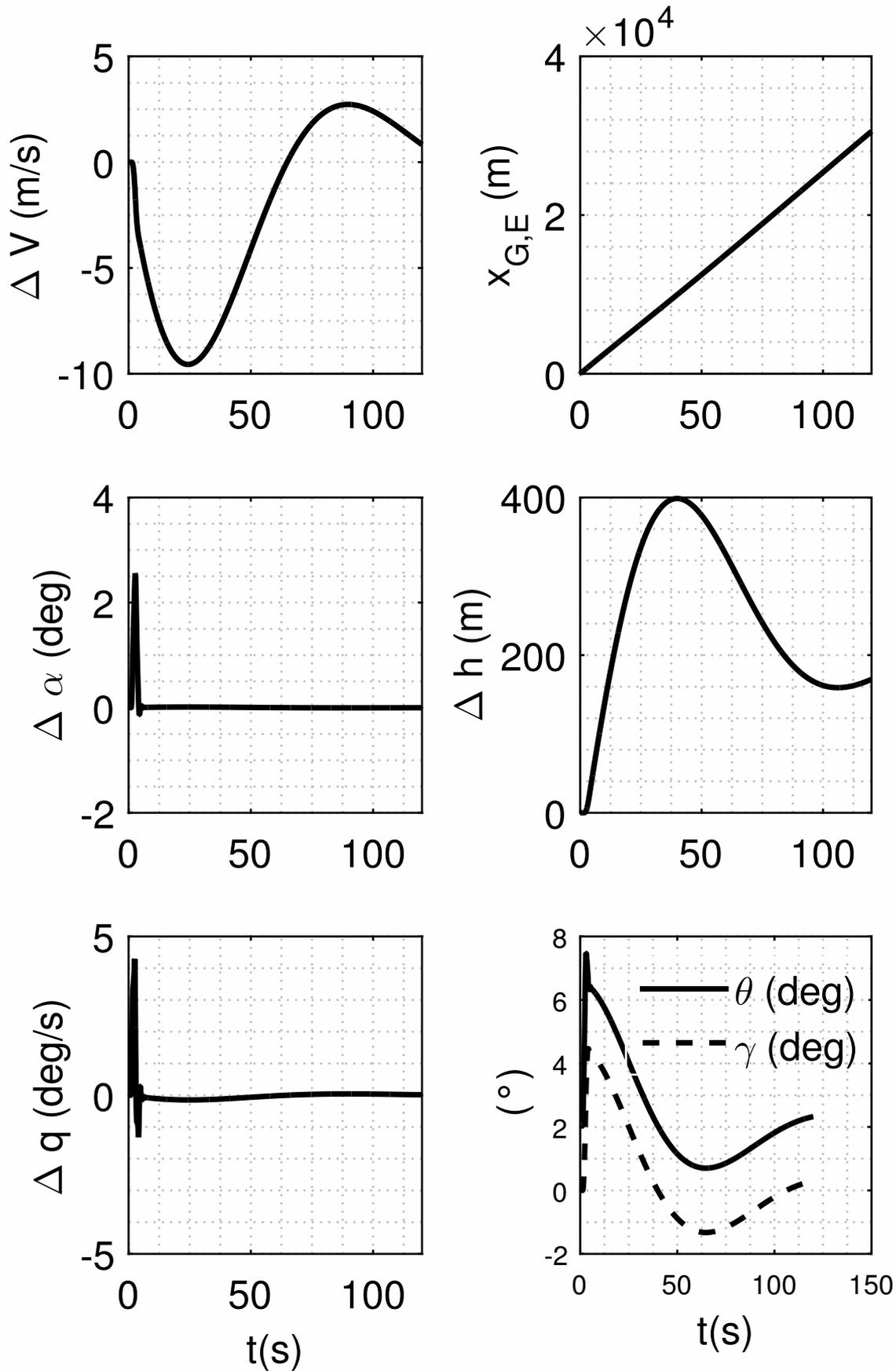


Figura 7.10 Storie temporali delle variabili di stato per la legge di comando "cabra-picchia" rappresentata in figura 7.8 a pagina 97, a partire dalla condizione di trim proposta nel listato 7.9 a pagina 94.

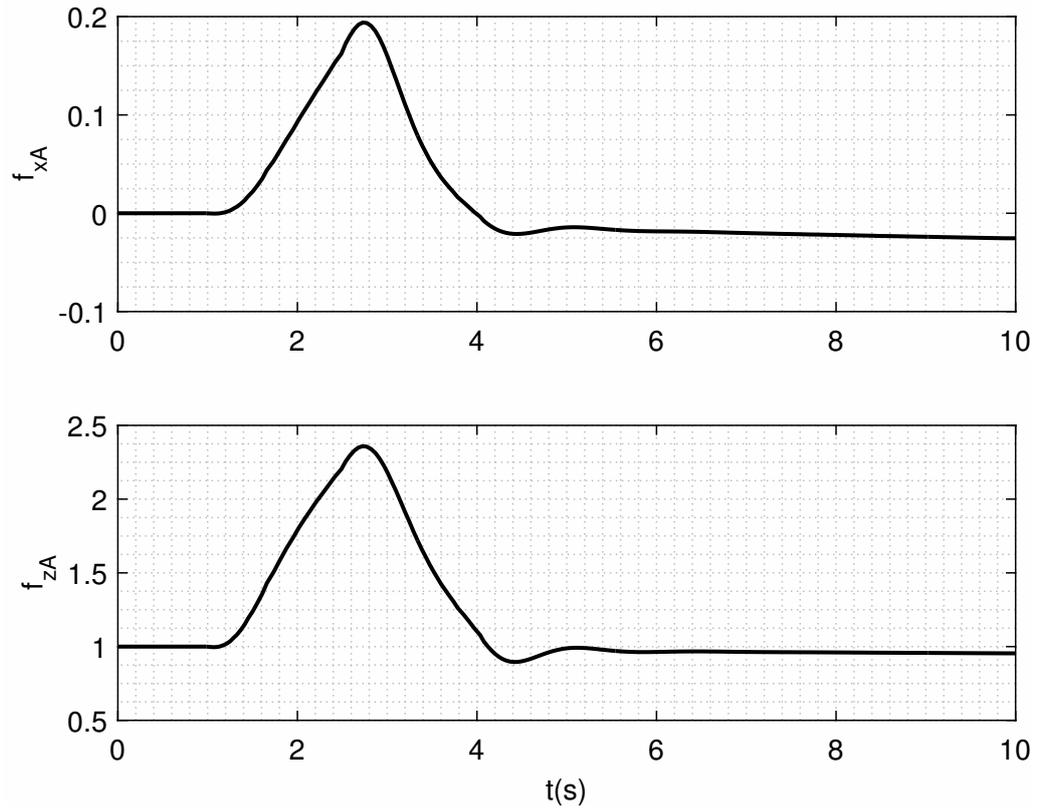


Figura 7.11 Andamento delle fattori di carico per la legge "cabra-picchia" nel breve periodo, a partire dalla condizione di trim riportata nel listato e per il velivolo assegnato precedentemente.

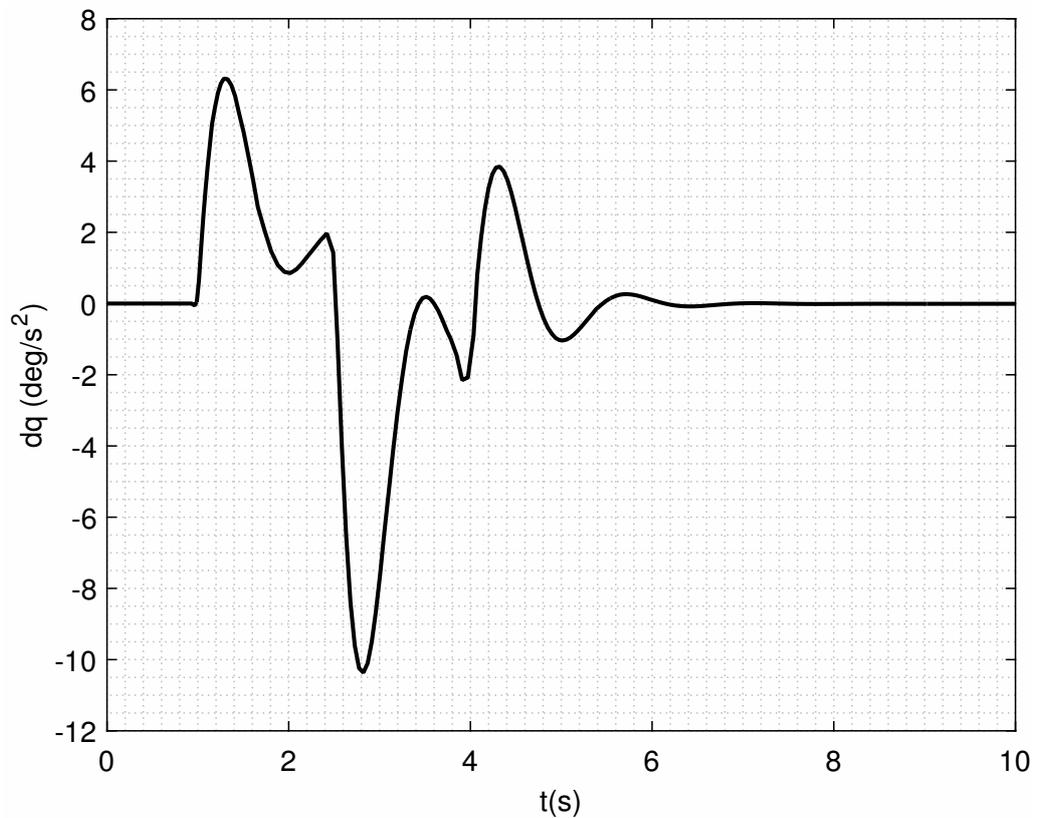


Figura 7.12 Andamento dell'accelerazione angolare di beccheggio \dot{q} nei primi 10 secondi della manovra, a partire dalla condizione di trim riportata nel listato e per il velivolo assegnato precedentemente.

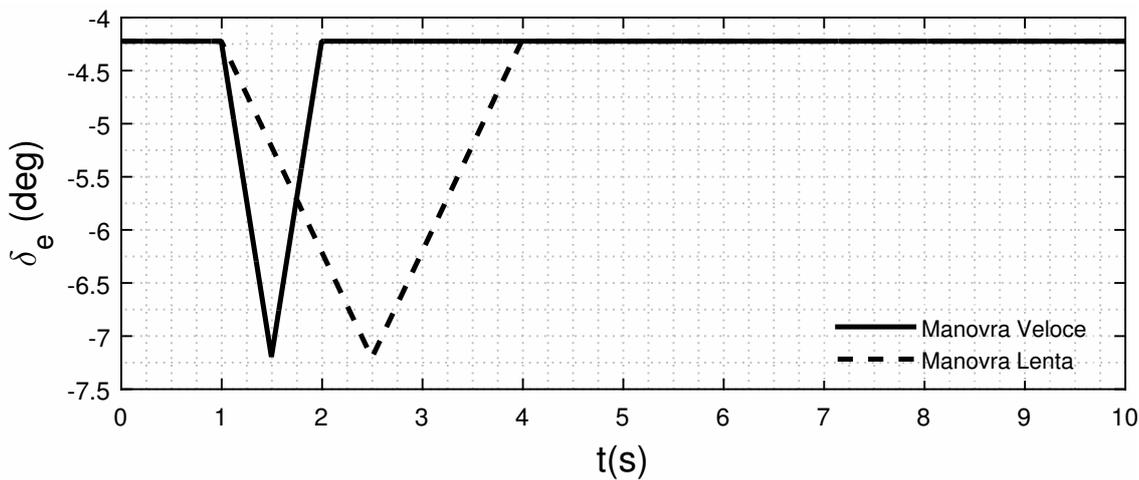


Figura 7.13 Confronto tra le due diverse leggi di comando per la deflessione dell'equilibratore δ_e .

7.7 Variazione della legge del δ_e - Esercizio 7.10

Osservati gli andamenti temporali delle variabili di stato per una manovra *cabra-picchia* particolare, si valutano adesso gli effetti sulle variabili di stato provocati da rotazioni dell'equilibratore con diversa intensità nell'intervallo $t_3 - t_1$.

Le leggi temporali dei comandi longitudinali che sono state confrontate sono la legge nella figura 7.8 a pagina 97 e la legge che si ottiene per gli assegnati valori:

$$t_1 = 1s, t_2 = 1.5s, t_3 = 2s \text{ e } \Delta\delta_e = -3deg \quad (7.17)$$

che corrisponde ad una manovra più rapida e impulsiva.

Il confronto tra le due diverse leggi temporali è stato effettuato riportato nella figura 7.13.

Sono stati confrontati gli andamenti delle variabili di stato a seguito di una manovra Lenta e di una manovra Veloce e per due periodi di tempo, uno breve pari a 10s e per uno lungo pari a 120s. Non sono stati riportati i codici di calcolo eseguiti in Matlab, in quanto ricalcano fedelmente il listato rappresentato 7.10 a pagina 94 in cui una volta è stato eseguito lanciando la manovra per un tempo di simulazione finale di 10s e un'altra volta per una manovra di 120s.

Dall'analisi dei risultati prodotti, confrontando i grafici per le variabili di stato nelle figure 7.14 a pagina 103 e 7.15 a pagina 104 si possono trarre le seguenti considerazioni:

- Si nota che nel breve periodo e nel lungo periodo le oscillazioni della variazione di velocità sono maggiormente influenzate da una manovra lenta, mentre invece una manovra veloce produce oscillazioni di velocità inferiori sia nel lungo che nel breve periodo;
- Si nota che nel breve periodo e nel lungo periodo le oscillazioni della variazione dell'angolo di attacco α sono maggiormente influenzate da una manovra lenta, mentre una manovra veloce produce oscillazioni di angolo di attacco inferiori sia nel lungo che nel breve periodo;
- Si nota che nel breve periodo una manovra lenta produce un'oscillazione minore della variazione della variazione della velocità di beccheggio, mentre nel lungo

periodo accade il contrario. Infatti, la variazione della velocità di beccheggio nel lungo periodo è maggiormente influenzata da una manovra veloce;

- Si nota una perfetta sovrapposizione della posizione del baricentro a seguito di una manovra veloce e di una manovra lenta sia nel breve che nel lungo periodo;
- Si nota che nel breve periodo e nel lungo periodo una manovra lenta produce una maggiore variazione di quota rispetto al valore iniziale di equilibrio;

Nel breve periodo si hanno oscillazioni della variazione dell'angolo di attacco α e della variazione della velocità V più apprezzabili per una manovra veloce. Nel caso di manovra più veloce, con legge riportata nell'equazione 7.17 nella pagina precedente si notano oscillazioni con minore ampiezza della velocità, della quota e dell'angolo di elevazione nel lungo periodo, vedi figura 7.15 a pagina 104. Mentre, nel breve periodo, si hanno oscillazioni dell'angolo d'attacco e della velocità di beccheggio più significative, ovvero caratterizzate da una frequenza maggiore e uno smorzamento minore vedi figura 7.14 a fronte. Si può desumere che una manovra "cabra-picchia" con una durata del forzamento $t_3 - t_1$ molto piccola (al limite infinitesima) eccita prevalentemente (al limite soltanto) il breve periodo.

Dalla figura 7.16 a pagina 105 si desume che nel caso di manovra veloce l'andamento dell'accelerazione angolare di beccheggio asseconda più fedelmente la manovra, presentando un massimo in corrispondenza del valore minimo (massimo in valore assoluto) dell'angolo di deflessione dell'equilibratore ed un minimo all'istante di tempo in cui termina la manovra (δ_e torna al valore $\delta_{e,0}$). Invece il fattore di carico normale tende a raggiungere il massimo alla fine della fase di picchiata.

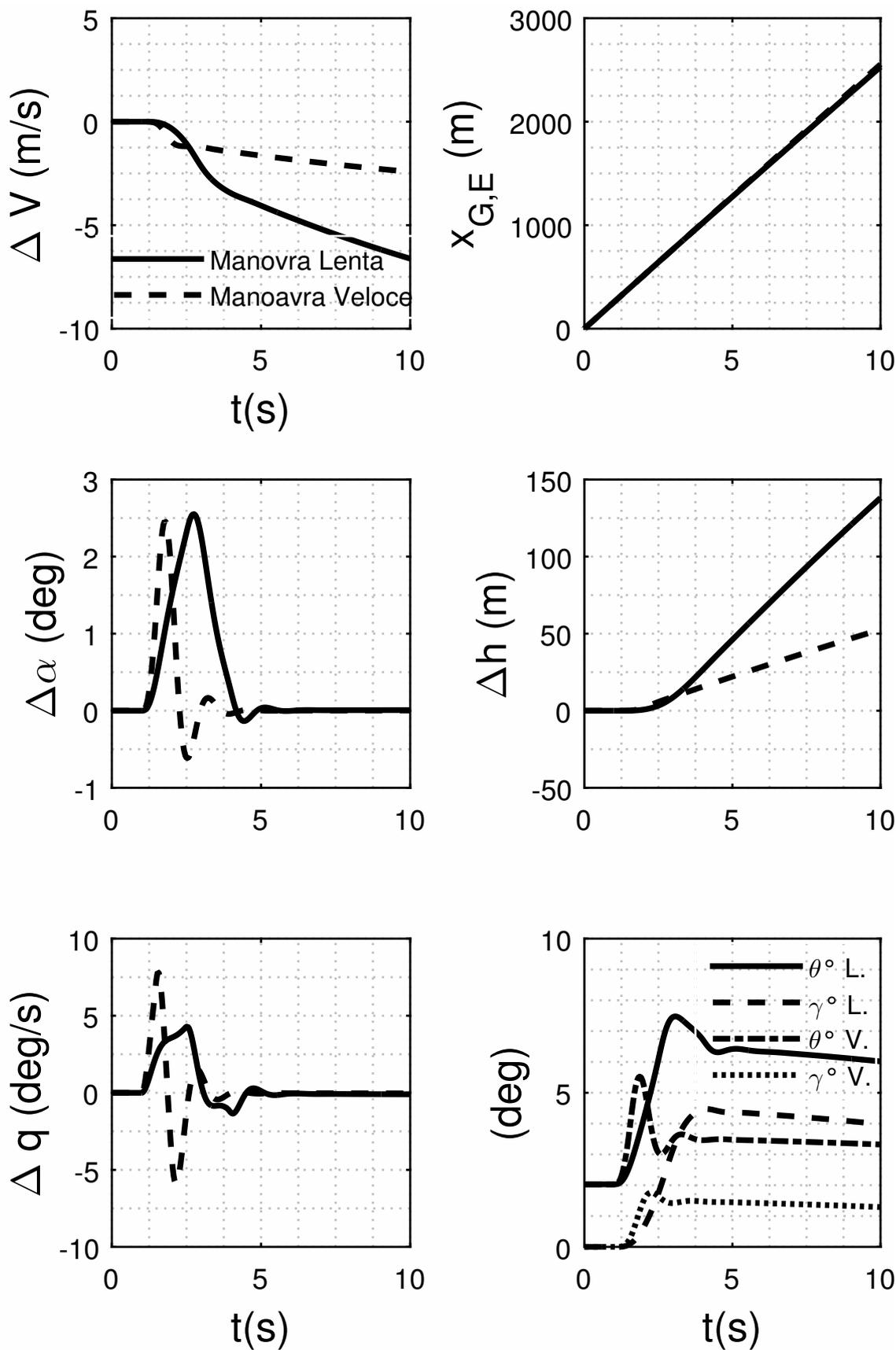


Figura 7.14 Confronto tra le storie temporali delle variabili di stato per due leggi di comando "cabra-picchia" rappresentate nella figura 7.13 a pagina 101, a partire dalla condizione di trim riportata nel listato 7.9 a pagina 94. Il velivolo è sempre lo stesso assegnato nel listato 7.3 a pagina 81. Sono rappresentate le due diverse leggi di comando nel Breve Periodo e le variazioni sono rispetto ai valori iniziali di trim 7.9 a pagina 94.

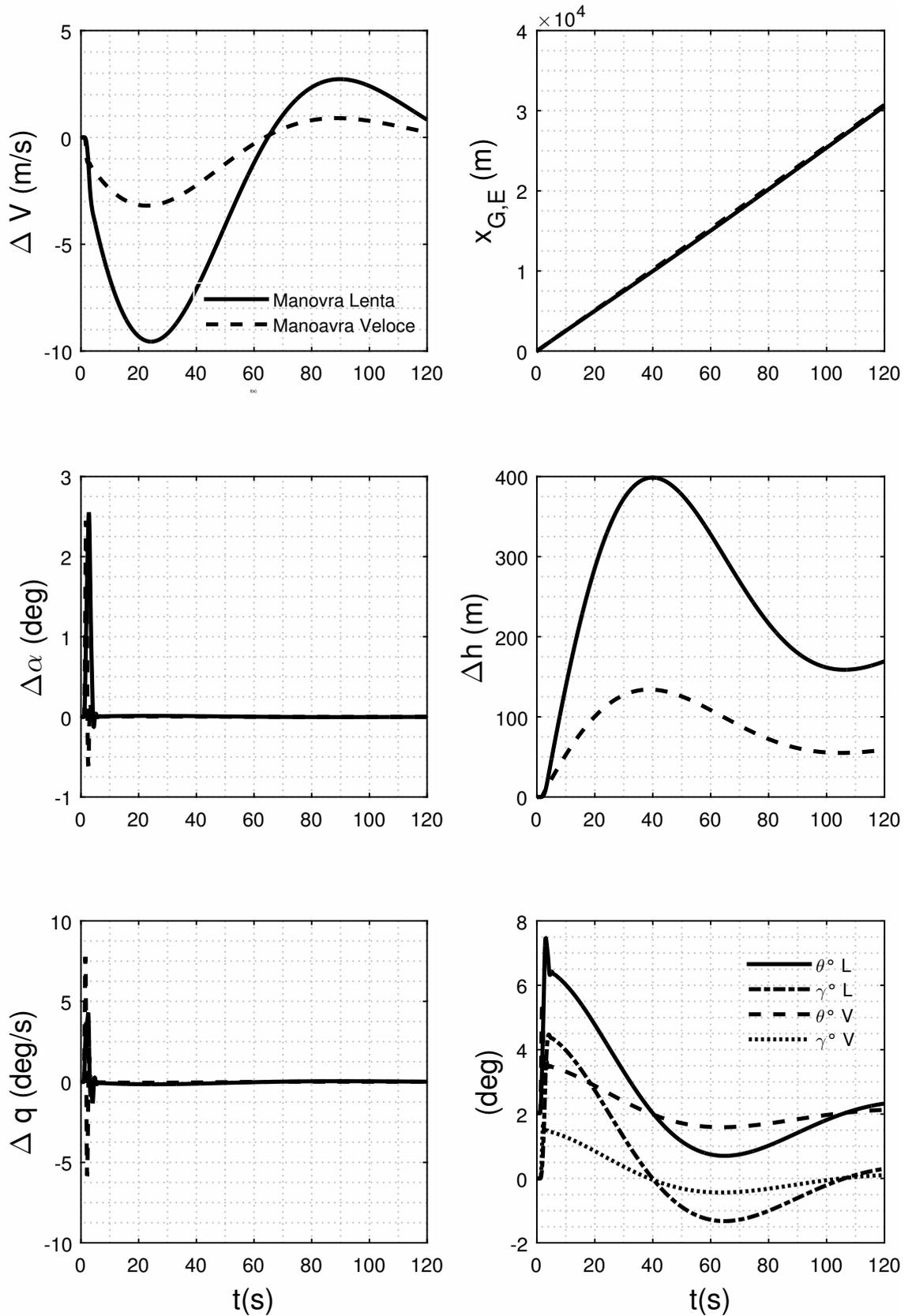


Figura 7.15 Confronto tra le storie temporali delle variabili di stato per due leggi di comando "cabra-picchia" rappresentate nella figura 7.13 a pagina 101, a partire dalla stessa condizione di trim riportata nel listato 7.9 a pagina 94. Il velivolo è quello assegnato nel listato 7.3 a pagina 81. Sono rappresentate le due diverse leggi di comando nel Lungo Periodo e le variazioni sono rispetto ai valori iniziali di trim 7.9 a pagina 94.

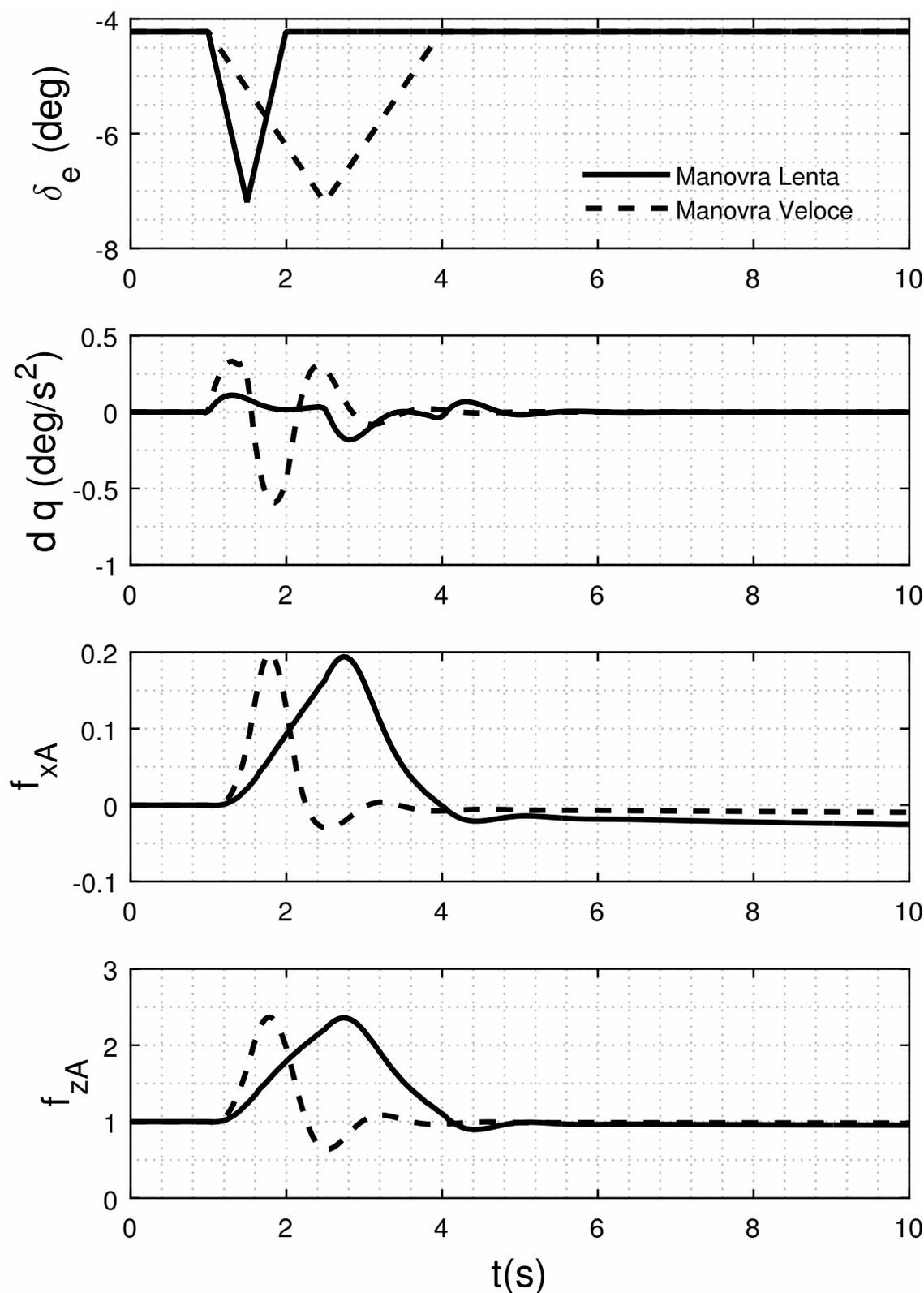


Figura 7.16 Due leggi di comando "cabra-picchia" a confronto. Si riportano nella stessa scala temporale le storie dei fattori di carico e dell'accelerazione angolare corrispondenti, a partire dalla condizione di trim riportata nel listato. Il velivolo è quello assegnato nel paragrafo precedente.

Modello aerodinamico di un velivolo completo con Digital DATCOM

Indice

9.1	Introduzione	107
9.2	Uso del programma	108
9.3	Formato del file di input	108
9.4	Un esempio di file di input	111
9.5	Caratteristiche aerodinamiche con Digital DATCOM - <i>Esercizio 9-1</i>	113
9.6	Elevatore e flap con <i>Digital DATCOM+</i> - <i>Esercizio 9-2</i>	121
9.7	Aerodinamica del velivolo <i>Tecnam P2006T</i> - <i>Esercizio 9.3</i>	126
9.8	Caratteristiche aerodinamiche del velivolo <i>ATR72</i> - <i>Esercizio 9.4</i> . .	139

9.1 Introduzione

Lo scopo di questa esercitazione è il calcolo delle caratteristiche aerodinamiche di diversi velivoli con l'ausilio del software *Digital DATCOM*.

Digital DATCOM implementa una serie di metodi semiempirici che forniscono stime dei coefficienti aerodinamici basate su risultati sperimentali ottenuti in galleria del vento. Essi si riferiscono a numerose prove eseguite su varie configurazioni aerodinamiche e tengono conto degli effetti della viscosità, della comprimibilità e delle interferenze tra i vari componenti dell'aeromobile (aspetti spesso trascurati dai metodi puramente teorici).

Nel presente elaborato si procede all'analisi aerodinamica del velivolo per diverse configurazioni: (a) *Wing*, (b) *Wing-Body*, (c) *Wing-Body-Vertical Tail* e (d) *velivolo completo*. Per ciascuna di queste configurazioni si determina l'andamento dei coefficienti di forza e momento in funzione degli angoli della corrente f

Sono analizzati due velivoli: il primo è uno dei velivoli presenti nella cartella degli esempi di Digital DATCOM+, il Cessna Citation e il secondo è un Tecnam P2006T, per il quale è stato creato un file di input *ex-novo*.

9.2 Uso del programma

L'eseguibile di Digital DATCOM, una volta avviato, cerca di aprire un file input la cui estensione dipende dalla versione del software che si sta utilizzando. Per gli esercizi di questo elaborato si è impiegata la versione Digital DATCOM+, che accetta in input un file di estensione .dcm e genera, in seguito alla sua esecuzione, una serie di file di output fra i quali il principale è quello di estensione .out contenente i risultati dell'analisi ed eventuali messaggi diagnostici di errore.

Affinché il file venga letto con successo (così da ottenere un output privo di errori), è necessario compilarlo adeguatamente. Il file .dcm è sostanzialmente un file di testo, strutturato in blocche che prendono il nome di *namelist statements*. una *namelist* è una lista di nomi di variabili. Nel file input è necessario assegnare un valore a tutte quelle variabili che consentono di definire la configurazione del velivolo e la condizione di volo di interesse e tanto può essere fatto mediante la scrittura di opportune istruzioni secondo una precisa sintassi. L'insieme delle istruzioni che assegnano i parametri di una stessa *namelist* costituisce un *namelist statement*.

Oltre i *namelist statements*, il file letto dal sorgente contiene anche altre istruzioni, le cosiddette *command card*, che servono per selezionare determinate impostazioni, come l'unità di misura delle lunghezze o delle derivate istazionarie, il salvataggio dei risultati parziali, il calcolo delle condizioni di trim, eccetera.

9.3 Formato del file di input

Segue un elenco delle principali *namelist* accettate da Digital DATCOM ed utilizzate nell'esercitazione per compilare i file di input. Per ciascuna di esse sono riportate le variabili da assegnare e tra parentesi le corrispondenti *keywords* che le rappresentano.

\$FLTCON ... \$ *Flight conditions*. Assegna i parametri che definiscono la condizione di volo ed il peso del velivolo. In particolare, in tutti gli esercizi sono stati assegnati i seguenti parametri:

- peso (WT);
- quota (NALT, ALT);
- angolo d'attacco (NALPHA, ALPHA);
- numero di Mach (NMACH, MACH);
- numero di Reynolds per unità di lunghezza (RNNUB).

\$OPTIONS ... \$ *Reference parameters*. Assegna le grandezze di riferimento:

- apertura alare (BLREF);
- corda media aerodinamica (CBARR);

- superficie alare (SREF).

\$SYNTHS ... \$ *Synthesis parameters*. Assegna le grandezze necessarie alla sintesi dei componenti della configurazione geometrica del velivolo. A seconda della configurazione di interesse, si definiscono i parametri seguenti:

- posizione longitudinale e verticale del baricentro come polo dei momenti (XCG, YCG);
- posizione longitudinale e verticale dell'ala rispetto alla fusoliera (XW, YW);
- angolo di calettamento della corda di radice dell'ala rispetto al *datum* XY (ALIW);
- posizione longitudinale e verticale del piano di coda orizzontale rispetto alla fusoliera (XH, YH);
- angolo di calettamento della corda di radice del piano di coda orizzontale rispetto al *datum* XY (ALIH);
- posizione longitudinale e verticale del piano di coda verticale rispetto alla fusoliera (XV, YV);
- posizione longitudinale e verticale della pinna ventrale verticale (XVF, YVF).

\$BODY ... \$ *Body parameters*. Assegna le principali grandezze geometriche della fusoliera:

- posizione e limiti di ingombro trasversale delle sezioni di fusoliera (NX, X, ZU, ZL);
- area delle sezioni di fusoliera (S);
- larghezza (ingombro laterale) delle sezioni di fusoliera (R);
- forma del *nose* e della parte posteriore (BNOSE, BTAIL, BLA).

\$WGPLNF ... \$ *Wing planform parameters*. Consente di definire la forma in pianta dell'ala:

- tipo della forma in pianta, trapezia, a doppia delta o *cranked* (TYPE);
- corda di radice, corda di estremità e corda al *breakpoint* (CHRDR, CHRDT, CHRDBP);
- semi-apertura alare, semi-apertura della parte esposta, lunghezza del tronco di ala esterna (SSPN, SPPNE, SSPNO);
- angoli di svergolamento geometrico del profilo all'estremità (TWISTA);
- angoli diedro dei due tronchi di ala (DHDADI, DHDADO).

\$HTPLNF ... \$ *Horizontal tail planform parameters*. Consente di definire la forma in pianta del piano di coda orizzontale. I parametri da assegnare sono gli stessi della *namelist* \$WGPLNF ...\$.

\$VTPLNF ... \$ *Vertical tail planform parameters*. Consente di definire la forma in pianta del piano di coda verticale. I parametri da assegnare sono gli stessi della *namelist* \$WGPLNF ...\$.

\$VFPLNF ...\$ *Vertical Finner planform parameters.* Consente di definire la forma in pianta dell'eventuale pinna ventrale verticale. I parametri da assegnare sono gli stessi della *namelist \$WGPLNF ...\$*.

\$SYMFLP ...\$ *Symmetrical Flap Deflection Parameters.* Consente di impostare le grandezze geometriche e aerodinamiche di tutte le superfici a deflessione simmetrica (flaps, elevatore). Le variabili da assegnare sono:

- tipologia di flap (FTYPE, NTYPS);
- angolo di deflessione della superficie (NDELTA, DETLA);
- forma del *nose* e geometria del bordo d'uscita del profilo (NTYPE, PHETE, PHETEP);
- stazioni lungo l'apertura di inizio e di estremità della superficie mobile (SPANF, SPANFO);
- corda dei profili alle stazioni di inizio e di estremità (CHRDFI, DHRDFO).

\$PROPWR ...\$ *Propulsion parameters for Propeller Power Effect.* Consente di definire i parametri di un sistema di propulsione ad elica:

- angolo di calettamento dell'asse di spinta (AIETLP);
- numero di motori (NENGSP);
- coefficiente di spinta (THSTCP);
- posizione longitudinale e verticale del mozzo, posizione laterale del motore (PHALOC, PHVLOC, YP);
- numero di pale per elica, raggio del disco (NOPBPE, PRPRAD).

\$JETPWR ...\$ *Jet Power Effect Parameters.* Consente di definire i parametri di un sistema propulsivo a getto:

- angolo di calettamento dell'asse di spinta (AIETLJ);
- numero di motori (NENG SJ);
- coefficiente di spinta (THSTCJ);
- posizione longitudinale ed area della sezione di ingresso (JIALOC, JINTLA);
- posizione longitudinale e verticale della sezione di uscita del getto, raggio della sezione di uscita (JEALOC, JEVLOC, JERAD);
- posizione laterale del motore (JELLOC);
- temperatura ambiente (AMBSTP, AMBTMP).

\$ASYFLP ...\$ *Asymmetrical Control Deflection parameters.* Definisce le aerosuperfici a deflessione asimmetrica (alettoni e spoiler). Le variabili di questa *namelist* sono identiche a quelle della *namelist \$SYMFLP...\$*, ma bisogna distinguere fra l'angolo di deflessione della superficie destra (DELTAR) e quello della superficie sinistra (DELTAL).

Si noti che le *namelist* sono sempre delimitate dal carattere \$.

L'elenco riportato non è comprensivo di tutte le *namelist* del programma. Ve ne sono altre che non sono state presentate, in quanto non se ne è fatto uso nello svolgimento degli esercizi. Fra queste si ricordano, tuttavia, \$*WGNDEF*...\$, che consente di includere l'eventuale effetto suolo, e le tre *namelist* \$*WGSCHR*...\$, \$*HTSCHR*...\$, e \$*VTSCHR*...\$, che permettono di definire la geometria dei profili dell'ala e degli impennaggi di coda. Nella maggior parte dei casi esaminati, trattandosi di profili NACA, i profili delle aerosuperfici sono stati assegnati tramite la *command card*: NACA x y z ...z. In cui x è una lettera che indica la superficie su cui è montato il profilo designato (W per l'ala, H per il piano di coda orizzontale, V per l'impennaggio verticale, eccetera), y è il numero d'ordine della serie (4, 5, 6 rispettivamente per la quarta, quinta e sesta serie) e z ...z è la sigla del profilo stesso (il numero di cifre dipende dalla serie). Infine, le principali *command card* sono:

CASEID Seguito da un nome o una locuzione che identifica il caso in esame. Assegna la denominazione del caso, consentendo all'utente di distinguere nel file di output i risultati relativi a configurazioni diverse. item NEXTCASE Termina la lettura delle istruzioni contenute nel file input ed esegue l'analisi delle configurazioni così definita. I dati relativi a ciascun Case ID vengono distrutti dopo l'esecuzione dello stesso a meno che non sia presente la *command card* SAVE.

SAVE Salva i dati input assegnati nella *namelisti* del caso in esame per l'uso degli stessi nella risoluzione dei casi successivi.

DERIV DEG Alcune derivate aerodinamiche sono espresse in (1/deg).

DIM Seguito da Ft, IN, M o CM specifica l'unità di misura delle lunghezze.

DUMP ALL Stampa nel file output tutte le derivate calcolate dal programma.

PART Output parziale per ogni Case ID.

9.4 Un esempio di file di input

Di seguito è riportato un tipico esempio di file pronto per essere dato in input al programma DATCOM+.

Listato 9.1 file di input per Digital DATCOM+

```

DIM FT
DERIV DEG
DAMP
PART
$FLTCON WT=7000.0, LOOP=2.0,
        NMACH=1.0, MACH(1)=0.4,
        NALT=1.0, ALT(1)=0.0,
        NALPHA=20.0,
        ALSCHD(1)= -16.0, -8.0, -6.0, -4.0, -2.0, 0.0, 2.0, 4.0, 8.0, 9.0,
                10.0, 12.0, 14.0, 16.0, 18.0, 19.0, 20.0, 21.0, 22.0, 24.0,
        STMACH=0.6, TSMACH=1.4, TR=1.0$
$OPTINS SREF=320.8, CBARR=6.75, BLREF=51.7, ROUGFC=0.25E-3$
$SYNTHS XCG=21.9, ZCG=3.125,

```

```

XW=19.1, ZW=3.125, ALIW=2.5,
XH=39.2, ZH=7.75, ALIH=0.0,
XV=36.0, ZV=6.0,
XVF=28.0, ZVF=7.4,
SCALE=1.0, VERTUP=.TRUE.$
$BODY NX=8.0,
X(1)=0.0,1.0,2.7,6.0,8.8,28.5,39.4,44.8,
R(1)=0.0,1.25,2.1,2.7,2.76,2.7,1.25,0.0,
ZU(1)=3.5,4.3,4.8,5.5,7.4,7.4,6.5,5.7,
ZL(1)=3.5,2.5,2.25,2.1,2.0,2.2,4.3,5.7,
BNOSE=1.0, BLN=8.8,
BTAIL=1.0, BLA=19.7,
ITYPE=1.0, METHOD=1.0$
$WGPLNF CHRDR=9.4, CHRDTP=3.01,
SSPN=25.85, SSPNE=23.46,
SAVSI=1.3,
CHSTAT=0.25, TWISTA=-3.0,
DHDADI=3.6,
TYPE=1.0$
NACA W 5 23014
NACA H 4 0010
NACA V 4 0012
NACA F 4 0012
$HTPLNF CHRDR=4.99, CHRDTP=2.48,
SSPN=9.42, SSPNE=9.21,
SAVSI=5.32,
CHSTAT=0.25, TWISTA=0.0,
DHDADI=9.2,
TYPE=1.0$
$VTPLNF CHRDTP=3.63, SSPNE=8.85, SSPN=9.42, CHRDR=8.3,
SAVSI=32.3, CHSTAT=0.25, TYPE=1.0$
$VFPLNF CHRDR=11.8, CHRDTP=0.0, CHSTAT=0.0, DHDADO=0.0,
SAVSI=80.0, SSPN=2.3, SSPNE=2.1, TYPE=1.0$
$JETPWR NENG SJ=2.0, AIETLJ=2.0, THSTCJ=0.0,
JIALOC=25.8, JELLOC=4.33, JEVLOC=5.625,
JEALOC=33.3, JINLTA=2.243,
AMBTMP=59.7, AMBSTP=2116.8, JERAD=0.755$
CASEID TOTAL: TOTAL: Citation II Model 550 Aircraft (Simple)
1 THE FOLLOWING IS A LIST OF ALL INPUT CARDS FOR THIS CASE.
0

```

9.5 Caratteristiche aerodinamiche con Digital DATCOM - *Esercizio 9-1*

Si è scelto uno dei velivoli presenti nella cartella degli esempi di Digital DATCOM+. È stato lanciato il programma di analisi aerodinamica per le diverse configurazioni: (a) *Wing*, (b) *Wing-Body*, (c) *Wing-Body-Vertical Tail*, (d) velivolo completo. Sono state confrontate le caratteristiche aerodinamiche delle suddette configurazioni. Si riportano in grafico i coefficienti di forza e momento in funzione di α_B o di β ove applicabile.

Il primo velivolo preso in esame è il Cessna Citation. Le figure ?? a pagina ?? e ?? a pagina ?? nella pagina successiva mostrano il profilo della fusoliera e le viste laterale e in pianta dell'aeromobile, disegnate a partire dai dati contenuti nel file di input.

Le figure da ?? a pagina ?? a ?? a pagina ?? mostrano le principali caratteristiche aerodinamiche, desumibili dai risultati dell'analisi eseguita da Digital DATCOM+, relativamente alle configurazioni *Wing*, *Wing-Body*, *Wing-Body-Vertical Tail* e velivolo completo.

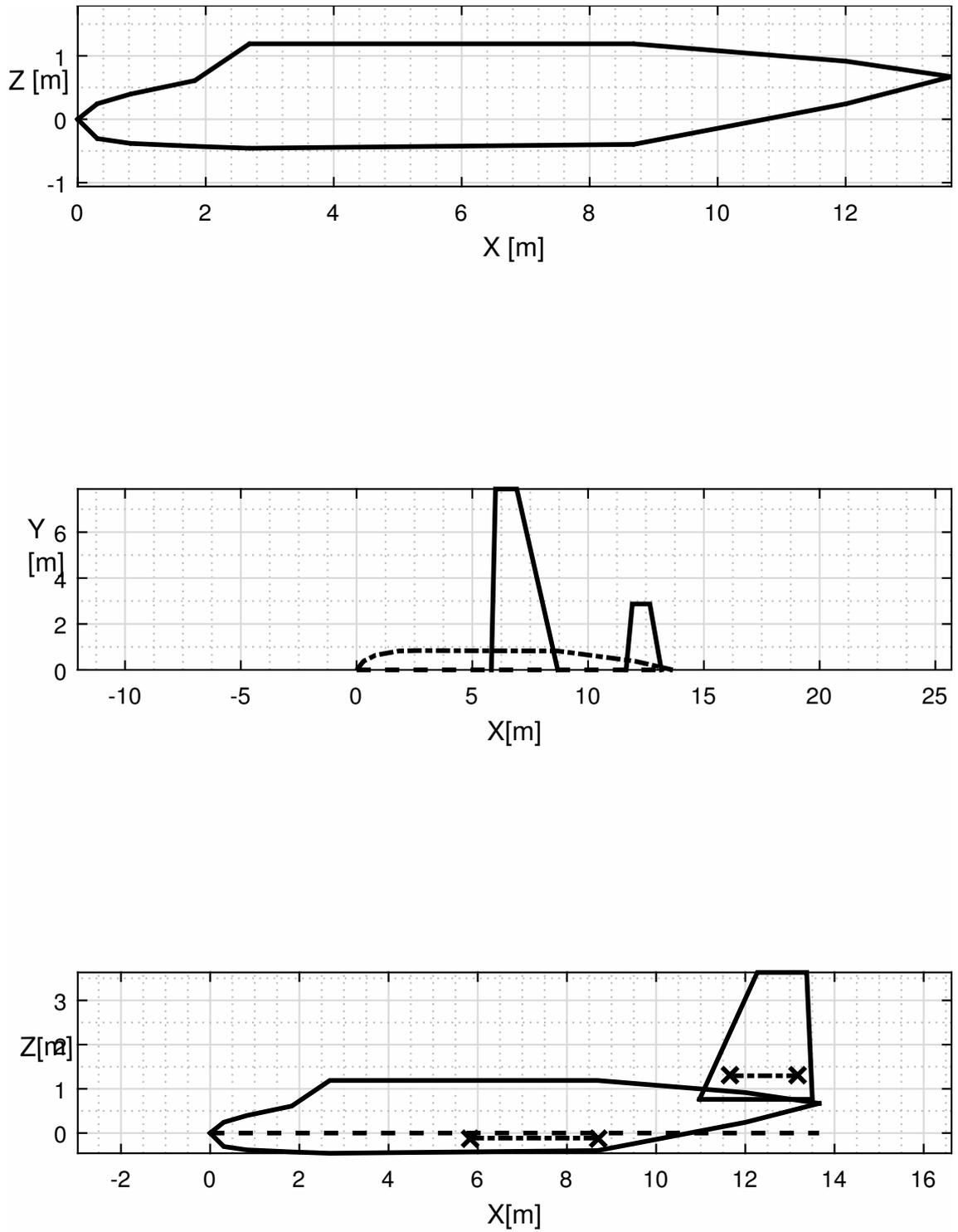


Figura 9.1 Viste Cessna Citation

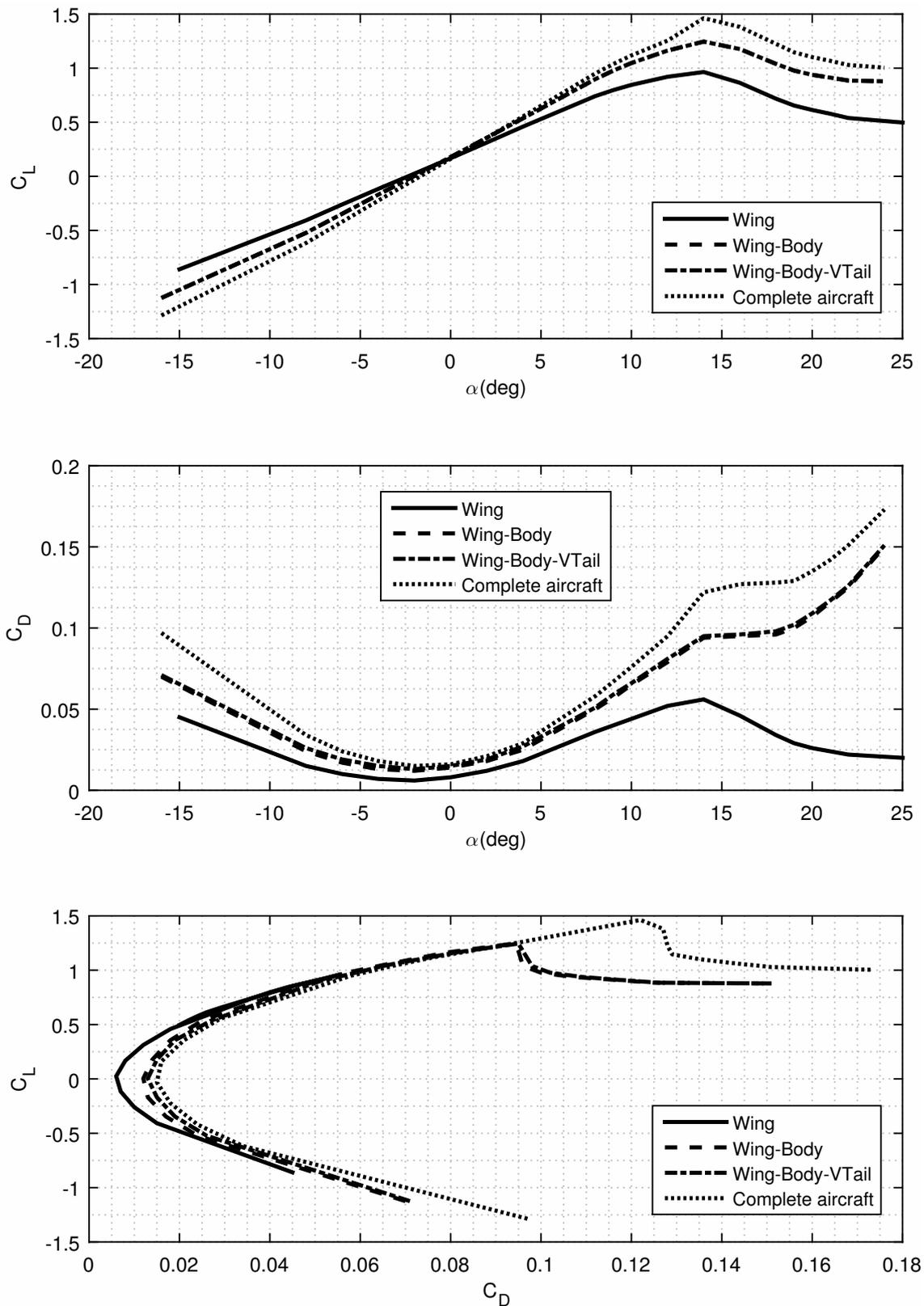


Figura 9.2 Velivolo Cessna Citation II 500. Curve del coefficiente di portanza C_L , del coefficiente di resistenza C_D e della polare per le configurazioni *Wing*, *Wing-Body*, *Wing-Body-Vertical Tail* e velivolo completo.

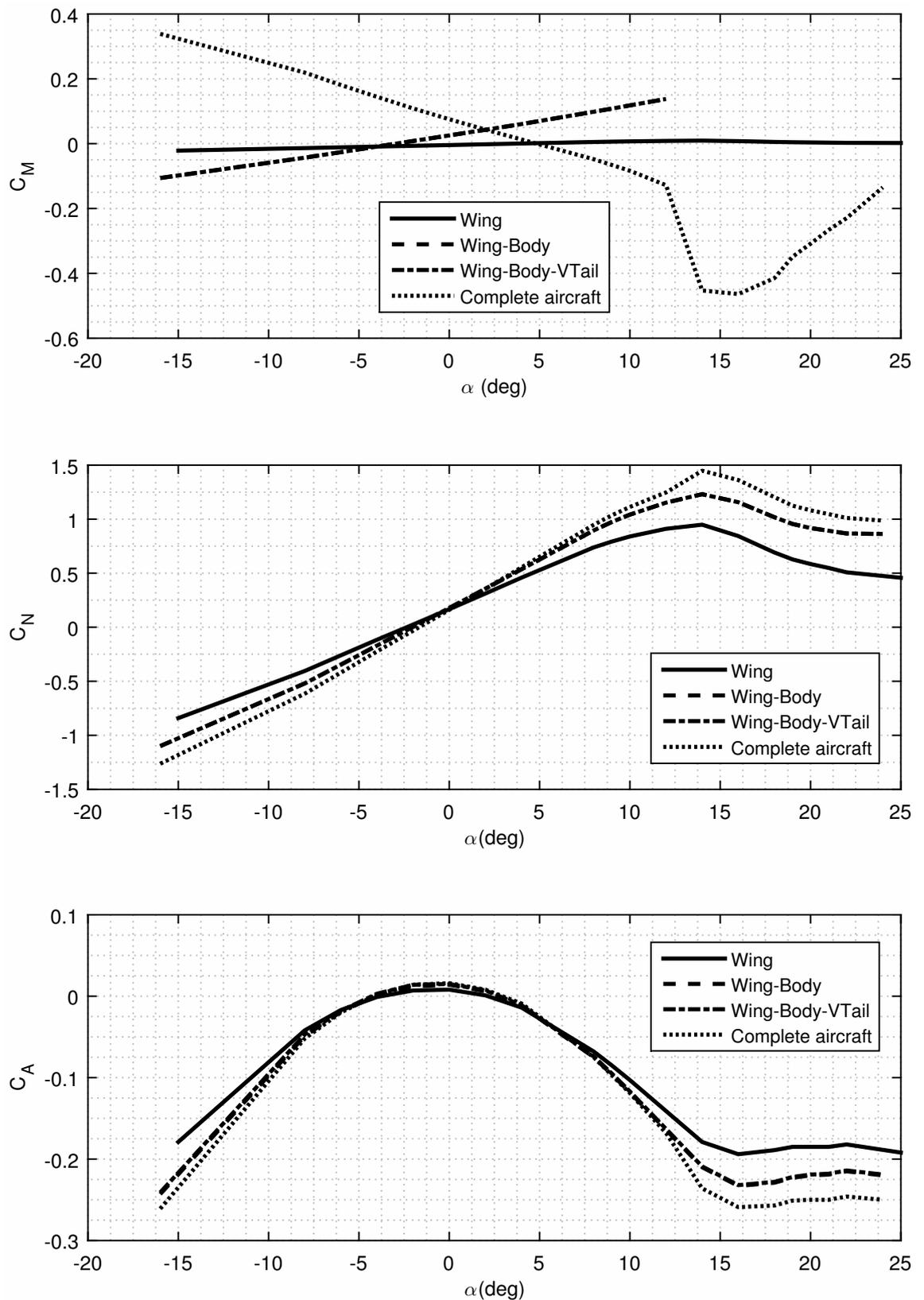


Figura 9.3 Velivolo Cessna Citation II 500. Curve del coefficiente di momento C_M e dei coefficienti di forza assiale e normale C_A e C_N al variare dell'angolo d'attacco per le configurazioni *Wing*, *Wing-Body*, *Wing-Body-Vertical Tail* e velivolo completo.

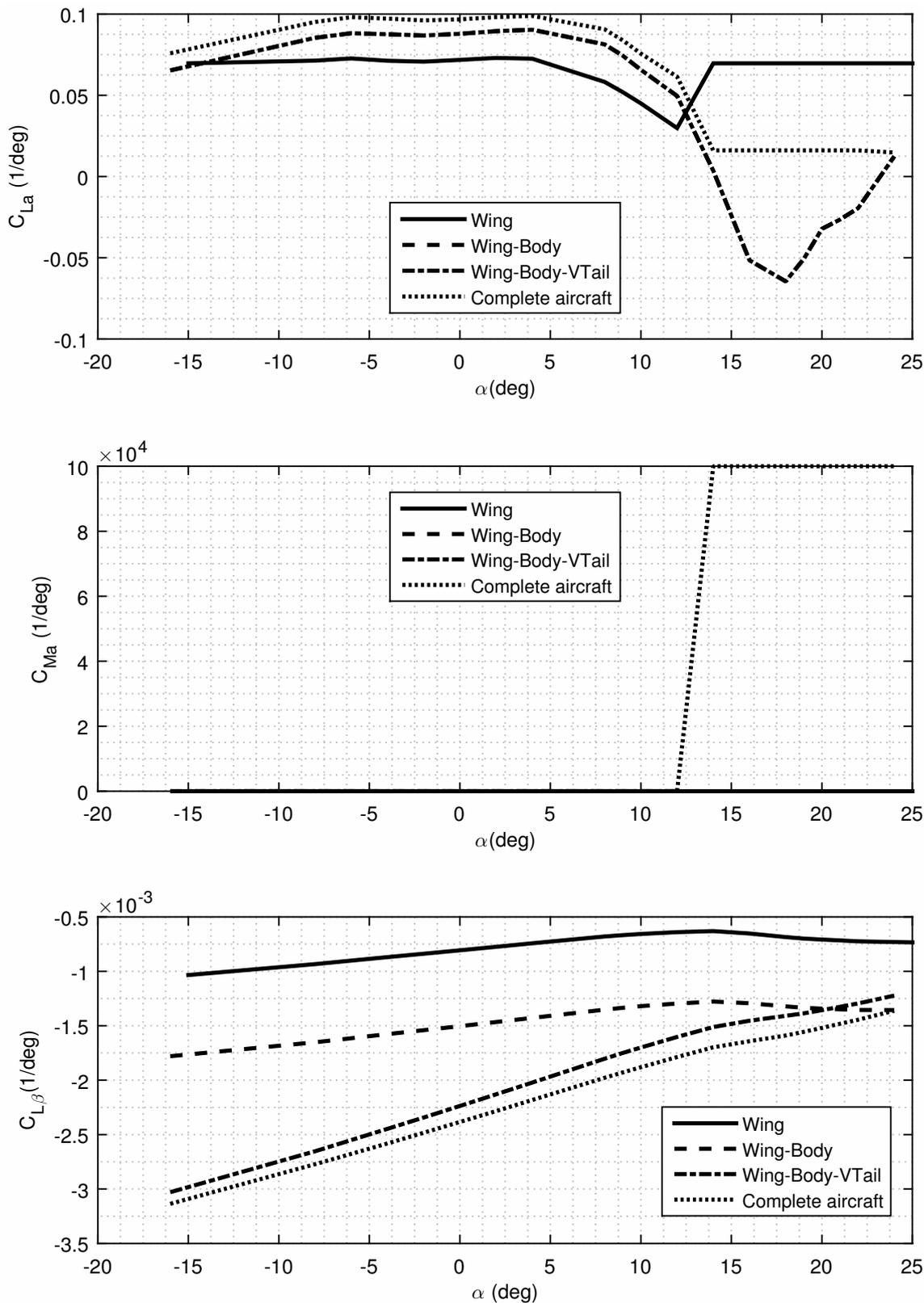


Figura 9.4 Velivolo Cessna Citation II 500. Variazione delle derivate dei coefficiente aerodinamici di momento $C_{L\alpha}$, $C_{M\alpha}$ e $C_{L\beta}$ al variare dell'angolo d'attacco per le configurazioni *Wing*, *Wing-Body*, *Wing-Body-Vertical Tail* e velivolo completo.

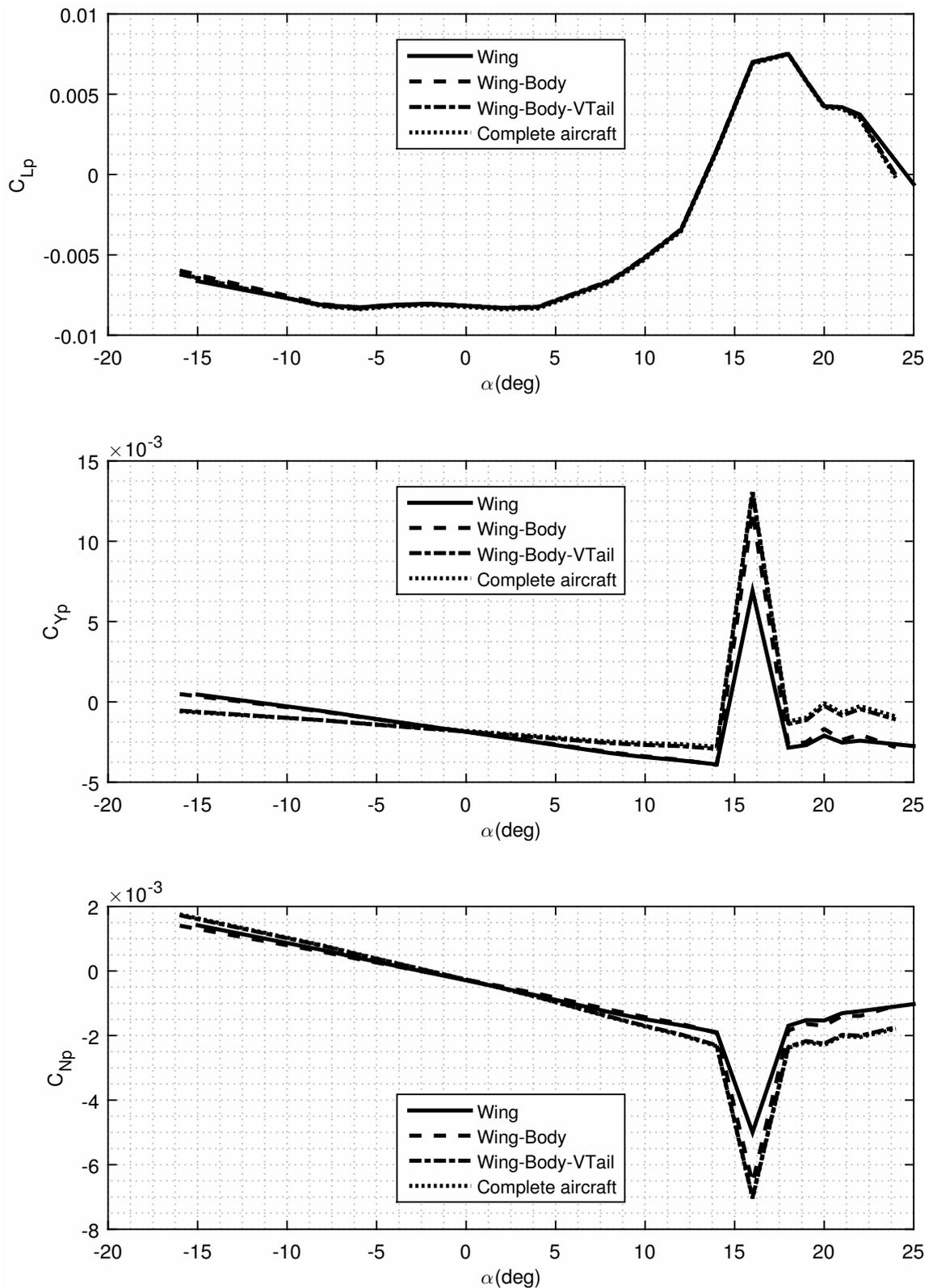


Figura 9.5 Velivolo Cessna Citation II 500. Variazione delle derivate dei coefficienti aerodinamici di momento C_{Lp} , C_{Yp} e C_{Np} al variare dell'angolo d'attacco per le configurazioni *Wing*, *Wing-Body*, *Wing-Body-Vertical Tail* e velivolo completo.

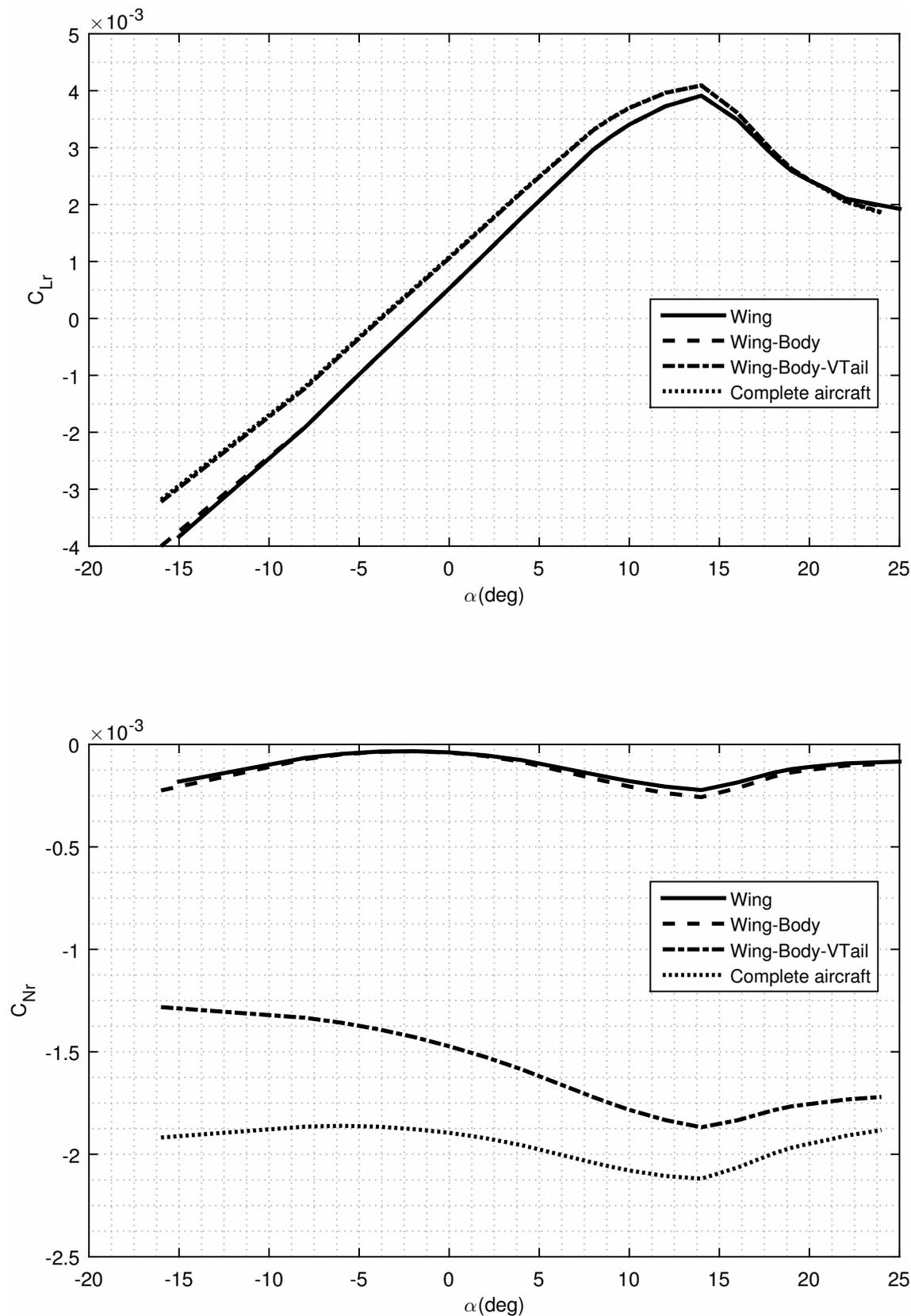


Figura 9.6 Velivolo Cessna Citation II 500. Variazione delle derivate dei coefficienti aerodinamici di momento C_{Lr} e C_{Nr} al variare dell'angolo d'attacco per le configurazioni *Wing*, *Wing-Body*, *Wing-Body-Vertical Tail* e velivolo completo.

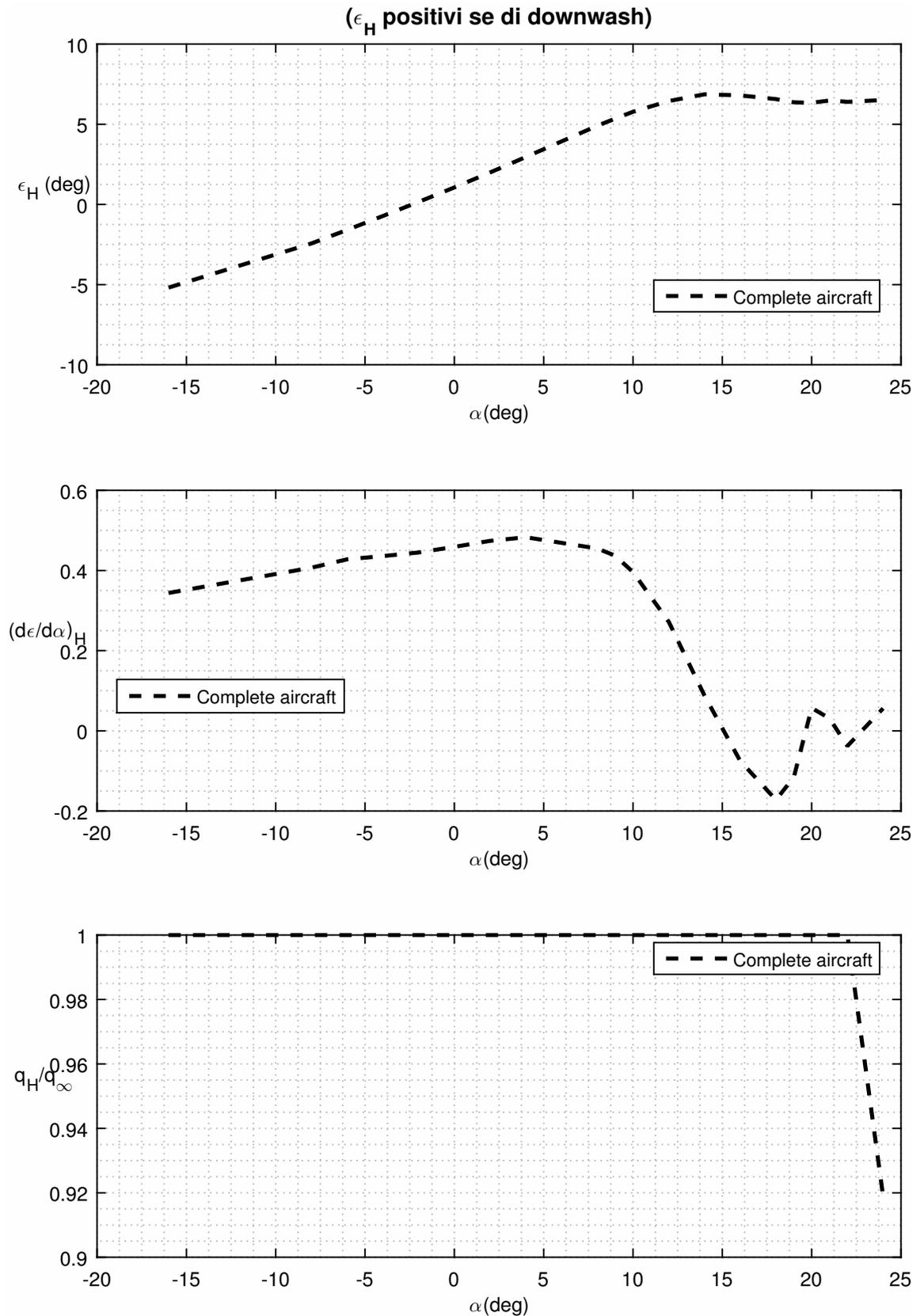


Figura 9.7 Velivolo Cessna Citation II 500. Variazione dell'angolo di downwash mesio ϵ_H , della sua derivata $(\frac{d\epsilon}{d\alpha})_H$ in funzione dell'angolo d'attacco per la configurazione del velivolo completo.

9.6 Elevatore e flap con *Digital DATCOM+* - Esercizio 9-2

Sulla base dell'esercizio precedente e relativamente alla configurazione completa, sono state analizzate le caratteristiche aerodinamiche al variare della deflessione dei flap e dell'elevatore. Sono stati riportati i grafici delle curve aerodinamiche corrispondenti ai diversi valori di δ_f per fissato δ_e e, viceversa, ai diversi valori di δ_e per fissato δ_f . Nel listato 9.2 è stato riportato com'è stato preparato il file di input per produrre i risultati richiesti.

Listato 9.2 file di input per Digital DATCOM+ per la configurazione richiesta

```
* File : CITATION.dat
*
DIM FT
DERIV DEG
DAMP
PART
* DUMP IOM

*****
* Flight Conditions *
*****

$FLTCON WT=7000.0, LOOP=2.0,
        NMACH=1.0, MACH(1)=0.4,
        NALT=1.0, ALT(1)=0.0,
        NALPHA=20.0,
        ALSCHD(1)= -16.0, -8.0, -6.0, -4.0, -2.0, 0.0, 2.0, 4.0, 8.0, 9.0,
                10.0, 12.0, 14.0, 16.0, 18.0, 19.0, 20.0, 21.0, 22.0, 24.0,
        STMACH=0.6, TSMACH=1.4, TR=1.0$

*****
* Reference Parameters *
*****

$OPTINS SREF=320.8, CBARR=6.75, BLREF=51.7, ROUGFC=0.25E-3$

*****
* Group II      Synthesis Parameters *
*****

$SYNTHS XCG=21.9, ZCG=3.125,
        XW=19.1, ZW=3.125, ALIW=2.5,
        XH=39.2, ZH=7.75, ALIH=0.0,
        XV=36.0, ZV=6.0,
        XVF=28.0, ZVF=7.4,
        SCALE=1.0, VERTUP=.TRUE.$

*****
* Body Configuration Parameters *
*****

$BODY NX=8.0,
      X(1)=0.0,1.0,2.7,6.0,8.8,28.5,39.4,44.8,
      R(1)=0.0,1.25,2.1,2.7,2.76,2.7,1.25,0.0,
      ZU(1)=3.5,4.3,4.8,5.5,7.4,7.4,6.5,5.7,
      ZL(1)=3.5,2.5,2.25,2.1,2.0,2.2,4.3,5.7,
      BNOSE=1.0, BLN=8.8,
      BTAIL=1.0, BLA=19.7,
      ITYPE=1.0, METHOD=1.0$

*****
* Wing planform variables
```

```

*****
$WGPLNF CHRDR=9.4,   CHRDTP=3.01,
        SSPN=25.85,  SSPNE=23.46,
        SAVSI=1.3,
        CHSTAT=0.25, TWISTA=-3.0,
        DHDADI=3.6,
        TYPE=1.0$

*****
* Wing Sectional Characteristics Parameters *
*****

NACA W 5 23014

SAVE

*****
* Symmetrical Flap Deflection parameters
*****

$SYMFLP FTYPE=2.0,   NDELTA=9.0,
        DELTA(1)=0.0,5.0,10.0,15.0,20.0,25.0,30.0,35.0,40.0,
        PHETE=0.0522, PHETEP=0.0391,
        CHRDFI=2.0,   CHRDFO=1.6,
        SPANFI=5.78,  SPANFO=15.3,
        CPRMEI(1)=8.1,8.1,8.2,8.2,8.3,8.3,8.3,8.4,8.4,
        CPRMEO(1)=3.7,3.7,3.8,3.8,3.9,3.9,3.9,4.0,4.0,
        NTYPE=1.0$

CASEID FLAPS: Citation II Model 550 Aircraft
SAVE
NEXT CASE

*****
* Horizontal Tail Sectional Characteristics
*****

NACA H 4 0010

*****
* Horizontal Tail planform variables
*****

$HTPLNF CHRDR=4.99,  CHRDTP=2.48,
        SSPN=9.42,  SSPNE=9.21,
        SAVSI=5.32,
        CHSTAT=0.25, TWISTA=0.0,
        DHDADI=9.2,
        TYPE=1.0$

*****
* Vertical Tail planform variables
*****

$VTPLNF CHRDTP=3.63, SSPNE=8.85,  SSPN=9.42, CHRDR=8.3,
        SAVSI=32.3,  CHSTAT=0.25, TYPE=1.0$

*****
* Vertical Fin planform variables
*****

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

```

```

*****
*   Elevator Deflection parameters
*****

$SYMFLP FTYPE=1.0,
        NDELTA=9.0, DELTA(1)=-20.0,-15.0,-10.0,-5.0,0.0,5.0,10.0,13.0,16.0,
        PHETE=0.0522, PHETEP=0.0523,
        CHRDFI=1.94,  CHRDFO=1.03,
        SPANFI=0.7,   SPANFO=9.21,
        CB=0.84,     TC=0.3,         NTYPE=1.0$

*****
*   Vertical Tail Sectional Characteristics
*****

NACA V 4 0012

*****
*   Ventral Fin Sectional Characteristics
*****

NACA F 4 0012

*****
*   Jet Power Effects parameters
*****
* $JETPWR NENGSJ=2.0, AIETLJ=2.0, THSTCJ=0.0,
*         JIALOC=25.8, JELLOC=4.33, JEVLOC=5.625,
*         JEALOC=33.3, JINLTA=2.243,
*         AMBTMP=59.7, AMBSTP=2116.8, JERAD=0.755$

CASEID TOTAL: Citation II Model 550 Aircraft

```

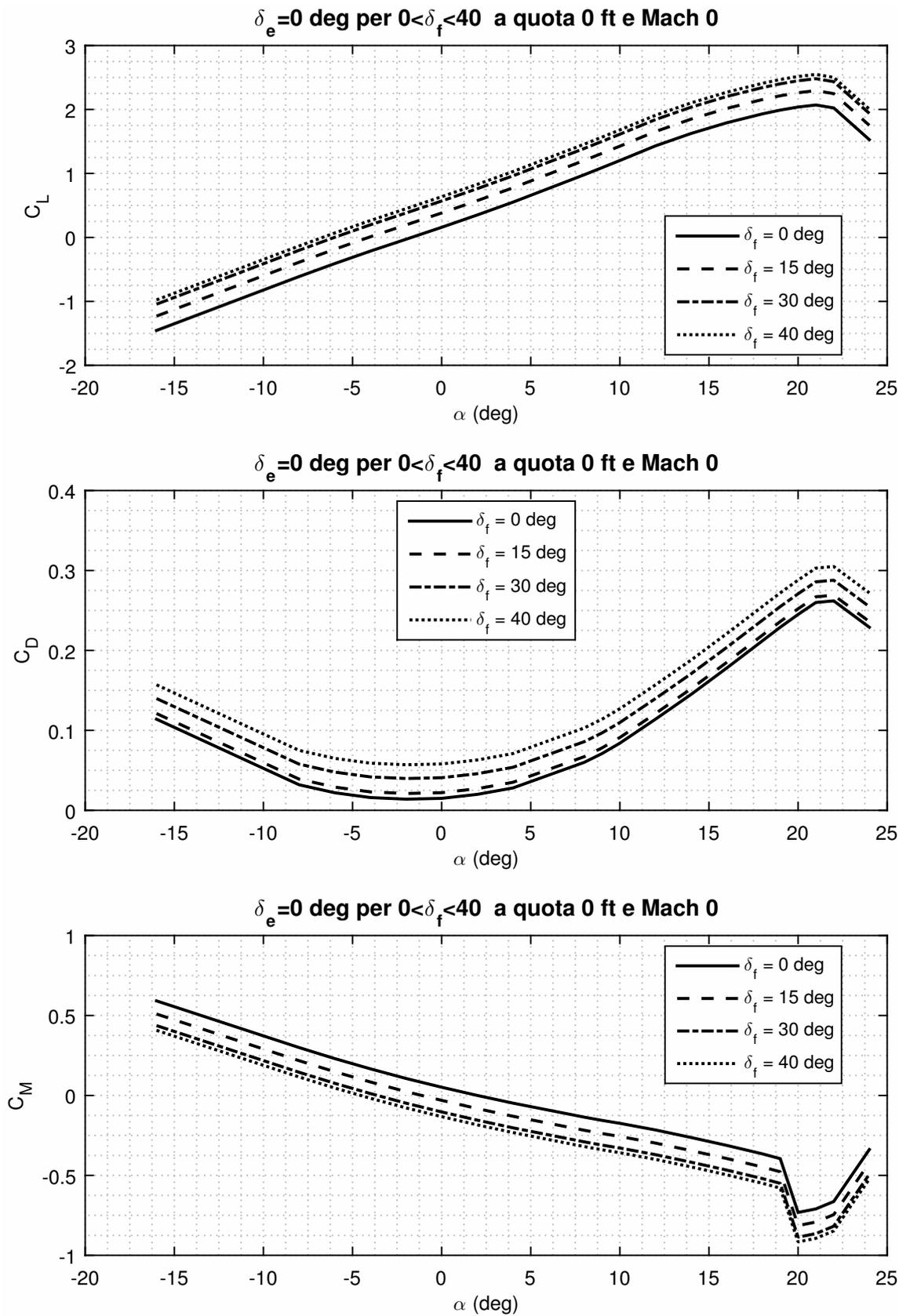


Figura 9.8 Andamento dei coefficienti C_L , C_D e C_M espressi per $\delta_e = 0$ deg e $\delta_f \in [0, 40]$.

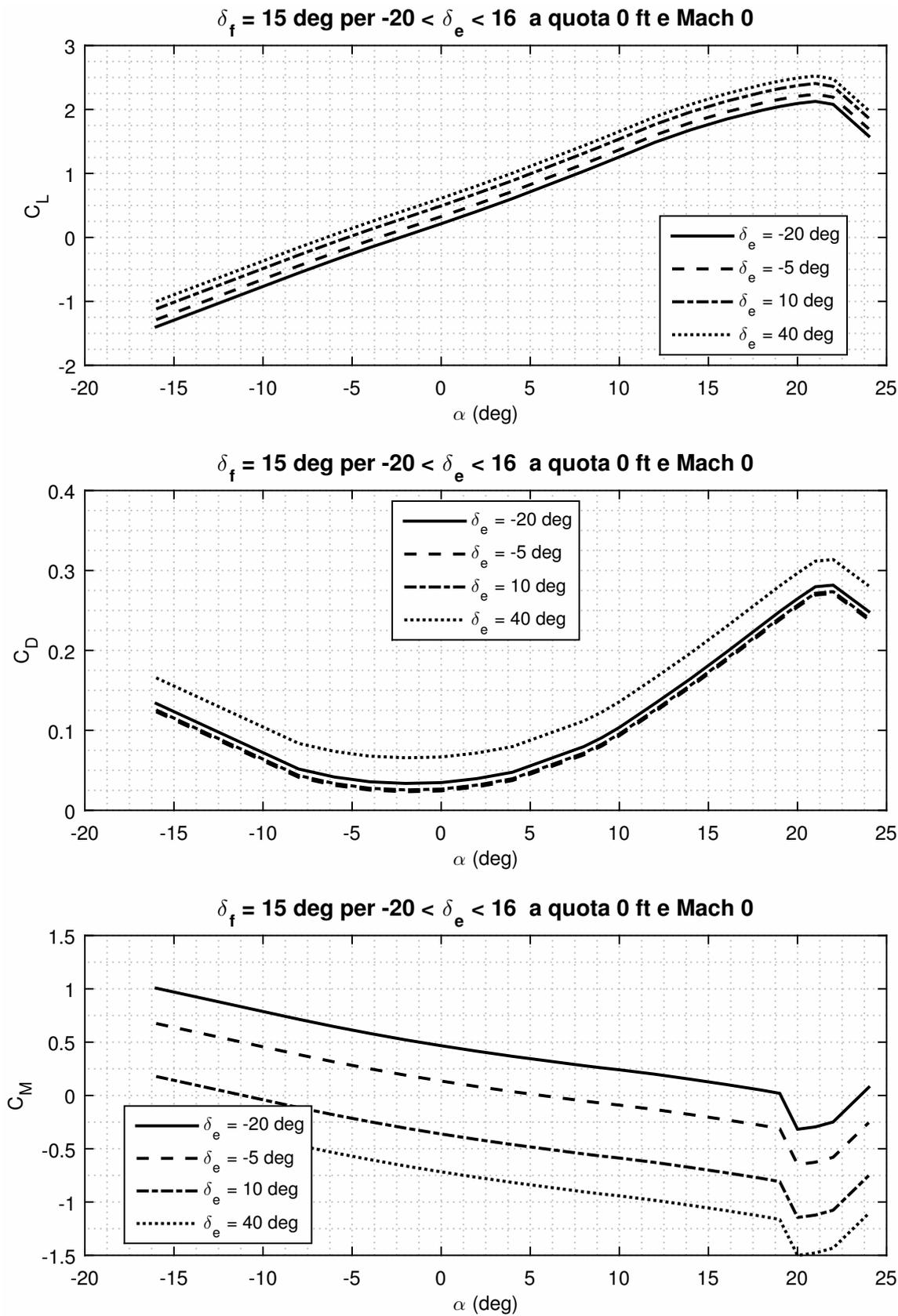


Figura 9.9 Andamento dei coefficienti C_L , C_D e C_M espressi per $\delta_f = 15 \text{ deg}$ e $\delta_e \in [-20, 16]$.

9.7 Aerodinamica del velivolo *Tecnam P2006T* - Esercizio 9.3

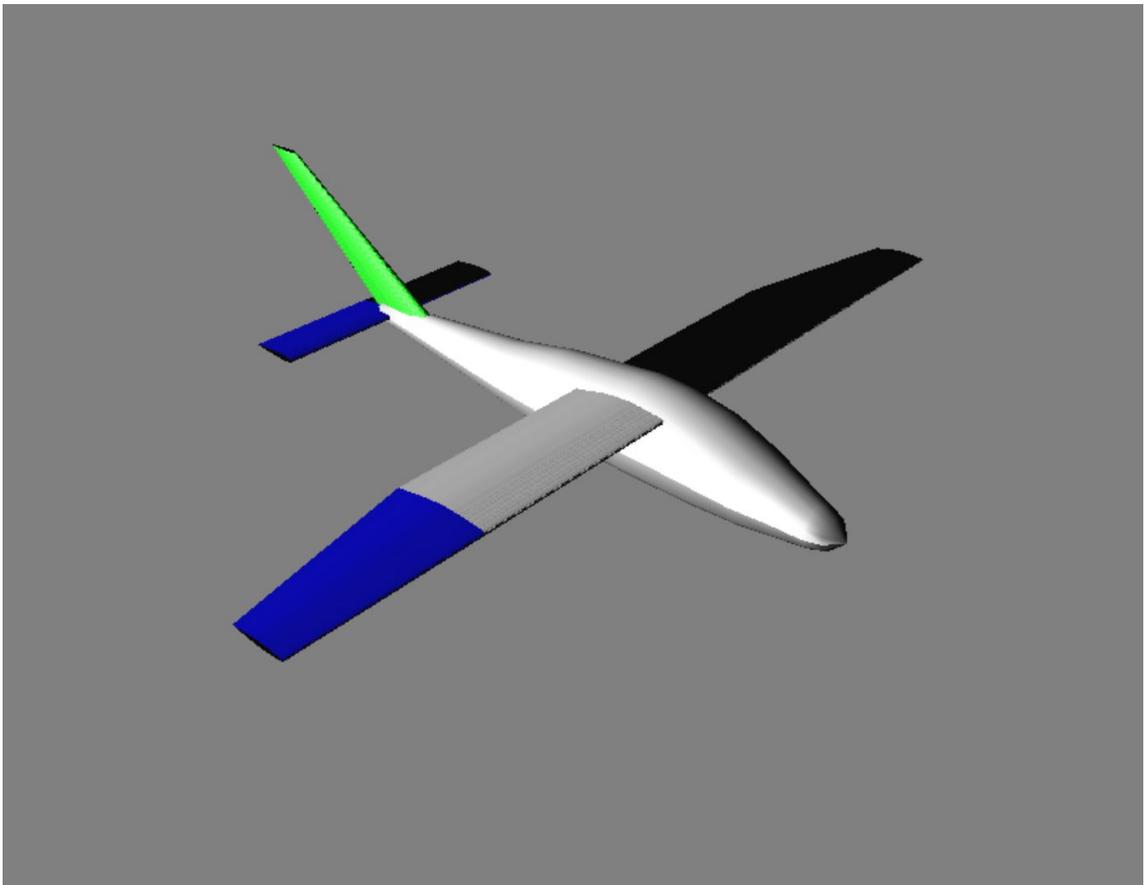


Figura 9.10 Velivolo TecnamP2006

In maniera analoga a quanto fatto per il velivolo Cessna Citation nel paragrafo precedente, si è analizzato il velivolo Tecnam P2006T. Per tale velivolo è stato creato un file di input *ex novo*, riportato nel listato 9.3.

I parametri geometrici introdotti della *namelist* BODY e altre variabili sono stati desunti dai dati messi a disposizione dal docente. Di seguito viene riproposto il listato 9.3 con il quale sono stati ricavati gli andamenti dei coefficienti aerodinamici per il velivolo Tecnam P2006T proposto per questa esercitazione.

Le viste 9.11 a pagina 129, 9.12 a pagina 130 e 9.13 a pagina 130 del velivolo sono state ricavate mediante la realizzazione di opportuni codici Matlab, in quanto la funzione *datcomimport* di Matlab non riporta in output i valori dei dati geometrici.

Nella figura 9.14 a pagina 131 sono mostrati gli andamenti dei coefficienti aerodinamici di portanza C_L e di resistenza C_D al variare dell'angolo di attacco e la polare, per le configurazioni *Wing*, *Wing-Body*, *Wing-Body-Vertical Tail* e velivolo completo.

Nella figura 9.15 a pagina 132 sono mostrati gli andamenti delle curva del coefficiente di momento C_M , del coefficiente di forza assiale C_A e normale C_N al variare dell'angolo d'attacco per le varie configurazioni di velivolo.

Listato 9.3 file di input per Digital DATCOM+ il velivolo Tecnam P2006T

*Tecnam P2006T

```

DIM M !
  Input in metri
DERIV DEG !
  derivate in deg
PART

*****
*Condizioni di volo:*
*****
$FLTCN WT=760.0,          LOOP=2.0,
      NMACH=1.0,          MACH(1)=0.6,
      NALT=1.0,           ALT(1)=2500.0,
      NALPHA=20.0,
      ALSCHD(1)= -16.0, -8.0, -6.0, -4.0, -2.0, 0.0, 2.0, 4.0, 8.0, 9.0,
      10.0, 12.0, 14.0, 16.0, 18.0, 19.0, 20.0, 21.0, 22.0, 24.0,
      TR=1.0,             GAMMA=0.0$
*RNNUB(1)=6706963.3,5010853.3,63421800.0

*****
*Parametri di riferimento, per l'adimensionalizzazione:*
*****
$OPTINS SREF=14.23,          !S = superficie
      alare in m^2
      CBARR=1.36,           !c_bar = corda
      media aerodinamica in m
      BLREF=10.6$          !b = apertura
      alare in m

*****
*Parametri di sintesi:*
*****
$SYNTHS XCG=2.6,           ZCG=0.7,          !X_CG =
      Longitudinal location of cg
      XW=2.3,              ZW=0.9,          ALIW=1.5,
      XH=7.87,            ZH=0.42,        ALIH=0.0,
      XV=6.88,            ZV=0.45,          !continua
      la pinna senza raccordo
      XVF=6.63,           ZVF=0.686,       !fine della curva
      della fusoliera
      SCALE=1.0,          VERTUP=.TRUE.$

*****
*Parametri fusoliera:*
*****
$BODY NX=18.0,
      X(1)
      =0.0,0.1950,0.4040,0.8430,1.2740,1.8680,2.4370,2.7150,2.9560,4.4160,
      4.8330,5.3640,5.8890,6.4690,7.1020,7.7080,8.1130,8.3450,
      ZU(1)
      =0.0,0.2150,0.3090,0.5260,0.7130,0.9470,1.0640,1.0820,1.0900,
      0.9450,0.8660,0.7730,0.7320,0.6860,0.6360,0.5890,0.5510,0.4170,
      ZL(1)
      =0.0,-0.1610,-0.2020,-0.2430,-0.2570,-0.2610,-0.2610,-0.2610,-0.2610,
      -0.2080,-0.1600,-0.0990,-0.0380,0.0290,0.1020,0.1720,0.2430,0.4170,
      R(1)
      =0.0,0.2250,0.3050,0.3900,0.5050,0.5850,0.6100,0.6100,0.6100,0.5250,
      0.4800,0.4150,0.3550,0.2850,0.2100,0.1400,0.1400,0.0$

*****
*Forma in pianta dell'ala:*
*****
$WGPLNF CHRDR=1.44,        CHRDTP=0.83,        CHROBP=1.44,
      SSPN=5.30,           SSPNE=4.7,          SSPNOP=2.15,

```

```

TYPE=1.0$

*****
*Profilo alare considerato:*
*****
NACA-W-63-A-012

SAVE
      !Salvo le configurazioni impostate

*****
* Flap*
*****
$SYMFLP FTYPE=2.0,      NDELTA=9.0,
        DELTA(1)=0.0,5.0,10.0,15.0,20.0,30.0,35.0,40.0,45.0,
        CHRDFI=0.28,   CHRDFO=0.28,
        SPANFI=0.51,   SPANFO=3.12,
        NTYPE=1.0$

CASEID FLAPS: TECHNAM P2006T
NEXT CASE

*****
*PIANO DI CODA ORIZZONTALE*
*****
$HTPLNF CHRDR=0.79,     CHRDTP=0.79,
        SSPN=1.65,     SSPNE=1.55,
        TYPE=1.0$

*****
*Profilo utilizzato:*
*****
NACA H 4 0012

*****
*PIANO DI CODA VERTICALE:*
*****
$VTPLNF CHRDTP=0.5730,  CHRDR=1.4508,
        SSPNE=1.4112,  SSPN=1.9812,
        SAVSI=50.0,   SAVSO=75.6,
        CHSTAT=0.0,
        TYPE=1.0$

*****
*Profilo utilizzato:*
*****
NACA V 4 0012

*****
*Elevatore:*
*****
$SYMFLP FTYPE=1.0,      NDELTA=9.0,
        DELTA(1)=-30.0,-20.0,-10.0,-5.0,0.0,5.0,10.0,20.0,30.0,
        CHRDFI=0.44,   CHRDFO=0.40,
        SPANFI=0.20,   SPANFO=0.747,
        CB=0.8,       TC=0.1,
        NTYPE=1.0$

CASEID TOTAL: P2006T

```

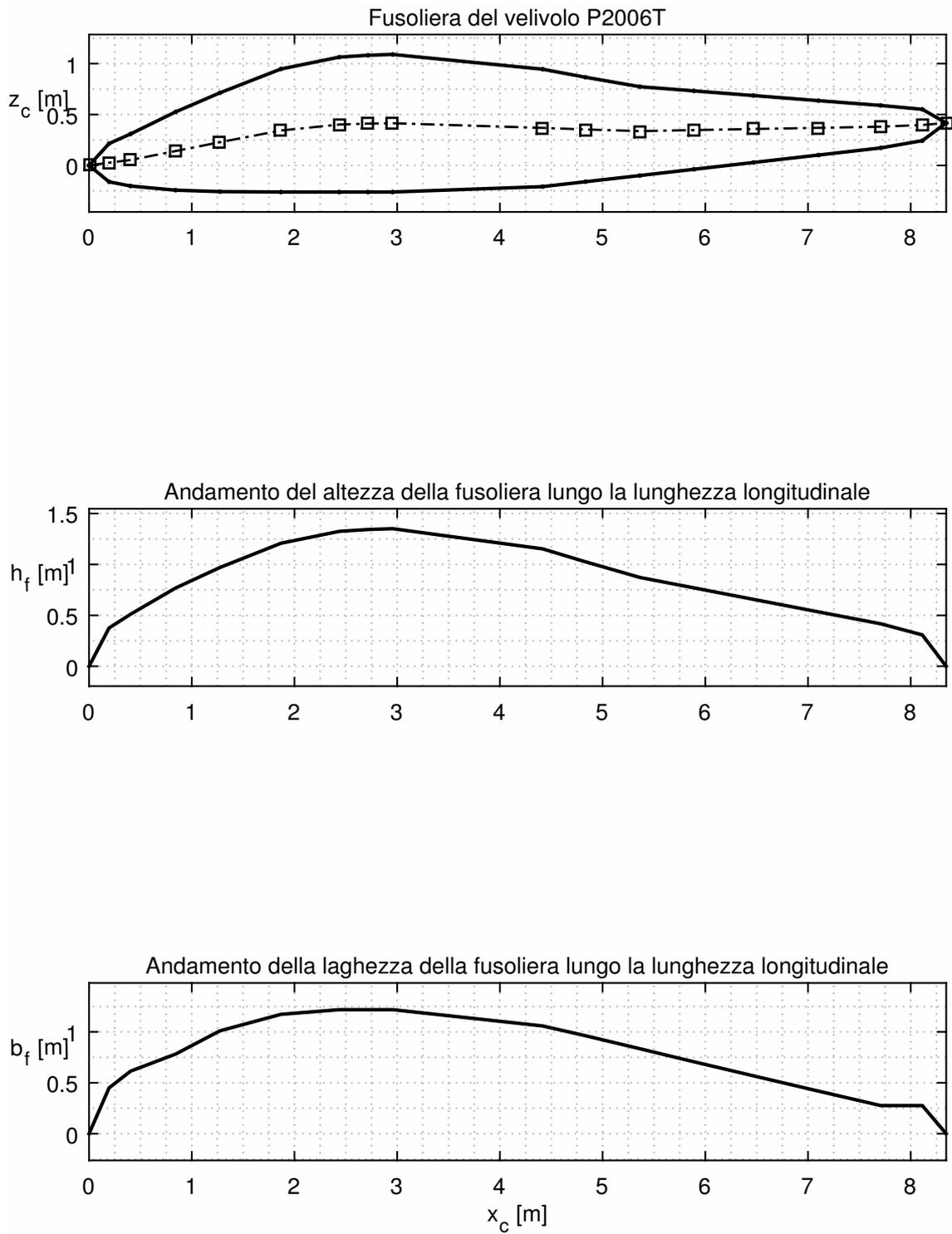


Figura 9.11 Profilo della fusoliera e ingombri del velivolo Tecnam P2006T.

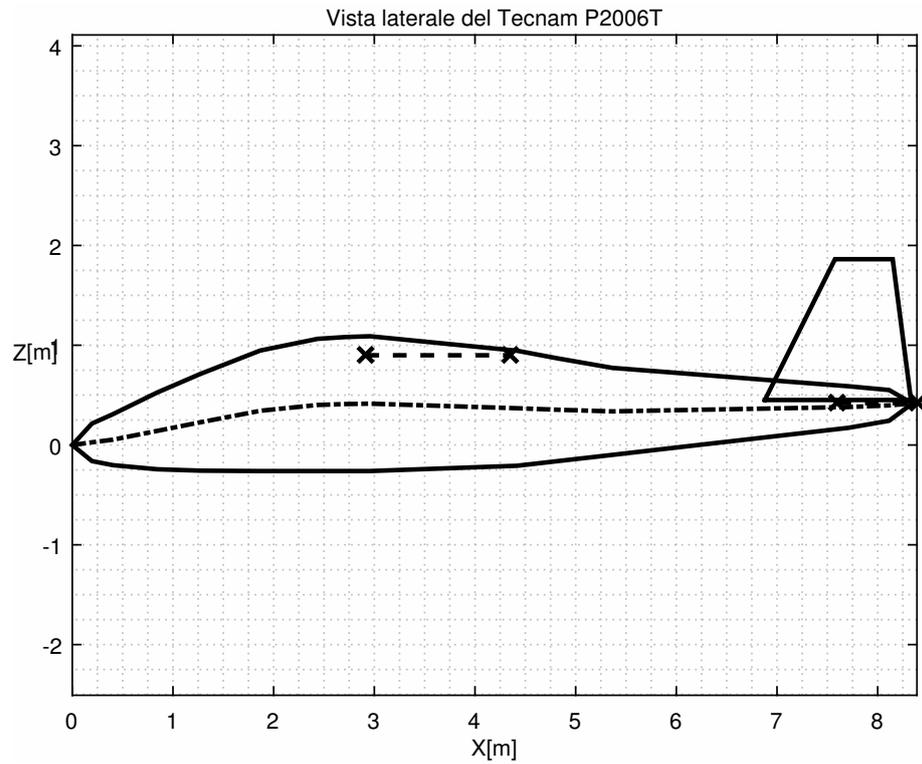


Figura 9.12 Vista laterale del velivolo Tecnam P2006T. Sono riportate le corde delle sezioni di radice dell'ala e del piano orizzontale di coda.

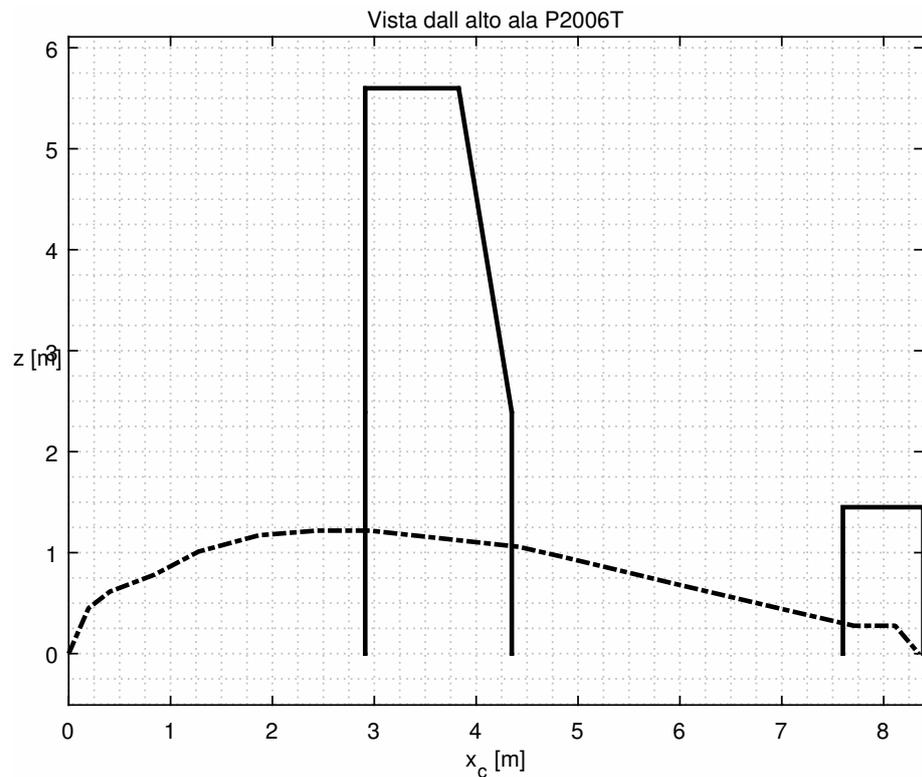


Figura 9.13 Vista Alto TECNAM P2006T.

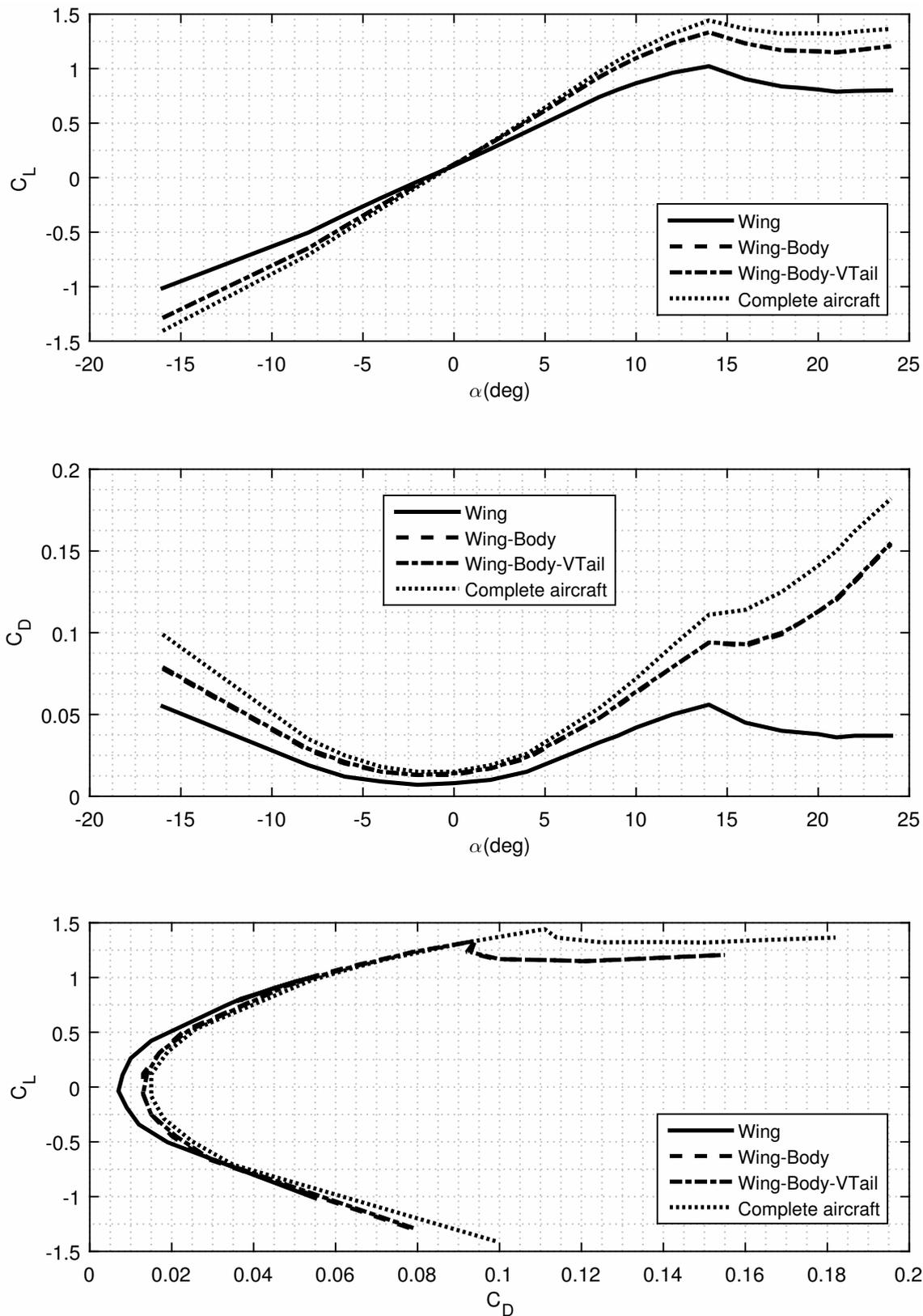


Figura 9.14 Velivolo Tecnam P2006T. Curve del coefficiente di portanza C_L e del coefficiente di resistenza C_D al variare dell'angolo d'attacco e polare per le configurazioni *Wing*, *Wing-Body*, *Wing-Body-Vertical Tail* e velivolo completo.

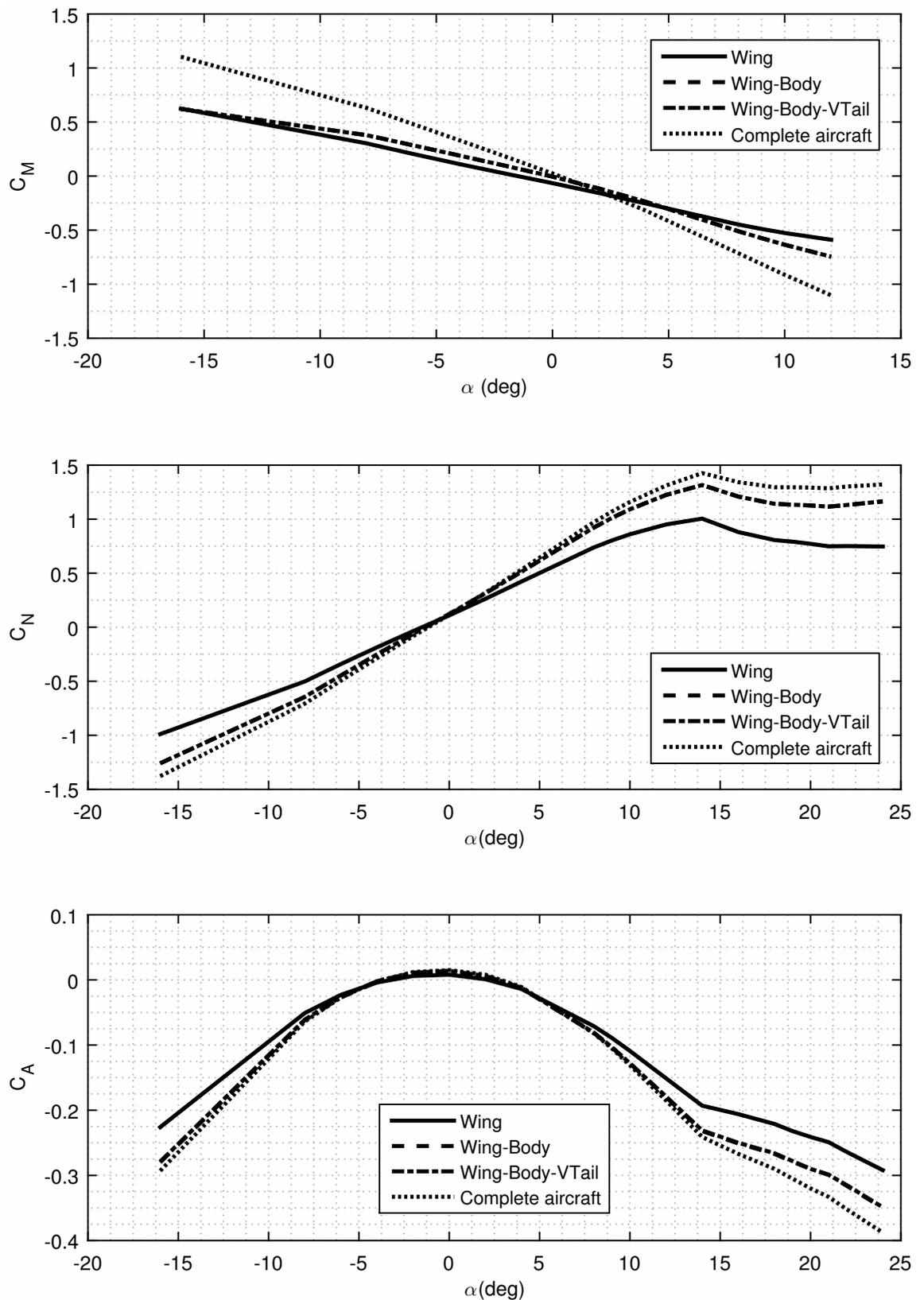


Figura 9.15 Velivolo Tecnam P2006T. Curve del coefficiente di momento C_M e del coefficienti di forza assiale e normale C_A e C_N al variare dell'angolo d'attacco per le configurazioni *Wing*, *Wing-Body*, *Wing-Body-Vertical Tail* e velivolo completo.

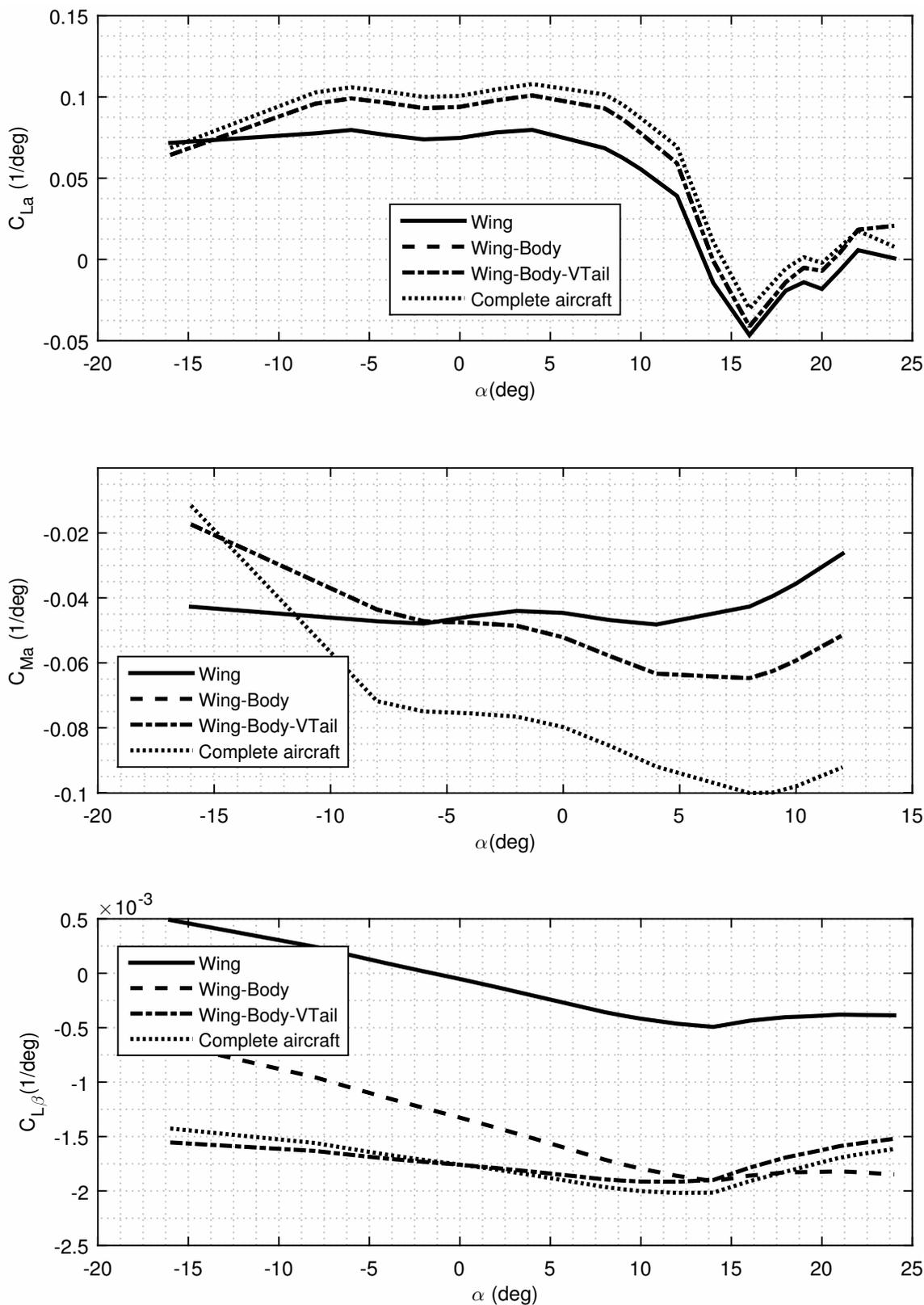


Figura 9.16 Velivolo Tecnam P2006T. Curve del coefficiente di momento C_M e dei coefficienti di forza assiale e normale C_A e C_N al variare dell'angolo d'attacco per le configurazioni *Wing*, *Wing-Body*, *Wing-Body-Vertical Tail* e velivolo completo.

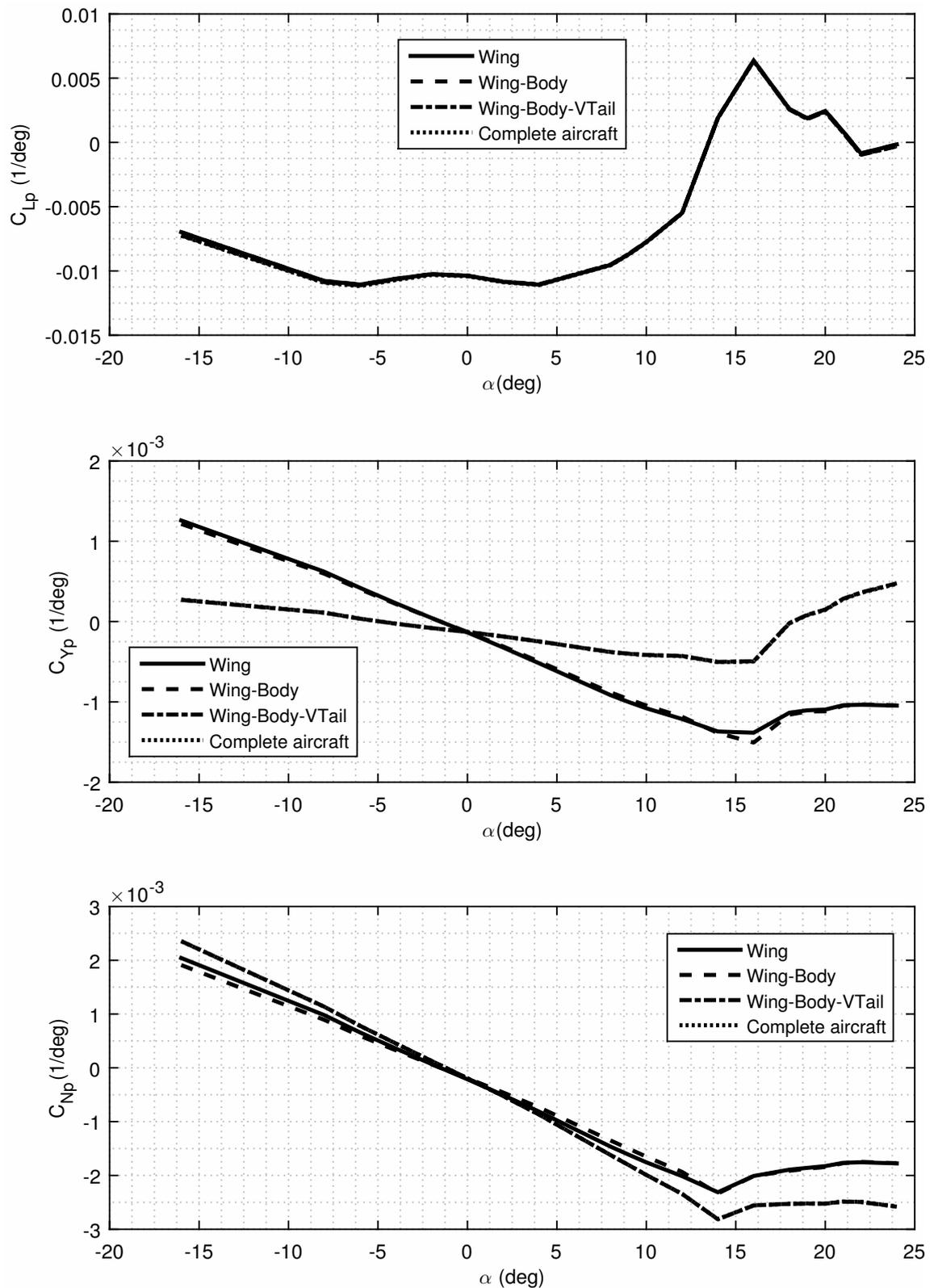


Figura 9.17 Velivolo Tecnam P2006T. Variazione delle derivate dei coefficienti aerodinamici di momento C_{Lp} , C_{Yp} e C_{Np} in funzione dell'angolo d'attacco per le configurazioni *Wing*, *Wing-Body*, *Wing-Body-Vertical Tail* e velivolo completo.

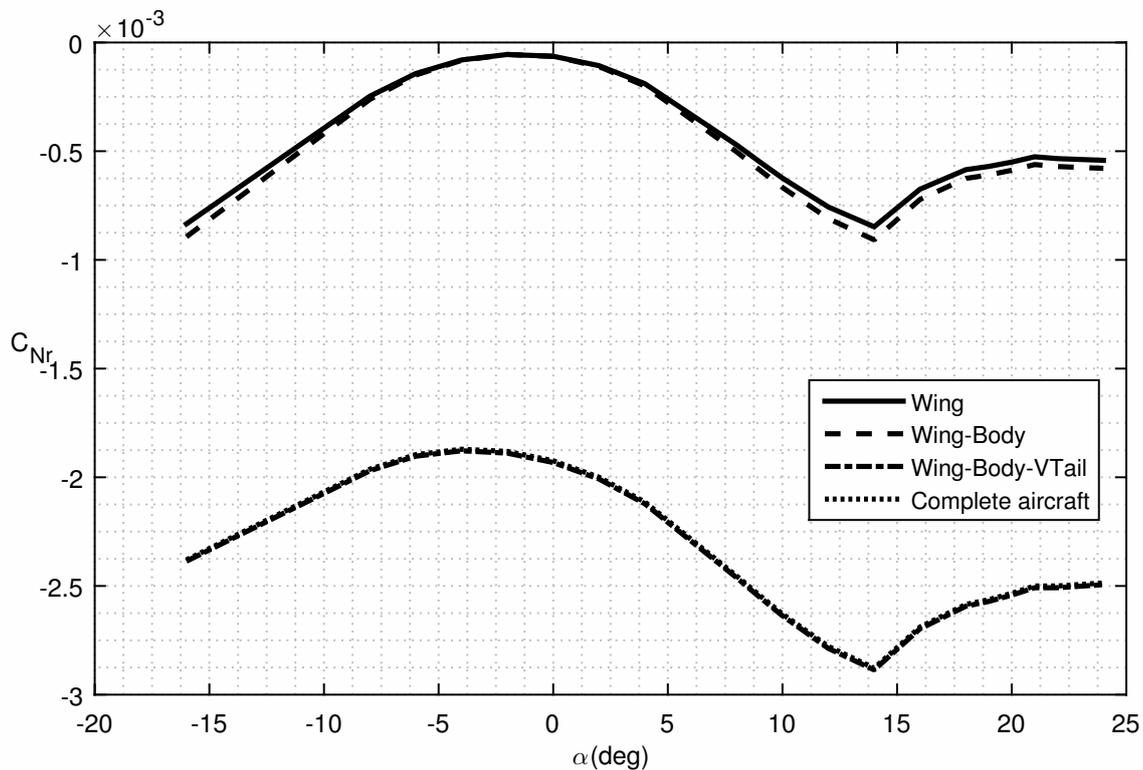
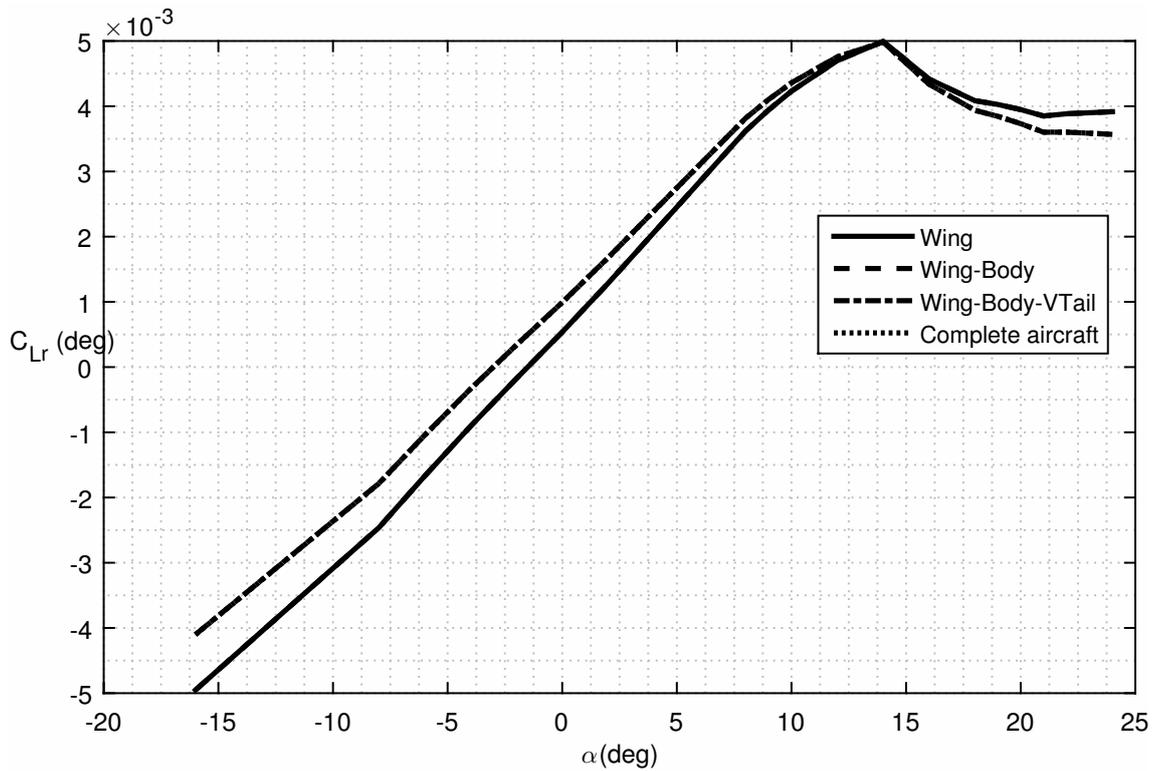


Figura 9.18 Velivolo Tecnam P2006T. Variazione delle derivate dei coefficienti aerodinamici di momento C_{L_r} e C_{N_r} in funzione dell'angolo d'attacco per le configurazioni *Wing*, *Wing-Body*, *Wing-Body-Vertical Tail* e velivolo completo.

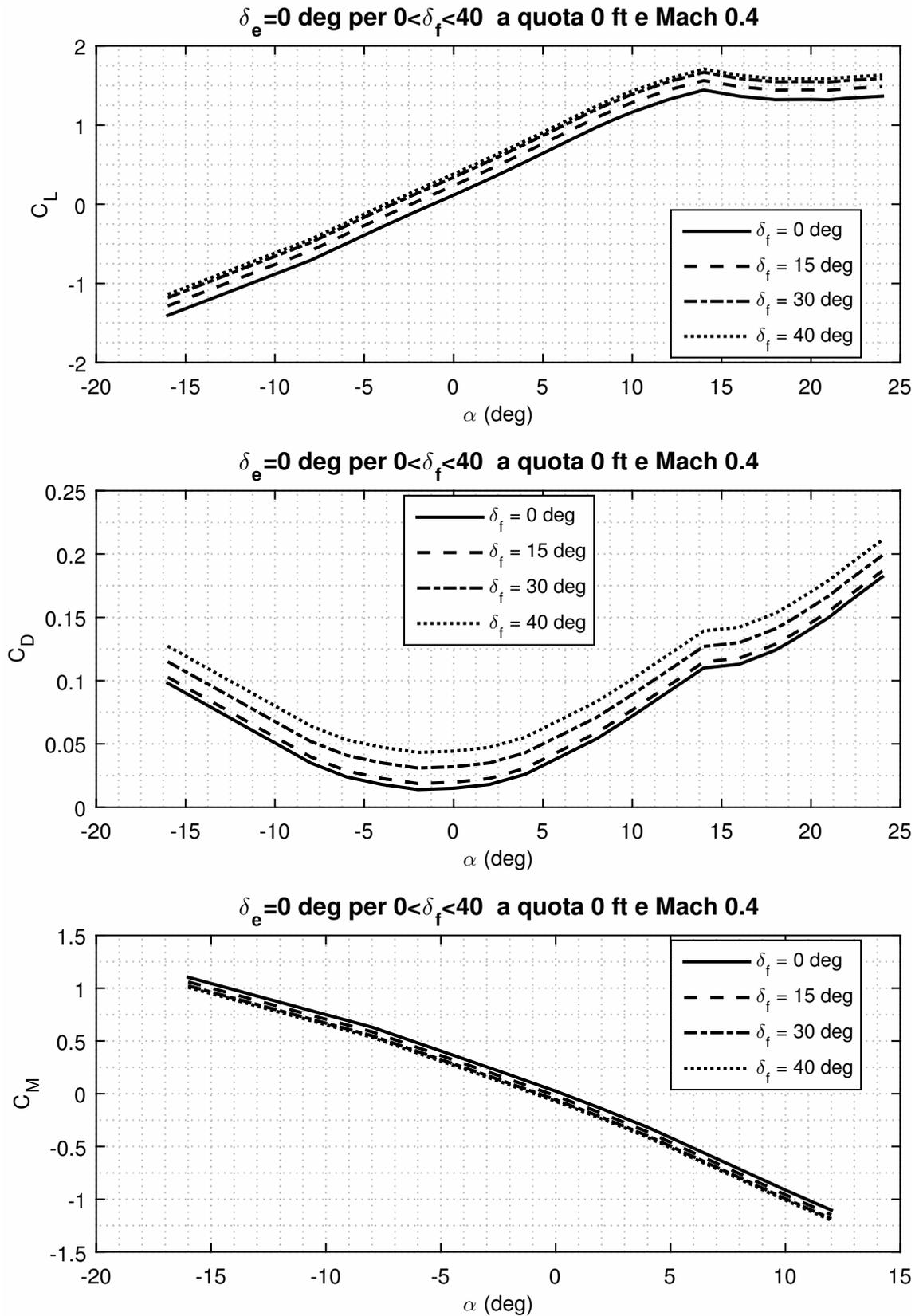


Figura 9.20 Velivolo Tecnam P2006T. Curve del coefficiente di portanza C_L , del coefficiente di resistenza C_D e del coefficiente di momento di beccheggio C_M al variare dell'angolo di deflessione dei flap per fissato valore dell'angolo di deflessione dell'eelevatore parti a $\delta_e = 0$ nel caso della conigurazione del velivolo completo.

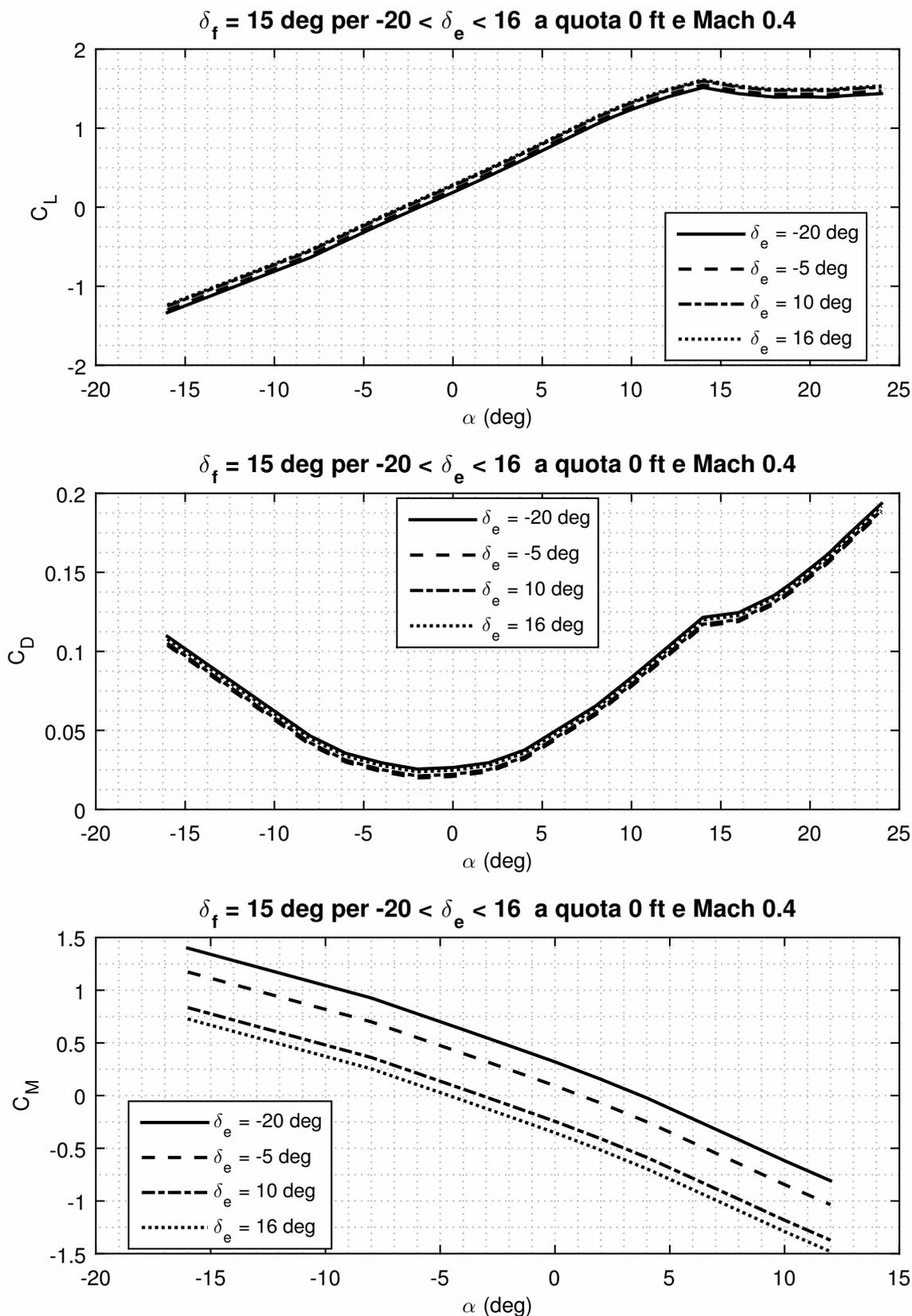


Figura 9.21 Velivolo Tecnam P2006T. Curve del coefficiente di portanza C_L , del coefficiente di resistenza C_D e del coefficiente di momento di beccheggio C_M al variare dell'angolo di deflessione dell'elevatore per fissato angolo di deflessione dei flap pari a $\delta_e=15 \text{ deg}$ nel caso della configurazione del velivolo completo.

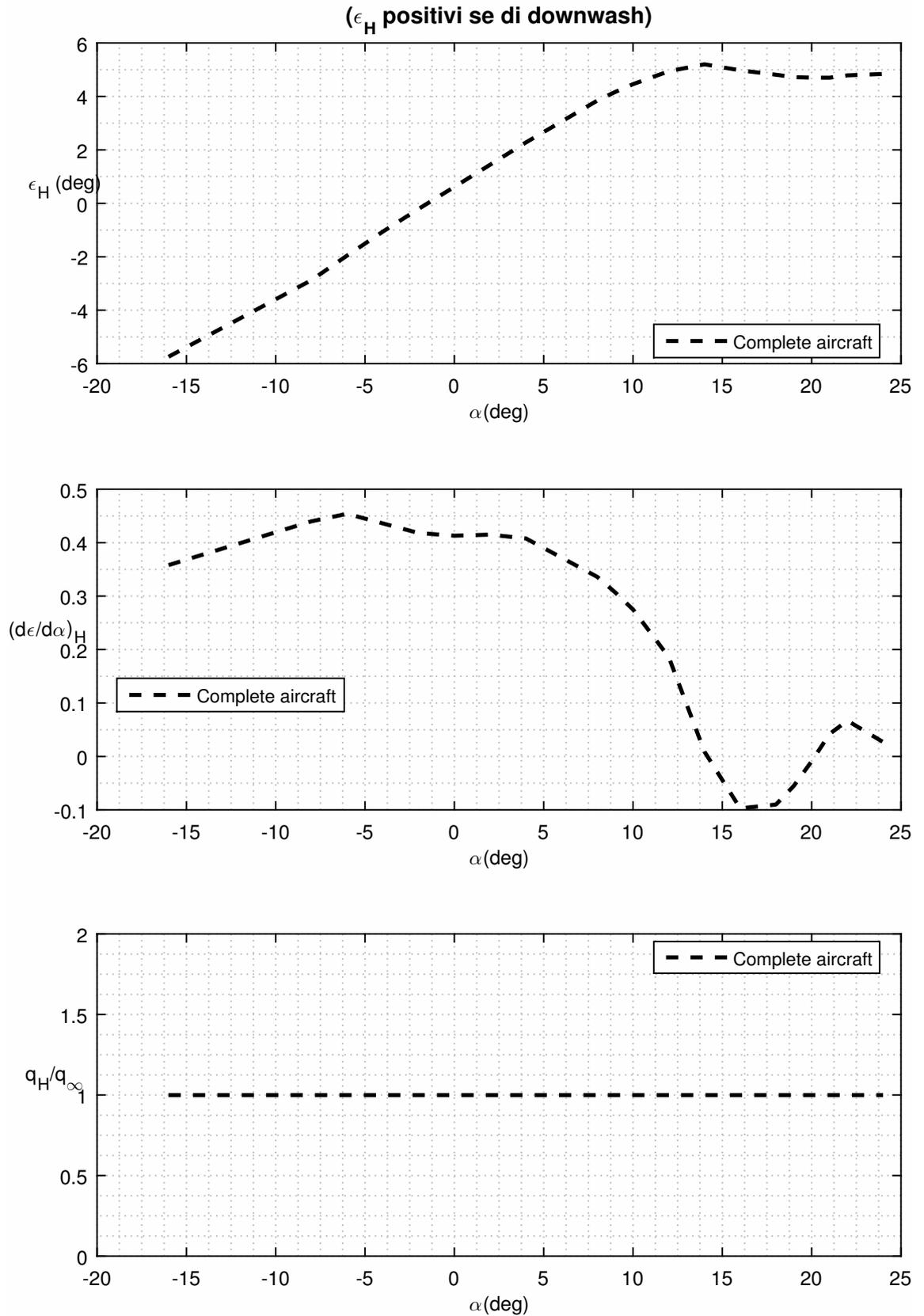


Figura 9.19 Velivolo Tecnam P2006T. Variazione dell'angolo di downwash mesio ϵ_H , della sua derivata $(\frac{d\epsilon}{d\alpha})_H$ in funzione dell'angolo d'attacco per la configurazione del velivolo completo.

9.8 Caratteristiche aerodinamiche del velivolo ATR72 - *Esercizio 9.4*

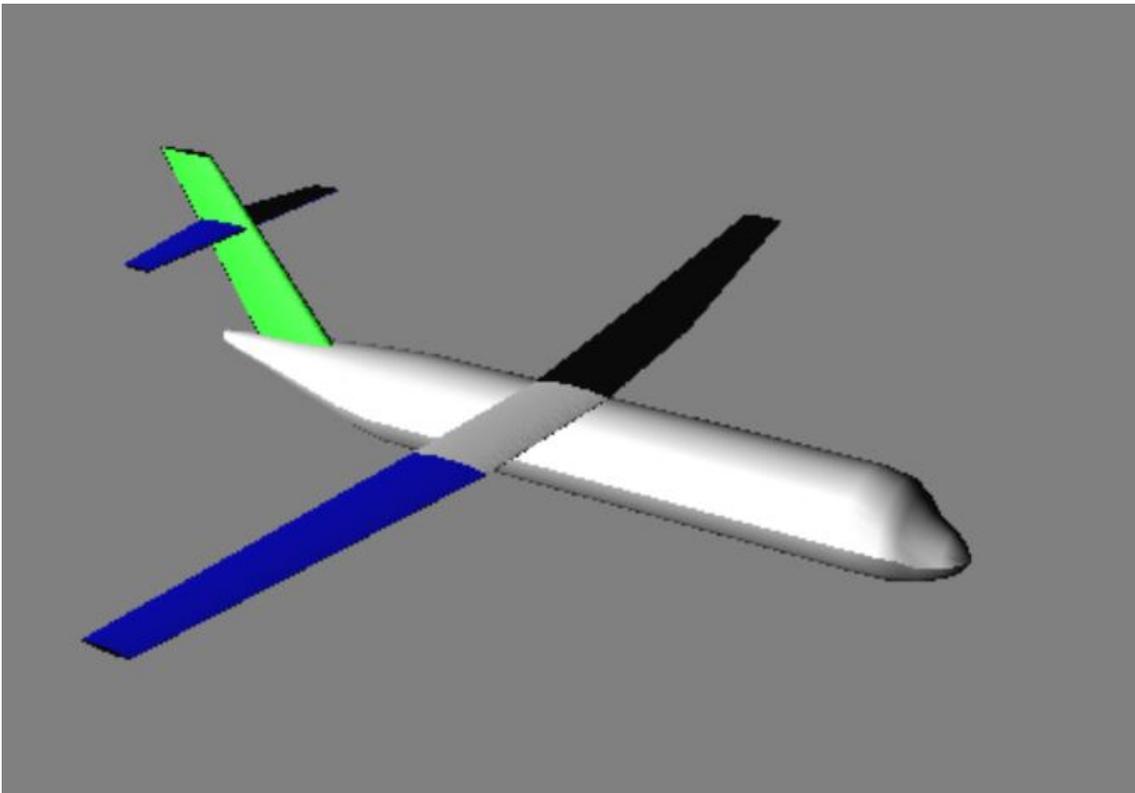


Figura 9.22 Velivolo ATR72 proposto nella rappresentazione Digital DATCOM

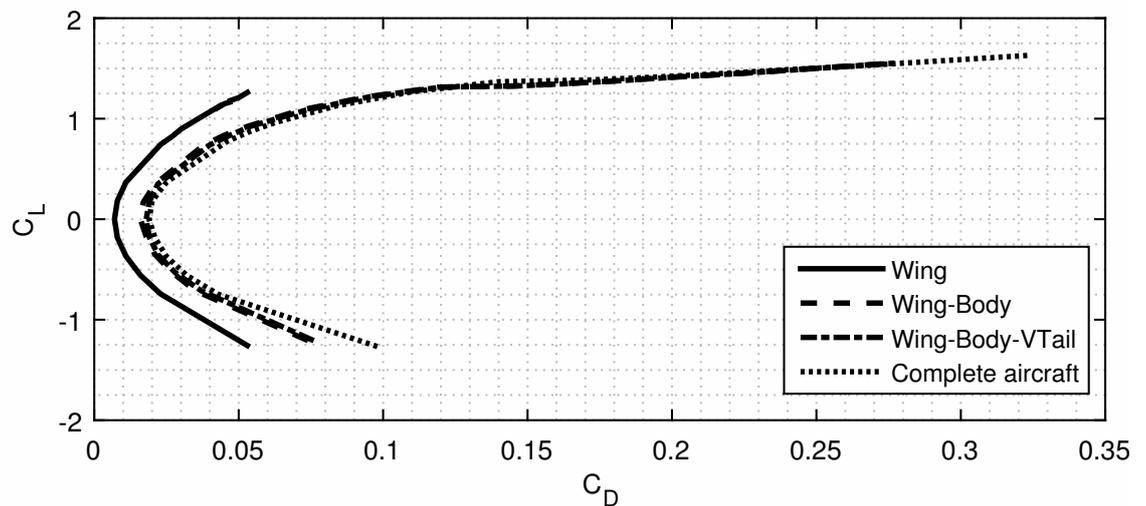
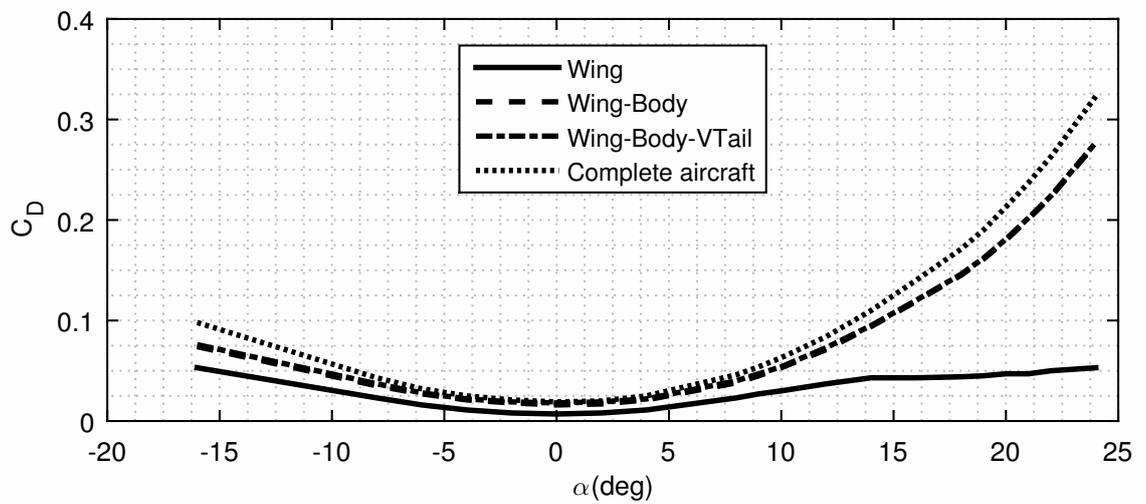
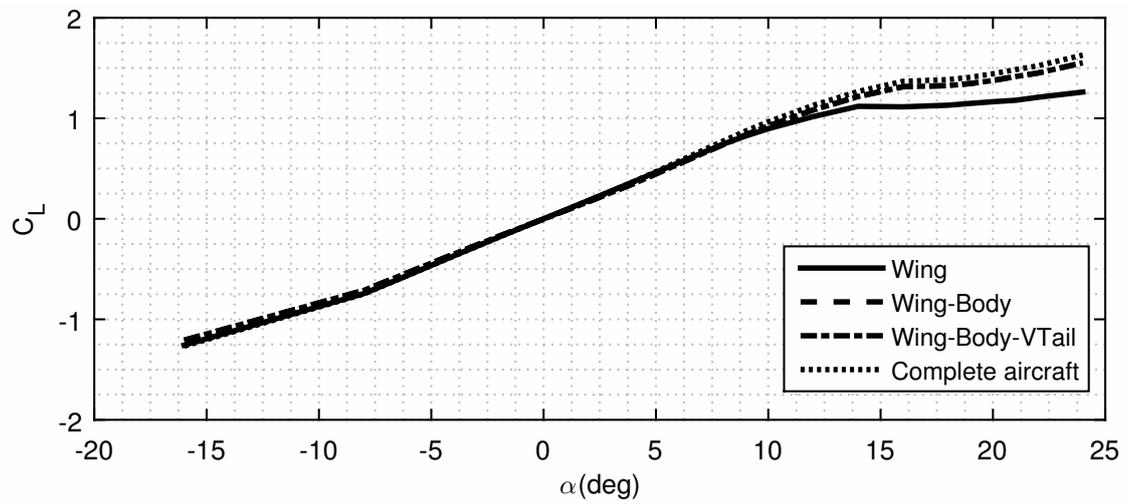


Figura 9.23 Velivolo ATR72. Curve del coefficiente di portanza C_L e del coefficiente di resistenza C_D al variare dell'angolo d'attacco e polare per le configurazioni *Wing*, *Wing-Body*, *Wing-Body-Vertical Tail* e velivolo completo.

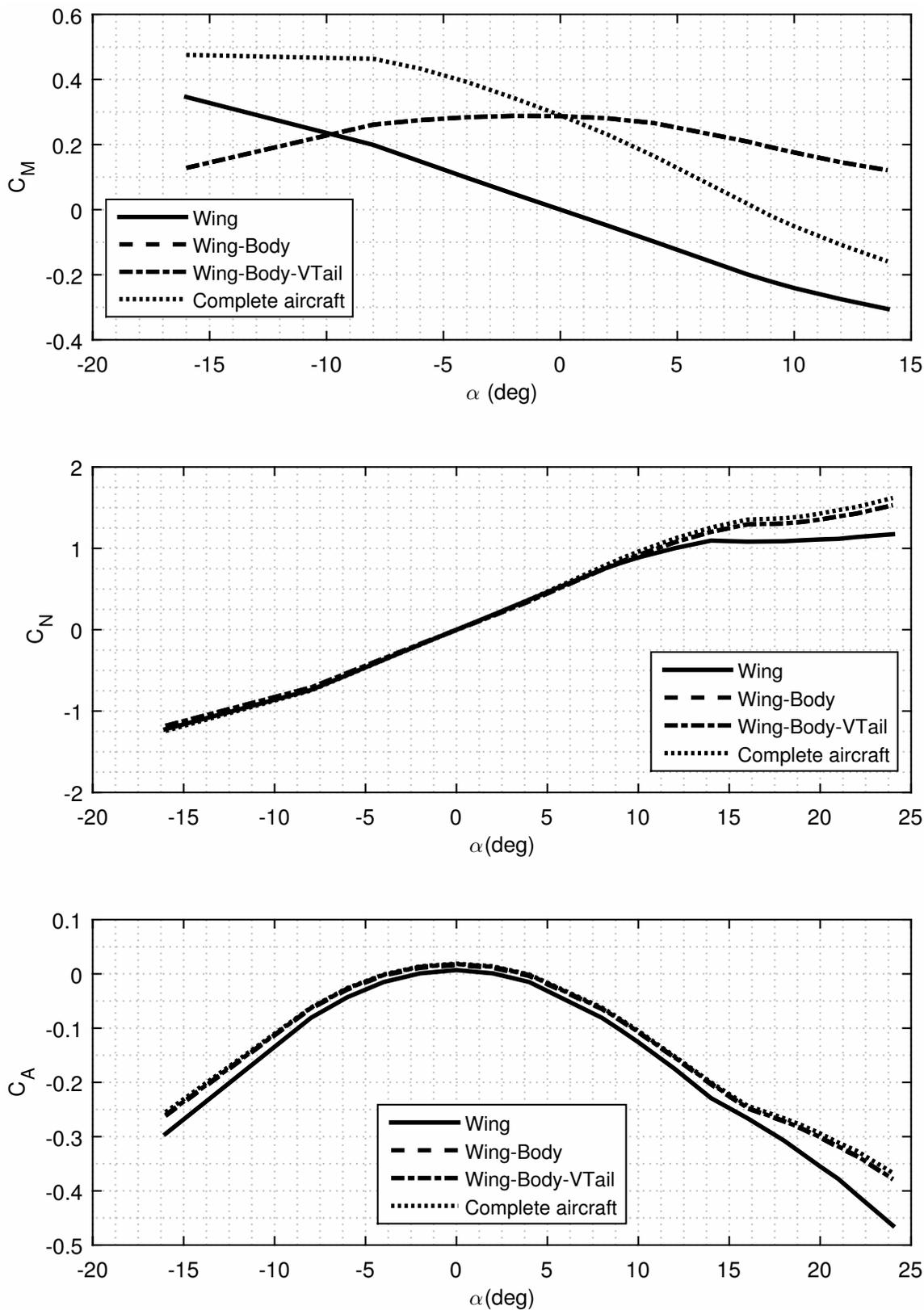


Figura 9.24 Velivolo ATR72. Curve del coefficiente di momento C_M e dei coefficienti di forza assiale e normale C_A e C_N al variare dell'angolo d'attacco per le configurazioni *Wing*, *Wing-Body*, *Wing-Body-Vertical Tail* e velivolo completo.

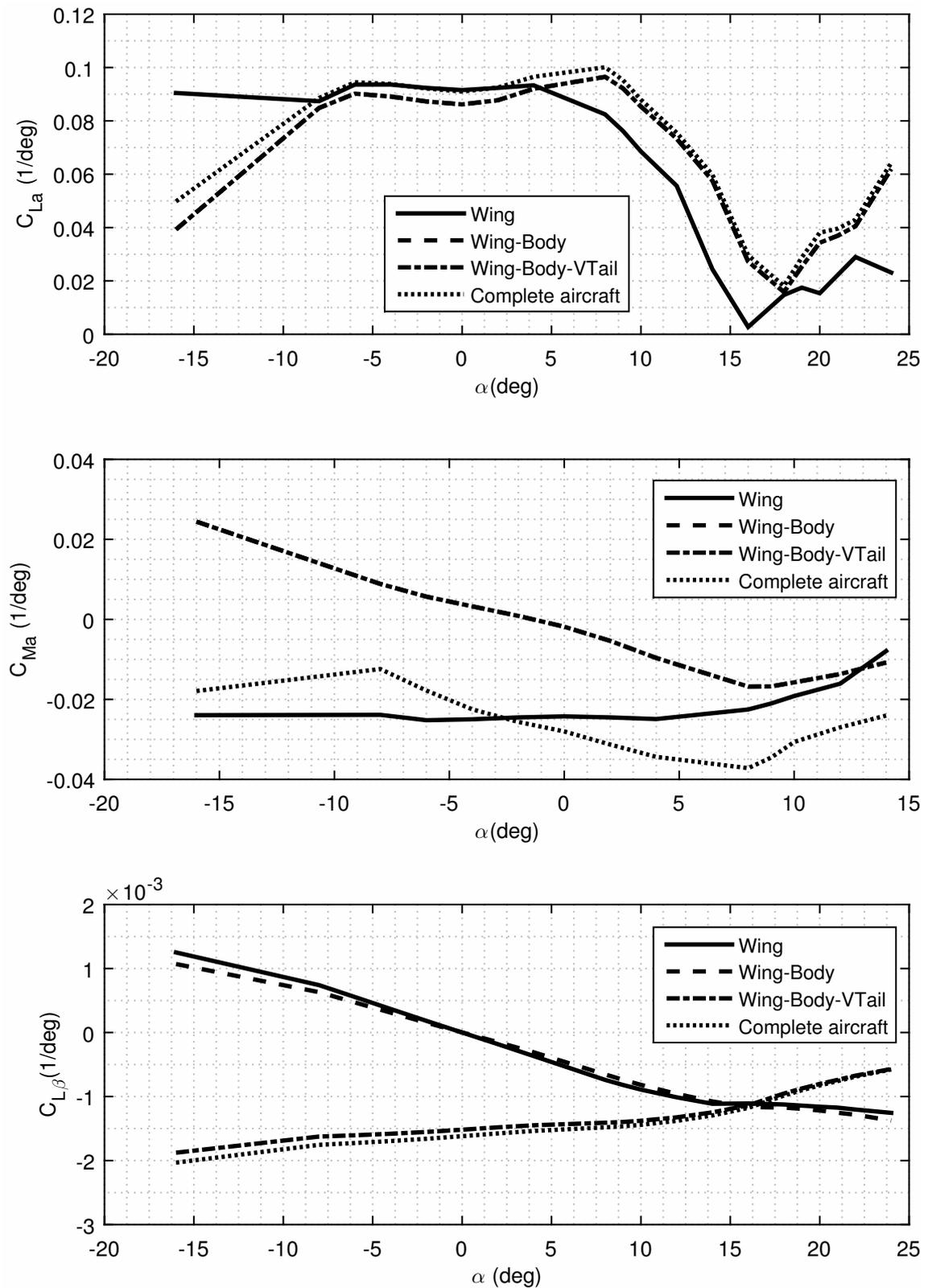


Figura 9.25 Velivolo ATR72. Curve del coefficiente di momento C_M e dei coefficienti di forza assiale e normale C_A e C_N al variare dell'angolo d'attacco per le configurazioni *Wing*, *Wing-Body*, *Wing-Body-Vertical Tail* e velivolo completo.

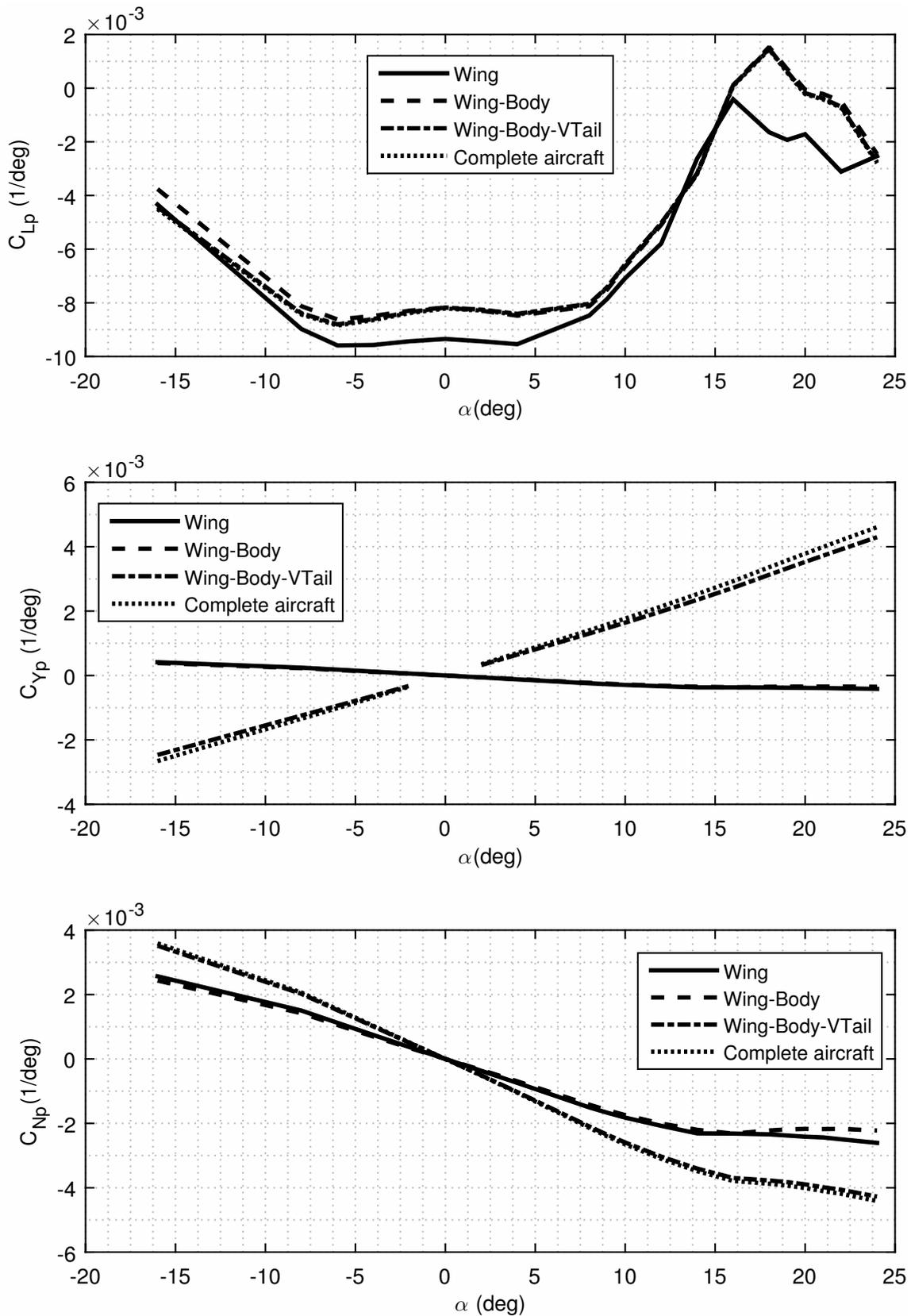


Figura 9.26 Velivolo ATR72. Variazione delle derivate dei coefficienti aerodinamici di momento C_{Lp} , C_{Yp} e C_{Np} in funzione dell'angolo d'attacco per le configurazioni *Wing*, *Wing-Body*, *Wing-Body-Vertical Tail* e velivolo completo.

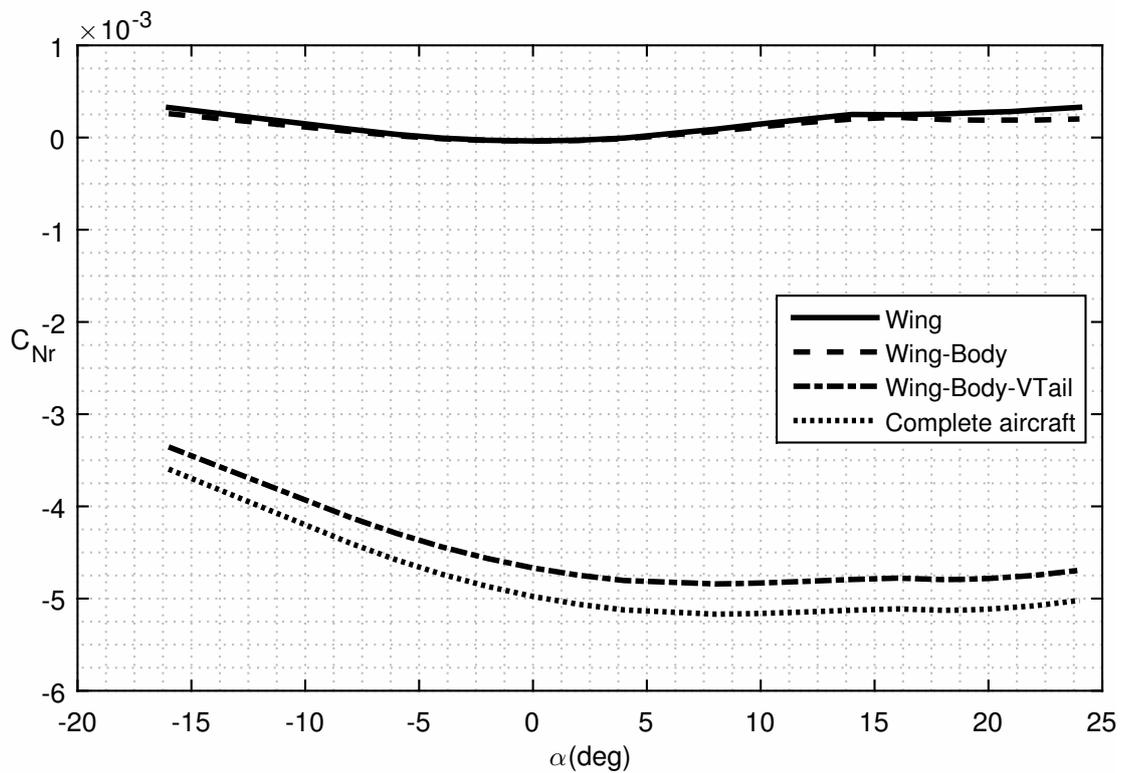
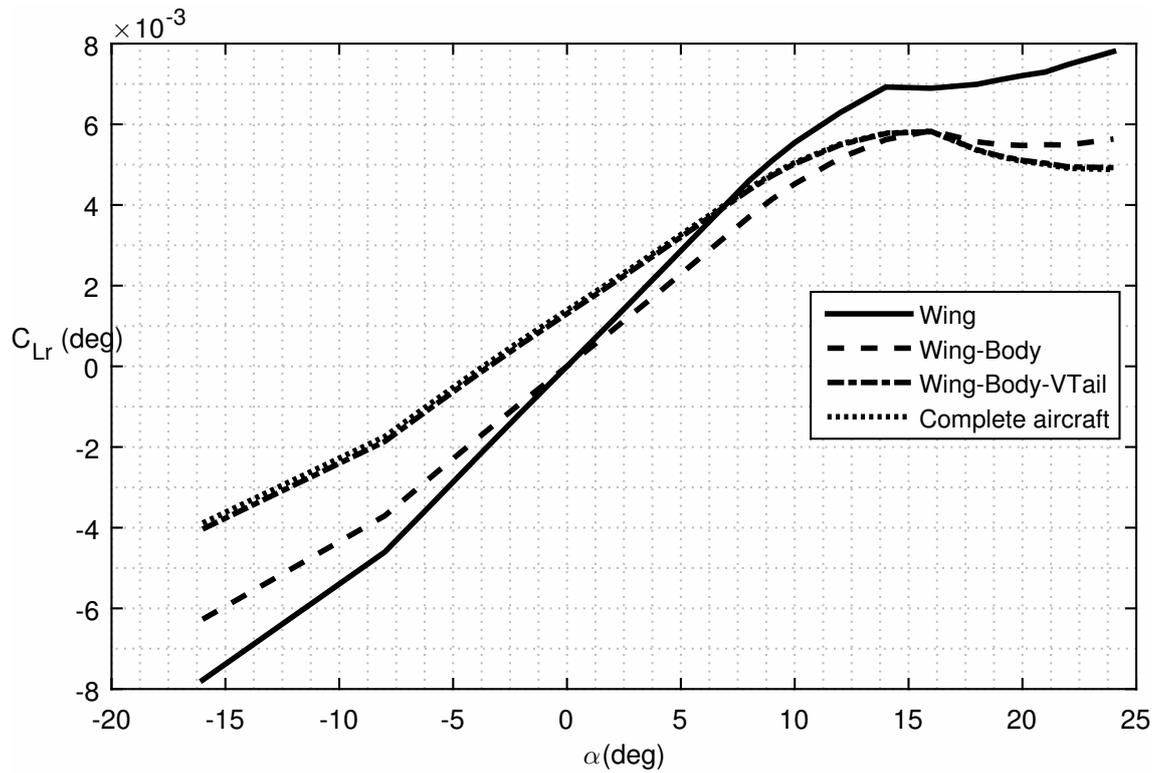


Figura 9.27 Velivolo ATR72. Variazione delle derivate dei coefficienti aerodinamici di momento C_{Lr} e C_{Nr} in funzione dell'angolo d'attacco per le configurazioni *Wing*, *Wing-Body*, *Wing-Body-Vertical Tail* e velivolo completo.

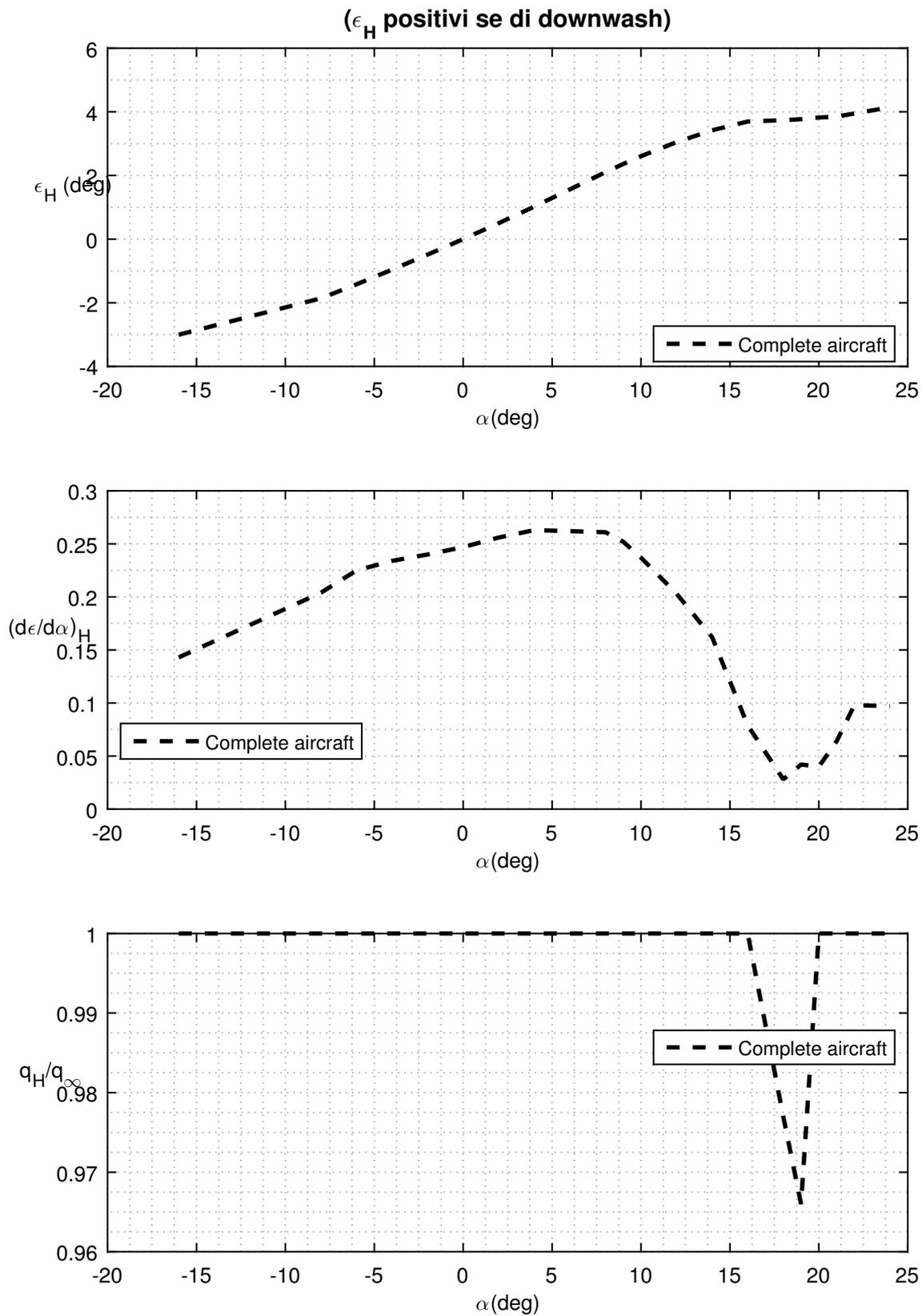


Figura 9.28 Velivolo ATR72. Variazione dell'angolo di downwash mesio ϵ_H , della sua derivata $(\frac{d\epsilon}{d\alpha})_H$ in funzione dell'angolo d'attacco per la configurazione del velivolo completo.

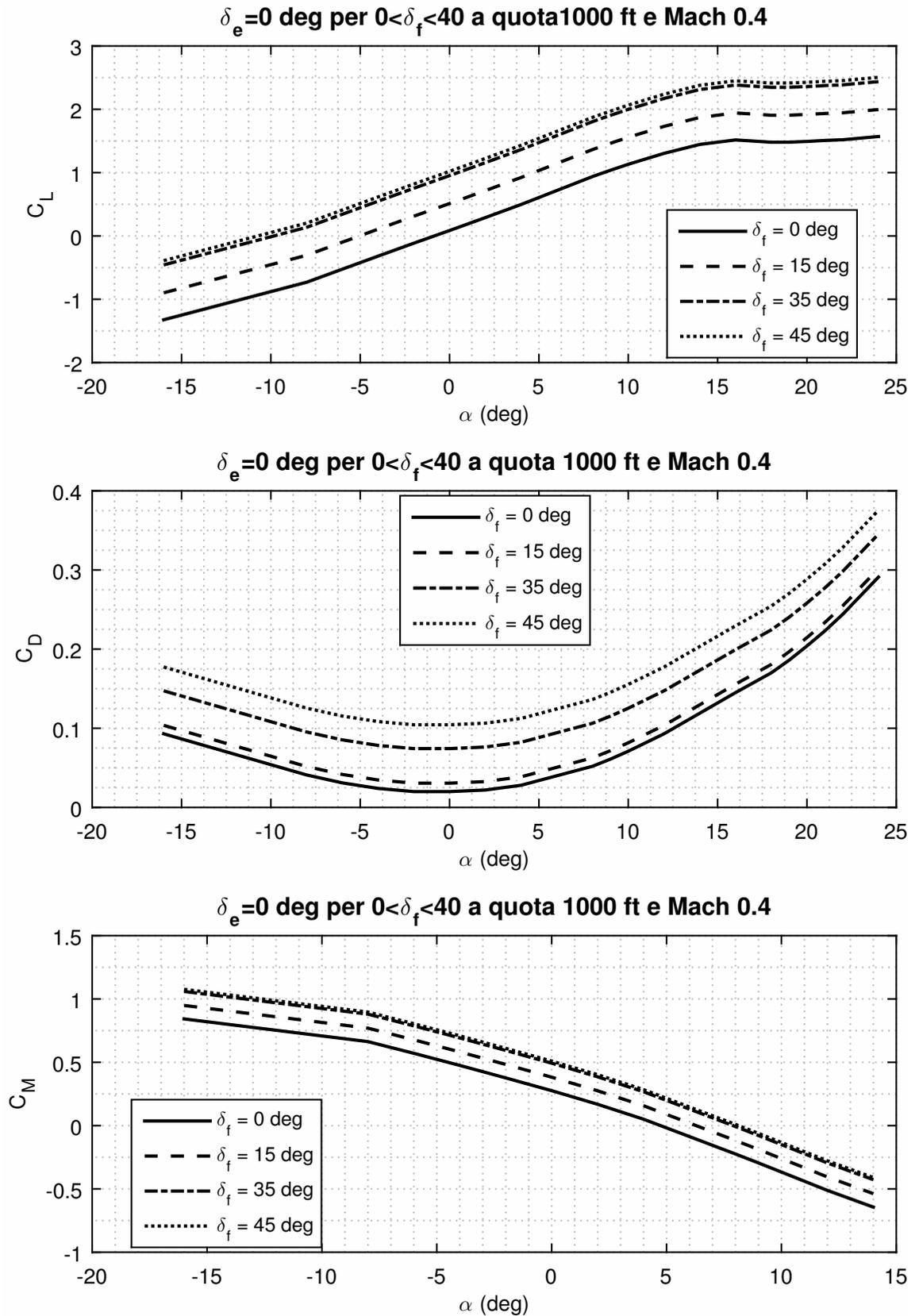


Figura 9.29 Velivolo ATR72. Curve del coefficiente di portanza C_L , del coefficiente di resistenza C_D e del coefficiente di momento di beccheggio C_M al variare dell'angolo di deflessione dei flap per fissato valore dell'angolo di deflessione dell'elevatore parti a $\delta_e = 0$ nel caso della configurazione del velivolo completo.

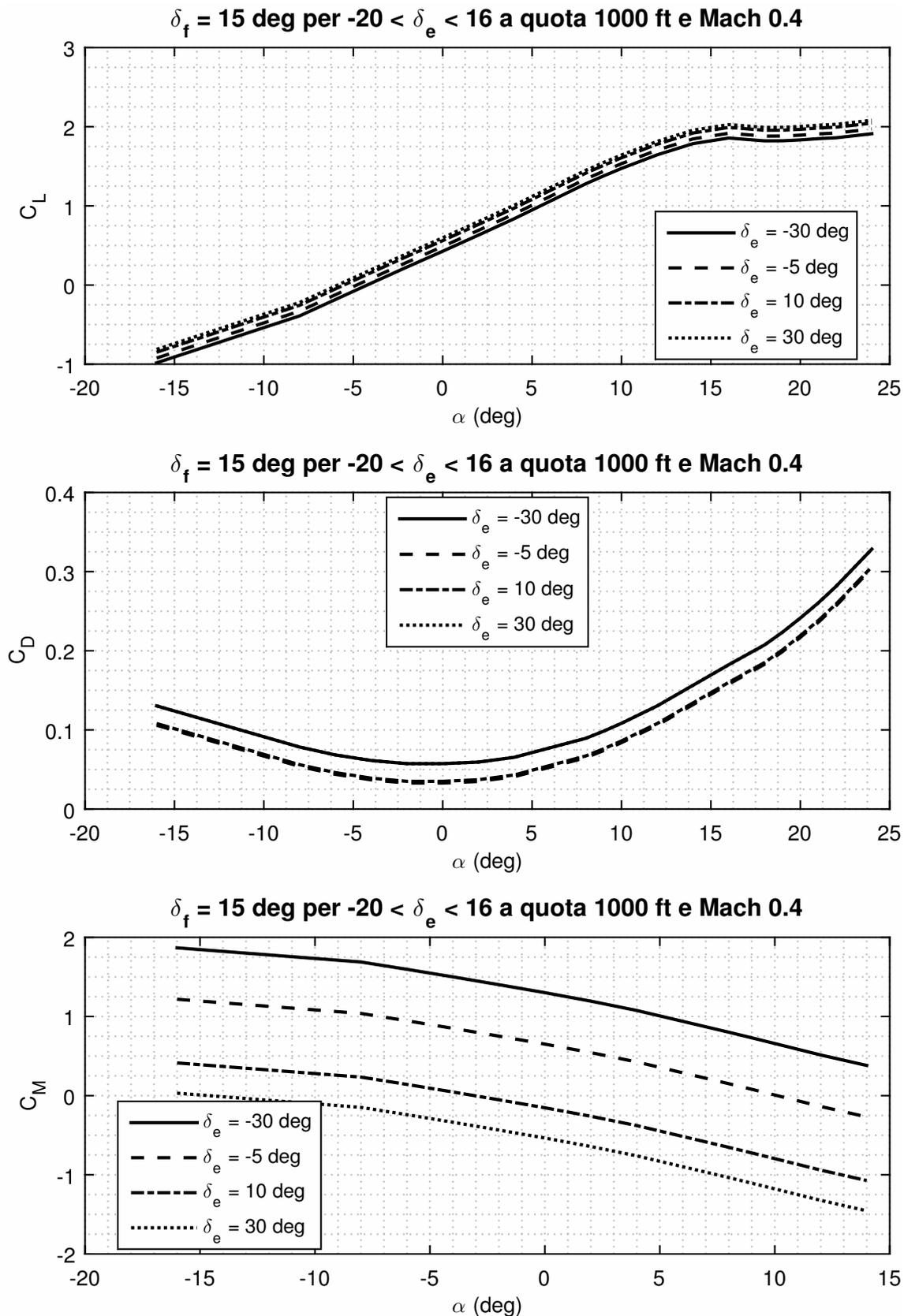


Figura 9.30 Velivolo ATR72. Curve del coefficiente di portanza C_L , del coefficiente di resistenza C_D e del coefficiente di momento di beccheggio C_M al variare dell'angolo di deflessione dell'elevatore per fissato angolo di deflessione dei flap pari a $\delta_e = 15 \text{ deg}$ nel caso della configurazione del velivolo completo.

Moto Longitudinal-simmetrico a comandi liberi

Indice

11.1	Introduzione	149
11.2	Impostazione del problema	150
11.3	Soluzioni moto a comandi liberi - <i>Esercizio 11-1</i>	155
11.3.1	Listato integrale per la risoluzione dell'esercizio 11-1	159
11.4	Angolo di flottaggio dell'equilibratore <i>Esercizio 11-2</i>	162
11.4.1	Listato integrale per la risoluzione dell'esercizio 11-2	168
11.5	Effetto del <i>trim tab</i> - <i>Esercizio 11-3</i>	172
11.6	Manovra <i>cabra-picchia</i> ed evoluzione a comandi liberi - <i>Esercizio 11-4</i>	176

11.1 Introduzione

Lo scopo di questa esercitazione è la rappresentazione del moto longitudinal-simmetrico a comandi liberi di un velivolo rigido.

Nell'ipotesi di comandi liberi, il pilota non esercita alcuna azione sui comandi e le superfici di governo dell'aeromobile sono libere di ruotare nella corrente. Le variazioni di forma conseguenti a tali rotazioni tipicamente non determinano una significativa redistribuzione delle masse del velivolo, sicchè è lecito, nello studio dell'evoluzione dinamica, continuare ad utilizzare le equazioni del moto valide per corpo rigido. Tuttavia, ai sei gradi di libertà rappresentati dalle deflessioni delle superfici di controllo, che, influenzano l'entità delle forze e dei momenti aerodinamici, hanno effetto sull'evoluzione del velivolo stesso e, dunque, vengono a costituire nuove variabili di stato, ovvero nuove incognite del problema.

Perciò, il sistema di equazioni differenziali presentato nell'ambito dell'esercitazione sui moti longitudinali a comandi bloccati deve essere amplificato con l'introduzione

di tante equazioni quanti sono i comandi longitudinali dell'aeromobile. Queste ultime esprimono l'equilibrio alla rotazione delle superfici di governo attorno al proprio asse di cerniera.

Nel presente elaborato si procede alla risoluzione di un problema ai valori iniziali per tale sistema di equazioni. Si assume lo stesso modello aerodinamico e propulsivo per il volo a 3-DoF impiegato nel caso di moto a comandi bloccati. In particolare sarà descritta e discussa l'evoluzione a comandi liberi successiva a una certa manovra (evoluzione a comandi bloccati) e valutato l'effetto del *trim tab*.

11.2 Impostazione del problema

Il sistema di equazioni del volo longitudinal-simmetrico a comandi bloccati in forma matriciale assume la seguente forma:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & M_{3,2} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \dot{V} \\ \dot{\alpha} \\ \dot{q} \\ \dot{x}_{\hat{E},G} \\ \dot{z}_{\hat{E},G} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} f_1(V, \alpha, Z_{\hat{E},G}, \theta, \delta_T, \delta_e, \delta_s) \\ f_2(V, \alpha, q, Z_{\hat{E},G}, \theta, \delta_T, \delta_e, \delta_s) \\ f_3(V, \alpha, q, Z_{\hat{E},G}, \delta_T, \delta_e, \delta_s, \dot{\delta}_t, \dot{\delta}_e) \\ f_4(V, \alpha, \theta) \\ f_5(V, \alpha, \theta) \\ f_6(q) \end{pmatrix} \quad (11.1)$$

tenendo conto anche degli effetti dovuti alla velocità di deflessione $\dot{\delta}_e$ dell'equilibratore e della deflessione $\dot{\delta}_t$ di un'eventuale aletta tab.

Si riportano di seguito le espressioni del termine di massa addizionale apparente M_{32} e dei secondi membri f_i (con $i=1, \dots, 6$):

$$M_{3,2} = -\frac{1}{4\mu} \frac{\bar{c}^2 V}{K_y^2 b} C_{M\dot{\alpha}} \quad (11.2)$$

$$f_1 = g \left[\frac{\delta_T T_{MAX}}{W} \cos(\alpha_0 - \mu_x + \mu_T) - \sin(\theta - \mu_x + \alpha) \right]$$

$$-\frac{\rho V^2}{2(W/S)} \left[C_{D_0} + k \left(C_{L_\alpha} \alpha + C_{L_{\delta_e}} \delta_e + C_{L_{\delta_s}} \delta_s \right)^m \right] \quad (11.3a)$$

$$f_2 = \frac{1}{1 + \frac{c/b}{4\mu} C_{L\dot{\alpha}}} \left[q \left(\frac{1 + \bar{c}/b}{4\mu} C_{Lq} \right) - \frac{\delta_T \Gamma_{MAX}}{W} \frac{g}{V} \sin(\alpha_0 - \mu_x + \mu_T) + \frac{g}{V} \cos(\alpha - \mu_x + \mu_T) - \frac{\rho V^2}{2(W/S)} \frac{g}{V} \left(C_{L_\alpha} \alpha + C_{L_{\delta_e}} \delta_e + C_{L_{\delta_s}} \delta_s \right) \right] \quad (11.3b)$$

$$f_3 = \frac{V^2}{K_y^2} \frac{\bar{c}/b}{2\mu Z_{E,G}} \left[C_{\mathcal{M},T} + C_{\mathcal{M}} + C_{\mathcal{M}_\alpha} \alpha + C_{\mathcal{M}_{\delta_e}} \delta_e + C_{\mathcal{M}_{\delta_s}} \delta_s + \frac{\bar{c}}{2V} \left(C_{\mathcal{M}_q} + C_{\mathcal{M}_{\dot{\delta}_e}} \dot{\delta}_e \right) \right] \quad (11.3c)$$

$$f_4 = V \cos(\theta + \mu_x - \alpha) \quad (11.3d)$$

$$f_5 = V \sin(\theta + \mu_x - \alpha) \quad (11.3e)$$

$$f_6 = q \quad (11.3f)$$

Sono visibili nelle equazioni le dipendenze del δ_e dell'equilibratore e la sua derivata $\dot{\delta}_e$, che nel caso di comandi bloccati costituiscono dei parametri di controllo determinati dalla legge di comando impartita dal pilota e, dunque, sono termini noti da un punto di vista matematico.

Al contrario, nel moto a comandi liberi δ_e e $\dot{\delta}_e$ sono incognite del problema, essendo correlati alla dinamica del velivolo. Per tale motivo, è necessario aggiungere al sistema 11.3 a fronte un'equazione in più, che esprime l'equilibrio alla rotazione dell'equilibratore attorno al suo asse di cerniera. Tale equazione nel caso più generale ha la seguente espressione:

$$I_{h_e} \ddot{\delta}_e e - \dot{q} \left(m_e e e_{x_{B,C_e}} - I_{h_e} \cos(\Lambda_{h_e}) \right) + m_e \left(a_{G_{z_B}} - g_{z_B} \right) e_e = \mathcal{H}_{A,e} \quad (11.4)$$

dove $\mathcal{H}_{A,e}$ rappresenta il momento di cerniera aerodinamica agente sull'equilibratore per il quale si assume la seguente espressione:

$$\begin{aligned} \mathcal{H}_{A,e} &= \bar{q}_\infty S_e \bar{c}_e C_{\mathcal{H}_{A,e}} = \\ &= \bar{q}_\infty S_e \bar{c}_e \left[C_{\mathcal{H}_0} + C_{\mathcal{H}_\alpha} \alpha_H + C_{\mathcal{H}_{\delta_e}} \delta_e C_{\mathcal{H}_{\delta_s}} \delta_s + C_{\mathcal{H}_{\delta_{st}}} \delta_t \right. \\ &\quad \left. + \frac{\bar{c}_e}{2V} \left(C_{\mathcal{H}_{\dot{\alpha}}} \dot{\alpha}_H + C_{\mathcal{H}_q} q + C_{\mathcal{H}_{\dot{\delta}_e}} \dot{\delta}_e \right) \right] \end{aligned} \quad (11.5a)$$

L'equazione 11.4 è un'equazione differenziale ordinaria del secondo ordine, che va affiancarsi a un sistema di equazioni differenziali ordinarie del primo ordine. Al fine di non aumentare l'ordine del sistema, si può riscrivere come insieme di due equazioni differenziali del primo ordine, tenendo conto che:

$$\ddot{\delta}_e = \frac{d(\dot{\delta}_e)}{dt}$$

In più consideranto che:

$$a_{G_{ZB}} - g_{ZB} = \dot{V} \sin(\alpha - \mu_x) - V(\dot{\theta} - \dot{\alpha}) \cos(\alpha - \mu_x) - g \cos \theta \quad (11.6)$$

$$\alpha_H = \left(1 - \frac{d\varepsilon}{dt}\right) \alpha_B - \varepsilon_0 + \delta_s + \mu_x \quad (11.7)$$

e avendo posto $I_{h,e} = m_e k_e^2$, la 11.4 nella pagina precedente si trasforma nelle equazioni:

$$\begin{aligned} \frac{d(\dot{\delta}_e)}{dt} = f_7 = & \left(\frac{e_e x_{B,C_e}}{K_e^2} - \cos \Lambda_{h_e} \right) \dot{q} \\ & - \frac{e_e}{k_e^2} \left[\dot{V} \sin(\alpha - \mu_x) - V(\dot{\theta} - \dot{\alpha}) \cos(\alpha - \mu_x) - g \cos \theta \right] \\ & - \frac{\rho V^2 S_e \bar{c}_e}{2m_e K_e^2} \left\{ C_{\mathcal{H}_0} + C_{\mathcal{H}_\alpha} \left[\left(1 - \frac{d\varepsilon}{dt}\right) (\alpha - \mu_x) - \varepsilon_0 + \delta_s + \mu_x \right] \right. \\ & \quad + C_{\mathcal{H}_{\delta_e}} \delta_e + C_{\mathcal{H}_{\delta_s}} \delta_s + C_{\mathcal{H}_{\delta_t}} \delta_t \\ & \quad \left. + \frac{\bar{c}_e}{2V} \left(C_{\mathcal{H}_{\dot{\alpha}}} \left(1 - \frac{d\varepsilon}{dt}\right) \dot{\alpha} + C_{\mathcal{H}_q} q + C_{\mathcal{H}_{\dot{\delta}_e}} \dot{\delta}_e \right) \right. \end{aligned} \quad (11.8)$$

$$\frac{d(\delta_e)}{dt} = f_8 = \dot{\delta}_e \quad (11.9)$$

Aggiungendo le equazioni 11.8 e 11.9 al sistema 11.1 a pagina 150 si arriva al sistema di equazioni del moto longitudinal-simmetrico a comandi liberi.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & M_{3,2} & 1 & 0 & 0 & 0 & 0 & M_{38} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ M_{71} & M_{72} & M_{73} & 0 & 0 & M_{76} & 1 & M_{78} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \end{pmatrix} \quad (11.10)$$

con le seguenti posizioni:

$$\begin{aligned} V = x_1, & \quad \alpha = x_2, & \quad q = x_3, & \quad x_{E,G} = x_4, & \quad (11.11) \\ z_{E,G} = x_5, & \quad \theta = x_6, & \quad \ddot{\delta}_e = x_7, & \quad \dot{\delta}_e = x_8 \end{aligned}$$

Le espressioni degli elementi della matrice delle masse aggiuntive apparenti sono:

$$M_{38} = -\frac{1}{4\mu} \frac{\bar{c}^2}{K_y^2} \frac{V}{b} C_{\mathcal{H}\delta_e} \quad (11.12a)$$

$$M_{71} = -\frac{e_e}{K_e^2} \sin(\alpha - \mu_x) \quad (11.12b)$$

$$M_{72} = \frac{e_e}{K_e^2} V \cos(\alpha - \mu_x) - \frac{\rho V S_e \bar{c}_e^2}{4m_e K_e^2} \left(1 - \frac{d\varepsilon}{dt}\right) \quad (11.12c)$$

$$M_{72} = \frac{e_e}{K_e^2} V \cos(\alpha - \mu_x) - \frac{\rho V S_e \bar{c}_e^2}{4m_e K_e^2} \left(1 - \frac{d\varepsilon}{dt}\right) \quad (11.12d)$$

$$M_{73} = -\left(\frac{e_e x_{B,C_e}}{K_e^2} - \cos \Lambda_{h_e}\right) \quad (11.12e)$$

$$M_{76} = -\frac{e_e}{K_e^2} V \cos(\alpha - \mu_x) \quad (11.12f)$$

$$M_{78} = -\frac{\rho V S_e \bar{c}_e^2}{4m_e K_e^2} C_{\mathcal{H}\delta_e} \quad (11.12g)$$

In effetti, gli elementi M_{7j} con $j = 1, 2, 3, 6, 8$ sono i coefficienti, cambiati di segno, delle corrispondenti derivate delle variabili di stato \dot{x}_j all'interno dell'equazione 11.8 a fronte. Allo stesso modo, l'elemento M_{38} è il coefficiente, cambiato di segno, della derivata \dot{x}_8 nell'equazione 11.12e.

Infine, le f_i con $i = 1, 2, 4, 5, 6, 8$ sono date dalle equazioni 11.15 nella pagina seguente e 11.9 nella pagina precedente, mentre le f_3 e f_7 vanno ridefinite eliminando in esse i termini contenenti le derivate \dot{x}_j delle variabili di stato. In questo modo la matrice delle masse apparenti non risulterà essere più identica.

Con una notazione più sintetica il sistema di equazioni differenziali 11.15 nella pagina seguente si può riscrivere come:

$$[M(x, t)] \left\{ \dot{x}(t) \right\} = \left\{ f(x, u, t) \right\} \quad (11.13)$$

Essendo la matrice delle masse apparenti $M(x, t)$ non identica, il sistema non è scritto in forma normale. Per ricondursi alla forma normale, bisogna premoltiplicare ambo i membri per l'inversa della matrice $M(x, t)$. L'equazione che si ottiene è:

$$\left\{ \dot{x}(t) \right\} = [M(x, t)]^{-1} \left\{ f(x, u, t) \right\} \quad (11.14)$$

dove:

$$\begin{bmatrix}
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & -M_{3,2} & 1 & 0 & 0 & 0 & 0 & -M_{38} \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 -M_{71} & M_{32}M_{73} - M_{72} & -M_{73} & 0 & 0 & -M_{76} & 1 & M_{38}M_{73} - M_{78} \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{bmatrix} \quad (11.15)$$

Per brevità e per la scarsa utilità che avrebbe nella presente trattazione, si tralascia la forma scalare del sistema di equazioni 11.16 nella pagina successiva

11.3 Soluzioni moto a comandi liberi - *Esercizio 11-1*

E' stato sviluppato un codice di calcolo che andasse a implementare le equazioni del sistema 11.14 a pagina 153 propoteste precedentemente per lo studio del moto longitudinal simmetrico nel caso di comandi liberi.

Per completezza si riporta la forma del tipo di sistema che è stata risolta nel seguito.

$$\left\{ \dot{x}(t) \right\} = \left[M(x, t) \right]^{-1} \left\{ f(x, u, t) \right\} \quad (11.16)$$

Questo sistema può essere svolto secondo due procedure differenti, esplicitati di seguito:

- Si implemena la matrice di massa, tramite la funzione apposita fornita dal docente, **MassStickFree**, dopo di ciò si inverte all'interno della seconda funzione scritta appositamente per implementare la funzione $f(t,x)$, definita con il nome **LongEquilibriumDynamicStickFree**.
- Si decide di predisporre la funzione ode45 per risolvere sistemi algebrico-differenziali (*Differential-Algebraic Equation, DAE*) che hanno la forma dell'equazione 11.16

Si è scelto di procedere con il primo dei due modi mensionati sopra. Di seguito viene riportato il listato scritto per la risoluzione di tale esercizio.

Listato 11.1 Codice per la risoluzione del sistema 11.14 a pagina 153

```

...continua dalla condizione di Trim per il velivolo proposto
%Risoluzione del sistema di equazioni 7.42
t_0 = 0.0;           % tempo iniziale
t_f = 10.0;         % tempo di osservazione totale      (s)

%Leggi di comando
v_time = [t_0:0.001:t_f]';
n_times = length(v_time);

v_t_delta_T = [v_time, ones(n_times,1)*delta_T_0];
v_t_delta_s = [v_time, ones(n_times,1)*delta_s_0];
v_t_delta_e = [v_time, ones(n_times,1)*delta_e_0];
v_t_delta_tab = [v_time, ones(n_times,1)*delta_tab_0];

%Condizioni Iniziali
x0 = [V_0; alpha_0; q_0; 0.0; z_0; theta_0; dot_delta_e_0; delta_e_0];

%Risoluzione ODE
[T,Y] = ode45(@longEquilibriumDynamicStickFree, v_time, x0);

```

Tale codice prende origine dalla condizioni di trim, che per sintesi è stata omessa, in quanto il codice relativo al trim è identico a quello presentato nell'Esercizio 7.6.

Si è scelto quindi di definire le leggi temporali dei parametri di controllo forniti in input al sistema ($\delta_e, \delta_s, \delta_T, \delta_t$), da eseguire durante la simulazione, si è poi integrato numericamente il sistema mediante la funzione `ode45` per il sistema 11.16. Per completezza espositiva è stata riportata la funzione **MassStickFree**, fornita dal docente, che viene passata col puntatore alla funzione `ode45`.

Listato 11.2 **LongEquilibriumDynamicStickFree**

```
function dxdt = longEquilibriumDynamicStickFree (t,x)
```

```

%Dichiarazioni delle variabili globali
global g ...
    z_0 V_0 q_0 gamma_0... %condizioni iniziali
    rho_0... %densità dell'aria Sea Level
    myAC... %oggetto my aircraft da file t
    v_t_delta_s... %legge dei comandi e vettore tempo da interpolare
    v_t_delta_T ...
    delta_tab_0

%rinomino le variabili x in ingresso per facilitare le espressioni
%-- Vettore di Stato
V = x(1);
alpha = x(2);
q = x(3);
x_G_E = x(4);
z_G_E = x(5);
theta = x(6);
dot_delta_e = x(7);
delta_e = x(8);

%% check if elevator is acceptable
if (delta_e < myAC.delta_e_min)
    delta_e = myAC.delta_e_min;
    x(8) = delta_e;
    if (dot_delta_e < 0)
        dot_delta_e = 0;
        x(7) = 0;
    end
end
if (delta_e > myAC.delta_e_max)
    delta_e = myAC.delta_e_max;
    x(8) = delta_e;
    if (dot_delta_e > 0)
        dot_delta_e = 0;
        x(7) = 0;
    end
end
end

gamma = x(6) + myAC.mu_x - x(2); %angolo di rampa
%Richiamo il modello dell'atmosfera
[air_Temp, sound_speed, air_pressure, rho] = atmosisa(-z_G_E);
%preallocazione

mu_rel = (myAC.W / g) / (rho_0* myAC.S* myAC.b); %densità relativa

%interpolazione del vettore dei comandi
delta_s = interp1(v_t_delta_s(:,1),v_t_delta_s(:,2),t);
delta_T = interp1(v_t_delta_T(:,1),v_t_delta_T(:,2),t);

F1 = g*( ...
    (delta_T*myAC.T/myAC.W)*cos(x(2) - myAC.mu_x + myAC.mu_T) ...
    -sin(gamma) ...
    -((rho_0*x(1)^2)/(2*(myAC.W/myAC.S))) ...
    *(myAC.CD_0 + myAC.K*((myAC.CL_alpha*x(2) ...
        + myAC.CL_delta_e*x(8) ...
        + myAC.CL_delta_s*delta_s)^myAC.m)) ...
    );

F2 = ( ...
    ( 1. - myAC.CL_q*(myAC.mac/myAC.b)/(4*mu_rel) ) * x(3) ...
    - (delta_T*myAC.T/myAC.W)*(g/x(1))*sin(x(2) - myAC.mu_x + myAC.mu_T) ...
    +(g/x(1))*cos(gamma) ...

```

```

-(g/x(1))*((rho_0*x(1)^2)/(2*(myAC.W/myAC.S))) ...
*( ...
  myAC.CL_alpha*x(2) ...
  + myAC.CL_delta_e*x(8) ...
  + myAC.CL_delta_s*delta_s) ...
)/( 1. + myAC.CL_alpha_dot*(myAC.mac/myAC.b)/(4*mu_rel) );

F3 = ((x(1)^2/myAC.k_y^2)*(myAC.mac/myAC.b)/(2*mu_rel)) ...
*( ... % Aerodynamics
  myAC.Cm_0 + myAC.Cm_alpha*x(2) ...
  + myAC.Cm_delta_s*delta_s ...
  + myAC.Cm_delta_e*x(8) ...
  + (myAC.mac/(2*x(1)))*(myAC.Cm_q*x(3)+ myAC.Cm_alpha_dot*F2) ...
  ... % Thrust
  + (myAC.Cm_T_0 + myAC.Cm_T_alpha*x(2))*delta_T ...
);

F4 = x(1)*cos(gamma) ;

F5 = -x(1)*sin(gamma) ;

F6 = x(3) ;

F7 = (myAC.ec_adim*g*cos(x(6))/myAC.k_e^2) ...
+ (rho_0 * (x(1)^2) * myAC.S_e * myAC.mac_e...
*(1/(2*(myAC.k_e^2) * myAC.mass_e)))...
*(myAC.Ch_e_0 + myAC.Ch_e_delta_e*x(8) + myAC.Ch_e_delta_s*delta_s ...
+ myAC.Ch_e_delta_tab*delta_tab_0 +(myAC.mac_e/(2*x(1)))*myAC.Ch_e_q*x(3)
...
+ myAC.Ch_e_alpha*((1-myAC.DepsDalpha)*(x(2)-myAC.mu_x)-myAC.eps_0 ...
+ myAC.mu_x + delta_s) ...
);

F8 = x(7);

f = [F1;F2;F3;F4;F5;F6;F7;F8];

M=MassStickFree(t,x);
nM=inv(M);
dxdt=nM*f;

end

```

Listato 11.3 MassStickFree

```

function M = MassStickFree(t,x)
global ...
  g ...           % gravity acceleration
  myAC           % the aircraft object

%% Give variable proper names
V           = x(1);
alpha      = x(2);
q          = x(3);
xEG       = x(4);
zEG       = x(5);
theta     = x(6);
dot_delta_e = x(7);
delta_e   = x(8);

%% Using Matlab built-in ISA model for density
[air_Temp, sound_speed, air_pressure, rho] = atmosisa(-zEG);

%% Aircraft relative density
mu_rel = (myAC.W/g)/(rho*myAC.S*myAC.b);

```

```

%% Mass matrix
M=eye(8);

M(3,2) = ...
    -(1/(4*mu_rel))*(myAC.mac^2/myAC.k_y^2) ...
    *(V/myAC.b)*myAC.Cm_alpha_dot;

M(3,8) = ...
    -(1/(4*mu_rel))*(myAC.mac^2/myAC.k_y^2) ...
    *(V/myAC.b)*myAC.Cm_delta_e_dot;

M(7,1) = ...
    (myAC.mac_e*myAC.ec_adim)/myAC.k_e^2 ...
    *sin(alpha-myAC.mu_x);

M(7,2) = ...
    ((myAC.mac_e*myAC.ec_adim)/myAC.k_e^2) ...
    *V*cos(alpha-myAC.mu_x) ...
    -rho*V*myAC.S_e*myAC.mac_e^2/(4*myAC.I_e) ...
    *(1 - myAC.DepsDalpha)*myAC.Ch_e_alpha_dot;

M(7,3) = ...
    cos(myAC.Lambda_e) ...
    - myAC.mac_e*myAC.ec_adim*myAC.x_C_e/myAC.k_e^2;

M(7,6) = ...
    -(myAC.mac_e*myAC.ec_adim)/myAC.k_e^2 ...
    *V*cos(alpha-myAC.mu_x);

M(7,8) = ...
    -rho*V*myAC.S_e*myAC.mac_e^2/(4*myAC.I_e) ...
    *myAC.Ch_e_delta_e_dot;

end

```

11.3.1 Listato integrale per la risoluzione dell'esercizio 11-1

Listato 11.4 Listato integrale per la risoluzione dell'esercizio 11-2

```

=====
%Esercizio 11.1
=====
clear all; clc; close all;

%% Script di risoluzione del problema di valori iniziali, moto a 3DoF, comandi
  bloccati

disp(['Aircraft motion, 3DoF.']);
disp(['Solving the problem of trim at given altitude and flight speed.']);

%% Declarations
global ...
g ...                % accelerazione di gravità
z_0 V_0 q_0 gamma_0 ... % condizioni iniziali
rho_0 ...            % densità aria quota z_0
myAC ...             % aereo
v_t_delta_T ...     % time histories comandi
v_t_delta_s ...
delta_tab_0

%% Populate aircraft data
aircraftDataFileName = 'DSV_Aircraft_data.txt';

%% Aircraft object
myAC = DSVACraft(aircraftDataFileName);

if (myAC.err == -1)
    disp(['... terminating.']);
else
    disp(['File ', aircraftDataFileName, ' letto correttamente.']);
end

%% Condizioni iniziali
z_0 = -4000.;          % altitudine dal livello mare (m)
V_0 = 257.;           % Velocità
q_0 = 0.;             % velocità angolare beccheggio (rad/s)
gamma_0 = convang(0, 'deg', 'rad'); % angolo di rampa (rad)

%% Using Matlab built-in ISA model for density
[air_Temp_0, sound_speed_0, air_pressure_0, rho_0] = atmosisa(-z_0);

%% Accelerazione di gravità
g = 9.81; % (m/s^2)
%% Condizioni design per valori iniziali
xi0 = [ ...
    0; ... % alpha_0
    0; ... % delta_e_0
    0; ... % delta_s_0
    0.5 ... % delta_T_0
    ];

%% Minimizzazione della funzione di costo - Condizioni Trim
% Confini
lb = [convang(-15, 'deg', 'rad'), ... % minimo alpha
      convang(-20, 'deg', 'rad'), ... % minima deflessione equilibratore
      convang(-5, 'deg', 'rad'), ... % minima incidenza stabilizzatore
      0.2 ... % minimo rateo spinta
      ];
ub = [convang( 15, 'deg', 'rad'), ... % massimo alpha
      convang( 13, 'deg', 'rad'), ... % massima deflessione equilibratore
      convang( 5, 'deg', 'rad'), ... % massima incidenza stabilizzatore
      1.0 ... % massimo rateo di spinta
      ];

```

```

];

options = optimset('tolfun',1e-9,'Algorithm','interior-point');
[xi,fval] = ...
    fmincon(@costLongEquilibriumStaticStickFixed, ...
        xi0, ...
        [], ...
        [], ...
        [], ...
        [], ...
        lb,ub, ...
        @myNonLinearConstraint, ...
        options);

alpha_0      = xi(1);
alpha_0_deg  = convang(alpha_0,'rad','deg');

theta_0      = gamma_0 + alpha_0 + myAC.mu_x;
theta_0_deg  = convang(theta_0,'rad','deg');

delta_e_0    = xi(2);
delta_e_0_deg = convang(delta_e_0,'rad','deg');
dot_delta_e_0 = 0.0;

delta_s_0    = xi(3);
delta_s_0_deg = convang(xi(3),'rad','deg');

delta_T_0    = xi(4);

delta_tab_0  = 0.0;
delta_tab_0_deg = convang(delta_tab_0,'rad','deg');

% Visualizzazione Monitor valori di TRIM
disp('')
disp('Condizione di trim:')
disp(['Velocità V_0= ',num2str(V_0),' m/s'])
disp(['Angolo d'attacco alpha_0= ',num2str(alpha_0_deg),'deg'])
disp(['Elevatore delta_e_0= ',num2str(delta_e_0_deg),'deg'])
disp(['Stabilizzatore delta_s_0= ',num2str(delta_s_0_deg),'deg'])
disp(['Manetta delta_T_0= ',num2str(delta_T_0)])
disp(['Andolo del Tab= ',num2str(delta_tab_0_deg)])

%%Risoluzione del sistema di equazioni 7.42
t_0 = 0.0;           % tempo iniziale
t_f = 10.0;         % tempo di osservazione totale      (s)

%Leggi di comando
v_time = [t_0:0.001:t_f]';
n_times = length(v_time);

v_t_delta_T = [v_time,ones(n_times,1)*delta_T_0];
v_t_delta_s = [v_time,ones(n_times,1)*delta_s_0];
v_t_delta_e = [v_time,ones(n_times,1)*delta_e_0];
v_t_delta_tab = [v_time,ones(n_times,1)*delta_tab_0];

%Condizioni Iniziali
x0 = [V_0;alpha_0;q_0;0.0;z_0;theta_0;dot_delta_e_0;delta_e_0];

%Risoluzione ODE
[T,Y] = ode45(@longEquilibriumDynamicStickFree,v_time,x0);

% derived vars
v_gamma = Y(:,6) + myAC.mu_x - Y(:,2);
v_dgamma = gradient(v_gamma,T);
v_dV      = gradient(Y(:,1),T);

```

```

v_fxa = -sin(v_gamma)-v_dV./g;
v_fza = cos(v_gamma)+(v_dgamma.*Y(:,1))./g;
v_dq = gradient(Y(:,3),T);

%Verifica MOMENTO DI CERNIERA

delta_s = interp1(v_t_delta_s(:,1),v_t_delta_s(:,2),T);
delta_T = interp1(v_t_delta_T(:,1),v_t_delta_T(:,2),T);

Ch_e = myAC.Ch_e_0 + myAC.Ch_e_delta_e*Y(:,8) + myAC.Ch_e_delta_s*delta_s ...
      + myAC.Ch_e_delta_tab*delta_tab_0 +(myAC.mac_e/(2*Y(:,1)))*myAC.Ch_e_q*Y
      (:,3)...
      + myAC.Ch_e_alpha*((1-myAC.DepsDalpha)*(Y(:,2)-myAC.mu_x)-myAC.eps_0 ...
      + myAC.mu_x + delta_s);

%% PLOT RISULTATI

```

11.4 Angolo di flottaggio dell'equilibratore *Esercizio 11-2*

Per l'esercizio proposto ci è stato richiesto di ricavare la deflessione dell'equilibratore necessaria per il mantenimento del volo equilibrato per una condizione di comandi liberi. Si è pensato quindi di partire da una condizione di volo longitudinal simmetrico a comandi bloccati e a partire da un certo istante di tempo, si immagina che il pilota lasci il comando longitudinale, continuando così il volo a comandi liberi, per tale condizione di volo ci è stato richiesto di ricavare il valore della deflessione dell'equilibratore.

Le caratteristiche del velivolo, quindi di massa, geometria, propulsione e aerodinamica sono state fornite al nostro codice mediante un file di testo opportunamente realizzato per poter essere letto dalla funzione `matlab Aircraft.m` realizzata dal docente. Un'esempio delle caratteristiche aerodinamiche fornite in input alla nostra simulazione e contenute nel file `DSV_Aircraft_Data.txt` è riportato nel listato seguente.

Listato 11.5 Caratteristiche aerodinamiche dell'equilibratore

```
AERODYNAMICS
  0.0000  Ch_e_0
 -0.130   Ch_e_alpha (1/rad)
 -0.110   Ch_e_delta_s (1/rad)
 -0.300   Ch_e_delta_e (1/rad)
 -0.550   Ch_e_delta_tab (1/rad)
 -1.000   Ch_e_delta_e_dot (1/rad)
 -1.00    Ch_e_q (1/rad)
 -0.50    Ch_e_alpha_dot (1/rad)
```

Le condizini iniziali assegnate, vengono riportate nel seguente listato:

Listato 11.6 Condizioni iniziali assegnate

```
Quota z_0 = -4000m
Velocità V_0 = 257 m/s
Velocità di beccheggio q(0)=q_0=0.00 rad/s
Angolo di rampa, gamma(0)=gamma_0=0.00 deg
```

Per queste condizioni iniziali, si determinano le condizioni di volo equilibrato a comandi bloccati, supponendo che:

- l'angolo del piano di coda sia fissato a $\delta_{s,0} = -1.00$ deg;
- grado di ammissione della manetta fisso a $\delta_{T,0} = 0.5$.

I valori di α_0 e $\delta_{e,0}$ corrispondenti a condizioni di volo equilibrato sono ottenuti con la solita procedura vista a partire dagli esercizi del quaderno 7.

Listato 11.7 Condizioni di Trim

```
Condizione di trim:
Velocità V_0 = 257 m/s
Angolo d'attacco alpha_0 = 3.6294 deg
Elevatore delta_e_0 = -4.3021 deg
Stabilizzatore delta_s_0 = -1 deg          %<--Parametro Bloccato
Manetta delta_T_0 = 0.5                   %<--Parametro Bloccato
Angolo del Tab = 0
```

Per la risoluzione dell'equazione del moto, è stato supposto che:

- Il volo avvenga a comandi bloccati dal tempo $t_0 = 0$ s fino al tempo $t_1 = 1$ s;

- Stabilito il tempo finale della simulazione $t_f = 30\text{s}$, si considera dal tempo t_1 al tempo finale t_f un volo a comandi liberi. In altri termini è come se il pilota lasciasse i comandi liberi all'istante t_1 . Per tale esercitazione è stata ritenuta nulla la deflessione dell'aletta tab $\delta_t = 0.00\text{ deg}$.

Una possibile risoluzione del problema è riportata di seguito.

Listato 11.8 Codice per risoluzione esercizio 11.2

```

=====
%Esercizio 11.2
=====

%% PASSO1 RICERCA CONDIZIONE DI TRIM - funzione minimizzazione di costo
...porzione di codice omessa in quanto uguale a quanto visto fino a questo
momento...

%% PASSO2 Integrazione dell'istante t0=0s all'istante t1=1s comandi bloccati

t_0=0;      %[s]
t_1=1;      %[s]

v_time1=t_0:0.01:t_1;
n_times1 = length(v_time1);
v_t_delta_T_1 = [v_time1;ones(1,n_times1)*delta_T_0];
v_t_delta_s_1 = [v_time1;ones(1,n_times1)*convang(delta_s_0_deg,'deg','rad')];
v_t_delta_e_1 = [v_time1;ones(1,n_times1)*convang(delta_e_0_deg,'deg','rad')];

xeg0=0.;
x01=[V_0,alpha_0,q_0,xeg0,z_0,gamma_0];

[T1,Y1]=ode45(@eqLongDynamicStickFixed,v_time1,x01);

V1=Y1(:,1);
alpha1=Y1(:,2);
q1=Y1(:,3);
xeg1=Y1(:,4);
zeg1=Y1(:,5);
gamma1=Y1(:,6);
theta1=gamma1+alpha1-myAC.mu_x;

%% PASSO3 Risoluzione del sistema di equazioni 7.42, dall'istante t1=1s a
% tf=30s per il caso di comandi liberi

t_i=1;      %[s]
t_f = 30.0;  %[s]

v_time = [t_i:0.001:t_f]';
n_times = length(v_time);
%Vettori riga tempi
v_t_delta_T = [v_time,ones(n_times,1)*delta_T_0];
v_t_delta_s = [v_time,ones(n_times,1)*delta_s_0];
v_t_delta_e = [v_time,ones(n_times,1)*delta_e_0];
v_t_delta_tab = [v_time,ones(n_times,1)*delta_tab_0];

% Condizioni Iniziali, a partire dalle condizioni di comandi bloccati
x02 = [V1(end),alpha1(end),q1(end),xeg1(end),zeg1(end),theta1(end),...
      dot_delta_e_0,delta_e_0];

%options_ODE = odeset('AbsTol',1e-6,'Mass',@MassStickFree);

%Risoluzione per comandi liberi
[T2,Y2] = ode45(@longEquilibriumDynamicStickFree,v_time,x02);

```

Si è proceduto nel seguente modo:

- Nella prima parte dello script si è integrato il sistema relativo ai comandi bloccati, mediante la funzione `ode45` alla quale viene passata la funzione `eqLongDynamicStickFixed`, riportata nel listato 7.5 a pagina 88 per la risoluzione dell'esercizio 7.8, che implementa i secondi membri del suddetto sistema.
- Nella seconda parte dello script si è integrato, invece il sistema relativo ai comandi liberi 11.15 a pagina 154. In tal caso, invece, si è ricorso alla `ode45` ma passandole la funzione `LongEquilibriumDynamicStickFree`, utilizzata nell'esercizio 11.1, 11.2 a pagina 155 la quale implementa i secondi membri del suddetto sistema. Da notare che le condizioni iniziali, in questo caso sono gli ultimi termini del vettore di stato Y_1 , ottenuto dall'integrazione precedente.

La figura 11.1 a pagina 166 mostra la storia temporale delle variabili di stato del velivolo considerato.

La figura 11.2 a pagina 166 riporta la storia temporale dell'andamento dell'angolo di deflessione, dei fattori di carico e dell'accelerazione angolare nei primi 5s successivi all'istante t_0 . Si nota che la superficie lasciata libera di ruotare tende a disporsi ad una posizione angolare $\delta_e \approx 2$ deg, il cosiddetto *angolo di flottaggio* dell'equilibratore. In questa condizione asintotica si può verificare che il momento di cerniera tende ad annullarsi. In particolare, il momento di cerniera \mathcal{H}_c è dato dalla somma del momento aerodinamico espresso nella 11.5a a pagina 151 e dal momento dovuto all'eccentricità del baricentro rispetto all'asse di cerniera, essendo nullo il momento di comando, per ipotesi di comandi liberi. Per calcolare il momento di cerniera è bastato aggiungere il seguente codice.

Listato 11.9 Codice per il calcolo del momento di cerniera, esercizio 11.2

```
%Estrazione grandezze d'interesse:
V2=Y2(:,1);
alpha2=Y2(:,2);
q2=Y2(:,3);
xeg2=Y2(:,4);
zeg2=Y2(:,5);
theta2=Y2(:,6);
dot_delta_e1=Y2(:,7);
delta_e1=Y2(:,8);

%Composizione dei vettori di stato di tutta la manovra:(vettori riga)
T=[T1' T2']; %[s]

V=[V1' V2']; %[m/s]
alpha=[alpha1' alpha2']; %[rad]
q=[q1' q2']; %[rad/s]
xeg=[xeg1' xeg2']; %[m]
zeg=[zeg1' zeg2']; %[m]
theta=[theta1' theta2']; %[rad]
gamma=theta-alpha+myAC.mu_x; %[rad]

dot_delta_e=[zeros(1,length(T1)) dot_delta_e1'];
delta_e=[ones(1,length(T1))*delta_e_0 delta_e1'];

%Valuto gli scostamenti dalle condizioni iniziali
Dzeg=zeg-z_0;
Dalphi=alpha-alpha_0;
DV=V-V_0;
Dq=q-q_0;

%Trasformazione delle variabili:
Dalphi_deg=convang(Dalphi,'rad','deg');
```

```

q_deg=convangvel(Dq, 'rad/s', 'deg/s');
theta_deg=convang(theta, 'rad', 'deg');
gamma_deg=convang(gamma, 'rad', 'deg');
delta_e_deg=convang(delta_e, 'rad', 'deg');

%Valuto i fattori di carico e la variazione di velocita angolare di
%beccheggio
v_dot=gradient(V,T);
q_dot=gradient(q,T); %[rad/s]
q_dot_deg_s=convangvel(q_dot, 'rad/s', 'deg/s');
gamma_dot=gradient(gamma,T);
fxa = -sin(gamma)-v_dot./g;
fza = cos(gamma)+(gamma_dot.*V)./g;

%Verifica condizioni del momento di cerniera, che in tali condizioni di
%volo deve tendere a 0:
delta_s_1=interp1(v_t_delta_s_1(1,:),v_t_delta_s_1(2,:),T1);
delta_T_1=interp1(v_t_delta_T_1(1,:),v_t_delta_T_1(2,:),T1);
delta_s_2=interp1(v_t_delta_s(:,1),v_t_delta_s(:,2),T2);
delta_T_2=interp1(v_t_delta_T(:,1),v_t_delta_T(:,2),T2);

%momento di cerniera a comandi bloccati:
Ch_e_1 = myAC.Ch_e_0 + myAC.Ch_e_delta_e*delta_e_0 + myAC.Ch_e_delta_s*
delta_s_1 ...
+ myAC.Ch_e_delta_tab*delta_tab_0 +(myAC.mac_e/(2*Y1(:,1)))*myAC.
Ch_e_q*Y1(:,3)...
+ myAC.Ch_e_alpha*((1-myAC.DepsDalpha)*(Y1(:,2)-myAC.mu_x)-myAC.eps_0
...
+ myAC.mu_x + delta_s_1);

%Momento di cerniera a comandi liberi:
Ch_e_2 = myAC.Ch_e_0 + myAC.Ch_e_delta_e*Y2(:,8) + myAC.Ch_e_delta_s*
delta_s_2 ...
+ myAC.Ch_e_delta_tab*delta_tab_0 +(myAC.mac_e/(2*Y2(:,1)))*myAC.
Ch_e_q*Y2(:,3)...
+ myAC.Ch_e_alpha*((1-myAC.DepsDalpha)*(Y2(:,2)-myAC.mu_x)-myAC.eps_0
...
+ myAC.mu_x + delta_s_2);

```

Nella figura 11.3 a pagina 167, si riporta l'andamento del momento di cerniera lungo tutta la simulazione, e si può notare, che tale valore tende a zero alla fine di essa, mentre nella figura 11.4 a pagina 167 sono riportati gli andamenti delle due variabili di stato aggiuntive, nel caso di sistema a comandi liberi. Si noti che l'angolo di flottaggio sopra definito, è del valore di circa $\delta_e \approx 2$ deg.

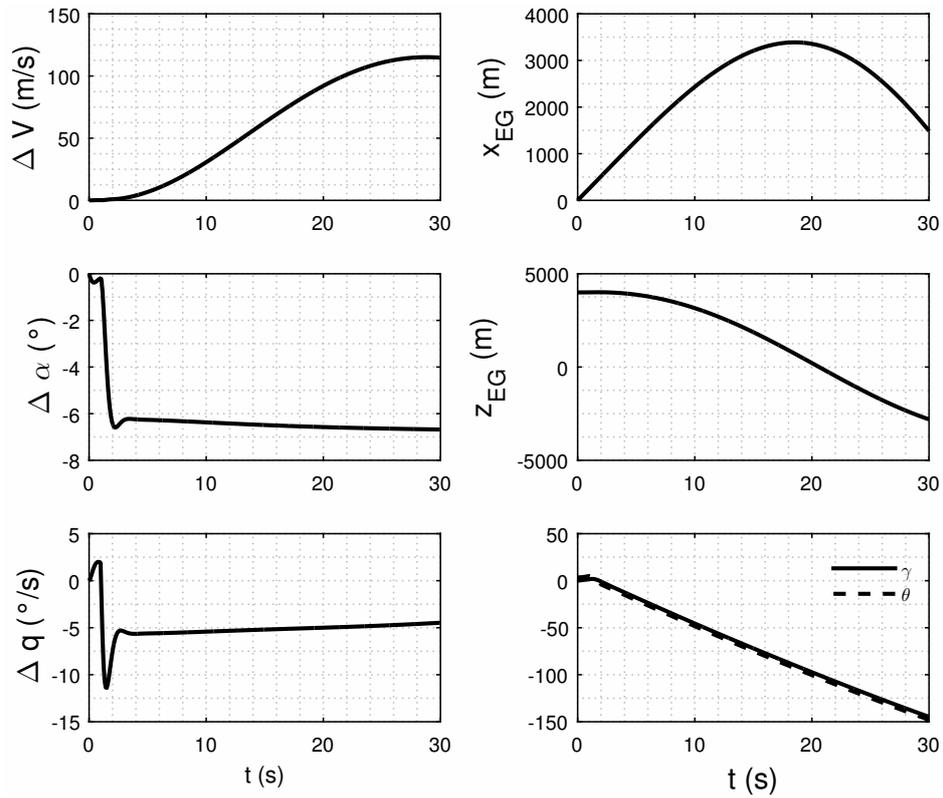


Figura 11.1 Storie temporali delle variabili di stato per comandi bloccati fino all'istante $t_1 = 1$ s e comandi liberi nei successivi istanti di tempo. Le grandezze $\Delta(*)$ rappresentano gli scostamenti rispetto ai valori iniziali.

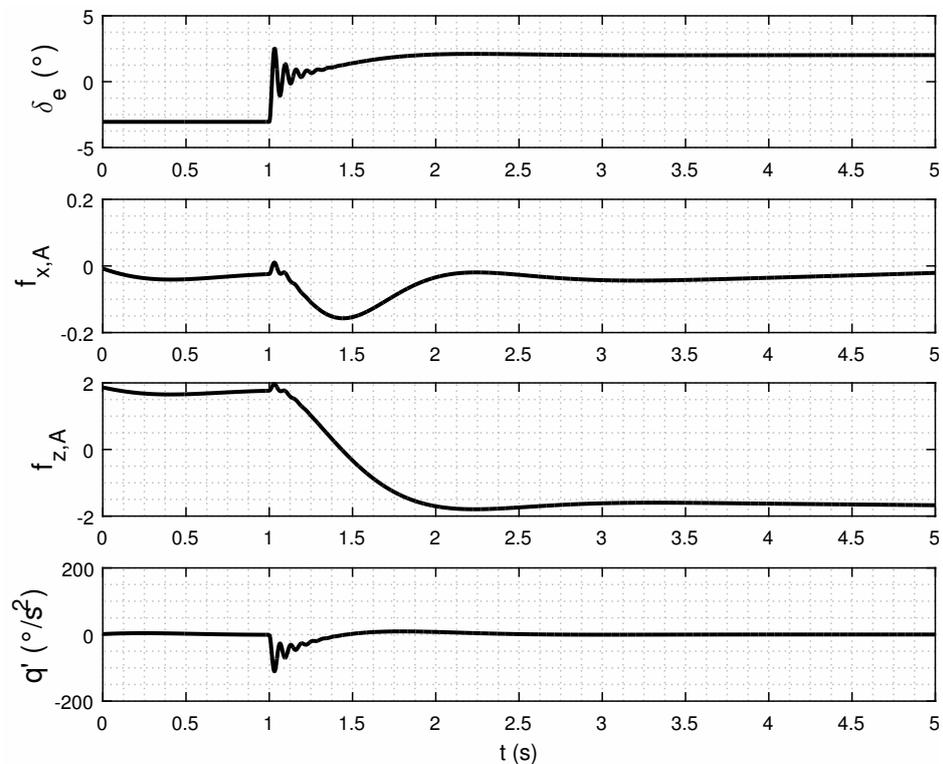


Figura 11.2 Storia temporale dell'angolo di deflessione dell'equilibratore, dei fattori di carico e dell'accelerazione angolare per comandi bloccati fino all'istante $t_1 = 1$ s e comandi liberi negli istanti di tempo successivi, a partire dalle condizioni di trime ricavate mediante la funzione `fmincon`.

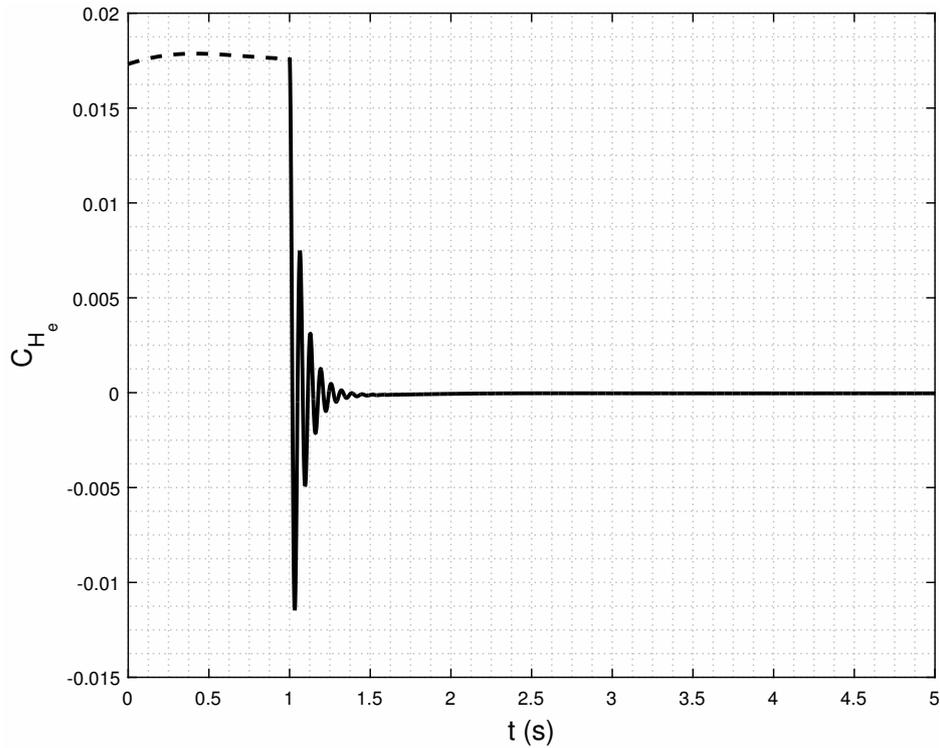


Figura 11.3 Storia temporale del momento di cerniera \mathcal{H}_e , per comandi bloccati fino all'istante $t_1 = 1$ s e comandi liberi nei successivi istanti di tempo, a partire dalle condizioni di trim ricavate mediante la funzione `fmincon`.

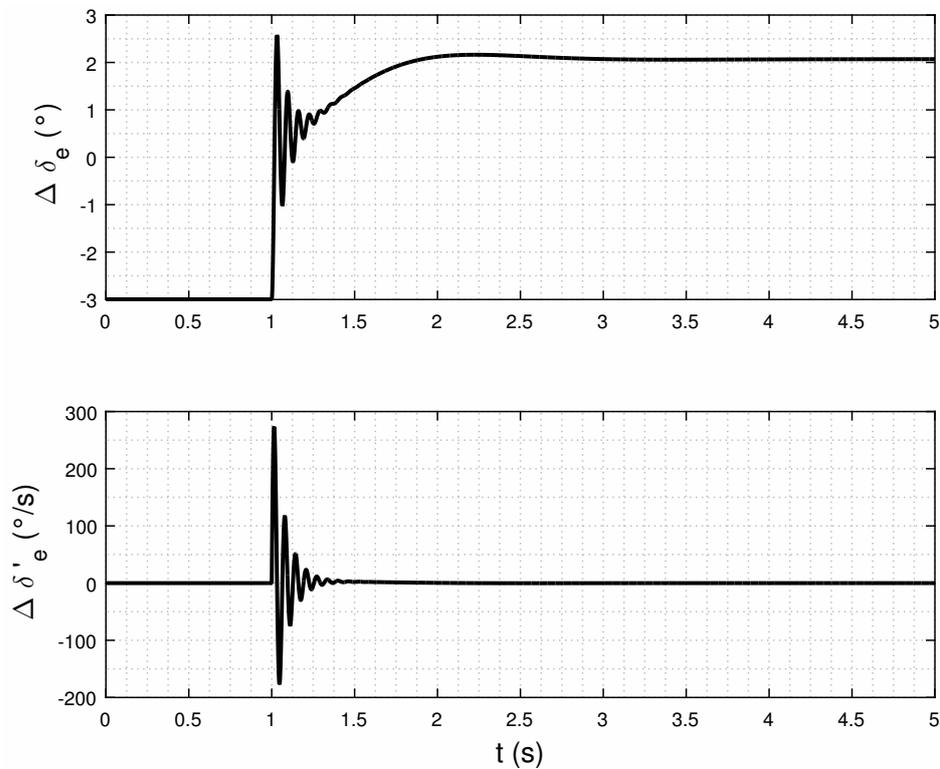


Figura 11.4 Storia temporale della variazione dell'angolo di deflessione dell'elevatore e della sua velocità, per comandi bloccati fino all'istante $t_1 = 1$ s e comandi liberi nei successivi istanti di tempo, a partire dalle condizioni di trim ricavate con la funzione `fmincon` e velivolo assegnato.

11.4.1 Listato integrale per la risoluzione dell'esercizio 11-2

Listato 11.10 Listato integrale esercizio 11-2

```

=====
%Esercizio 11.2
=====

clear all; clc; close all;

%% Script di risoluzione del problema di valori iniziali, moto a 3DoF, comandi
  bloccati

disp(['Aircraft motion, 3DoF.']);
disp(['Solving the problem of trim at given altitude and flight speed.']);

%% Dichiarazioni variabili globali
global ...
    g ...                % gravity acceleration
    z_0 V_0 q_0 gamma_0 ... % initial conditions
    rho_0 ...            % air density at z_0
    myAC ...             % the aircraft object
    v_t_delta_T ...     % command time histories
    v_t_delta_s ...
    delta_tab_0 ...
    delta_e_0 ...
    delta_s_0 ...
    delta_T_0

%% Caricamento dell'aeromobile considerato.
aircraftDataFileName = 'DSV_Aircraft_data.txt';

%% Aircraft object
myAC = DSVAircraft(aircraftDataFileName);
%% Initial conditions
z_0    = -4000.;          % a.s.l. altitude (m)
V_0    = 257.;           % flight speed
q_0    = 0.;             % pitching angular speed (rad/s)
gamma_0 = convang(0, 'deg', 'rad'); % path climb angle (rad)

%% Using Matlab built-in ISA model for density
[air_Temp_0, sound_speed_0, air_pressure_0, rho_0] = atmosisa(-z_0);

%% Gravity acceleration
g = 9.81; % (m/s^2)
%% initial guess for the design vector
xi0 = [ ...
    0; ... % alpha_0
    0; ... % delta_e_0
    0; ... % delta_s_0
    0.5 ... % delta_T_0
    ];

%% PASS01 RICERCA CONDIZIONE DI TRIM - funzione minimizzazione di costo
% Fisso la posizione dello stabilizzatore delta_s_0=-1 deg
% Aeq, in Aeq*x=beq linear constraint
Aeq    = zeros(4,4);
beq    = zeros(4,1);
Aeq(3,3) = 1; % pick delta_s
Aeq(4,4) = 1;
delta_T_0 = 0.5;
delta_s_0 = convang(-1., 'deg', 'rad');
beq(3,1) = delta_s_0; % keep delta_s fixed
beq(4,1) = delta_T_0;

```

```

% vincoli
lb =[convang(-15,'deg','rad'), ... % minimum alpha
     convang(-20,'deg','rad'), ... % minimum elevator deflection
     convang(-5,'deg','rad'), ... % minimum stabilizer incidence
     0.2 ... % minimum thrust fraction
    ];
ub =[convang( 15,'deg','rad'), ... % maximum alpha
     convang( 13,'deg','rad'), ... % maximum elevator deflection
     convang( 5,'deg','rad'), ... % maximum stabilizer incidence
     1.0 ... % maximum thrust fraction
    ];

options = optimset( ...
    'tolfun',1e-9, ... % threshold
    'Algorithm','interior-point' ... % algor. type
);
[xi,fval] = ...
    fmincon(@costLongEquilibriumStaticStickFixed, ...
        xi0, ...
        [], ... % A, A*x<=b
        [], ... % b
        Aeq, ... % Aeq, Aeq*x=beq
        beq, ... % beq
        lb,ub, ...
        @myNonLinearConstraint, ...
        options);

alpha_0 = xi(1);
alpha_0_deg = convang(alpha_0,'rad','deg');

theta_0 = gamma_0 + alpha_0 + myAC.mu_x;
theta_0_deg = convang(theta_0,'rad','deg');

delta_e_0 = xi(2);
delta_e_0_deg = convang(delta_e_0,'rad','deg');
dot_delta_e_0 = 0.0;

delta_s_0 = xi(3);
delta_s_0_deg = convang(xi(3),'rad','deg');

delta_T_0 = xi(4);

delta_tab_0 = 0.0;
delta_tab_0_deg = convang(delta_tab_0,'rad','deg');
%visualizzazione a video delle condizioni di trim:
% Visualizzazione Monitor valori di TRIM
disp('')
disp('Condizione di trim:')
disp(['Velocità V_0= ',num2str(V_0),' m/s'])
disp(['Angolo d''attacco alpha_0= ',num2str(alpha_0_deg),'deg'])
disp(['Elevatore delta_e_0= ',num2str(delta_e_0_deg),'deg'])
disp(['Stabilizzatore delta_s_0= ',num2str(delta_s_0_deg),'deg'])
disp(['Manetta delta_T_0= ',num2str(delta_T_0)])
disp(['Anolo del Tab= ',num2str(delta_tab_0_deg)])

% PASS02 Integrazione dell'istante t0=0s all'istante t1=1s comandi bloccati

t_0=0; %[s]
t_1=1; %[s]

v_time1=t_0:0.01:t_1;
n_times1 = length(v_time1);
v_t_delta_T_1 = [v_time1;ones(1,n_times1)*delta_T_0];
v_t_delta_s_1 = [v_time1;ones(1,n_times1)*convang(delta_s_0_deg,'deg','rad')];
v_t_delta_e_1 = [v_time1;ones(1,n_times1)*convang(delta_e_0_deg,'deg','rad')];

```

```

xeg0=0.;
x01=[V_0,alpha_0,q_0,xeg0,z_0,gamma_0];

[T1,Y1]=ode45(@eqLongDynamicStickFixed,v_time1,x01);

V1=Y1(:,1);
alpha1=Y1(:,2);
q1=Y1(:,3);
xeg1=Y1(:,4);
zeg1=Y1(:,5);
gamma1=Y1(:,6);
theta1=gamma1+alpha1-myAC.mu_x;

%% PASS03 Risoluzione del sistema di equazioni 7.42, dall'istante t1=1s a
% tf=30s per il caso di comandi liberi

t_i=1;           %[s]
t_f = 30.0;      %[s]

v_time = [t_i:0.001:t_f]';
n_times = length(v_time);
%Vettori riga tempi
v_t_delta_T = [v_time,ones(n_times,1)*delta_T_0];
v_t_delta_s = [v_time,ones(n_times,1)*delta_s_0];
v_t_delta_e = [v_time,ones(n_times,1)*delta_e_0];
v_t_delta_tab = [v_time,ones(n_times,1)*delta_tab_0];

% Condizioni Iniziali, a partire dalle condizioni di comandi bloccati
x02 = [...
    V1(end);      ... %[m/s]
    alpha1(end);  ... %[rad]
    q1(end);      ... %[rad/s]
    xeg1(end);    ... %[m]
    zeg1(end);    ... %[m]
    theta1(end);  ... %[rad]
    dot_delta_e_0; ... %[rad/s]
    delta_e_0 ...  %[rad]
    ];

%options_ODE = odeset('AbsTol',1e-6,'Mass',@MassStickFree);

%Risoluzione per comandi liberi
[T2,Y2] = ode45(@longEquilibriumDynamicStickFree,v_time,x02);

%Estrazione grandezze d'interesse:
V2=Y2(:,1);
alpha2=Y2(:,2);
q2=Y2(:,3);
xeg2=Y2(:,4);
zeg2=Y2(:,5);
theta2=Y2(:,6);
dot_delta_e1=Y2(:,7);
delta_e1=Y2(:,8);

%Composizione dei vettori di stato di tutta la manovra:(vettori riga)
T=[T1' T2'];           %[s]

V=[V1' V2'];           %[m/s]
alpha=[alpha1' alpha2']; %[rad]
q=[q1' q2'];           %[rad/s]
xeg=[xeg1' xeg2'];     %[m]
zeg=[zeg1' zeg2'];     %[m]
theta=[theta1' theta2']; %[rad]
gamma=theta-alpha+myAC.mu_x; %[rad]

```

```

dot_delta_e=[zeros(1,length(T1)) dot_delta_e1'];
delta_e=[ones(1,length(T1))*delta_e_0 delta_e1'];

%Valuto gli scostamenti dalle condizioni iniziali
Dzeg=zeg-z_0;
Dalpha=alpha-alpha_0;
DV=V-V_0;
Dq=q-q_0;

%Trasformazione delle variabili:
Dalpha_deg=convang(Dalpha,'rad','deg');
q_deg=convangvel(Dq,'rad/s','deg/s');
theta_deg=convang(theta,'rad','deg');
gamma_deg=convang(gamma,'rad','deg');
delta_e_deg=convang(delta_e,'rad','deg');

%Valuto i fattori di carico e la variazione di velocita angolare di
%beccheggio
v_dot=gradient(V,T);
q_dot=gradient(q,T); %[rad/s]
q_dot_deg_s=convangvel(q_dot,'rad/s','deg/s');
gamma_dot=gradient(gamma,T);
fxa = -sin(gamma)-v_dot./g;
fza = cos(gamma)+(gamma_dot.*V)./g;

%Verifica condizioni del momento di cerniera, che in tali condizioni di
%volo deve tendere a 0:
delta_s_1=interp1(v_t_delta_s_1(1,:),v_t_delta_s_1(2,:),T1);
delta_T_1=interp1(v_t_delta_T_1(1,:),v_t_delta_T_1(2,:),T1);
delta_s_2=interp1(v_t_delta_s(:,1),v_t_delta_s(:,2),T2);
delta_T_2=interp1(v_t_delta_T(:,1),v_t_delta_T(:,2),T2);

%momento di cerniera a comandi bloccati:
Ch_e_1 = myAC.Ch_e_0 + myAC.Ch_e_delta_e*delta_e_0 + myAC.Ch_e_delta_s*
delta_s_1 ...
      + myAC.Ch_e_delta_tab*delta_tab_0 +(myAC.mac_e/(2*Y1(:,1)))*myAC.
Ch_e_q*Y1(:,3)...
      + myAC.Ch_e_alpha*((1-myAC.DepsDalpha)*(Y1(:,2)-myAC.mu_x)-myAC.eps_0
...
      + myAC.mu_x + delta_s_1);

%Momento di cerniera a comandi liberi:
Ch_e_2 = myAC.Ch_e_0 + myAC.Ch_e_delta_e*Y2(:,8) + myAC.Ch_e_delta_s*
delta_s_2 ...
      + myAC.Ch_e_delta_tab*delta_tab_0 +(myAC.mac_e/(2*Y2(:,1)))*myAC.
Ch_e_q*Y2(:,3)...
      + myAC.Ch_e_alpha*((1-myAC.DepsDalpha)*(Y2(:,2)-myAC.mu_x)-myAC.eps_0
...
      + myAC.mu_x + delta_s_2);

```

11.5 Effetto del *trim tab* - Esercizio 11-3

L'uso di alette di compensazione aerodinamica, alette *tab*, è dovuto al tentativo di riduzione del momento aerodinamico di cerniera che si viene a generare in seguito ad una coppia di comando sulla superficie di governo. Questa aletta si trova sull'estremità finale della superficie mobile e definisce angoli di rotazione δ_t opposti a quelli della superficie di governo su cui agisce. Una deflessione verso l'alto della *tab*, determina una variazione di momento positiva. Praticamente l'aletta *tab* genera un momento di cerniera che è opposto al momento aerodinamico, in modo tale da rendere il volantino del pilota sgravato di parte del momento aerodinamico trasmesso al volantino. Per l'esercizio proposto si vuole calcolare il valore della deflessione dell'aletta *tab* δ_t , affinché si abbia un valore della deflessione dell'equilibratore δ_e asintotico, pari al valore della deflessione $\delta_{e,0}$ ricavato dalla condizione di trim. Per ricavare il valore del δ_t opportuno per lo scopo prefissato, si è proceduto in maniera iterativa imponendo di volta in volta un valore della deflessione dell'aletta *tab* e andando a risolvere le equazioni del moto longitudinal simmetrico a comandi liberi. Di volta in volta è stato comparato il valore della deflessione dell'equilibratore, δ_e ottenuto per la data deflessione del *tab*, con il valore dell' deflessione dell'equilibratore per le condizioni di trim, $\delta_{e,0}$.

$$\Delta \mathcal{H}_{A,e} = \bar{q} \bar{S}_e \bar{c}_e (C_{\mathcal{H}_{\delta_t}} \delta_t) \quad (11.17)$$

Per cui il momento di cerniera aerodinamico risultante relativo all'impennaggio orizzontale sarà:

$$\Delta \mathcal{H}_{A,e} = \bar{q} \bar{S}_e \bar{c}_e (C_{\mathcal{H}_\alpha} \alpha_{\mathcal{H}} + C_{\mathcal{H}_{\delta_e}} \delta_e + C_{\mathcal{H}_{\delta_t}} \delta_t) \quad (11.18)$$

A partire dalle condizioni di trim si è valutato l'effetto dell'aletta *tab* sulla deflessione dell'equilibratore. In particolare si è trovata l'incidenza opportuna dell'aletta *tab* affinché la deflessione dell'equilibratore in flottaggio sia pari a quella del trim $\delta_{e,0}$.

Listato 11.11 Legge temporale imposta alla *tab*

```
%Risoluzione del sistema di equazioni 7.42
t_0 = 0.0; %Tempo iniziale (s)
t_1 = 1.0; %Primo tempo di osservazione (s)
duration_sec = 1.0; %Durata movimento elevatore (s)
t_2 = t_1+duration_sec/2.; %2nd breakpoint (s)
t_3 = t_2+duration_sec/2.; %3rd breakpoint (s)

% Ampiezza movimento del tab (deg)
delta_tab_increment_deg = 2.3;
grad_delta_tab = ... % time rate of delta_tab (rad/s)
    convang(delta_tab_increment_deg, 'deg', 'rad')/(t_2 - t_1);

t_f = 30.0; %Tempo di osservazione totale (s)

freq_Hz = 220.; %Frequenza dello stato di propagazione, 1/delta t

n0 = floor((t_1 - t_0)*freq_Hz);
n1 = floor((t_2 - t_1)*freq_Hz);
n2 = floor((t_3 - t_2)*freq_Hz);
n3 = floor((t_f - t_3)*freq_Hz);
v_time = [linspace(t_0, t_1, n0), linspace(t_1, t_2, n1), linspace(t_2, t_3, n2),
linspace(t_3, t_f, n3)]';
v_time = unique(v_time);
n_times = length(v_time);

v_t_delta_T = [v_time, ones(n_times, 1)*delta_T_0];
v_t_delta_e = [v_time, ones(n_times, 1)*convang(delta_e_0_deg, 'deg', 'rad')];
```

```

v_t_delta_s = [v_time,ones(n_times,1)*convang(delta_s_0_deg,'deg','rad')];
v_t_delta_tab = [v_time,ones(n_times,1)*convang(delta_tab_0_deg,'deg','rad')];

% Legge di comando del TAB
for kt=1:length(v_time)
    if ((v_t_delta_tab(kt,1)>=t_1)&&(v_t_delta_e(kt,1)<t_2))
        v_t_delta_tab(kt,2)=v_t_delta_tab(kt-1,2)+(grad_delta_tab/freq_Hz);
    end
    if (v_t_delta_tab(kt,1) >= t_2)
        v_t_delta_tab(kt,2)=delta_tab_0+convang(delta_tab_increment_deg,'deg',
'rad');
    end
end

%Inizializzazione del vettore di stato
x0=[ ...
    V_0;      ...      % x1
    alpha_0; ...      % x2
    q_0;      ...      % x3
    0.0;      ...      % x4 = xEG_0
    z_0;      ...      % x5
    theta_0; ...      % x6
    dot_delta_e_0; ... % x7
    delta_e_0 ];      % x8

options_ODE = odeset('AbsTol',1e-6,'Mass',@MassStickFree);

%Risoluzione ODE
[T,Y] = ode45(@longEquilibriumDynamicStickFree,v_time',x0',options_ODE);

```

Dalla figura 11.5 nella pagina successiva si evince che l'incidenza dell'aletta tab affinché l'elevatore abbia un'incidenza asintotica pari a $\delta_{e,0} = -4,2 \text{ deg}$ è di circa $\delta_t = 2.2 \text{ deg}$.

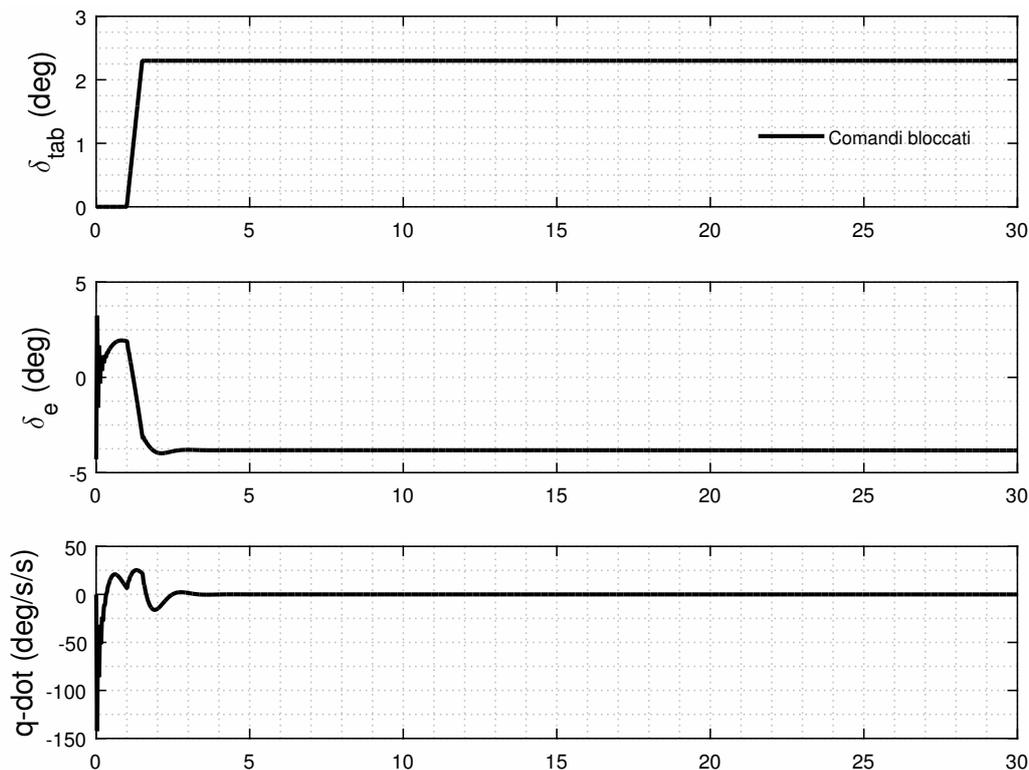


Figura 11.5 Storie temporale dell'angolo di deflessione dell'equilibratore, del tab e dell'accelerazione angolare, per comandi bloccati fino all'istante $t_1 = 1s$ e comandi liberi per i successivi istanti di tempo, a partire dalle condizioni di trim e per velivolo assegnato. Si nota come il δ_t che restituisce il valore della deflessione prossima a quella della condizione di trim è di ≈ 2.2 deg.

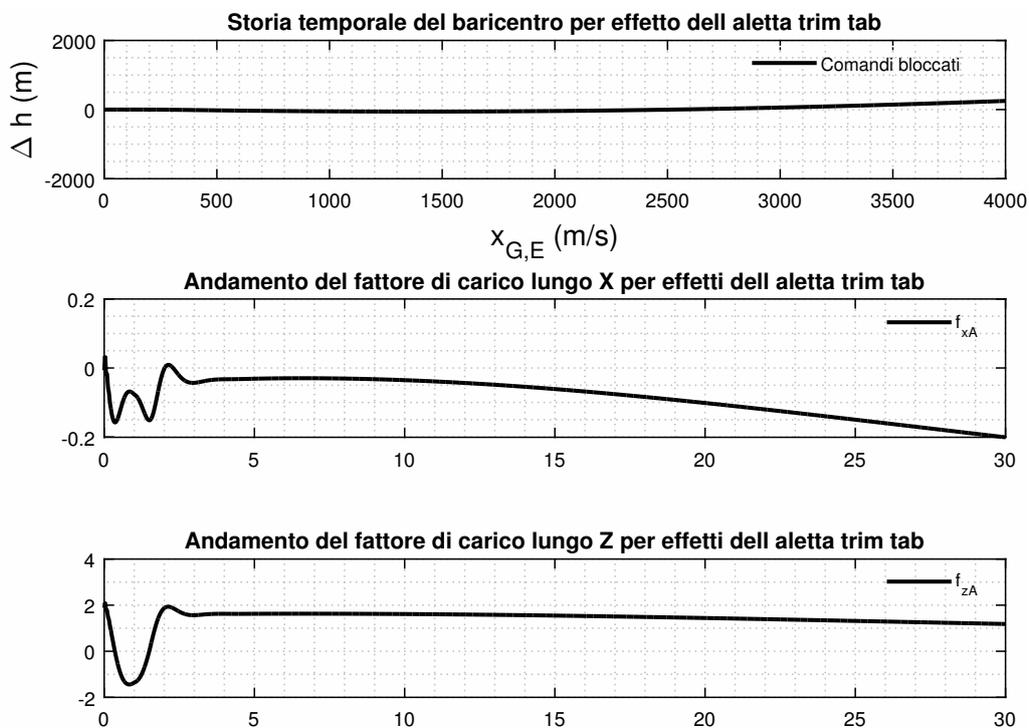


Figura 11.6 Storie temporali degli output dell'integrazione, effetto dell'aletta *trim tab*.

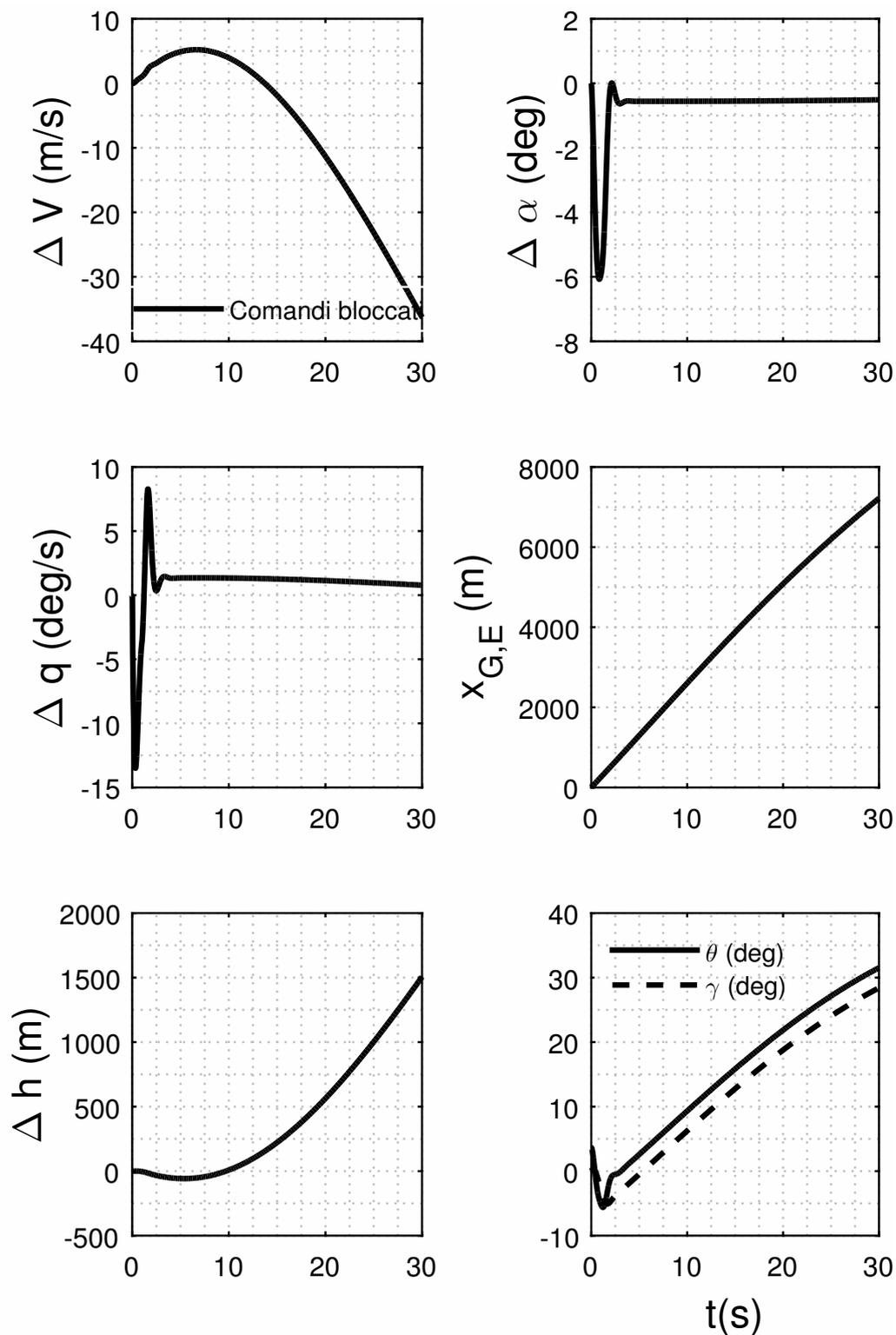


Figura 11.7 Storie temporali delle variabili di stato per la legge di deflessione del tab fornita al fine di ricercare la deflessione dell'equilibratore δ_e per alla deflessione fornita per le condizioni di trim $\delta_{e,0}$.

11.6 Manovra *cabra-picchia* ed evoluzione a comandi liberi - *Esercizio 11-4*

In questo esercizio si risolve nuovamente il problema del volo longitudinal-simmetrico, assegnando una legge di comando di δ_e del tipo *cabra-picchia* nella prima fase del moto, e nella fase successiva si suppone che il comando longitudinale venga rilasciato dal pilota nel momento in cui l'equilibratore è riportato alla posizione iniziale. In particolare, come nel caso dell'esercizio 7.10, si consideri la legge di comando δ_e costruita in questo modo:

- Al tempo $t_1 = 1s$, si mantenga l'escursione fissa $\delta_{e,0}$ (dipendente dalle condizioni di trim) fino all'istante $t_2 = 2s$.
- Dall'istante $t_2 = 2s$ si applica una legge $\delta_e(t)$ che incrementa linearmente l'escursione dell'elevatore dal valore $\delta_{e,0}$ al valore $\delta_{e,3} = -3$ deg, fino all'istante $t_3 = 2.5s$.
- Dall'istante t_3 all'istante $t_4 = 4s$, si riporta l'elevatore in posizione iniziale $\delta_{e,4} = \delta_{e,0}$.
- Per il resto della manovra ovvero dall'istante t_4 a t_{fin} il pilota lascia il volantino.

Così come avvenuto nell'esercizio precedente, anche questa volta bisognerà effettuare una doppia integrazione. I parametri in uscita dalla prima integrazione a comandi bloccati corrispondono ai parametri in ingresso per la seconda integrazione a comandi liberi.

Listato 11.12 Legge di comando cabra picchia

```

...Questa porzione di codice é da intendersi sempre a valle della condizione di
trim...
%% CABRATA-PICCHIATA

%costruzione legge
t_0 = 0.0;           %Tempo iniziale
t_1 = 1.0;           %Primo tempo di osservazione (s)
t_2cp = 2.5;         %Tempo di cabrata (s)
t_3cp = 4.0;         %Tempo di picchiata (s)
t_2 = 1.5;           %2nd breakpoint (s)
t_3 = 2.0;           %3rd breakpoint (s)
t_f = t_3cp;         %total observation time (s)

freq_Hz = 220.;     % frequency of state propagation, 1/delta t

n0 = floor((t_1 - t_0)*freq_Hz);
n1 = floor((t_2 - t_1)*freq_Hz);
n2 = floor((t_3 - t_2)*freq_Hz);
n3 = floor((t_f - t_3)*freq_Hz);
v_time = [linspace(t_0,t_1,n0),linspace(t_1,t_2,n1),linspace(t_2,t_3,n2),
linspace(t_3,t_f,n3)]';
v_time = unique(v_time);
n_times = length(v_time);

%Costruzione per punti delle curve di comando
v_t_delta_T = [v_time,ones(n_times,1)*delta_T_0];
v_t_delta_s = [v_time,ones(n_times,1)*convang(delta_s_0_deg,'deg','rad')];
v_t_delta_e = [v_time,ones(n_times,1)*convang(delta_e_0_deg,'deg','rad')];
v_t_delta_e_dot = [v_time,ones(n_times,1)*convang(dot_delta_e_0,'deg','rad')];

%CABRAPICCHIA

```

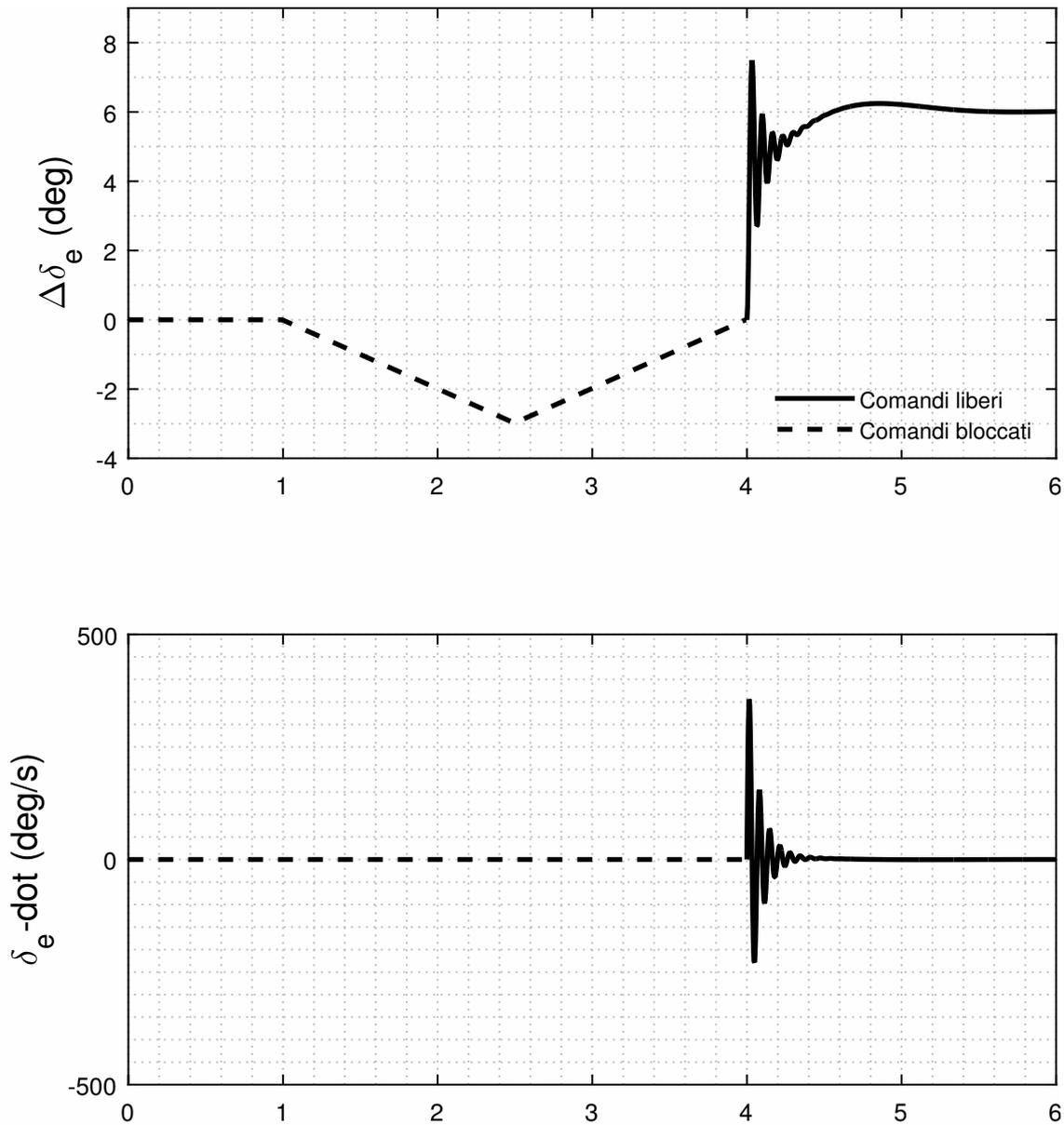


Figura 11.8 Legge temporale impartita alla deflessione dell'equilibratore, aletta tab = 0.

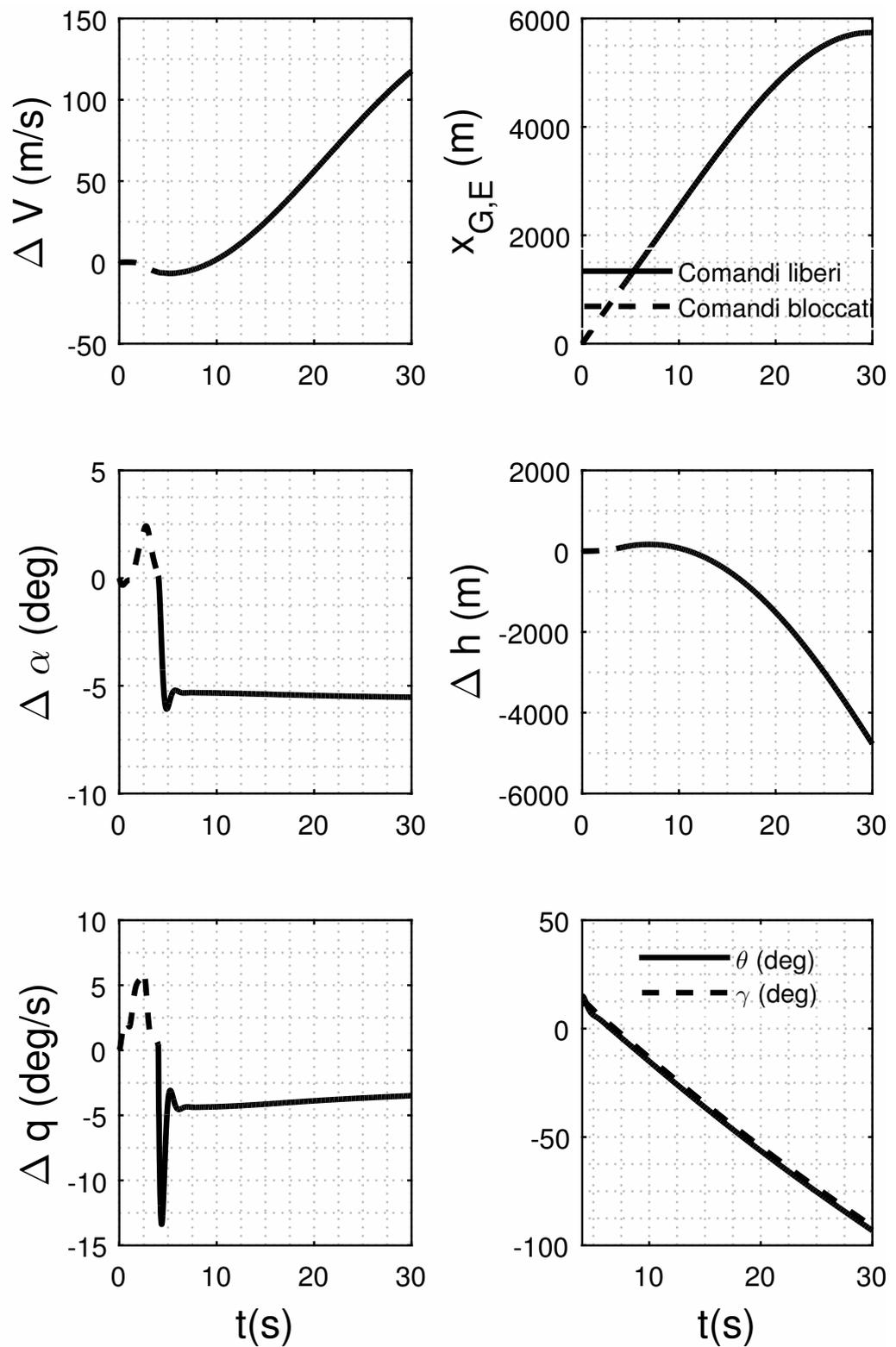


Figura 11.9 Storie temporali delle variabili di stato per comandi bloccati fino all'istante $t_1 = 1$ s e comandi liberi nei successivi istanti di tempo. Le grandezze $\Delta()$ rappresentano gli scostamenti rispetto ai valori iniziali, in particolare per il valore di δ_t che mi restituisce un valore asintotico del $\delta_e \approx \delta_{e,0}$.

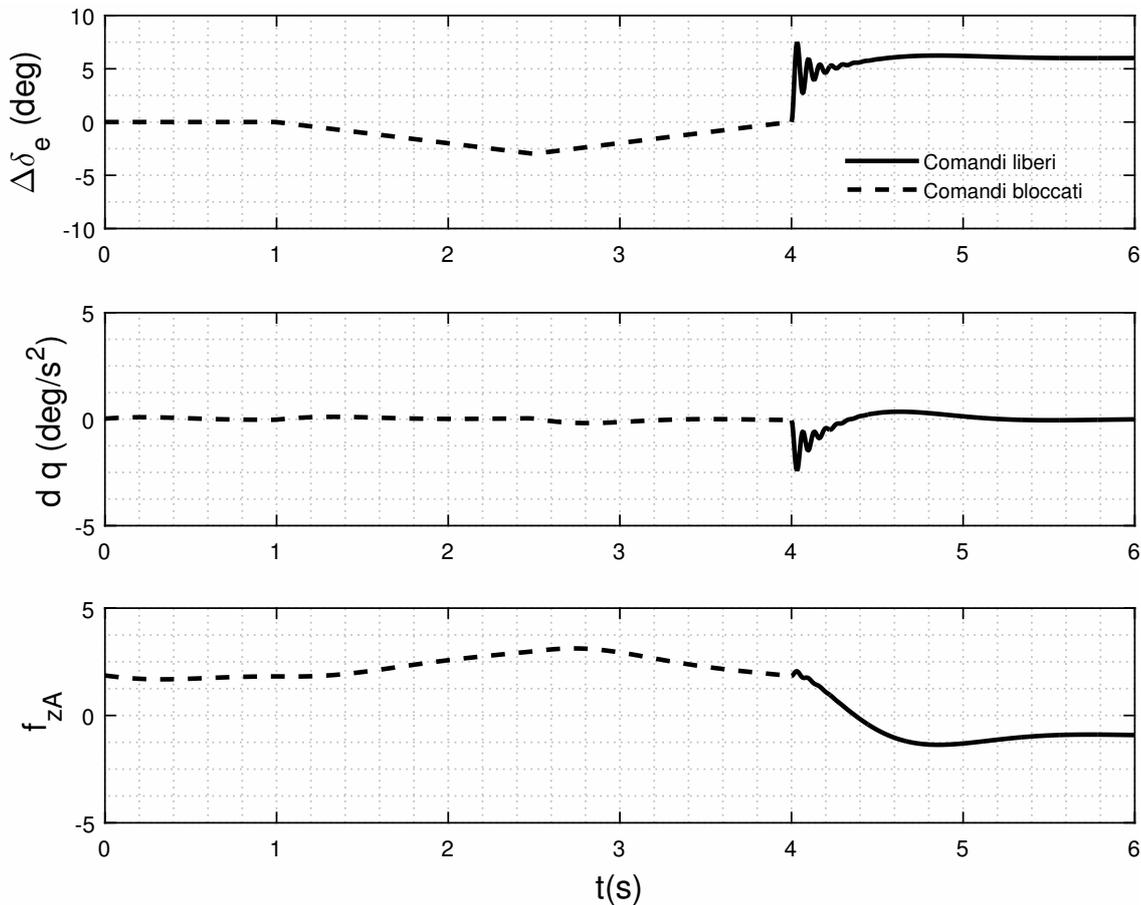


Figura 11.10 Grandezze d'interesse per l'assegnata legge cabra bicchia

```

delta_e_increment_deg = -3.0;
grad_delta_e=convang(delta_e_increment_deg,'deg','rad')/(t_2cp - t_1);

for ke=1:length(v_time)
    if ( (v_t_delta_e(ke,1)>=t_1) && (v_t_delta_e(ke,1)<t_2cp))
        v_t_delta_e(ke,2) = ...
            v_t_delta_e(ke-1,2) ...
            + (grad_delta_e/freq_Hz);
    elseif ( (v_t_delta_e(ke,1)>=t_2cp) && (v_t_delta_e(ke,1)<t_3cp))
        v_t_delta_e(ke,2) = ...
            v_t_delta_e(ke-1,2) ...
            - (grad_delta_e/freq_Hz);
    end
end

%Condizioni Iniziali
x0 = [ ...
    V_0;    ...
    alpha_0; ...
    q_0;    ...
    0.0;    ... % x_G_E_0
    z_0;    ...
    theta_0 ...
    ];

%Risoluzione Equazione differenziale, aggiungendo queste opzioni si
%nota l'attenuarsi delle oscillazioni.
options = odeset('RelTol', 1e-12,'AbsTol', 1e-12);

[T1, Y1] = ode45(@eqLongDynamicStickFixed,v_time, x0');

```

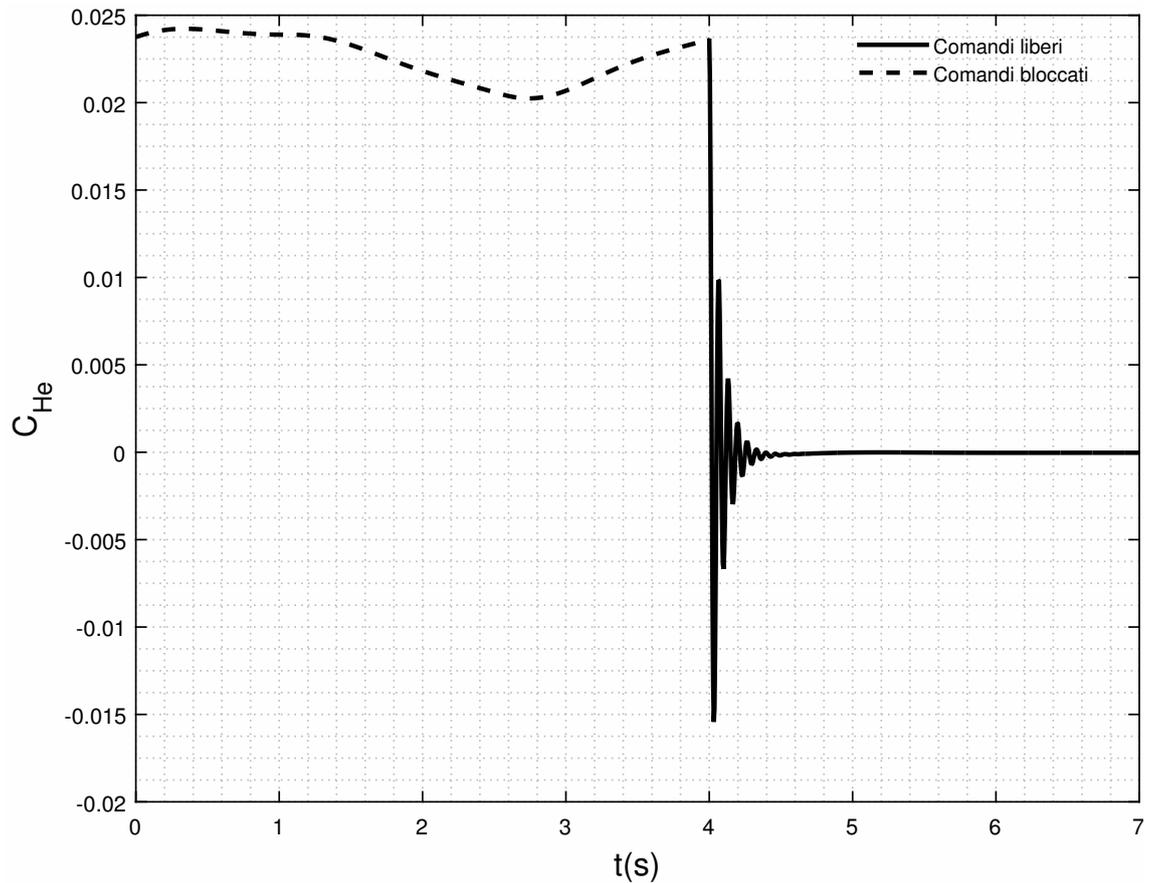


Figura 11.11 Grandezze d'interesse per l'assegnata legge cabra bicchia, variazione del momento di Cerniera

```

%derived vars
v_gamma1 = Y1(:,6) + myAC.mu_x - Y1(:,2);
v_dgamma1 = gradient(v_gamma1,T1);
v_dV1     = gradient(Y1(:,1),T1);
v_fxa1    = -sin(v_gamma1)-v_dV1./g;
v_fza1    = cos(v_gamma1)+(v_dgamma1.*Y1(:,1))./g;
v_dq1     = gradient(Y1(:,3),T1);
delta_s1  = interp1(v_t_delta_s(:,1),v_t_delta_s(:,2),T1);
delta_T1  = interp1(v_t_delta_T(:,1),v_t_delta_T(:,2),T1);
delta_e1  = interp1(v_t_delta_e(:,1),v_t_delta_e(:,2),T1);
delta_e_tab1 = interp1(v_t_delta_e_dot(:,1),v_t_delta_e_dot(:,2),T1);

%COMANDI LIBERI

t_f = 30.0;           % total observation time           (s)

freq_Hz = 220.;      % frequency of state propagation, 1/delta t

n0      = floor((t_f - t_3cp)*freq_Hz);

v_time  = [linspace(t_3cp,t_f,n0)...
           ]';

v_time  = unique(v_time);
n_times = length(v_time);

v_t_delta_T = [v_time,ones(n_times,1)*delta_T_0];
v_t_delta_s = [v_time,ones(n_times,1)*convang(delta_s_0_deg,'deg','rad')];
v_t_delta_e = [v_time,ones(n_times,1)*convang(delta_e_0_deg,'deg','rad')];
v_t_delta_tab = [v_time,ones(n_times,1)*convang(delta_tab_0_deg,'deg','rad')];

```

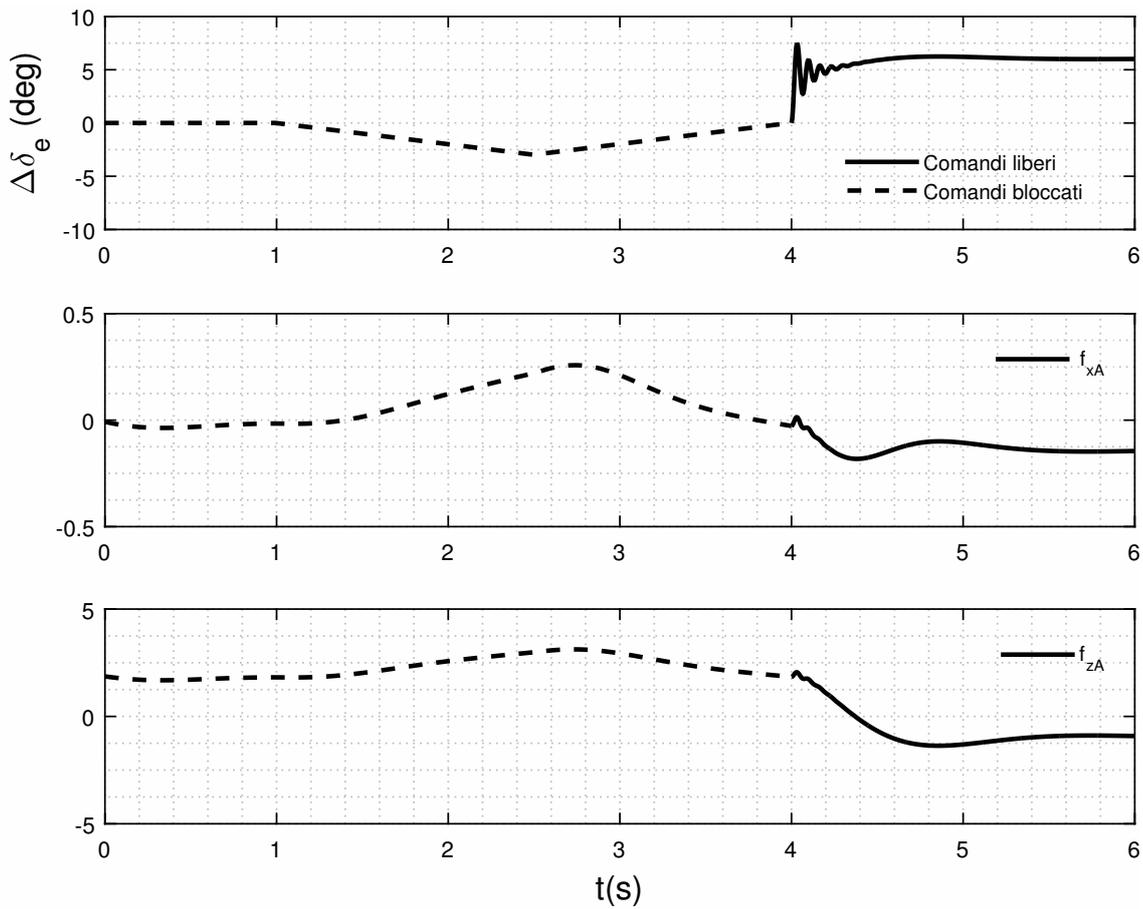


Figura 11.12 Andamento Carichi

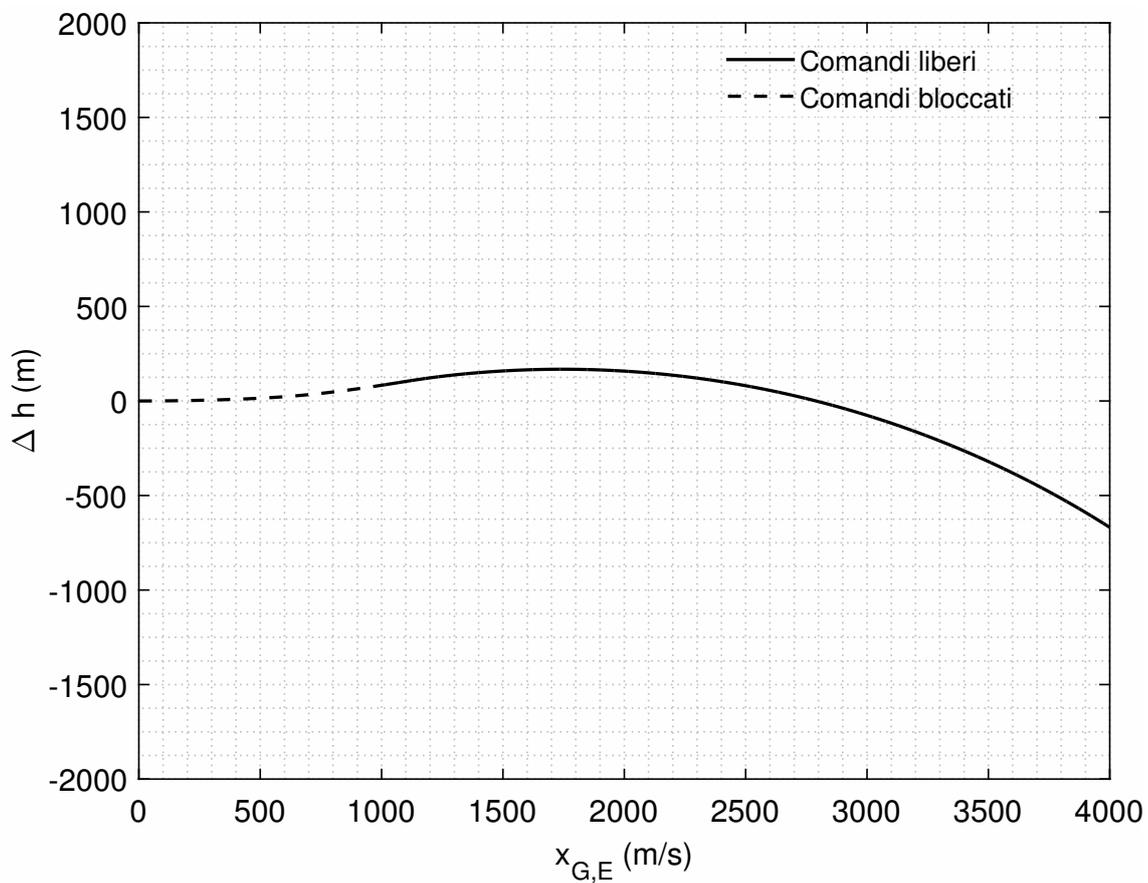


Figura 11.13 Variazione della quota a seguito della manovra cabra picchia

```

%Condizioni Iniziali
x0 = [ ...
      Y1(end,1);      ...
      Y1(end,2);      ...
      Y1(end,3);      ...
      Y1(end,4);      ... % x_G_E_0
      Y1(end,5);      ...
      Y1(end,6);      ...
      dot_delta_e_0;  ...
      delta_e_0 ...
      ];

options_ODE = odeset('AbsTol',1e-12,'Mass',@MassStickFree);

%Risoluzione
[T,Y] = ode45(@longEquilibriumDynamicStickFree,v_time,x0,options_ODE);

%derived vars
v_gamma = Y(:,6) + myAC.mu_x - Y(:,2);
v_dgamma = gradient(v_gamma,T);
v_dV      = gradient(Y(:,1),T);
v_fxa     = -sin(v_gamma)-v_dV./g;
v_fza     = cos(v_gamma)+(v_dgamma.*Y(:,1))./g;
v_dq      = gradient(Y(:,3),T);

%Verifica MOMENTO DI CERNIERA

delta_s = interp1(v_t_delta_s(:,1),v_t_delta_s(:,2),T);
delta_T = interp1(v_t_delta_T(:,1),v_t_delta_T(:,2),T);

Ch_e = myAC.Ch_e_0 + myAC.Ch_e_delta_e*Y(:,8) + myAC.Ch_e_delta_s*delta_s ...
      + myAC.Ch_e_delta_tab*delta_tab_0 +(myAC.mac_e/(2*Y(:,1)))*myAC.Ch_e_q*Y
      (:,3)...
      + myAC.Ch_e_alpha*((1-myAC.DepsDalpha)*(Y(:,2)-myAC.mu_x)-myAC.eps_0 ...
      + myAC.mu_x + delta_s);

Ch_e1 = myAC.Ch_e_0 + myAC.Ch_e_delta_e*delta_e_0 + myAC.Ch_e_delta_s*delta_s1
      ...
      + myAC.Ch_e_delta_tab*delta_tab_0 +(myAC.mac_e/(2*Y1(:,1)))*myAC.Ch_e_q*Y1
      (:,3)...
      + myAC.Ch_e_alpha*((1-myAC.DepsDalpha)*(Y1(:,2)-myAC.mu_x)-myAC.eps_0 ...
      + myAC.mu_x + delta_s1);

```

Analisi delle piccole perturbazioni del volo longitudinal simmetrico

Indice

16.1	Introduzione	183
16.2	Impostazione del problema	184
16.3	Dinamica Longitudinale	184
16.3.1	Caratteristiche longitudinali di un Boeing 747 in Matlab - <i>Eser-</i> <i>cizio 16.1</i>	185
16.3.2	Caratteristiche longitudinali di un Boeing 747 - <i>Esercizio 16.2</i> .	194

16.1 Introduzione

L'obiettivo della presente esercitazione è l'analisi della dinamica longitudinale del velivolo, ovvero dell'evoluzione delle piccole perturbazioni delle variabili simmetriche intorno a una condizione nominale di moto equilibrato, ad ali livellate e a velocità angolare nulla.

L'ipotesi di piccole perturbazioni ci consente di linearizzare il sistema delle equazioni del moto, costituito dall'insieme dell' *equazioni dinamiche* e dell' *equazioni cinematiche ausiliarie*, pervenendo a un sistema di equazioni lineari che può essere studiato con i metodi classici della teoria dei sistemi lineari tempo invarianti (LTI). Il sistema lineare, inoltre, tramite le ipotesi di *Braian*, può essere studiato separando la dinamica longitudinale da quella latero-direzionale.

In questo contesto, ci si sofferma soltanto sulla dinamica longitudinale. Si trascurano, inoltre, la determinazione, onerosa dal punto di vista dei passaggi e della trattazione, del sistema delle equazioni linearizzate del moto longitudinal simmetrico. Piuttosto, si dedica spazio e attenzione alla risoluzione numerica del suddetto sistema e alla determinazione

e rappresentazione dei modi di risposta del velivolo (di corto periodo e di fugoide), e più in generale allo studio della risposta libera longitudinale.

In particolare, in una prima parte si analizza la dinamica longitudinale del velivolo Boeing 747, assegnato dal docente, dopodiché si ripete la stessa analisi per il velivolo Cessna Citation già considerato nelle precedenti esercitazioni sull'analisi aerodinamica con Digital DATCOM+ e sui moti longitudinali a comandi liberi e a comandi bloccati. Si conclude con un confronto della risposta lineare e 3-DoF e la risposta del modello linearizzato.

16.2 Impostazione del problema

Il sistema di equazioni del volo longitudinal-simmetrico a comandi bloccati in forma matriciale assume la seguente forma:

$$\begin{Bmatrix} \dot{x}_{LON} \\ \dot{x}_{LD} \end{Bmatrix} = \begin{bmatrix} A_{LON} & 0 \\ 0 & A_{LD} \end{bmatrix} \begin{Bmatrix} x_{LON} \\ x_{LD} \end{Bmatrix} + \begin{bmatrix} B_{LON} & 0 \\ 0 & B_{LD} \end{bmatrix} \begin{Bmatrix} u_{LON} \\ u_{LD} \end{Bmatrix} \quad (16.1)$$

La A definisce la *matrice del sistema*, mentre la B definisce la *matrice degli ingressi*. Le due dinamiche potranno quindi essere analizzate con gli strumenti della Teoria dei sistemi dinamici Lineari Tempo Invarianti (LTI). Questo implica che la stabilità del sistema dinamico velivolo può essere studiata a partire dalla *risposta libera* del sistema con una condizione iniziale assegnata.

16.3 Dinamica Longitudinale

$$\Delta_{LON}(s) = (s - \lambda_{RANGE}) (s - \lambda_{HEIGHT}) (s - \lambda_{PH}) (s - \lambda_{PH}^*) (s - \lambda_{SP}) (s - \lambda_{SP}^*) \quad (16.2)$$

Il polinomio caratteristico (derivante da $\det[sI - A_{LON}] = 0$) sarà caratterizzato da due radici reali e distinte λ_{RANGE} , λ_{HEIGHT} associate alle equazioni linearizzate della perturbazione $x_{E,G}$ e $z_{E,G}$.

Tipicamente queste due radici sono trascurabili e quindi i corrispettivi modi non verranno presi in considerazione. Mentre le altre due radici, λ_{PH} e λ_{SP} , rappresentano due modi oscillatori: in particolare caratterizzano un modo di *lungo periodo* (o di *fugoide*) e uno di *corto periodo*.

Nel caso di modo di corto periodo si otterrà uno smorzamento e una frequenza naturale molto più elevati di quelli che si ottengono nel lungo periodo.

Alla luce di quanto detto, il vettore di stato relativo alle variabili longitudinali è tipicamente caratterizzato dall'interazione dei disturbi di velocità, angolo d'attacco e angolo di beccheggio.

Quanto esposto verrà chiarificato nei successivi paragrafi andando a sviluppare gli esercizi proposti dal docente.

Tabella 16.1 Caratteristiche inerziali e derivate aerodinamiche longitudinali del Boeing 747

Boeing 747 - Condizione volo 2 (Rif.Tab.16.3)	
Quota, h	SL
Massa, m = W/g	564000 lbf
Mach	0,25
Superficie alare, S	5500 ft^2
Corda Aerodinamica media, \bar{c}	27,3 ft
C_L	1,108 rad $C_{L\alpha}$ 5,70 rad^{-1}
$C_{L\dot{\alpha}}$	6,70 rad^{-1}
$C_{D\alpha}$	0,66 rad^{-1}
$C_{L\delta_e}$	0,617 rad^{-1}
C_{Lq}	5,4 rad^{-1}
C_{m_q}	-20,8 rad^{-1}
$C_{m\dot{\alpha}}$	-3,2 rad^{-1}
$C_{m\alpha}$	-1,26 rad^{-1}
$C_{m\delta_e}$	-1.34 rad^{-1}

16.3.1 Caratteristiche longitudinali di un Boeing 747 in Matlab - Esercizio 16.1

Mediante uno script proposto dal docente sono stati calcolati gli autovalori e gli autovettori caratterizzanti la dinamica longitudinale di un Boeing 747. Nella Tabella 16.1 sono riportate le caratteristiche inerziali e le derivate aerodinamiche longitudinali di un Boeing 747 per la specifica condizione di volo 2 in *powered approach* (caso caratterizzato da bassi numeri di Mach ed elevati coefficienti di portanza).

Listato 16.1 Codice per la ricerca delle caratteristiche longitudinali di un Boeing 747

```

clc; clear all; close all;

%% Condizione volo 2 (tabella 16.3 Pag.64)
% Definizione delle grandezze da inserire nella matrice del sistema STI

Weight = convforce(564032, 'lbf', 'N');
g_0 = 9.81; % m/s^2
mass = Weight/g_0;

SLUGFT2toKGM2 = convmass(1, 'slug', 'kg')*(convlengh(1, 'ft', 'm')^2);
Iyy = 32.3e6 * SLUGFT2toKGM2; % Kg*m^2

h_0 = 0; % m
[T_0, a_0, p_0, rho_0] = atmosisa(h_0);

Mach_0 = 0.25;

% U_0 = convvel(279.1, 'ft/s', 'm/s'); % m/s
U_0 = Mach_0 * a_0;

S = 5500*(convlengh(1, 'ft', 'm')^2); % m^2
cbar = convlengh(27.3, 'ft', 'm'); % m

qbar_0 = 0.5*rho_0*(U_0^2);
mu_0 = mass/(0.5*rho_0*cbar*S);

Gamma_0 = 0.0; % rad

```

```

C_L      = 1.108;
C_D      = 0.102;
C_L_Mach = 0.0;
C_L_alpha = 5.70;    % 1/rad
C_L_alphadot = 6.7;    % 1/rad
C_L_q     = 5.4;    % 1/rad
C_D_alpha = 0.66;    % 1/rad
C_D_Mach  = 0.0;
C_D_alphadot = 0.0;    % 1/rad
C_D_q     = 0.0;    % 1/rad

SM_0 = 0.22;
C_m_alpha_0 = -1.26;    % 1/rad
SM = 0.22;    % Try also: 0.0158  0.0021  0.0  -0.0145
C_m_alpha = C_m_alpha_0*(SM/SM_0);    % 1/rad

C_m_alphadot = -3.2;    % 1/rad
C_m_q        = -20.8;    % 1/rad
C_m_Mach     = 0.0;
C_L_de       = 0.338;    % 1/rad
C_m_de       = -1.34;    % 1/rad

X_u = -(qbar_0*S/(mass*U_0))*(2*C_D + Mach_0*C_D_Mach);    % constant thrust
X_w = (qbar_0*S/(mass*U_0))*(C_L - C_D_alpha);
X_wdot = 0;
X_q = 0;
X_de = 0;
X_dT = 0;

Z_u = -(qbar_0*S/(mass*U_0))*(...
    2*C_L + (Mach_0^2/(1-Mach_0^2))*C_L_Mach);
Z_w = -(qbar_0*S/(mass*U_0))*(C_D + C_L_alpha);
Z_wdot = -(1/(2*mu_0))*C_L_alphadot;
Z_q = -(U_0/(2*mu_0))*C_L_q;
Z_de = 0;
Z_dT = 0;

M_u = (qbar_0*S*cbar/(Iyy*U_0))*Mach_0*C_m_Mach;
M_w = (qbar_0*S*cbar/(Iyy*U_0))*C_m_alpha;
M_wdot = (rho_0*S*(cbar^2)/(Iyy*4))*C_m_alphadot;
M_q = (rho_0*U_0*S*(cbar^2)/(Iyy*4))*C_m_q;
M_de = 0;
M_dT = 0;

k_hat = M_wdot/(1-Z_wdot);

%% Costruzione delle Matrici
% Plan matrix 16.147b
A_lon(1,1) = X_u;
A_lon(1,2) = X_w;
A_lon(1,3) = 0;
A_lon(1,4) = -g_0*cos(Gamma_0);
A_lon(2,1) = Z_u/(1-Z_wdot);
A_lon(2,2) = Z_w/(1-Z_wdot);
A_lon(2,3) = (Z_q + U_0)/(1-Z_wdot);
A_lon(2,4) = -g_0*sin(Gamma_0)/(1-Z_wdot);
A_lon(3,1) = M_u + k_hat*Z_u;
A_lon(3,2) = M_w + k_hat*Z_w;
A_lon(3,3) = M_q + k_hat*(Z_q+U_0);
A_lon(3,4) = -k_hat*g_0*sin(Gamma_0);
A_lon(4,1) = 0;
A_lon(4,2) = 0;
A_lon(4,3) = 1;

```

```

A_lon(4,4) = 0;

% Input matrix 16.147c
B_lon = [ ...
    X_de/mass,          X_dT/mass; ...
    Z_de/(mass-Z_wdot), Z_dT/(mass-Z_wdot); ...
    (M_de + Z_de*M_wdot)/(mass-Z_wdot)/Iyy, ...
    (M_dT + Z_dT*M_wdot)/(mass-Z_wdot)/Iyy; ...
    0,                  0];

% Output matrices
C_lon = eye(3,4);
D_lon = zeros(3,2);

%% Symbolic determinant ---> characteristic equation
syms lambda

characteristic_sym_polynomial_lon = det(A_lon-lambda*eye(4));
char_poly_lon = sym2poly(characteristic_sym_polynomial_lon);

a1 = char_poly_lon(1);
a2 = char_poly_lon(2); a2 = a2/a1;
a3 = char_poly_lon(3); a3 = a3/a1;
a4 = char_poly_lon(4); a4 = a4/a1;
a5 = char_poly_lon(5); a5 = a5/a1; a1 = 1;

%% Eigenvector, eigenvalues
[V,D] = eig(A_lon);

W = inv(V);

V_SP = V(:,1);
V_SP = V_SP/V_SP(4);
V_SP(1,1) = V_SP(1,1)/U_0;
V_SP(2,1) = V_SP(2,1)/U_0;
V_SP(3,1) = V_SP(3,1)/(2*U_0/cbar);

V_Ph = V(:,3);
V_Ph = V_Ph/V_Ph(4);
V_Ph(1,1) = V_Ph(1,1)/U_0;
V_Ph(2,1) = V_Ph(2,1)/U_0;
V_Ph(3,1) = V_Ph(3,1)/(2*U_0/cbar);

%% State variables transformation
syms a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 a33 a34 a41 a42 a43 a44

AA = [ ...
    a11, a12, a13, a14; ...
    a21, a22, a23, a24; ...
    a31, a32, a33, a34; ...
    a41, a42, a43, a44];

% transform plant matrix 16.208 - 16.210
syms U0 Cbar
TT = diag([1/U0,1/U0,Cbar/(2*U0),1]);

AA1 = TT*AA*inv(TT);

% evaluate AA1 by substituting numbers to symbols
A_lon1 = ...
double(...
    subs(...
        AA1, ...
        [a11, a12, a13, a14, ...
         a21, a22, a23, a24, ...
         a31, a32, a33, a34, ...

```

```

        a41, a42, a43, a44 ...
    U0, Cbar], ...
    [A_lon(1,1), A_lon(1,2), A_lon(1,3), A_lon(1,4), ...
    A_lon(2,1), A_lon(2,2), A_lon(2,3), A_lon(2,4), ...
    A_lon(3,1), A_lon(3,2), A_lon(3,3), A_lon(3,4), ...
    A_lon(4,1), A_lon(4,2), A_lon(4,3), A_lon(4,4), ...
    U_0, cbar] ...
    ) ...
);

% right eigenvectors (columns) and eigenvalues
[V1,D1] = eig(A_lon1);

% left eigenvectors (rows)
W1 = inv(V1);

% make a state-space representation
sys = ss(A_lon1, ...           %A
        zeros(size(A_lon1,1),1), ... %B
        eye(4,4), ...         %C
        zeros(4,1) ...        %D
        );

% Short-Period/Phugoid, normalized left eigenvector
V1SP = V1(:,1);
V1SP = V1SP/V1SP(4,1);
V1Ph = V1(:,3);
V1Ph = V1Ph/V1Ph(4,1);

% time response taking the initial state coincident
% with the original eigenvectors

x_init_1 = real(V1SP);
[y1,t1,x1] = initial(sys,x_init_1);

x_init_3 = real(V1Ph);
[y3,t3,x3] = initial(sys,x_init_3);

%% Costruzione delle MATRICI APPROSSIMATE
% Plan matrix 16.215
A_lon_AP(1,1) = Z_w;
A_lon_AP(1,2) = Z_q ;
A_lon_AP(2,1) = M_w + M_wdot*Z_w;
A_lon_AP(2,2) = M_q + M_wdot*U_0;

% Input matrix 16.147c
B_lon_AP = [ ...
            X_dT*Z_dT,                Z_de; ...
            (M_dT + Z_dT*M_wdot),      (M_de + Z_de*M_wdot)...
            ];

%% Eigenvector Plot

h1=figure(1)
subplot(211)
compass(V_SP)
axis('square')

```

```

grid minor
xlabel(num2str(abs(V_SP)))
subplot(212)
plot(t1,y1(:,1),'-','LineWidth',2)
hold on
plot(t1,y1(:,2),'--','LineWidth',2)
plot(t1,(cbar/(2*U_0))*y1(:,3),'-','LineWidth',2)
plot(t1,y1(:,4),'-','LineWidth',2)
legend('u/U_0','w/U_0','q c/(2 U_0)','theta','Location','northoutside','
    Orientation','horizontal');
%set(legend,'color','none');
ylabel('Perturbation States Short ');
xlabel('time (sec)');
grid minor
axis ([0 10 -1 1.0])
print(h1,'-deps','Immagini\CortoPeriodoBoeing747');
print(h1,'-deps','C:\Users\Giuseppe\Dropbox\DSV\__template_esercitazioni\
    quaderno16\images\CortoPeriodoBoeing747');
%prop={'PaperType','PaperSize','PaperPosition','PaperPositionMode','PaperUnits','
    Units'};
%valori={'<custom>',[18,26],[0,0,18,26],'manual','Inches','Inches'};
%set(h1,prop,valori);

h2=figure(2)
subplot(211)
compass(V_Ph)
axis('square')
xlabel(num2str(abs(V_Ph)))
subplot(212)
plot(t3,y3(:,1),'-','LineWidth',2)
hold on
plot(t3,y3(:,2),'--','LineWidth',2)
plot(t3,(cbar/(2*U_0))*y3(:,3),'-','LineWidth',2)
plot(t3,y3(:,4),'-','LineWidth',2)
legend('u/U_0','w/U_0 (deg)','q c/(2 U_0)','theta (deg)','Location','northoutside'
    , 'Orientation','horizontal');
%set(legend,'color','none');
ylabel('Perturbation States Phugoid');
xlabel('time (sec)');
grid minor
axis ([0 600 -1 1])
print(h2,'-deps','Immagini\LungoPeriodoBoeing747');
print(h2,'-deps','C:\Users\Giuseppe\Dropbox\DSV\__template_esercitazioni\
    quaderno16\images\LungoPeriodoBoeing747');
%prop={'PaperType','PaperSize','PaperPosition','PaperPositionMode','PaperUnits','
    Units'};
%valori={'<custom>',[18,26],[0,0,18,26],'manual','Inches','Inches'};
%set(h1,prop,valori);

%% Anonymous function
% Calcolo caratteristiche modali: coefficiente di smorzamento,
% pulsazione naturale, periodo T, tempo di dimezzamento( o raddoppio),
% numero di cicli fino al tempo di dimezzamento

zeta = @(sigma,omega) ...
    sqrt(1/(1+(omega/sigma)^2));
omega_n = @(sigma,omega) ...
    -sigma/zeta(sigma,omega);
Period = @(sigma,omega) ...
    2*pi/(omega_n(sigma,omega)*sqrt(1-(zeta(sigma,omega))^2));
t_half = @(sigma,omega) ...
    log(2)/(omega_n(sigma,omega)*zeta(sigma,omega));
N_half = @(sigma,omega) ...
    t_half(sigma,omega)/Period(sigma,omega);

```

```

% SHORT PERIOD
lambda_SP = D(1,1);
sigma_SP = real(lambda_SP);
omega_SP = imag(lambda_SP);

zeta_SP = zeta(sigma_SP,omega_SP);
omega_n_SP = omega_n(sigma_SP,omega_SP);
T_SP = Period(sigma_SP,omega_SP);
t_half_SP = t_half(sigma_SP,omega_SP);
N_half_SP = N_half(sigma_SP,omega_SP);

% SHORT PERIOD APPROX
omega_n_SP_AP = sqrt(Z_w*M_q - M_w*U_0);
zeta_SP_AP = -(Z_w + M_q + M_wdot*U_0)/(2*omega_n_SP_AP);
% SHORT PERIOD COARSE APPROX
omega_n_SP_CAP = sqrt(- M_w*U_0);
zeta_SP_CAP = - M_q/(2*omega_n_SP_CAP);

disp('');
disp('Caratteristiche Modali SHORT PERIOD:');
disp(['Periodo SP= ',num2str(T_SP)]);
disp(['Tempo dimezzamento SP= ',num2str(t_half_SP)]);
disp(['Numero cicli SP= ',num2str(N_half_SP)]);
disp(['Zeta SP= ',num2str(zeta_SP)]);
disp(['Frequenza naturale SP= ',num2str(omega_n_SP)]);
disp('.');
disp('SHORT PERIOD COARSE APPROXIMATION:');
disp(['Zeta SP= ',num2str(zeta_SP_AP)]);
disp(['Frequenza naturale SP= ',num2str(omega_n_SP_AP)]);
disp('.');
disp('SHORT PERIOD COARSE APPROXIMATION:');
disp(['Zeta SP= ',num2str(zeta_SP_CAP)]);
disp(['Frequenza naturale SP= ',num2str(omega_n_SP_CAP)]);

% PHUGOID
lambda_Ph = D(3,3);
sigma_Ph = real(lambda_Ph);
omega_Ph = imag(lambda_Ph);

zeta_Ph = zeta(sigma_Ph,omega_Ph);
omega_n_Ph = omega_n(sigma_Ph,omega_Ph);
T_Ph = Period(sigma_Ph,omega_Ph);
t_half_Ph = t_half(sigma_Ph,omega_Ph);
N_half_Ph = N_half(sigma_Ph,omega_Ph);

% PHUGOID APPROX
omega_n_Ph_AP = sqrt(g_0*((Z_u*M_w-M_u*Z_w)/(-M_w*U_0 + M_q*Z_w)));
zeta_Ph_AP = -(X_u + X_w*((M_u*U_0-M_q*Z_u)/(-M_w*U_0+M_q*Z_w)))/(2*omega_n_Ph_AP);
;
% PHUGOID COARSE APPROX
omega_n_Ph_CAP = sqrt(- Z_u*g_0/U_0);
zeta_Ph_CAP = - X_u/(2*omega_n_Ph_CAP);
disp('');
disp('');
disp('');
disp('Caratteristiche Modali PHUGOID:');
disp(['Periodo Ph= ',num2str(T_Ph)]);
disp(['Tempo dimezzamento Ph= ',num2str(t_half_Ph)]);
disp(['Numero cicli Ph= ',num2str(N_half_Ph)]);
disp(['Zeta Ph= ',num2str(zeta_Ph)]);
disp(['Frequenza naturale Ph= ',num2str(omega_n_Ph)]);
disp('.');
disp('PHUGOID COARSE APPROXIMATION:');
disp(['Zeta Ph= ',num2str(zeta_Ph_AP)]);
disp(['Frequenza naturale Ph= ',num2str(omega_n_Ph_AP)]);

```

```

disp(' ');
disp('PHUGOID COARSE APPROXIMATION:');
disp(['Zeta Ph= ', num2str(zeta_Ph_CAP)]);
disp(['Frequenza naturale Ph= ', num2str(omega_n_Ph_CAP)])

```

I dati contenuti nello script, necessari per riempire le matrici del sistema LTI, sono relativi alla tabella 16.3 fornita dal docente e si riferiscono al velivolo Boeing 747 e sono relativi alla condizione di volo 2, della tabella.

Nel codice vengono preliminarmente definite le grandezze da inserire nelle matrici del sistema LTI; si procede, dunque, alla costruzione delle stesse matrici caratteristiche del sistema, si calcolano poi gli autovalori e gli autovettori della matrice dinamica A_{LON} , sfruttando il tool simbolico di Matlab. L'ultima parte è dedicata alla rappresentazione dei risultati ottenuti ed inoltre si ripete il calcolo delle grandezze ζ_{SP} , $\omega_{n,SP}$ e ζ_{PH} , $\omega_{n,PH}$ utilizzando i modelli approssimati presenti nel paragrafo precedente.

Si osservi che per la rappresentazione degli autovettori del modo di corto periodo e di fugoide come fasori nel piano complesso si è ricorso ad una opportuna normalizzazione. Precisamente, ogni autovettore destro viene normalizzato dividendo per la sua quarta componente - perturbazione dell'angolo di Eulero θ - in modo che la quarta componente sia paria a 1 (fasore allineato con l'asse reale nel piano complesso), dopodichè si rendono adimensionali le prime tre componenti (perturbazioni u , w e q).

In questo modo i fasori delle prime tre componenti dell'autovettore normalizzato si possono interpretare come risposte periodiche sfasate in anticipo o in ritardo rispetto alla risposta oscillatoria $\Theta(t)$.

Nel codice questa normalizzazione è ottenuta introducendo la matrice \mathbf{T}_{LON} definita come segue:

$$\begin{bmatrix} \frac{1}{U_0} & 0 & 0 & 0 \\ 0 & \frac{1}{U_0} & 0 & 0 \\ 0 & 0 & \frac{\bar{c}}{2U_0} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16.3)$$

e le trasformazioni:

$$\hat{\mathbf{x}}_{LON} = \mathbf{T}_{LON} \mathbf{x}_{LON} \quad (16.4)$$

$$\hat{\mathbf{A}}_{LON} = \mathbf{T}_{LON}^{-1} \mathbf{A}_{LON} \mathbf{T}_{LON} \quad (16.5)$$

La matrice dinamica nelle unità del SI è data da:

$$\mathbf{A}_{LON} = \begin{bmatrix} -0,0212 & 0,0466 & 0 & -9,8100 \\ -0,2231 & -0,5841 & 80,0055 & 0 \\ -0,002 & -0,0059 & -0,5011 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (16.6)$$

Essa ha il seguente polinomio caratteristico:

$$\Delta_{LON}(s) = s^4 + 1,1065s^3 + 0,7992s^2 + 0,0225s + 0,0140 \quad (16.7)$$

le cui radici sono:

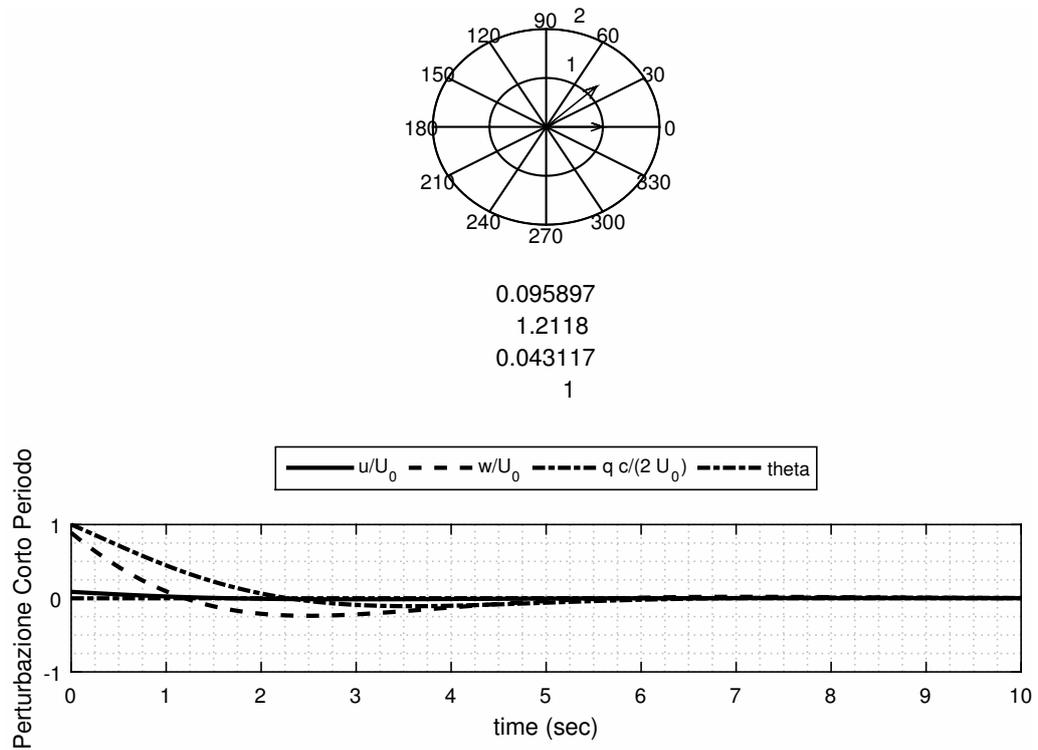


Figura 16.1 Risposta libera di un Boeing 747 a perturbazioni longitudinali della condizione iniziale. La risposta è stata ottenuta eccitando *solo* il modo di corto periodo.

$$\lambda_{1,2} = \sigma_{SP} \pm j\omega_{SP} = -0,5515s^{-1} \pm j0,6879s^{-1} \quad (16.8)$$

$$\lambda_{3,4} = \sigma_{PH} \pm j\omega_{PH} = -0,0018s^{-1} \pm j0,1340s^{-1} \quad (16.9)$$

Le componenti degli autovettori destri normalizzati hanno i seguenti valori numerici:

$$x_{SP} = \begin{pmatrix} 0,0870 + j0,0404 \\ 0,8882 + j0,8243 \\ -0,0270 + j0,0336 \\ 1 + j0 \end{pmatrix} \quad (16.10)$$

$$x_{PH} = \begin{pmatrix} -0,1576 + j0,8265 \\ 0,0327 - j0,1078 \\ 0,0002 + j0,0066 \\ 1 + j0 \end{pmatrix} \quad (16.11)$$

La rappresentazione in forma di fasori di questi autovettori e del modo di risposta del velivolo è riportata nelle figure 16.1 e 16.2 a fronte e rappresenta i modi di risposta del velivolo ottenute eccitando solo il modo di corto periodo e solo il modo di fugoide. Tali tipi di risposte sono la frutto della soluzione del sistema che abbiamo impostato 16.1 a pagina 184 e valide nelle ipotesi di perturbazioni a parametri di controllo nulle, assegnando delle condizioni iniziali coincidenti rispettivamente con l'autovettore (sinistro) del corto periodo ξ_1^T corrispondente all'autovalore λ_1 e l'autovettore di fugoide ξ_3^T corrispondente all'autovalore λ_3 .

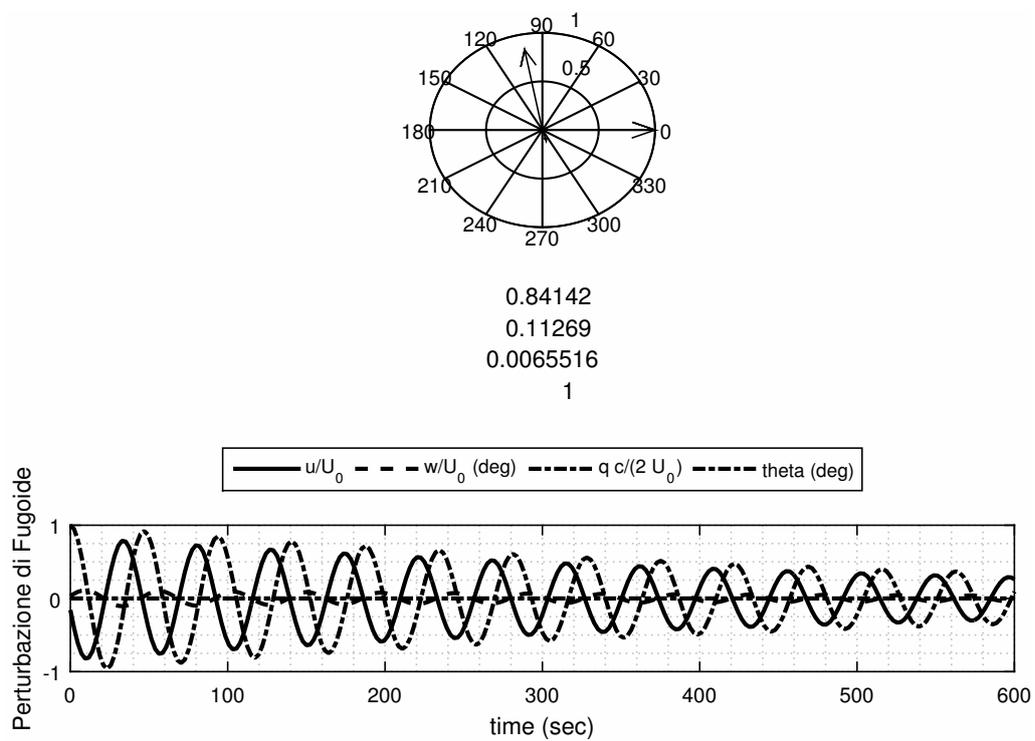


Figura 16.2 Risposta libera di un Boeing 747 a perturbazioni longitudinali della condizione iniziale. La risposta è stata ottenuta eccitando *solo* il modo di lungo periodo.

16.3.2 Caratteristiche longitudinali di un Boeing 747 - Esercizio 16.2

L'esercizio precedente è stato riproposto questa volta andando a supporre che il velivolo fosse in altre condizioni di volo. A tal fine si è provveduto a riscrivere le matrici andandole a riempire con nuovi coefficienti che, per le nuove condizioni di volo, risulteranno variati. I coefficienti aerodinamici sono stati selezionati dalla Tabella 16.3 proposta dal docente e si riferiscono alla condizione di volo 5.

La matrice dinamica nelle unità del SI è data da:

$$A_{\text{LON}} = \begin{bmatrix} -0,00776 & 0,0283 & 0 & -9,8100 \\ -0,1195 & -0,4233 & 153,6564 & 0 \\ 0,0003 & -0,0054 & -0,4870 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (16.12)$$

Essa ha il seguente polinomio caratteristico:

$$\Delta_{\text{LON}}(s) = S^4 + 0,9180s^3 + 1,0518s^2 + 0,0116s + 0,0078 \quad (16.13)$$

le cui radici sono:

$$\lambda_{1,2} = \sigma_{\text{SP}} \pm j\omega_{\text{SP}} = -0,4567s^{-1} \pm j0,9119s^{-1} \quad (16.14)$$

$$\lambda_{3,4} = \sigma_{\text{PH}} \pm j\omega_{\text{PH}} = -0,0023s^{-1} \pm j0,0866s^{-1} \quad (16.15)$$

Le componenti degli autovettori destri normalizzati hanno i seguenti valori numerici:

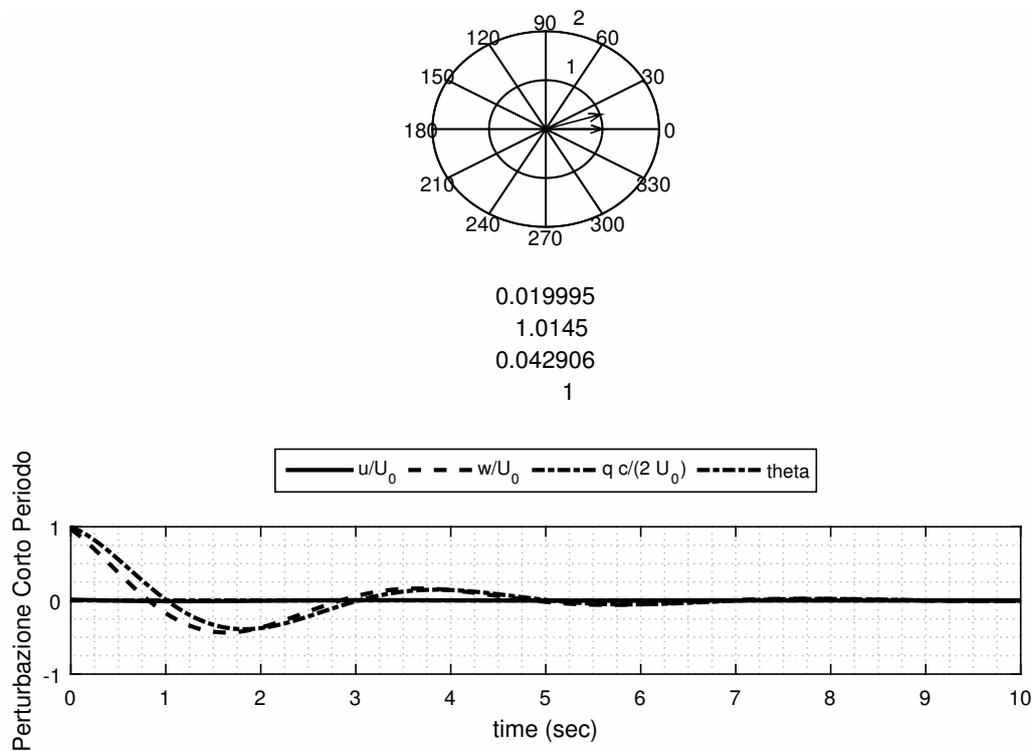
$$x_{\text{SP}} = \begin{pmatrix} 0,0262 + j0,0246 \\ 0,9858 + j0,4543 \\ -0,0120 + j0,0240 \\ 1 + j0 \end{pmatrix} \quad (16.16)$$

$$x_{\text{PH}} = \begin{pmatrix} -0,0461 + j0,7117 \\ 0,0071 - j0,0035 \\ -0,0000602 + j0,0023 \\ 1 + j0 \end{pmatrix} \quad (16.17)$$

La rappresentazione in forma di fasori di questi autovettori e del modo di risposta del velivolo è riportata nelle figure 16.3 a fronte e 16.4 a pagina 196 e rappresenta i modi di risposta del velivolo ottenute eccitando solo il modo di corto periodo e solo il modo di fugoide. Tali tipi di risposte sono la frutto della soluzione del sistema che abbiamo impostato 16.1 a pagina 184 e valide nelle ipotesi di perturbazioni a parametri di controllo nulle, assegnando delle condizioni iniziali coincidenti rispettivamente con l'autovettore (sinistro) del corto periodo ξ_1^T corrispondente all'autovalore λ_1 e l'autovettore di fugoide ξ_3^T corrispondente all'autovalore λ_3 .

Tabella 16.2 Caratteristiche inerziali e derivate aerodinamiche longitudinali del Boeing 747

Boeing 747 - Condizione volo 5 (Rif.Tab.16.3)	
Quota, h	20000 ft
Massa, $m = W/g$	564032 lbf
Mach	0,50
Superficie alare, S	5500 ft^2
Corda Aerodinamica media, \bar{c}	27,3 ft
C_L	0,68 rad
$C_{L\alpha}$	4,67 rad^{-1}
$C_{L\dot{\alpha}}$	6,53 rad^{-1}
$C_{D\alpha}$	0,04 rad^{-1}
$C_{L\delta_e}$	0,356 rad^{-1}
C_{Lq}	5,13 rad^{-1}
C_{mq}	-20,70 rad^{-1}
$C_{m\dot{\alpha}}$	-3,20 rad^{-1}
$C_{m\alpha}$	-1,15 rad^{-1}
$C_{m\delta_e}$	-0.356 rad^{-1}

**Figura 16.3** Risposta libera di un Boeing 747 a perturbazioni longitudinali della condizione iniziale. La risposta è stata ottenuta eccitando *solo* il modo di corto periodo.

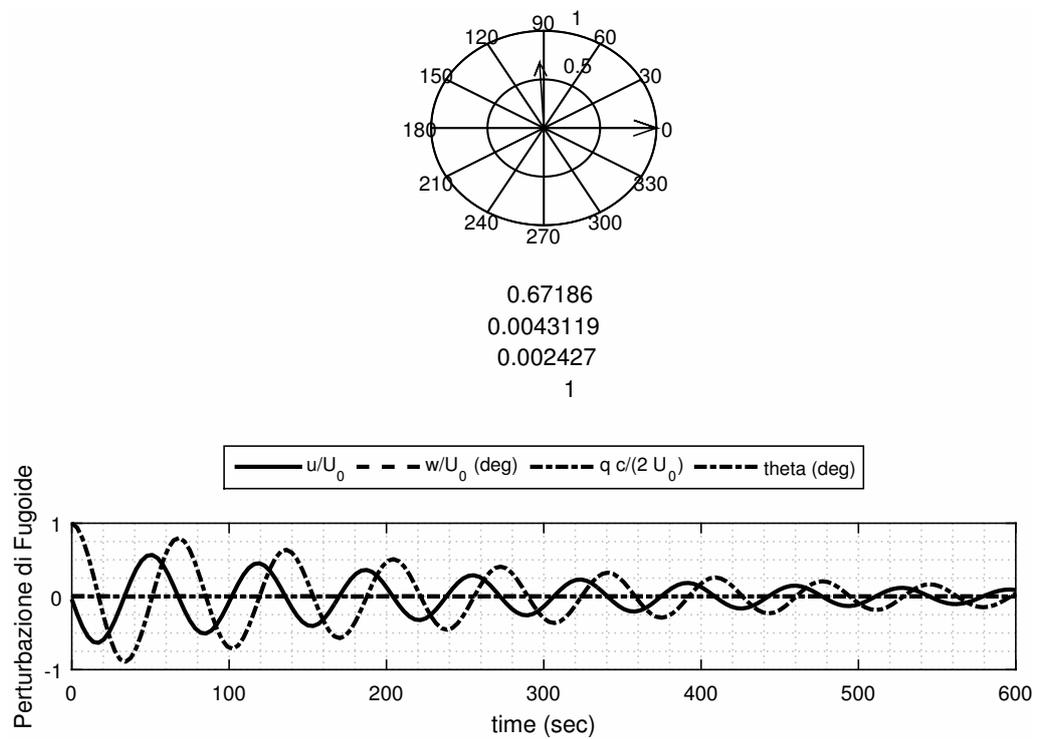


Figura 16.4 Risposta libera di un Boeing 747 a perturbazioni longitudinali della condizione iniziale. La risposta è stata ottenuta eccitando *solo* il modo di lungo periodo.

Evoluzione in piano orizzontale

Indice

17.1 Introduzione	197
17.2 Virata corretta	198
17.2.1 Virata corretta per assegnata legge $C_L(t)$ - Esercizi da 17.1 a	
17.3	198

17.1 Introduzione

In questo capitolo affronteremo una manovra che consente al velivolo di cambiare rotta, la *virata*. In generale la virata può essere effettuata lungo traiettorie in salita o discesa; noi ci concentreremo sul caso particolare di virata quota costante, in cui l'angolo di rampa $\gamma = 0$.

Dei 6 DoF che costituirebbero il problema nella condizione più generica, per via dell'ipotesi di traiettoria orizzontale, andremo ad eliminarne 1. Il problema passerà dalle precedenti 6 equazioni differenziali a 5 equazioni differenziali, 3 alla rotazione e 2 alla traslazione, e una algebrica per la quota. Per studiare tale tipo di problema si ipotizza che le equazioni alla rotazione siano automaticamente soddisfatte. Tale scelta si rende necessaria, in quanto si vuole garantire che $\gamma = 0$. Per garantire ciò il pilota deve usare tutti e 3 i comandi δ_a , δ_e , δ_r , δ_s e δ_T per far sì che l'ipotesi venga rispettata. A tal punto rimangono 3 equazioni, 2 differenziali (che contemplano le accelerazioni lungo x_w e lungo y_w) e 1 algebrica (che si riferisce alla quota fissata).

Queste 3 equazioni sono quelle che reggono il moto della virata generale semplificata. Bisogna essere in grado di risolverlo ricordando che le nostre incognite sono 5 (V , α_B , β , δ , ν) su 3 equazioni. Le due incognite in eccesso vengono determinate andando ad imporre o la legge della sopportazione (assegnando il fattore di carico lungo z), oppure quella della sustentazione (assegnando il C_L).

$$\frac{\dot{V}}{g} = \frac{T}{W} \cos(\alpha_B + \mu_T) \cos \beta - \frac{\bar{q}}{W/S} (C_D \cos \beta - C_Y \sin \beta) \quad (17.1a)$$

$$\begin{aligned} \frac{V \dot{\delta}}{g} = & -\frac{T}{W} \left[\cos(\alpha_B + \mu_T) \sin \beta \cos \nu - \sin(\alpha_B + \mu_T) \sin \nu \right] \\ & + \frac{\bar{q}}{W/S} \left[(C_D \sin \beta + C_Y \cos \beta) \cos \nu + C_L \sin \nu \right] \end{aligned} \quad (17.1b)$$

$$\begin{aligned} 1 = & \frac{T}{W} \left[\cos(\alpha_B + \mu_T) \sin \beta \sin \nu + \sin(\alpha_B + \mu_T) \cos \nu \right] \\ & - \frac{\bar{q}}{W/S} \left[(C_D \sin \beta + C_Y \cos \beta) \sin \nu - C_L \cos \nu \right] \end{aligned} \quad (17.1c)$$

17.2 Virata corretta

Mantenendo la deriva, β costantemente nulla durante la virata, si ottengono valori della C_Y praticamente trascurabili, per cui il sistema 17.1 si riduce al seguente:

$$\frac{\dot{V}}{g} = \frac{T}{W} \cos(\alpha_B + \mu_T) - \frac{\bar{q}}{W/S} C_D \quad (17.2a)$$

$$\frac{V \dot{\delta}}{g} = \left[\frac{T}{W} \sin(\alpha_B + \mu_T) + \frac{\bar{q}}{W/S} C_L \right] \sin \nu \quad (17.2b)$$

$$1 = \left[\frac{T}{W} \sin(\alpha_B + \mu_T) + \frac{\bar{q}}{W/S} C_L \right] \cos \nu \quad (17.2c)$$

Le incognite di questo problema sono la V , α , e δ , considerando che la ν è connessa al fattore di carico normale f_{z_A} e quindi a δ col raggio di virata.

$$f_{z_A} = 1 / \cos \nu$$

17.2.1 Virata corretta per assegnata legge $C_L(t)$ - Esercizi da 17.1 a 17.3

L'integrazione del sistema viene effettuata assegnando una legge temporale del $C_L(t)$, assumendo T e δ_e si farà ricorso, durante l'integrazione del sistema, all'equilibrio alla rotazione intorno all'asse di beccheggio, così da ottenere una legge temporale coerente con la manovra. E' stata effettuata un'analisi sul velivolo *Cessna Citation II 550*, con la legge di sustentazione riportata in 17.1 nella pagina successiva. Tale legge ha per $C_L(0)$ pari al coefficiente di portanza in volo equilibrato ad ali livellate.

E' stata ricercata una condizione di volo traslato in equilibrio ad ali livellate. Nell'istante $t = 0$ sono state assegnate le seguenti condizioni riportate nel listato:

Listato 17.1 Legge di sustentazione fornita

%% Settaggio legge di manovra

```
t_fin = 32.0; %Tempo di simulazione finale

% Legge del CL
vTime_bp = [0, t_fin*(1/10), t_fin*(1/3), t_fin*(2/4), t_fin*(2/3), t_fin ];
vCL_bp = [CL_eq, CL_eq+0.01 CL_eq+0.06, CL_eq+0.12, CL_eq+0.145, CL_eq
+0.15];
```

La soluzione del problema mediante codice Matlab proposto dal docente ha prodotto i seguenti andamenti delle variabili di stato:

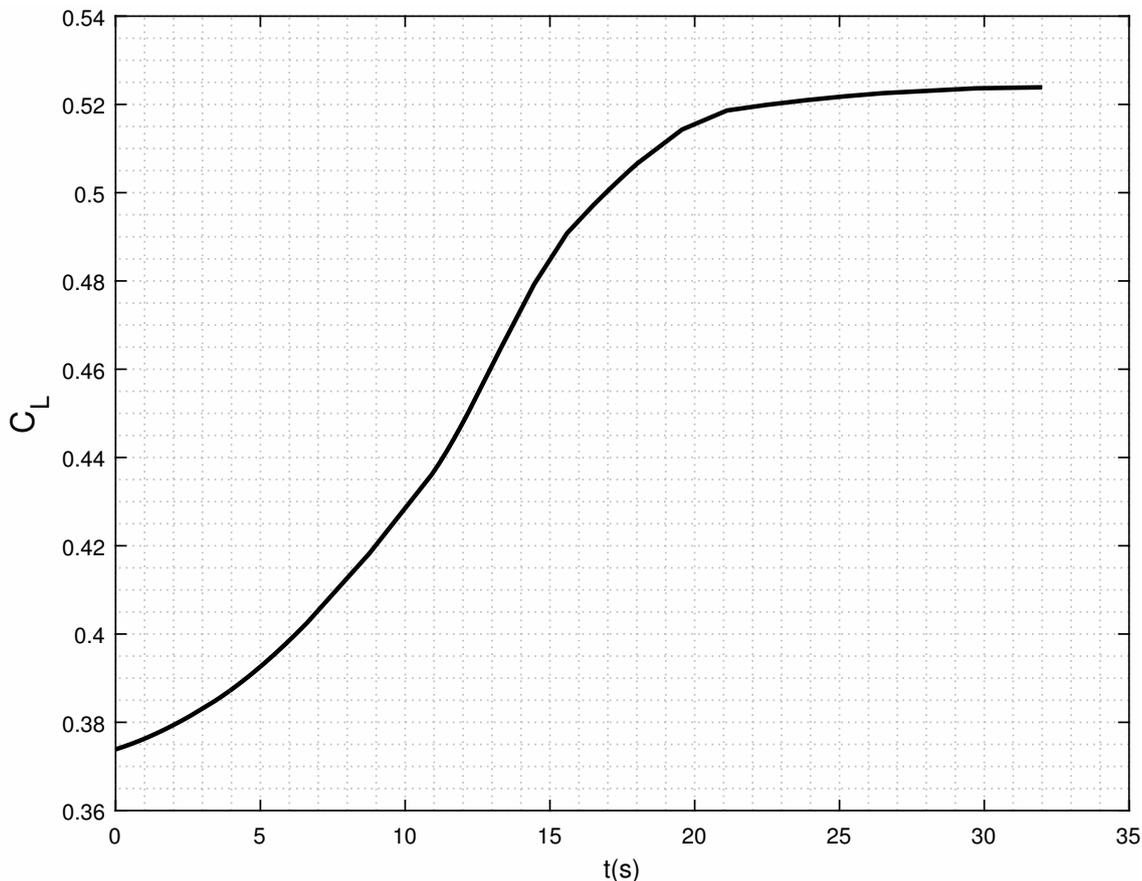


Figura 17.1 Legge di Sostentazione

Lo script col quale è stato possibile rappresentare il tipo di manovra è stato proposto dal docente ed è qui riproposto nel listato 17.2.

Nota bene le condizioni di trim sono state valutate mediante la procedura vista negli esercizi del quaderno 7. Le soluzioni sono state salvate nel file trim_result_h_1524m.mat.

Listato 17.2 Codice per la risoluzione della virata

```
close all; clear all; clc;

%%Declaring global variables
global ...
g ... % gravity acceleration
rho_0 ... % air density
delta_T_0 ...
delta_s_0 ...
vTime_bp ...
vCL_bp ...
myAC

%% Populate aircraft data
```

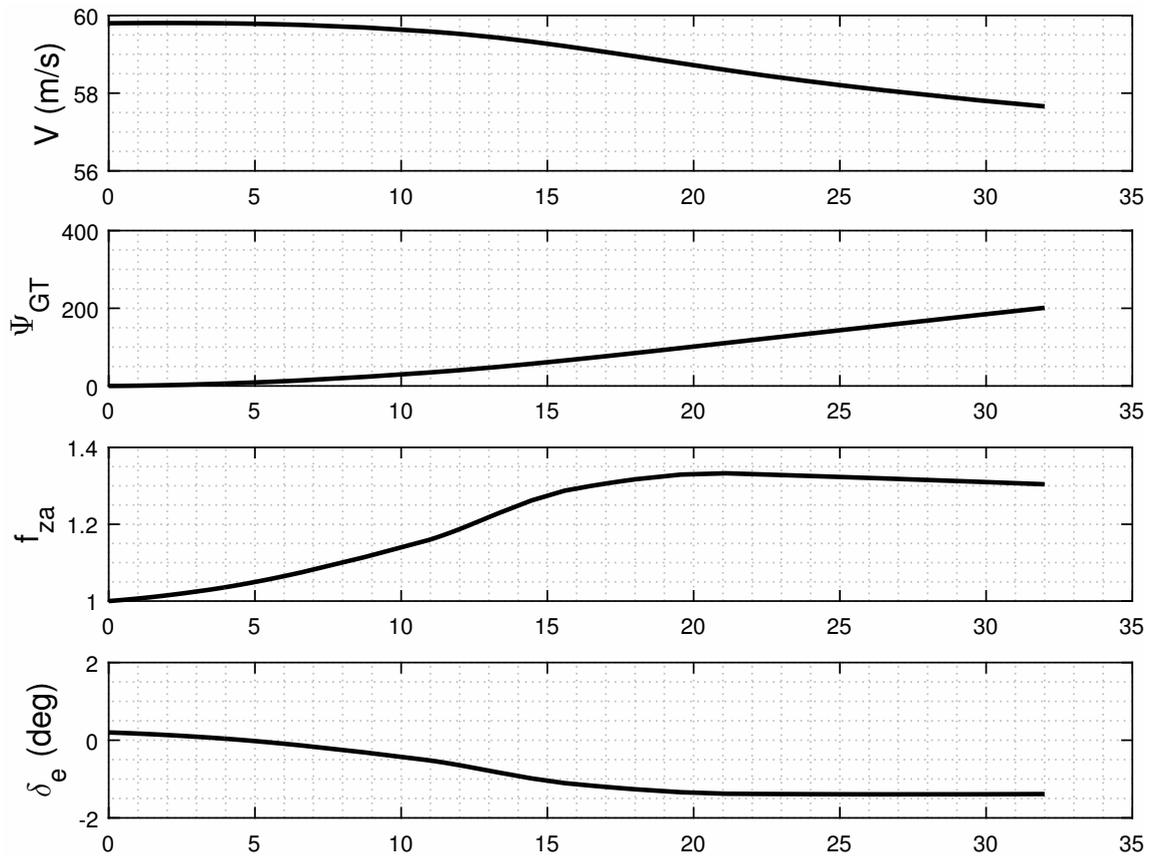


Figura 17.2 Storie Temporali per la legge di sostentazione.

```

aircraftDataFileName = 'DSV_Aircraft_Data2010.txt';

%% Aircraft object
myAC = DSVAircraft(aircraftDataFileName);

if (myAC.err == -1)
    disp(['... terminating.']);
else
    disp(['File ', aircraftDataFileName, ' letto correttamente.']);
end

%% Initial conditions
z_0 = 1524.; % Quota dal livello del mare (m)
q_0 = 0.; % Velocità angolare di beccheggio (rad/s)
gamma_0 = convang(0,'deg','rad'); % Angolo di rampa (rad)

%% Using Matlab built-in ISA model for density
[air_Temp_0, sound_speed_0, air_pressure_0, rho_0] = atmosisa(z_0);

%% Gravity acceleration
g = 9.81; % (m/s^2)

%% Solve the trim problem first, wings-level and zero R/C
% load trim results
load 'trim_results_h_1524m.mat'

% Data read from file:
V_0 = mySolution.V_0;
alpha_0_deg = mySolution.alpha_0_deg ;
alpha_0 = mySolution.alpha_0;
delta_e_0_deg = mySolution.delta_e_0_deg;
delta_e_0 = mySolution.delta_e_0;

```

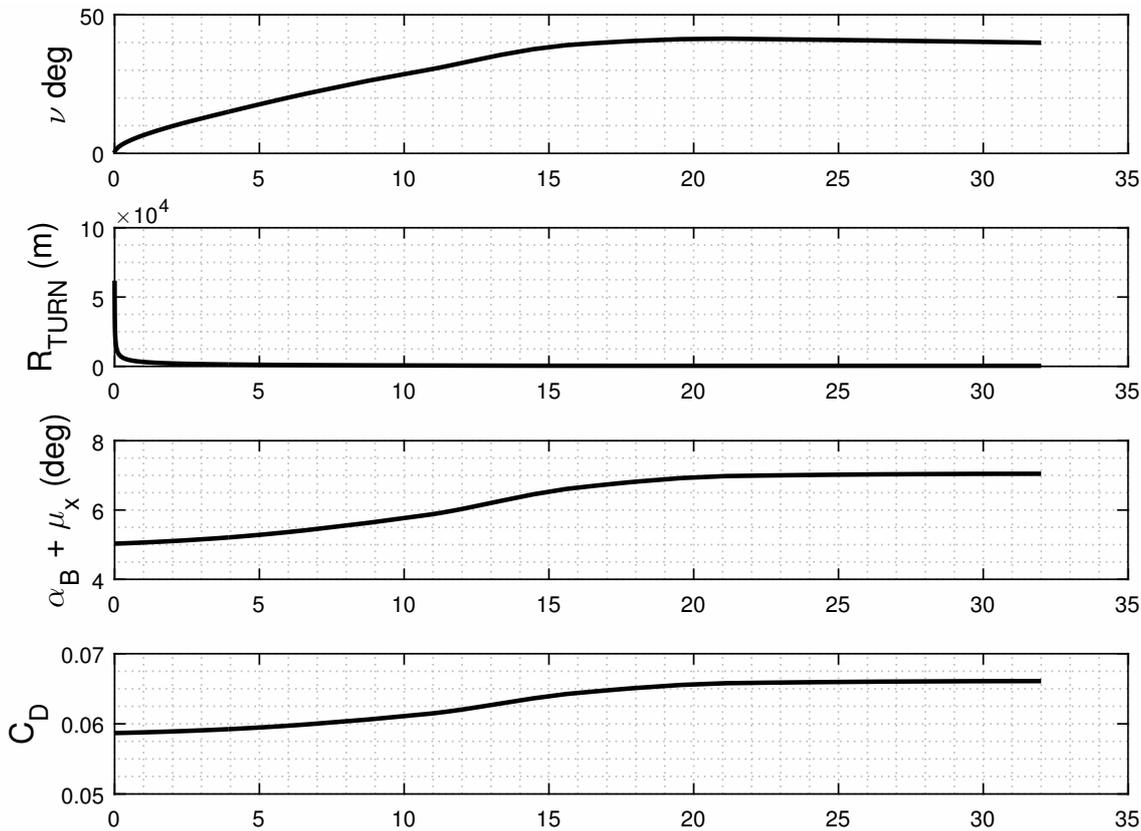


Figura 17.3 Storie Temporali per la legge di sustentazione.

```

delta_s_0_deg = mySolution.delta_s_0_deg ;
delta_s_0 = mySolution.delta_s_0;
delta_T_0 = mySolution.delta_T_0;

%Heading iniziale
psiGT_0 = convang(0,'deg','rad');
%Fattore di carico iniziale
fza_0 = 1.0;
%Cl iniziale
CL_eq = f_CL(alpha_0, 0.0, 0.0, delta_e_0, delta_s_0);

%% Settaggio per la manovra

t_fin = 32.0; %Tempo di simulazione finale

% Legge del Cl
vTime_bp = [0, t_fin*(1/10), t_fin*(1/3), t_fin*(2/4), t_fin*(2/3), t_fin ];
vCL_bp = [CL_eq, CL_eq+0.01 CL_eq+0.06, CL_eq+0.12, CL_eq+0.145, CL_eq
+0.15];

%% Soluzione dell'equazioni 17.61

% Matrice di Massa
M = diag([1,1,0,0]');

%Inizializzazione del vettore iniziale
x0 = [ ...
    V_0; ...
    psiGT_0; ...
    fza_0; ...
    delta_e_0];

options_ODE = odeset('AbsTol',1e-7,'Mass',M);

```

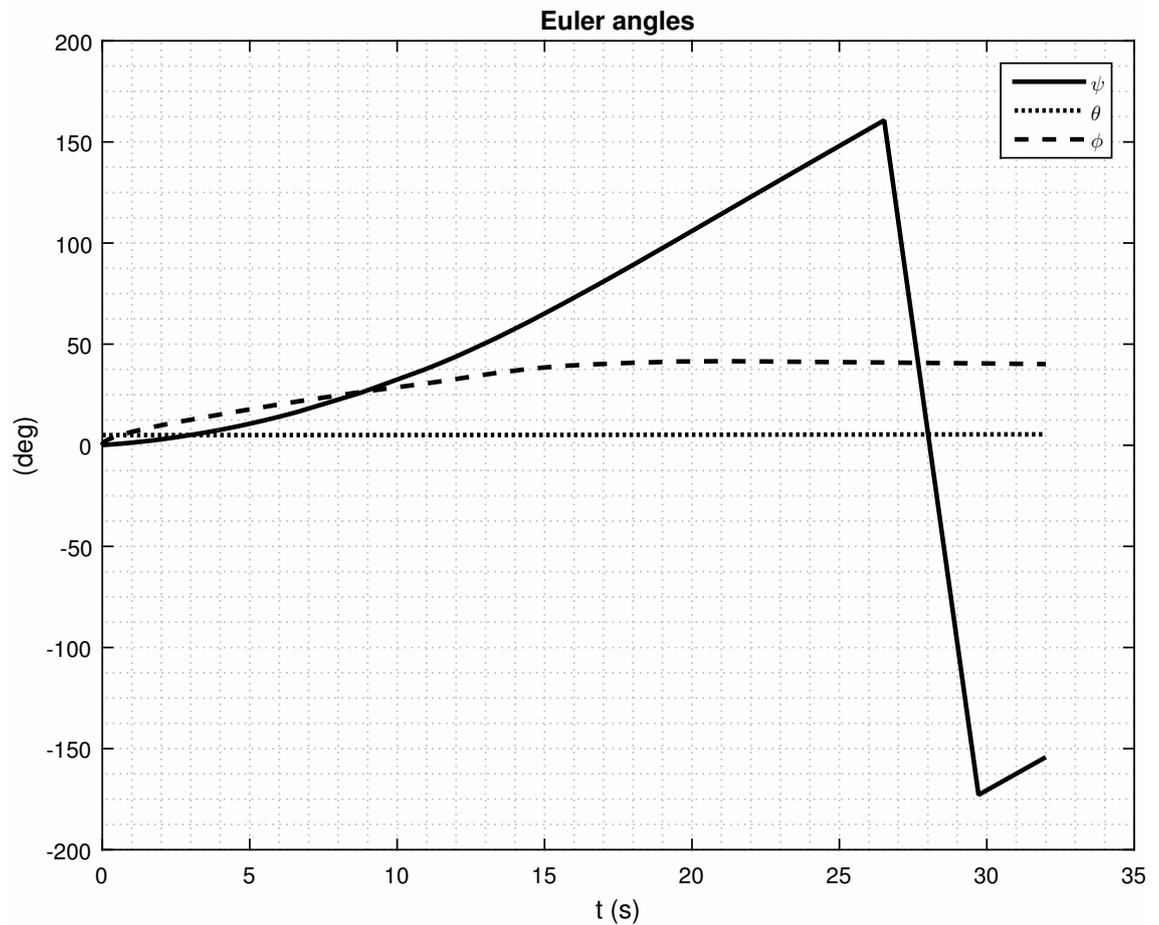


Figura 17.4 Angoli Eulero per la manovra di virata

```
[vTime, vX] = ode15s(@correctedTurnCLAssigned, [0 t_fin], x0, options_ODE);
```

```
%% auxiliary variables
vY(:,1) = ... % bank angle
    acos(1./vX(:,3));
vY(:,2) = ... % turn radius
    (vX(:,1).^2)./(g.*sqrt(vX(:,3).^2 - 1.));
vY(:,3) = ... % Angle of Attack
    f_AoA(...
        interp1(vTime_bp,vCL_bp,vTime,'pchip'), ...
        vX(:,4), ...
        delta_s_0*ones(length(vX(:,4)),1) ...
    );
vY(:,4) = ... % CD
    f_CD_polar( ...
        interp1(vTime_bp,vCL_bp, vTime,'pchip') ...
    );
```

```
CL = interp1(vTime_bp,vCL_bp,vTime,'pchip');
```

```
%% PLOT SOLUZIONE PROBLEMA VIRATA CORRETTA
```

```
h1=figure(1);
plot(vTime,CL,'-', 'LineWidth',2);
ylabel('C_L', 'fontsize',14);
xlabel('t(s)');
grid minor;
print(h1, '-deps', 'C:\Users\Giuseppe\Dropbox\DSV\__template_esercitazioni\
quaderno17\images\LeggeDiSostentazioneES17_1');
```

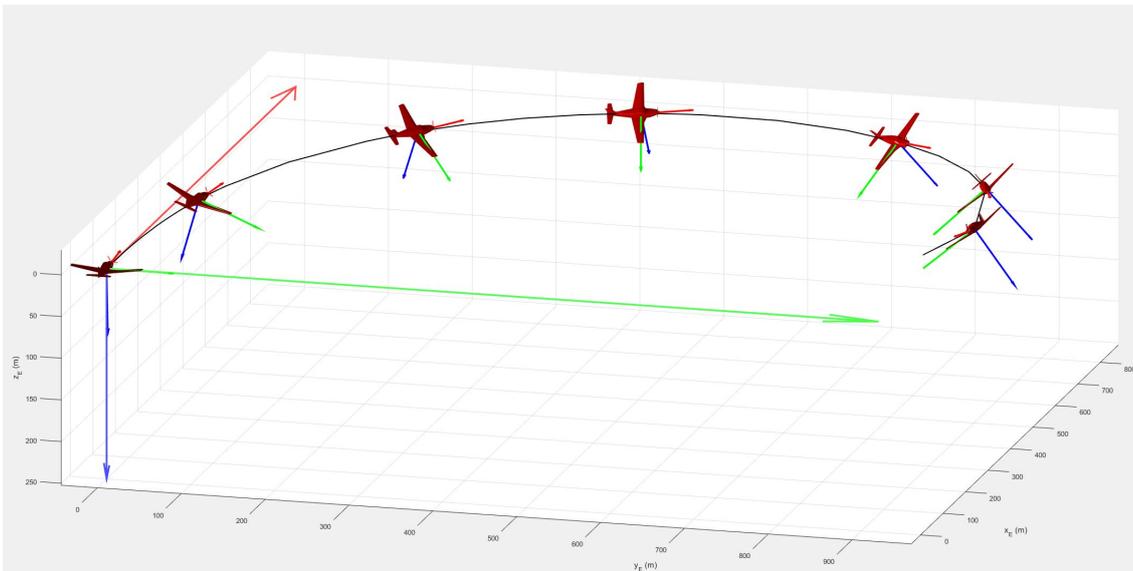


Figura 17.5 Angoli Euler per la manovra di virata

```

print(h1, '-deps', 'Immagini\LeggeDiSostentazioneES17_1');

h2=figure(2);
subplot 411
plot(vTime,vX(:,1),'-','LineWidth',2);
ylabel('V (m/s)', 'fontsize',14);
grid minor;
subplot 412
plot(vTime,convang(vX(:,2),'rad','deg'),'-','LineWidth',2);
ylabel('\Psi_{GT}', 'fontsize',14);
grid minor;
subplot 413
plot(vTime,vX(:,3),'-','LineWidth',2);
ylabel('f_{za}', 'fontsize',14);
grid minor;
subplot 414
plot(vTime,convang(vX(:,4),'rad','deg'),'-','LineWidth',2);
ylabel('\delta_e (deg)', 'fontsize',14);
grid minor;
print(h2, '-deps', 'C:\Users\Giuseppe\Dropbox\DSV\__template_esercitazioni\
quaderno17\images\StorieTemporalIA17_1');
print(h2, '-deps', 'Immagini\StorieTemporalIA17_1');

h3=figure(3);
subplot 411
plot(vTime,convang(vY(:,1),'rad','deg'),'-','LineWidth',2);
ylabel('\nu deg', 'fontsize',14);
grid minor;
subplot 412
plot(vTime,vY(:,2),'-','LineWidth',2);
ylabel('R_{TURN} (m)', 'fontsize',14);
grid minor;
subplot 413
plot(vTime,convang(vY(:,3),'rad','deg'),'-','LineWidth',2);
ylabel('\alpha_B + \mu_x (deg)', 'fontsize',14);
grid minor;
subplot 414
plot(vTime,vY(:,4),'-','LineWidth',2);
ylabel('C_D', 'fontsize',14);
grid minor;
print(h3, '-deps', 'C:\Users\Giuseppe\Dropbox\DSV\__template_esercitazioni\
quaderno17\images\StorieTemporalIB17_1');
print(h3, '-deps', 'Immagini\StorieTemporalIB17_1');

```

```

%% Angoli di Eulero - Esercizio 17.3
% Rotazione dell'angolo intorno al terzo asse
Rot3 = @(a)[cos(a),sin(a),0;-sin(a),cos(a),0; 0, 0,1];

% Rotazione dell'angolo intorno al secondo asse
Rot2 = @(a)[cos(a),0,-sin(a);0,1,0;sin(a),0,cos(a)];

% Rotazione dell'angolo intorno al primo asse
Rot1 = @(a)[ 1,0,0;0,cos(a),sin(a);0,-sin(a),cos(a)];

%% Pre-assign Euler angles arrays
vPsi = zeros(length(vTime),1);
vTheta = zeros(length(vTime),1);
vPhi = zeros(length(vTime),1);

% Applico la trasformazione 17.2 noti (psi,ni,alpha)

for k=1:length(vTime)
    psigt = vX(k,2); nu = vY(k,1); alpha = vY(k,3);
    %Rotation sequence 3-1-2
    Tbe = Rot2(alpha)*Rot1(nu)*Rot3(psigt);
    [psi,theta,phi] = dcm2angle(Tbe);
    vPsi(k) = psi; vTheta(k) = theta; vPhi(k) = phi;
end

%Time sequence of aircraft orientation quaternion
vQuat = angle2quat(vPsi,vTheta,vPhi);

% Euler angles time histories
h4=figure(4)
plot(vTime,convang(vPsi,'rad','deg'),'--',vTime,convang(vTheta,'rad','deg'),':',...
     vTime,convang(vPhi,'rad','deg'),'--','LineWidth',2)
grid minor;
legend('\psi','\theta','\phi')
xlabel('t (s)');
ylabel('(deg)');
title('Euler angles')
print(h4,'-deps','C:\Users\Giuseppe\Dropbox\DSV\__template_esercitazioni\
quaderno17\images\AngoliEuleroEs17_1');
print(h4,'-deps','Immagini\AngoliEuleroEs17_1');

%% TRAJECTORY - Solution of navigation equations

dPosEdt = @(t,Pos)[interp1(vTime,vX(:,1).*cos(vX(:,2)),t);interp1(vTime,vX(:,1).*
    sin(vX(:,2)),t);0];

options = odeset('RelTol', 1e-3,'AbsTol', 1e-3*ones(3,1));
PosE0 = [0;0;-z_0];
[vTime2, vPosE] = ode45(dPosEdt, vTime, PosE0, options);
N = length(vPosE);

vXe = vPosE(:,1);
vYe = vPosE(:,2);
vZe = z_0 + vPosE(:,3);

% CG coordinates time history
figure(5)
plot(vTime2,vXe, '--',vTime2,vYe, '-.',vTime2,vZe, '---')

% hold on, view(3)
legend('x_{G,E}(t)', 'y_{G,E}(t)', 'z_{G,E}(t)')
xlabel('t (s)'); ylabel('(m)');
title('CG coordinates in Earth axes');

%% Setup the figure/scene for 3D visualization

```

```

h_fig3 = figure(6);
grid on;
hold on;
light('Position',[1 0 4],'Style','local');
% Trick to have Ze pointing downward and correct visualization
set(gca,'XDir','reverse');
set(gca,'ZDir','reverse');
daspect([1 1 1]);

%% Load aircraft shape
shapeScaleFactor = 40.0;

shape = loadAircraftMAT('aircraft_pa24-250.mat', shapeScaleFactor);

mXYZe = [vPosE(:,1),vPosE(:,2),vPosE(:,3)+z_0];
mEul = [vPsi,vTheta,vPhi];

%% Settings
% General settings
options.samples = [1,55,60,65,70,75,77,80,81];
options.theView = [105 15];

% body axes settings
options.bodyAxes.show = true;
options.bodyAxes.magX = 1.5*shapeScaleFactor;
options.bodyAxes.magY = 2.0*shapeScaleFactor;
options.bodyAxes.magZ = 2.0*shapeScaleFactor;
options.bodyAxes.lineWidth = 2.5;

% helper lines
options.helperLines.show = true;
options.helperLines.lineStyle = ':';
options.helperLines.lineColor = 'k';
options.helperLines.lineWidth = 1.5;

% trajectory
options.trajectory.show = true;
options.trajectory.lineStyle = '-';
options.trajectory.lineColor = 'k';
options.trajectory.lineWidth = 1.5;

%% Plot body and trajectory
plotTrajectoryAndBodyE(h_fig3, shape, mXYZe, mEul, options);

%% Plot Earth axes
hold on;
xMax = max([max(abs(mXYZe(:,1))),5]);
yMax = max([max(abs(mXYZe(:,2))),5]);
zMax = 0.3*xMax; % max([abs(max(vXYZe(1))),0.18*xMax]);
vXYZ0 = [0,0,0];
vExtent = [xMax,yMax,zMax];
plotEarthAxes(h_fig3, vXYZ0, vExtent);
xlabel('x_E (m)'); ylabel('y_E (m)'); zlabel('z_E (m)')
hold on

```

Si nota come nel precedente listato viene passata alla funzione `ode15s` la funzione esterna `correctedTurnCLAssigned` che definisce 4 specifiche funzioni.

Listato 17.3 correctedTurnCLAssigned

```

function dx = correctedTurnCLAssigned(t,x)
% pass this to ode15s

%% Declaring global variables
global ...
g ... % gravity acceleration

```

```

rho_0 ... % air density
delta_T_0 ...
delta_s_0 ...
vTime_bp ...
vCL_bp ...
myAC % the aircraft object, populated outside this func

%% Give the state vector components proper names
V = x(1);
psiGT = x(2);
fza = x(3);
delta_e = x(4);

% give dimensions to the return vector
dx = zeros(4,1);

%% Right-hand-sides of equation of motion

rho = rho_0;
delta_T = delta_T_0;
delta_s = delta_s_0;
CL = interp1(vTime_bp,vCL_bp,t,'pchip');

ToW = delta_T*myAC.T/myAC.W;
WoS = myAC.W/myAC.S;
a_ = ...
    ( CL - myAC.CL_delta_e*delta_e ...
      - myAC.CL_delta_s*delta_s ...
      )/myAC.CL_alpha;

b_ =(rho*V^2)/(2*(myAC.W/myAC.S));

dx(1,1) = g*( ...
    ToW*( cos(myAC.mu_T) + a_*sin(myAC.mu_T) ) ...
    - b_*(myAC.CD_0 + myAC.K*CL^myAC.m ) ...
    );

dx(2,1) = (g/V)*sqrt(fza^2 - 1);

dx(3,1) = -fza ...
    + ToW*( a_*cos(myAC.mu_T) - sin(myAC.mu_T) ) ...
    +b_*CL;

c_ = 1/(myAC.Cm_alpha*(myAC.CL_delta_e/myAC.CL_alpha) - myAC.Cm_delta_e );

d_ = 2*(-400)*(g^2)/(rho*myAC.S*myAC.mac);

e_ = myAC.Cm_0 ...
    + myAC.Cm_alpha * CL / myAC.CL_alpha ...
    + ( ...
    myAC.Cm_delta_s ...
    - myAC.Cm_alpha*(myAC.CL_delta_s/myAC.CL_alpha) ...
    )*delta_s;

f_ = myAC.Cm_q*g*myAC.mac/2;

dx(4,1) = -delta_e ...
    +c_*( ...
    d_*(1 - 1/fza^2)/(V^4) ...
    + e_ + f_*(fza - 1/fza)/(V^2) ...
    );
end

```

La soluzione del problema numerico impostato sopra è memorizzata negli array vX e vY . Il loro contenuto è stato riportato nelle immagini ed è in funzione dei tempi discreti $vTime$. Si riportano per completezza anche alcune funzioni richiamate nel codice precedente. La funzione che calcola il valore di equilibrio $C_L(0)$ è la seguente.

Listato 17.4 f_CL

```
function res = f_CL(alpha, alpha_dot_hat, q_hat, delta_e, delta_s)
global myAC
res = myAC.CL_alpha.*alpha ...
    + myAC.CL_delta_e.*delta_e ...
    + myAC.CL_delta_s.*delta_s ...
    + myAC.CL_alpha_dot.*alpha_dot_hat ...
    + myAC.CL_q.*q_hat;
end
```

Listato 17.5 f_CD_polar

```
function res = f_CD_polar(CL)
global myAC
res = myAC.CD_0 + myAC.K .* CL.^myAC.m;
end
```

Listato 17.6 f_AoA

```
function res = f_AoA(CL, delta_e, delta_s)
global myAC
res = ...
    (1./myAC.CL_alpha) ...
    .* ( CL - myAC.CL_delta_s.*delta_s - myAC.CL_delta_e.*delta_e);
end
```


Carichi sul piano orizzontale di coda

Indice

18.1 Introduzione	209
18.2 Dati del velivolo	210
18.3 Realizzazione del Diagramma di manovra	211
18.4 Carico aerodinamico di equilibrio	215

18.1 Introduzione

Si vogliono calcolare in questa esercitazione proposta dal docente i carichi sul piano di coda orizzontale di un velivolo da trasporto a breve raggio e per condizioni di volo simmetriche. L'esercizio verrà articolato per punti.

Si è cominciato l'esercizio con il tracciare il diagramma di manovra completo, ovvero quello che fornisce la variazione del fattore di carico nominale f_{zA} in funzione della velocità di volo equivalente V_e , sia per volo dritto che per volo rovescio.

Si determineranno, successivamente, le curve del carico aerodinamico di equilibrio L_H in funzione di V_e nelle condizioni di volo fondamentali rappresentate dai punti sul contorno del diagramma di manovra (condizioni cosiddette al limite del diagramma di manovra). Queste saranno ottenute variando la posizione del baricentro del velivolo una volta al 25% e un'altra volta al 30% della corda media aerodinamica.

Le curve del carico aerodinamico di equilibrio saranno ottenute anche variando le condizioni del fattore di carico ponendolo pari a ± 1 .

Tabella 18.1 Caratteristiche del velivolo assegnato.

Massa, m	18 500 kg
Superficie alare, S	82 m ²
Corda media aerodinamica (m.a.c.), \bar{c}	3,5 m
Raggio d'inerzia rispetto l'asse di beccheggio, K_y	3,535
Distanza del paricentro del velivolo completo dal centro aerodinamico del piani orizzontale di coda, l_H (CG @ 0.25 \bar{c})	9,20 m
Distanza del paricentro del velivolo completo dal centro aerodinamico del piani orizzontale di coda, l_H (CG @ 0.30 \bar{c})	9,05 m
Distanza del baricentro del velivolo dal baricentro del piano orizzontale di coda, X_H (CG @ 0.25 \bar{c})	9,60 m
Distanza del baricentro del velivolo dal baricentro del piano orizzontale di coda, X_H (CG @ 0.30 \bar{c})	9,45 m
Massa del piano orizzontale di coda, m_H	250 Kg
$C_{L,max}$	1,4
$C_{L,max}$, in volo rovescio	-1,1
Massima velocità equivalente, $V_{e,max} \equiv V_D$	600 km/h
Fattore di carico limite, $f_{Z_A,max} \equiv n$	3,0
Fattore di carico limite in volo rovescio, n_t	-1,5
Escursione del baricentro	0,25 \bar{c} - 0,30 \bar{c}

Tabella 18.2 Punti caratteristici delagramma di manovra, riguardanti il velivolo assegnato.

<i>Punto</i>	<i>Velocità equivalente, V_e</i>	<i>Fattore di carico</i>
A (PHA)	di manovra, V_A	$n = 3,00$
B (PLA)	massima, V_D	3,00
C (NLA)	massima, V_D	0,00
D	massima rovescia, V_D	-1
E (NIA)	intermedia, $0,7 V_d$	-1,5
F (NHA)	di manovra rovescia, $V_{A,r}$	-1,5
G	di stallo rovescio, $V_{s1,r}$	-1
H	di stallo, $V_{s,1}$	1

18.2 Dati del velivolo

I dati caratteristici del velivolo oggetto di studio sono riportati nella tabella 18.1.

I punti caratteristici del diagramma di manovra sono definiti nella tabella 18.2 dove si sono indicati con i simboli:

- $V_{s,1}$, indica la velocità di stallo in volo diritto a fattore di carico unitario.
- $V_{s,1,r}$, indica la velocità di stallo in volo rovescio a fattore di carico -1,00.
- **PHA** (*Positive High Angle of attack*), indica la condizione di volo alla velocità di manovra V_A al fattore di carico limite positivo.

- PLA (*Positive Low Angle of attack*), indica la condizione di volo alla velocità massima V_D al fattore di carico limite positivo.
- NLA (*Negative Low Angle of attack*), indica la condizione di volo alla velocità massima V_D a fattore di carico nullo.
- NIA (*Negative Intermediate Angle of attack*), indica la condizione di volo alla velocità $0,7V_D$ a fattore di carico limite negativo.
- NHA (*Negative Low Angle of attack*), indica la la condizione di volo alla velocità $V_{s1,r}$ a fattore di carico limite negativo.

Infine nella 18.1 fornita dal docente, è riportata la curva del coefficiente di momento del velivolo parziale $C_{M,WB}$ rispetto al punto A_0 , bordo di attacco della corda media aerodinamica.

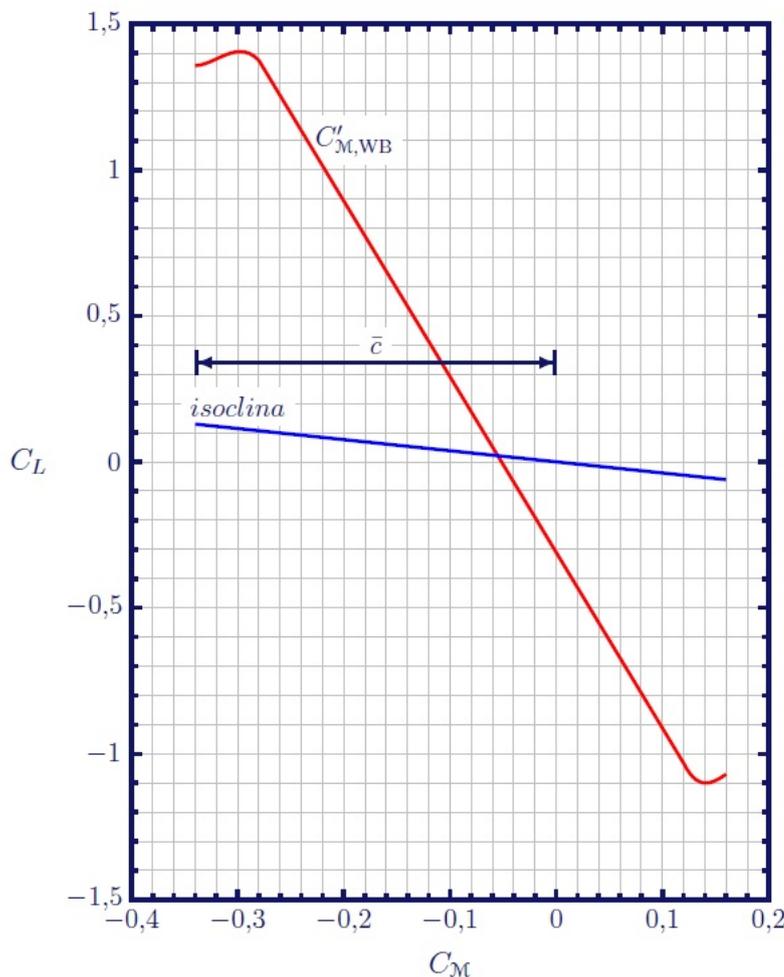


Figura 18.1 Diagramma del coefficiente di momento del velivolo parziale rispetto al punto A_0 , bordo di attacco della corda media aerodinamica. Nel piano $(C_{M_{WB}}, C_L)$ si è tracciata l'isoclina passante per il punto a $C_L = 0$ per baricentro posto al 25% di \bar{c} .

18.3 Realizzazione del Diagramma di manovra

Il diagramma di manovra fornisce la variazione del fattore di carico normale f_{zA} in funzione della velocità di volo equivalente V_e ($V_e = V_{TAS} \sqrt{\rho/\rho_0}$) per condizioni di volo

fondamentali, ovvero in cui è identicamente nulla la velocità angolare di beccheggio del velivolo q .

In termini di velocità, il limite inferiore del diagramma è determinato da condizioni aerodinamiche, precisamente dal raggiungimento degli assetti di stallo alle basse velocità di volo, mentre il limite superiore dipende dal valore di velocità massima (equivalente) conseguibile, che si ha in discesa e per il massimo grado di manetta del motore. I limiti in termini del fattore di carico derivano sia da condizioni aerodinamiche (limite di stallo per il tratto parabolico del bordo del diagramma, sia in volo dritto che rovescio) sia da condizioni strutturali (tratti orizzontali, corrispondenti ai *fattori di carico massimo* o *fattori di contingenza* in volo dritto e rovescio).

I punti interni al diagramma e sul bordo rappresentano l'insieme di tutte e sole le condizioni di volo navigabili in sicurezza e costituiscono il cosiddetto *inviluppo di volo in manovra* del velivolo.

Pertanto, conoscendo i dati del velivolo riportati in tabella 18.2 a pagina 210, sarà possibile realizzare il diagramma di manovra. Bisognerà andare a valutare l'andamento del diagramma sui due tratti parabolici per il volo dritto e volo rovescio. Questi sono caratteristici delle condizioni di stallo in volo dritto che in volo rovescio, ovvero quando si ha il coefficiente di portanza massima, $C_{L,max}$ e il coefficiente di portanza minimo, $C_{L,min}$ (massimo in volo rovescio).

Il listato seguente mostra un possibile codice realizzato in Matlab per la rappresentazione del diagramma di manovra.

Listato 18.1 Listato per la realizzazione del diagramma di manovra

```

clc; clear all; close all;

%%Dati Velivolo recuperati dalle tabelle fornita dal docente 18.1 e 18.2:
g      = 9.81;           %Accelerazione di gravità [m/s^2]
m      = 18500;         %massa [Kg]
W      = m*g;          %Peso in [N]
S      = 82;           %Superficie alare [m^2]
c      = 3.5;          %Corda media aerodinamica [m]
Ky     = 3.535;        %raggio di inerzia [m]
B      = m*Ky^2;       %Momento di inerzia [Kg*m^2]
mH     = 250;          %Massa del piano di coda orizzontale [kg]
CL_max = 1.4;          %Coeff. di portanza massimo in volo dritto
CL_min = -1.1;        %Coeff. di portanza minimo in volo rovescio
Ve_max = 600;          %Velocità equivalente massima [km/h]
V_D    = convvel(Ve_max, 'km/h', 'm/s'); %Massima velocità equivalente [m/s]
z      = 0;            %Quota di riferimento
[air_Temp, sound_speed, air_pressure, rho] = atmosisa(z);

%%Costruzione del Diagramma di manovra

%Assegnazione dei punti caratteristici del diagramma:
%Volo dritto
pro_f_H = 0;
f_H     = 1;
f_A     = 3;
f_B     = 3;
f_C     = 0;
%Volo rovescio
f_D     = -1;
f_E     = -1.5;
f_F     = -1.5;
f_G     = -1;
pro_f_G = 0;
%Costruzione della parte parabolica del diagramma, tratto non tratteggiato

```

```

%si infittiscono le zone paraboliche del fattore di carico, sia per volo
%Volo dritto
FattCarico_HA_dritto=linspace(f_H, f_A, 60);
Velocita_HA_dritto=sqrt((2*FattCarico_HA_dritto*(W/S))/(rho*CL_max));
%Volo rovescio
FattoreCarico_FG_rovescio=linspace(f_F, f_G, 60);
Velocita_FG_rovescio=sqrt((2*FattoreCarico_FG_rovescio*(W/S))/(rho*CL_min));
%Ascisse dei punti caratteristici, assegnati dalla tabella 18.2
%Volo dritto
V_H = Velocita_HA_dritto(1);
V_A = Velocita_HA_dritto(end);
V_B = V_D;
%Volo rovescio
V_C = V_D;
V_E = 0.7*V_D;
V_F = Velocita_FG_rovescio(1);
V_G = Velocita_FG_rovescio(end);
%Coordinate del bordo del diagramma di manovra
Velocita= [V_H Velocita_HA_dritto V_A V_B V_C V_D V_E V_F V_G V_G];
Fattore = [pro_f_H FattCarico_HA_dritto f_A f_B f_C f_D f_E f_F f_G pro_f_G];
%Coordinate dei punti caratteristici
x_punti = [V_H V_A V_B V_C V_D V_E V_F V_G];
y_punti = [f_H f_A f_B f_C f_D f_E f_F f_G];
%Tratto tratteggiato:
FattCarico_0H_dritto = linspace(0, f_H, 30);
Velocita_0H_dritto = sqrt((2*FattCarico_0H_dritto*(W/S))/(rho*CL_max));
FattCarico_0G_rovesc = linspace(0, f_G, 30);
Velocita_0G_rovescio = sqrt((2*FattCarico_0G_rovesc*(W/S))/(rho*CL_min));

%% Plot del diagramma di manovra
h1=figure(1);
plot(Velocita, Fattore, '-', 'Linewidth', 2.5)
grid minor
hold on
plot(Velocita_0H_dritto, FattCarico_0H_dritto, '--k',...
      Velocita_0G_rovescio, FattCarico_0G_rovesc, '--k',...
      [0 V_H], [f_H f_H], '--k',...
      [0 V_A], [f_A f_A], '--k',...
      [V_A V_A], [0 f_A], '--k',...
      [V_E V_E], [0, f_E], '--k',...
      [V_F V_F], [0, f_F], '--k',...
      [0 V_F], [f_F, f_F], '--k',...
      [0 V_G], [f_G, f_G], '--k',...
      [0 60 180], [0 0 0], 'k')
axis([0 180 -3 4]);
hold on
plot(x_punti, y_punti, 'o', 'MarkerEdgeColor', 'k',...
      'MarkerFaceColor', 'k', ...
      'MarkerSize', 4);
hold on
xlabel(['$' 'V_e \; (m/s)' '$'], 'interpreter', 'latex');
ylabel(['$' 'f_{z_{A}}' '$'], 'interpreter', 'latex');

%Definisco delle stinghe che delimitano i punti sul diagramma:
% Plot delle Lettere
text(V_A, f_A+0.3, 'A', 'fontsize', 13);
text(V_D +0.2, f_A+0.3, 'B', 'fontsize', 13);
text(V_D +0.5, 0+0.3, 'C', 'fontsize', 13);
text(V_D +1, f_G, 'D', 'fontsize', 13);
text(V_E +0.2, f_G-0.8, 'E', 'fontsize', 13);
text(V_F +0.2, f_G-0.8, 'F', 'fontsize', 13);
text(V_G-5, f_G-0.25, 'G', 'fontsize', 13);
text(V_H-5, f_A-1.7, 'H', 'fontsize', 13);
%Riferimento velocita
text(V_H-5, -0.4, ['$' 'V_{s,1}' '$'], 'interpreter', 'latex', 'fontsize', 12);
text(V_H+3, +0.4, ['$' 'V_{sr,1}' '$'], 'interpreter', 'latex', 'fontsize', 12);

```

```

text(V_A,-0.3,['\$ 'V_{A}' \$'],'interpreter','latex','fontsize',12);
text(V_F+2,+0.3,['\$ 'V_{A,r}' \$'],'interpreter','latex','fontsize',12);
text(V_E-6,+0.3,['\$ '0.7\;V_{D}' \$'],'interpreter','latex','fontsize',12);
text(V_D-10,-0.3,['\$ 'V_{D}' \$'],'interpreter','latex','fontsize',12);

print(h1,'-deps','Grafici\DiagrammaManovra');
print(h1,'-deps','C:\Users\Giuseppe\Dropbox\DSV\__template_esercitazioni\
quadernoCarichi\images\DiagrammaManovra');

```

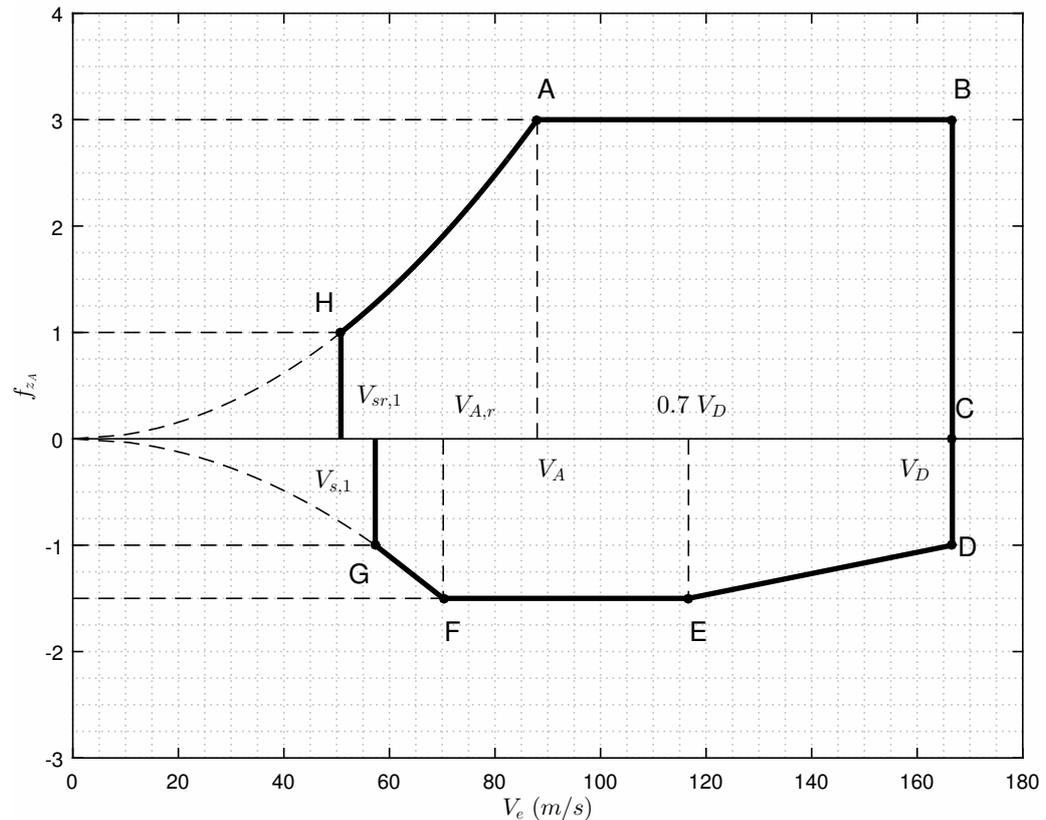


Figura 18.2 Diagramma di manovra del velivolo proposto per l'esercitazione

L'unica parte del codice che richiede un'elaborazione numerica è quella relativa alla determinazione delle coordinate dei punti dei tratti parabolici del bordo, sia nel caso di volo dritto che nel caso di volo rovescio. Questi tratti sono rappresentativi delle condizioni di stallo del velivolo, ovvero le condizioni in cui $C_L = C_{L,MAX}$ in volo dritto e $C_L = C_{L,min}$ in volo rovescio.

Si fissa un valore del fattore di carico lungo \bar{f}_{zA} compreso tra 0 e il valore massimo per il caso del volo dritto e tra 0 e il valore minimo per il caso di volo rovescio.

$$V_{s,\bar{f}_{zA}} = \sqrt{\frac{2 W}{\rho_0 S} \frac{\bar{f}_{zA}}{C_{L,MAX}}} \quad (18.1)$$

$$V_{s,\bar{f}_{zA}} = \sqrt{\frac{2 W}{\rho_0 S} \frac{\bar{f}_{zA}}{C_{L,min}}} \quad (18.2)$$

In figura 18.2 si è riportato il diagramma di manovra per il velivolo assegnato.

Tutti i regimi di volo sono rappresentati dai punti compresi nell'interno della poligonale A-B-C-D-E-F-G-H. Quindi, basterà effettuare una verifica nei punti al contorno del

poligono suddetto. Per completezza verrà descritto nel particolare i punti della poligonale mista di maggior interesse A-B-C-D ed E.

- Punto A: rappresenta la ripresa effettuata al coefficiente di contingenza n eseguita all'incidenza di portanza massima (in condizioni di velocità più basse compatibili con n , velocità di stallo compatibile con il fattore di carico massimo per il velivolo considerato);
- Punto B: corrisponde ad una ripresa dell'affondata eseguita alla massima velocità equivalente prescritta (al più basso coefficiente di portanza compatibile con n);
- Punto C: corrisponde ad una condizione di volo a portanza nulla alla massima velocità equivalente prescritta;
- Punto D: rappresenta la ripresa sul dorso per i velivoli acrobatici, la fase di volo imbarcato conseguente ad una manovra a picchiare, mentre per i rimanenti velivoli è sempre una condizione di volo imbarcato con una conseguente manovra di picchiata, ma con una massima velocità equivalente prescritta;
- Punto E: corrisponde alla ripresa sul dorso o alla fase di volo imbarcato conseguente alla manovra a picchiare, eseguita all'incidenza corrispondente alla massima portanza negativa (caso di volo rovescio)

18.4 Carico aerodinamico di equilibrio

In questa parte dell'esercizio si sono ricavate e diagrammate le curve del carico aerodinamico di equilibrio $L_{H,e}$ in funzione della velocità equivalente di volo, nelle condizioni di volo fondamentali corrispondenti ai punti limite del diagramma di manovra.

Il carico aerodinamico di equilibrio $L_{H,e}$, è quello che deve generare l'impennaggio di coda orizzontale per assicurare l'equilibrio dei momenti attorno all'asse di beccheggio del velivolo. Data l'esistenza di un braccio l_H fra il centro aerodinamico del piano di coda e il baricentro del velivolo, la forza L_H è responsabile di un momento di beccheggio rispetto al baricentro del velivolo $\mathcal{M}_{H,CG}$ che deve bilanciare gli altri momenti di natura aerodinamica, propulsiva e inerziale, secondo l'equazione dinamica:

$$\mathcal{M}_{WB,CG} + \mathcal{M}_{H,CG} + \mathcal{M}_{T,CG} = I_{yy}\dot{q} \quad (18.3)$$

In condizioni di volo fondamentali, e cioè quando sia verificato l'equilibrio alla rotazione attorno all'asse di beccheggio, la velocità angolare di beccheggio q è identicamente nulla e con essa anche l'accelerazione angolare di beccheggio \dot{q} . Allora dall'equazione 18.3, introducendo i coefficienti di momento e tenendo conto che $\mathcal{M}_{H,CG} = L_H l_H$ discende:

$$L_{H,e} = \frac{\bar{q}_\infty S \bar{c}}{l_H} \left(C_{\mathcal{M}_{WB}} + C_{\mathcal{M}_T} \right) \quad (18.4)$$

dove per leggerezza di notazione si è inteso che i momenti hanno per polo il baricentro del velivolo.

Nell'ipotesi di spinta baricentrica $C_{M_T} = 0$ e la valutazione del carico di equilibrio richiede il calcolo del coefficiente di momento di beccheggio del velivolo parziale $C_{M_{WB}}$ rispetto al baricentro. Quest'ultimo si può determinare con l'ausilio della curva (rossa) in figura 18.1 a pagina 211, che fornisce la variazione del coefficiente di momento del velivolo parziale rispetto al punto A' , bordo d'attacco della corda media aerodinamica.

Scelta una condizione di volo nel diagramma di manovra, individuata dalla coppia di valori (V_e, f_{z_A}) , è possibile determinare il coefficiente di portanza del velivolo completo tramite l'equazione:

$$C_L = \frac{f_{z_A} W}{\bar{q}_\infty} = \frac{2f_{z_A} W}{\rho_0 V_e^2 S} \quad (18.5)$$

Dalla conoscenza di C_L si può risalire a $C_{M_{WB,CG}}$ procedendo per via grafica.

Valutazione del $C_{M_{WB,CG}}$ per via grafica

Si procede alla valutazione del momento aerodinamico del velivolo parziale calcolato rispetto al baricentro. Nella figura 18.3 è riportata la procedura grafica che è stata applicata per calcolare il $C_{M_{WB,CG}}$ per il caso di baricentro posto al 25% della corda media aerodinamica e per il coefficiente di portanza $C_L = 1.0$, questi è un punto corrispondente ad un valore del diagramma di manovra avente: $V_e = 104, 12m/s$ e fattore di carico paria a $n = 3$.

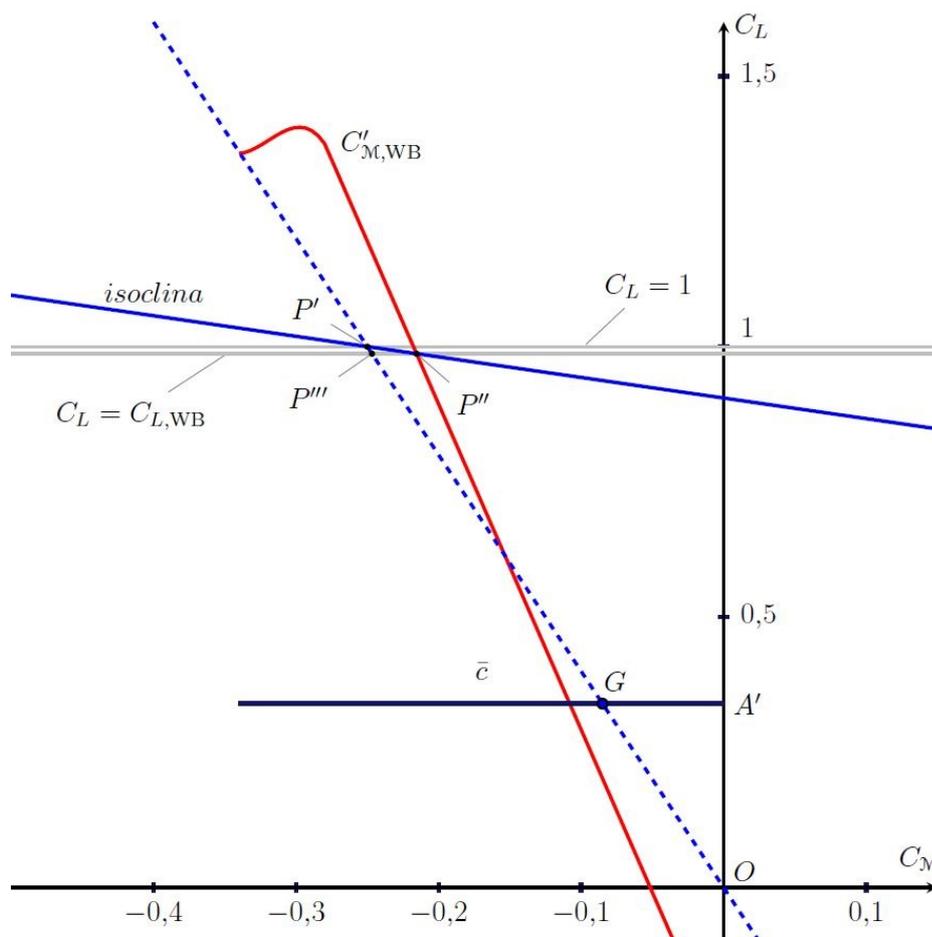


Figura 18.3 Determinazione grafica del coefficiente di momento del velivolo parziale $C_{M_{WB,CG}}$ rispetto al baricentro per un valore del coefficiente di portanza $C_L = 1.0$, nel caso di baricentro posto al 25% della corda media aerodinamica.

Il Segmento \bar{c} , riportato in figura 18.3 è stato tracciato tenendo conto della scala della

corda media ($\bar{c} = 0.34$) rispetto al momento ed alla portanza. Individuata su \bar{c} la posizione del baricentro $\bar{x}_{CG} = x_{CG}/\bar{c}$, ovvero il punto G di coordinate (\bar{x}_{CG}, h) , si prende la retta passante per l'origine O e per G (in tratteggio blu). Si individua il punto P_1 , intersezione della retta orizzontale corrispondente al cavolare del C_L del velivolo completo, calcolato dalla 18.5 a fronte e dalla retta baricentrale. A questo punto si traccia l'isoclina passante per P_1 . Si ricorda che la retta isoclina ha pendenza pari:

$$\frac{\Delta C_L}{\Delta C_M} = -\frac{\bar{c}}{l_H} \quad (18.6)$$

Tale retta interseca la curva del momento aerodinamico del velivolo parziale rispetto al bordo d'attacco del profilo, assegnata dal docente, nel punto P_2 , corrispondente alla coppia di valori $(C'_{M_{WB}}, C'_{L_{WB}})$, dove $C'_{M_{WB}}$ è il coefficiente di momento baricentrico del velivolo parziale rispetto al punto A_0 . Per determinare il coefficiente di momento baricentrico del velivolo parziale, è necessario ancora tracciare la retta di equazione $C_L = C_{L_{WB}}$ (parallela all'asse delle ascisse per P_2) e individuare il punto P_3 che essa intercetta sulla retta baricentrale. Si dimostra, infatti, che:

$$C_{M,WB,CG} = x_{P_2} - x_{P_3} \quad (18.7)$$

Tale procedura grafica, si è utilizzata per andare a valutare il carico d'equilibrio, L_{He} tramite la formula 18.4 a pagina 215. Un procedimento possibile di calcolo è riportato nel seguente listato di Matlab.

Listato 18.2 Listato per l'ottenimento del carico del piano di coda tramite il $C_{M_{WB,CG}}$ con relativo carico di Equilibrio.

```
%% Ricerca del Carico in funzione della velocita equivalente per x_cg 25% e 30%
```

```
%Ricerca CM_WBCG (Procedura Grafica)
```

```
% baricentro posto al 25%
```

```
xcg1 = 0.25;
lh1 = 9.20; %[m]
xh1 = 9.6; %[m]
```

```
%Si infittiscono i vettori delle coordinate dei punti del bordo del
%diagramma di manovra
```

```
%Tratto1 (Dal punto di proiezione di H fino a H)
```

```
w_V_H = V_H.*ones(1,30);
w_f_H = linspace(pro_f_H, f_H, length(w_V_H));
```

```
%Tratto2 (Dal punto H al punto A)
```

```
w_V_HA = Velocita_HA_dritto;
w_f_HA = FattCarico_HA_dritto;
```

```
%Tratto3 (Dal punto A al punto B)
```

```
w_V_AB = linspace(V_A, V_B, 60);
w_f_AB = f_A*ones(1, length(w_V_AB));
```

```
%Tratto4 (Dal punto B al punto D)
```

```
w_V_BD = V_D.*ones(1, 60);
w_f_BD = linspace(f_B, f_D, length(w_V_BD));
```

```
%Tratto5 (Dal punto D al punto F)
```

```
w_V_DE = linspace(V_D, V_E, 60);
w_f_DE = interp1([V_D, V_E], [f_D, f_E], w_V_DE, 'linear');
```

```
%Tratto6 (Dal punto E al punto F)
```

```
w_V_EF = linspace(V_E, V_F, 60);
w_f_EF = f_E*ones(1, length(w_V_EF));
```

```
%Tratto7 (Dal punto F al punto G)
```

```
w_V_FG = Velocita_FG_rovescio;
w_f_FG = FattoreCarico_FG_rovescio;
```

```
%Tratto8 (Dal punto G alla proiezione del tratto parabolico per volo rov)
```

```

w_V_G = V_G.*ones(1, 30);
w_f_G = linspace(f_G, pro_f_G, length(w_V_H));
%Assemblaggio di tutte le parti infittite in due vettori w_Vel e w_fatt:
w_Vel = [w_V_H w_V_HA w_V_AB w_V_BD w_V_DE w_V_EF w_V_FG w_V_G];
w_fatt = [w_f_H w_f_HA w_f_AB w_f_BD w_f_DE w_f_EF w_f_FG w_f_G];
%Valuto per tutti i contorni del diagramma di manovra la corrispondente
%pressione dinamica ed il rispettivo coefficiente di portanza.
w_q = 0.5.*rho.*w_Vel.^2;
w_CL = w_fatt.*(W/S)./w_q; %--> Relazione 18.5

% Valuto il Cm_WB_CG a partire dall'andamento del Cm_WB_A' (grafico
% dal docente) mediante procedura grafica.

%Assegno i valori caratteristici della curva assegnata dal docente:
Cm_WB_CG1=[];
CL_WB1=[];
for k = 2:length(w_CL)
C_L=w_CL(k);
C_L_0=-0.30; %Valore del CL quando il CM=0
C_M_0=-0.0515; %Valore del CM quando il CL=0
pend_CM_CL=-C_L_0/C_M_0; %Pendenza del tratto rettilineo curva CM-CL
%Costruzione curva del momento linearizzata:
%Definisco la variabile indipendente:
x_CM_CL = [-0.31;0.16];
%Definisco la variabile dipendente:
y_CM_CL = C_L_0 + pend_CM_CL.*x_CM_CL ;
%Costruzione della retta isoclina:
%Lunghezza in scala della corda media erodinamica:
c_scala=0.34; %fattore di scala
x_cg_scala=c_scala*xcg1;
pend_retta_isolina=-c/lh1; %<-----Relazione (18.6)
%Costruzione della Retta passante per l origine e il baricentro del velivolo
x_retta_G = [0 ; -x_cg_scala];
y_retta_G = [0 ; c_scala] ;
pend_retta_G = diff(y_retta_G)/diff(x_retta_G); % pendenza
% Per la grafica
X_retta_G = [ -0.5 ; 0.5];
Y_retta_G = pend_retta_G.*X_retta_G;
% Retta orizzontale per il valore di CL considerato
x_C_L = [ -0.5 ; 0.5 ];
y_C_L = [ C_L ; C_L] ;
%Valuto l Intersezione fra la retta baricentrale e la retta
% orizzontale per il valore di CL considerato
[xP_I,yP_I] = polyxpoly(x_C_L ,y_C_L ,X_retta_G ,Y_retta_G);
% Isoclina per il punto PI
x_iso = [ -0.5 ; 0.5 ];
y_iso = yP_I + pend_retta_isolina.*(x_iso - xP_I);
%Valuto l Intersezione fra l isoclina e la curva del momento
[xP_II,yP_II] = polyxpoly(x_iso ,y_iso ,x_CM_CL ,y_CM_CL);
% Retta orizzontale per PII
x_PII = [ -0.5 ; 0.5 ];
y_PII = [ yP_II ; yP_II];
%Valuto l Intersezione fra la retta baricentrale e la retta orizzontale
%per PII
[xP_III,yP_III] = polyxpoly(X_retta_G ,Y_retta_G ,x_PII ,y_PII);
% Calcolo del Cm_WB_CG
Cm_WB_CG1(k)= xP_II - xP_III; %<-----ralzione 18.7
CL_WB1(k)= yP_III;
end

%Conoscendo il momento del velivolo parziale rispetto al baricentro,
%si possono andare a valutare i carichi per tutte le condizioni di carico,
%che ho nel diagramma di manovra, per il piano di coda orizzontale.

w_LH_eq_1 =(w_q.*S.*c)./lh1.*Cm_WB_CG1; %<----- relazione 18.4
% Per una rappresentazione grafica

```

```

h2=figure(2)
plot(w_Vel , w_LH_eq_1 , '-k ', 'Linewidth',1.5);
hold on
xlabel('$V_e$, (m/s)$', 'interpreter', 'latex');
ylabel('$L_H$, (N)$', 'interpreter', 'latex');

% Ricerca CM_WBCG (Procedura Grafica)
% baricentro posto al 30%
xcg1 = 0.30;
lh1 = 9.05;
xh1 = 9.45;

%Si infittiscono i vettori delle coordinate dei punti del bordo del
%diagramma di manovra
%Tratto1 (Dal punto di proiezione di H fino a H)
w_V_H = V_H.*ones(1,30);
w_f_H = linspace(pro_f_H, f_H, length(w_V_H));
%Tratto2 (Dal punto H al punto A)
w_V_HA = Velocita_HA_dritto;
w_f_HA = FattCarico_HA_dritto;
%Tratto3 (Dal punto A al punto B)
w_V_AB = linspace(V_A, V_B, 60);
w_f_AB = f_A*ones(1,length(w_V_AB));
%Tratto4 (Dal punto B al punto D)
w_V_BD = V_D.*ones(1, 60);
w_f_BD = linspace(f_B, f_D, length(w_V_BD));
%Tratto5 (Dal punto D al punto F)
w_V_DE = linspace(V_D, V_E, 60);
w_f_DE = interp1([V_D, V_E], [f_D f_E], w_V_DE, 'linear');
%Tratto6 (Dal punto E al punto F)
w_V_EF = linspace(V_E, V_F, 60);
w_f_EF = f_E*ones(1, length(w_V_EF));
%Tratto7 (Dal punto F al punto G)
w_V_FG = Velocita_FG_rovescio;
w_f_FG = FattoreCarico_FG_rovescio;
%Tratto8 (Dal punto G alla proiezione del tratto parabolico per volo rov)
w_V_G = V_G.*ones(1, 30);
w_f_G = linspace(f_G, pro_f_G, length(w_V_H));
%Assemblaggio di tutte le parti infittite in due vettori w_Vel e w_fatt:
w_Vel = [w_V_H w_V_HA w_V_AB w_V_BD w_V_DE w_V_EF w_V_FG w_V_G];
w_fatt = [w_f_H w_f_HA w_f_AB w_f_BD w_f_DE w_f_EF w_f_FG w_f_G];
%Valuto per tutti i contorni del diagramma di manovra la corrispondente
%pressione dinamica ed il rispettivo coefficiente di portanza.
w_q = 0.5.*rho.*w_Vel.^2;
w_CL = w_fatt.*(W/S)./w_q;

% Valuto il Cm_WB_CG a partire dall'andamento del Cm_WB_A' (grafico
% dal docente) mediante procedura grafica.

%Assegno i valori caratteristici della curva assegnata dal docente:
Cm_WB_CG1=[];
CL_WB1=[];
for k = 2:length(w_CL)
C_L=w_CL(k);
C_L_0=-0.30; %Valore del CL quando il CM=0
C_M_0=-0.0515; %Valore del CM quando il CL=0
pend_CM_CL=-C_L_0/C_M_0; %Pendenza del tratto rettilineo curva CM-CL
%Costruzione curva del momento linearizzata:
%Definisco la variabile indipendente:
x_CM_CL = [-0.31;0.16];
%Definisco la variabile dipendente:
y_CM_CL = C_L_0 + pend_CM_CL.*x_CM_CL ;
%Costruzione della retta isoclina:
%Lunghezza in scala della corda media erodinamica:
c_scala=0.34; %fattore di scala
x_cg_scala=c_scala*xcg1;

```

```

pend_retta_isolina=-c/lh1; %<-----Relazione (18.6)
%Costruzione della Retta passante per l origine e il baricentro del velivolo
x_retta_G = [0 ; -x_cg_scala];
y_retta_G = [0 ; c_scala] ;
pend_retta_G = diff(y_retta_G)/diff(x_retta_G); % pendenza
% Per la grafica
X_retta_G = [ -0.5 ; 0.5];
Y_retta_G = pend_retta_G.*X_retta_G;
% Retta orizzontale per il valore di CL considerato
x_C_L = [ -0.5 ; 0.5 ];
y_C_L = [ C_L ; C_L] ;
%Valuto l Intersezione fra la retta baricentrale e la retta
% orizzontale per il valore di CL considerato
[xP_I,yP_I] = polyxpoly(x_C_L ,y_C_L ,X_retta_G ,Y_retta_G);
% Isoclina per il punto PI
x_iso = [ -0.5 ; 0.5 ];
y_iso = yP_I + pend_retta_isolina.*(x_iso - xP_I);
%Valuto l Intersezione fra l isoclina e la curva del momento
[xP_II,yP_II] = polyxpoly(x_iso ,y_iso ,x_CM_CL ,y_CM_CL);
% Retta orizzontale per PII
x_PII = [ -0.5 ; 0.5 ];
y_PII = [ yP_II ; yP_II];
%Valuto l Intersezione fra la retta baricentrale e la retta orizzontale
%per PII
[xP_III,yP_III] = polyxpoly(X_retta_G ,Y_retta_G ,x_PII ,y_PII);
% Calcolo del Cm_WB.CG
Cm_WB.CG1(k)= xP_II - xP_III; %<-----ralzione 18.
CL_WB1(k)= yP_III;
end

%Dopo aver definito il momento del velivolo parziale rispetto al
%baricentro, si possono andare a valutare i carichi per tutte le condizioni
%di carico, che ho nel diagramma di manovra, per il piano di coda
%orizzontale.
w_LH_eq_1 =(w.q.*S.*c)./lh1.*Cm_WB.CG1; %<----- relazione 18.4
% Per una rappresentazione grafica
h2=figure(2)
plot(w_Vel , w_LH_eq_1 , '--k ', 'Linewidth',2.5);
grid minor; hold on
xlabel('$V_e\,(m/s)$', 'interpreter', 'latex');
ylabel('$L_H\,(N)$', 'interpreter', 'latex');
legend('CG @ 25%', 'CG @ 30%', 'Location', 'best')
set(legend, 'color', 'none');

print(h2, '-deps', 'Grafici\CaricoEquilibrio');
print(h2, '-deps', 'C:\Users\Giuseppe\Dropbox\DSV\__template_esercitazioni\
quadernoCarichi\images\CaricoEquilibrio');

```

In particolare si sono valutati i carichi, per baricentro al 25% e al 30%, lungo tutto il diagramma d'involuppo. Per completezza, ne seguente listato è stata usata la funzione:

- `polyxpoly`, funzione che consente di determinare il punto o i punti di intersezione fra due poligonali, in pratica realizzando da un punto di vista numerico quanto si potrebbe eseguire manualmente con penna e righello.

Si può notare dallo script precedente che quanto fatto per il baricentro al 25% della corda media, è stato fatto esattamente per il baricentro posto al 30% della corda media, infatti, lo script precedentemente presentato è riproposto duplicato per determinare i valori dei carichi relativi ai due baricetri.

```

xcg1 = 0.30;
lh1 = 9.05;
xh1 = 9.45;

```

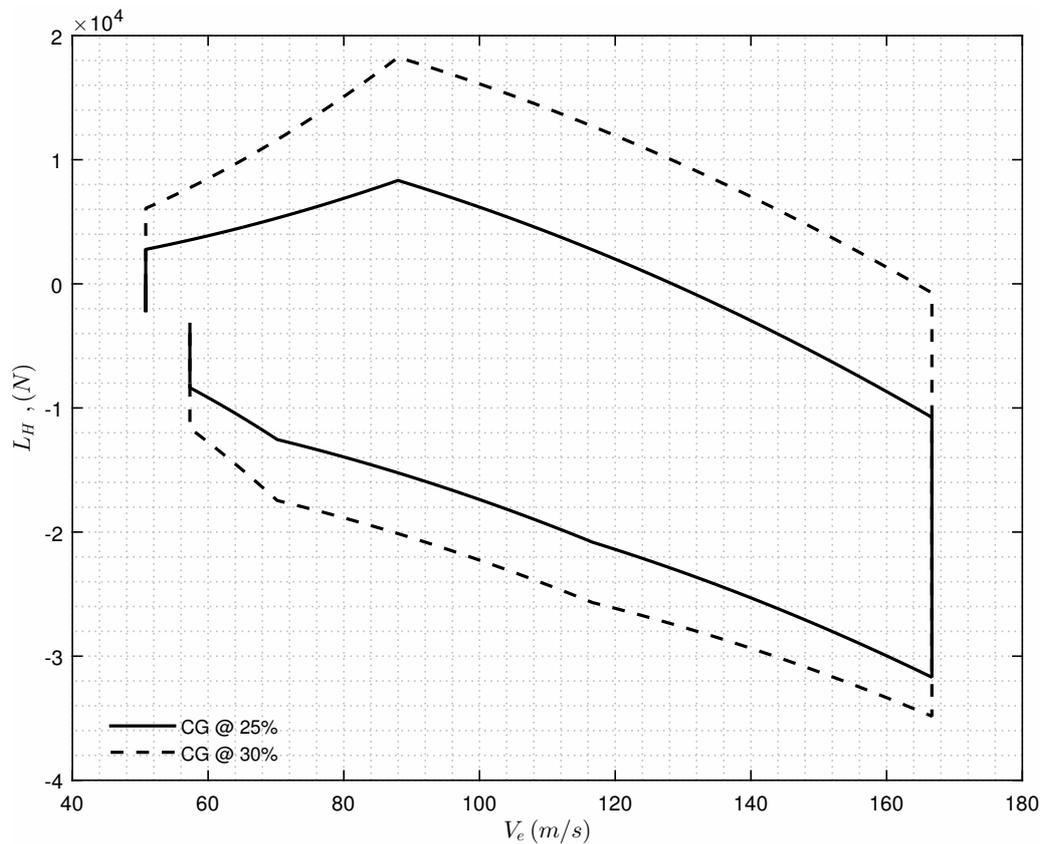


Figura 18.4 Curve del carico aerodinamico di equilibrio del piano di coda orizzontale ottenute al variare della velocità di volo per i punti sul bordo del diagramma di manovra. Il velivolo è un aereo da trasporto a breve raggio, i cui dati sono riportati in tabella 18.1 a pagina 210. Sono rappresentate le curve relative ai due diversi casi di baricentro posto al 25% e al 30% della corda media aerodinamica.

Ottenendo così, anche i carichi corrispondenti al baricentro posto al 30% della corda media aerodinamica. In figura 18.4 è riportato il confronto dei due andamenti, al variare della due posizioni del baricentro considerate.

Per concludere, sono stati tracciati grafici simili relativi alle condizioni di fattore di carico ± 1 . Tali curve, rappresentate nella figura 18.5 a pagina 226, sono utili quando si determinano i carichi di manovra per condizioni di volo complementare. Il listato seguente riporta le istruzioni da aggiungere al codice presentato sopra per ottenere i diagrammi di figura 18.5 a pagina 226.

Listato 18.3 Carico del LH in condizioni di volo fondamentale e fattore di carico ± 1

```
%% Curve del carico LH in condizioni di volo fondamentale e attore di carico
% uguale a +/- 1.
xcg1=0.25;
lh1=9.20;

v_V_HD_dir = [w_V_HA w_V_AB];
v_V_HD_dir = unique(v_V_HD_dir);
%-----
%Nota
%la funzione unique restituisce un vettore, andando ad eliminare tutte le
%ripetizioni presenti all interno del vettore d input
%-----
% Vettore delle velocità da VS1 a VD, volo rovescio
v_V_GD_rov = [w_V_DE w_V_EF w_V_FG];
v_V_GD_rov = unique(v_V_GD_rov);
% si calcolano le pressioni dinamiche
```

```

v_q_f1_dir = 0.5.*rho.*v_V_HD_dir.^2 ;
v_q_f1_rov = 0.5.*rho.*v_V_GD_rov.^2 ;
% e i coefficienti di portanza, a fattore di carico -1 ed 1
v_CL_f1_dir = 1.*(W/S)./v_q_f1_dir ;
v_CL_f1_rov = -1.*(W/S)./v_q_f1_rov ;
%=====
%Valuto il Cm_WB_CG, a partire dall andamento del Cm_WB_A' (grafico fornito
%dal docente) mediante procedura grafica.
%=====
%Assegno i valori caratteristici della curva assengata dal docente:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Volo dritto:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Cm_WB1_CG_f1_dritto=[];
CL_WB1_f1_dritto=[];
for k = 1:length(v_CL_f1_dir)
C_L=v_CL_f1_dir(k);
C_L_0=-0.30; %Valore del CL quando il CM=0
C_M_0=-0.0515; %Valore del CM quando il CL=0
pend_CM_CL=-C_L_0/C_M_0; %Pendenza del tratto rettilineo curva CM-CL
%Costruzione curva del momento linearizzata:
%Definisco la variabile indipendente:
x_CM_CL = [-0.31;0.16];
%Definisco la variabile dipendente:
y_CM_CL = C_L_0 + pend_CM_CL.*x_CM_CL ;
%Costruzione della retta isoclina:
%Lunghezza in scala della corda media erodinamica:
c_scala=0.34; %fattore di scala
x_cg_scala=c_scala*xcg1;
pend_retta_isolina=-c/lh1;
%Costruzione della Retta passante per l origine e il baricentro del velivolo
x_retta_G = [0 ; -x_cg_scala];
y_retta_G = [0 ; c_scala] ;
pend_retta_G = diff(y_retta_G)/diff(x_retta_G); % pendenza
% Per la grafica
X_retta_G = [ -0.5 ; 0.5];
Y_retta_G = pend_retta_G.*X_retta_G;
% Retta orizzontale per il valore di CL considerato
x_C_L = [ -0.5 ; 0.5 ];
y_C_L = [ C_L ; C_L] ;
%Valuto l Intersezione fra la retta baricentrale e la retta
% orizzontale per il valore di CL considerato
[xP_I,yP_I] = polyxpoly(x_C_L ,y_C_L ,X_retta_G ,Y_retta_G);
% Isoclina per il punto PI
x_iso = [ -0.5 ; 0.5 ];
y_iso = yP_I + pend_retta_isolina.*(x_iso - xP_I);
%Valuto l Intersezione fra l isoclina e la curva del momento
[xP_II,yP_II] = polyxpoly(x_iso ,y_iso ,x_CM_CL ,y_CM_CL);
% Retta orizzontale per PII
x_PII = [ -0.5 ; 0.5 ];
y_PII = [ yP_II ; yP_II];
%Valuto l Intersezione fra la retta baricentrale e la retta orizzontale
%per PII
[xP_III,yP_III] = polyxpoly(X_retta_G ,Y_retta_G ,x_PII ,y_PII);
% Calcolo del Cm_WB_CG
Cm_WB1_CG_f1_dritto(k)= xP_II - xP_III; %<-----relazione 18.
CL_WB1_f1_dritto(k)= yP_III;
end
%Dopo aver definito il momento del velivolo parziale rispetto al
%baricentro, si possono andare a valutare i carichi per tutte le condizioni
%di carico, che ho nel diagramma di manovra, per il piano di coda
%orizzontale.
w_LH_eq1_f1_dritto =(v_q_f1_dir.*S.*c)./lh1.*Cm_WB1_CG_f1_dritto; %<-----
relazione 18.4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Volo rovescio:

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Cm_WB1_CG_f2_rovescio=[];
CL_WB1_f2_rovescio=[];
for k = 1:length(v_CL_f1_rov)
C_L=v_CL_f1_rov(k);
C_L_0=-0.30; %Valore del CL quando il CM=0
C_M_0=-0.0515; %Valore del CM quando il CL=0
pend_CM_CL=-C_L_0/C_M_0; %Pendenza del tratto rettilineo curva CM-CL
%Costruzione curva del momento linearizzata:
%Definisco la variabile indipendente:
x_CM_CL = [-0.31;0.16];
%Definisco la variabile dipendente:
y_CM_CL = C_L_0 + pend_CM_CL.*x_CM_CL ;
%Costruzione della retta isoclina:
%Lunghezza in scala della corda media erodinamica:
c_scala=0.34; %fattore di scala
x_cg_scala=c_scala*xcg1;
pend_retta_isolina=-c/lh1;
%Costruzione della Retta passante per l origine e il baricentro del velivolo
x_retta_G = [0 ; -x_cg_scala];
y_retta_G = [0 ; c_scala] ;
pend_retta_G = diff(y_retta_G)/diff(x_retta_G); % pendenza
% Per la grafica
X_retta_G = [ -0.5 ; 0.5];
Y_retta_G = pend_retta_G.*X_retta_G;
% Retta orizzontale per il valore di CL considerato
x_C_L = [ -0.5 ; 0.5 ];
y_C_L = [ C_L ; C_L ] ;
%Valuto l Intersezione fra la retta baricentrale e la retta
% orizzontale per il valore di CL considerato
[xP_I,yP_I] = polyxpoly(x_C_L ,y_C_L ,X_retta_G ,Y_retta_G);
% Isoclina per il punto PI
x_iso = [ -0.5 ; 0.5 ];
y_iso = yP_I + pend_retta_isolina.*(x_iso - xP_I);
%Valuto l Intersezione fra l isoclina e la curva del momento
[xP_II,yP_II] = polyxpoly(x_iso ,y_iso ,x_CM_CL ,y_CM_CL);
% Retta orizzontale per PII
x_PII = [ -0.5 ; 0.5 ];
y_PII = [ yP_II ; yP_II];
%Valuto l Intersezione fra la retta baricentrale e la retta orizzontale
%per PII
[xP_III,yP_III] = polyxpoly(X_retta_G ,Y_retta_G ,x_PII ,y_PII);
% Calcolo del Cm_WB_CG
Cm_WB1_CG_f2_rovescio(k)= xP_II - xP_III; %<-----relazione 18.
CL_WB1_f2_rovescio(k)= yP_III;
end
%Dopo aver definito il momento del velivolo parziale rispetto al
%baricentro, si possono andare a valutare i carichi per tutte le condizioni
%di carico, che ho nel diagramma di manovra, per il piano di coda
%orizzontale.
w_LH_eq1_f2_rovescio =(v_q_f1_rov.*S.*c)./lh1.*Cm_WB1_CG_f2_rovescio; %<-----
relazione 18.4

figure(3)
plot(v_V_HD_dir, w_LH_eq1_f1_diritto,'-k','linewidth',1.5)
hold on
plot(v_V_GD_rov, w_LH_eq1_f2_rovescio,'-.k','linewidth',1.5)
hold on

%%Rieseguo lo script variando la posizione del baricentro
xcg1=0.30;
lh1=9.05;

v_V_HD_dir = [w_V_HA w_V_AB];
v_V_HD_dir = unique(v_V_HD_dir);

```

```

%-----
%Nota
%la funzione unique restituisce un vettore, andando ad eliminare tutte le
%ripetizioni presenti all'interno del vettore d input
%-----
% Vettore delle velocità da VS1 a VD, volo rovescio
v_V_GD_rov = [w_V_DE w_V_EF w_V_FG];
v_V_GD_rov = unique(v_V_GD_rov);
% si calcolano le pressioni dinamiche
v_q_f1_dir = 0.5.*rho.*v_V_HD_dir.^2 ;
v_q_f1_rov = 0.5.*rho.*v_V_GD_rov.^2 ;
% e i coefficienti di portanza, a fattore di carico -1 ed 1
v_CL_f1_dir = 1.*(W/S)./v_q_f1_dir ;
v_CL_f1_rov = -1.*(W/S)./v_q_f1_rov ;
%=====
%Valuto il Cm_WB_CG, a partire dall'andamento del Cm_WB_A' (grafico fornito
%dal docente) mediante procedura grafica.
%=====
%Assegno i valori caratteristici della curva assengata dal docente:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Volo dritto:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Cm_WB1_CG_f1_dritto=[];
CL_WB1_f1_dritto=[];
for k = 1:length(v_CL_f1_dir)
    C_L=v_CL_f1_dir(k);
    C_L_0=-0.30; %Valore del CL quando il CM=0
    C_M_0=-0.0515; %Valore del CM quando il CL=0
    pend_CM_CL=-C_L_0/C_M_0; %Pendenza del tratto rettilineo curva CM-CL
    %Costruzione curva del momento linearizzata:
    %Definisco la variabile indipendente:
    x_CM_CL = [-0.31;0.16];
    %Definisco la variabile dipendente:
    y_CM_CL = C_L_0 + pend_CM_CL.*x_CM_CL ;
    %Costruzione della retta isoclina:
    %Lunghezza in scala della corda media erodinamica:
    c_scala=0.34; %fattore di scala
    x_cg_scala=c_scala*xcg1;
    pend_retta_isolina=-c/lh1;
    %Costruzione della Retta passante per l'origine e il baricentro del velivolo
    x_retta_G = [0 ; -x_cg_scala];
    y_retta_G = [0 ; c_scala] ;
    pend_retta_G = diff(y_retta_G)/diff(x_retta_G); % pendenza
    % Per la grafica
    X_retta_G = [ -0.5 ; 0.5];
    Y_retta_G = pend_retta_G.*X_retta_G;
    % Retta orizzontale per il valore di CL considerato
    x_C_L = [ -0.5 ; 0.5 ];
    y_C_L = [ C_L ; C_L] ;
    %Valuto l'Intersezione fra la retta baricentrale e la retta
    % orizzontale per il valore di CL considerato
    [xP_I,yP_I] = polyxpoly(x_C_L ,y_C_L ,X_retta_G ,Y_retta_G);
    % Isoclina per il punto PI
    x_iso = [ -0.5 ; 0.5 ];
    y_iso = yP_I + pend_retta_isolina.*(x_iso - xP_I);
    %Valuto l'Intersezione fra l'isoclina e la curva del momento
    [xP_II,yP_II] = polyxpoly(x_iso ,y_iso ,x_CM_CL ,y_CM_CL);
    % Retta orizzontale per PII
    x_PII = [ -0.5 ; 0.5 ];
    y_PII = [ yP_II ; yP_II];
    %Valuto l'Intersezione fra la retta baricentrale e la retta orizzontale
    %per PII
    [xP_III,yP_III] = polyxpoly(X_retta_G ,Y_retta_G ,x_PII ,y_PII);
    % Calcolo del Cm_WB_CG
    Cm_WB1_CG_f1_dritto(k)= xP_II - xP_III; %<-----razione 18.
    CL_WB1_f1_dritto(k)= yP_III;

```

```

end
%Dopo aver definito il momento del velivolo parziale rispetto al
%baricentro, si possono andare a valutare i carichi per tutte le condizioni
%di carico, che ho nel diagramma di manovra, per il piano di coda
%orizzontale.
w_LH_eq1_f1_diritto =(v_q_f1_dir.*S.*c)./lh1.*Cm_WB1_CG_f1_dritto; %<-----
    relazione 18.4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Volo rovescio:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Cm_WB1_CG_f2_rovescio=[];
CL_WB1_f2_rovescio=[];
for k = 1:length(v_CL_f1_rov)
C_L=v_CL_f1_rov(k);
C_L_0=-0.30; %Valore del CL quando il CM=0
C_M_0=-0.0515; %Valore del CM quando il CL=0
pend_CM_CL=-C_L_0/C_M_0; %Pendenza del tratto rettilineo curva CM-CL
%Costruzione curva del momento linearizzata:
%Definisco la variabile indipendente:
x_CM_CL = [-0.31;0.16];
%Definisco la variabile dipendente:
y_CM_CL = C_L_0 + pend_CM_CL.*x_CM_CL ;
%Costruzione della retta isoclina:
%Lunghezza in scala della corda media erodinamica:
c_scala=0.34; %fattore di scala
x_cg_scala=c_scala*xcg1;
pend_retta_isolina=-c/lh1;
%Costruzione della Retta passante per l origine e il baricentro del velivolo
x_retta_G = [0 ; -x_cg_scala];
y_retta_G = [0 ; c_scala] ;
pend_retta_G = diff(y_retta_G)/diff(x_retta_G); % pendenza
% Per la grafica
X_retta_G = [ -0.5 ; 0.5];
Y_retta_G = pend_retta_G.*X_retta_G;
% Retta orizzontale per il valore di CL considerato
x_C_L = [ -0.5 ; 0.5 ];
y_C_L = [ C_L ; C_L] ;
%Valuto l Intersezione fra la retta baricentrale e la retta
% orizzontale per il valore di CL considerato
[xP_I,yP_I] = polyxpoly(x_C_L ,y_C_L ,X_retta_G ,Y_retta_G);
% Isoclina per il punto PI
x_iso = [ -0.5 ; 0.5 ];
y_iso = yP_I + pend_retta_isolina.*(x_iso - xP_I);
%Valuto l Intersezione fra l isoclina e la curva del momento
[xP_II,yP_II] = polyxpoly(x_iso ,y_iso ,x_CM_CL ,y_CM_CL);
% Retta orizzontale per PII
x_PII = [ -0.5 ; 0.5 ];
y_PII = [ yP_II ; yP_II];
%Valuto l Intersezione fra la retta baricentrale e la retta orizzontale
%per PII
[xP_III,yP_III] = polyxpoly(X_retta_G ,Y_retta_G ,x_PII ,y_PII);
% Calcolo del Cm_WB1_CG
Cm_WB1_CG_f2_rovescio(k)= xP_II - xP_III; %<-----ralzione 18.
CL_WB1_f2_rovescio(k)= yP_III;
end
%Dopo aver definito il momento del velivolo parziale rispetto al
%baricentro, si possono andare a valutare i carichi per tutte le condizioni
%di carico, che ho nel diagramma di manovra, per il piano di coda
%orizzontale.
w_LH_eq1_f2_rovescio =(v_q_f1_rov.*S.*c)./lh1.*Cm_WB1_CG_f2_rovescio; %<-----
    relazione 18.4

h3=figure(3)
plot(v_V_HD_dir, w_LH_eq1_f1_diritto,':k','linewidth',1.5)
hold on
plot(v_V_GD_rov, w_LH_eq1_f2_rovescio,'--k','linewidth',1.5);

```

```

hold on
legend('n = 1 CG @ 25%', 'n = - 1 CG @ 25%', 'n = 1 CG @ 30%', 'n = - 1 CG @ 30%', '
      Location', 'best')
grid minor

print(h3, '-deps', 'Grafici\CaricoEquilibrioFattoreCaricoCostante');
print(h3, '-deps', 'C:\Users\Giuseppe\Dropbox\DSV\__template_esercitazioni\
      quadernoCarichi\images\CaricoEquilibrioFattoreCaricoCostante');

```

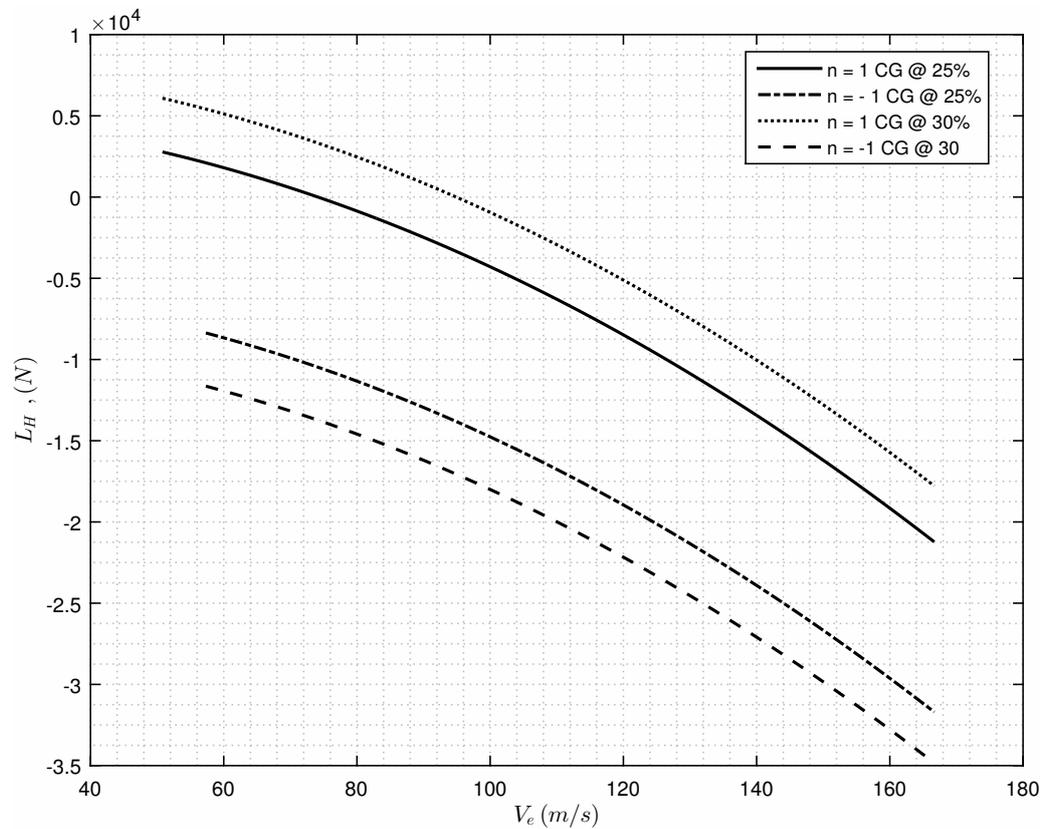


Figura 18.5 Curve del carico aerodinamico di equilibrio del piano di coda orizzontale ottenute al variare della velocità di volo equivalente per le condizioni di fattore di carico ± 1 . Il velivolo è un aereo da trasporto a breve raggio, i cui dati sono riportati in tabella 18.1 a pagina 210. Sono rappresentate le curve relative ai due diversi casi di baricentro posto al 25% e al 30% della corda media aerodinamica.

Data logging delle variabili si stato

Indice

19.1 Introduzione	227
19.2 Decollo	229
19.3 Crociera 1	231
19.4 Crociera 2	233
19.5 Atterraggio	235

19.1 Introduzione

Si è pensato che potesse essere istruttivo eseguire delle acquisizioni dati delle variabili di stato di un sistema dinamico.

Lo strumento utilizzato per l'acquisizione dati è stato uno smartphone il quale mediante un'applicazione esterna **SensorLog**, ha consentito di registrare molteplici parametri. I file contenenti i dati dell'acquisizione sono stati forniti dall'applicativo mediante estensione file.csv (*comma separated value*).

La campagna di acquisizione dati è stata condotta su di un velivolo durante le fasi salienti della missione di volo. Si è, quindi, ritenuto opportuno riportare le fasi del decollo, crociera e atterraggio.

Data la complessità del fenomeno analizzato, si è provato a dare delle risposte qualitative a quanto registrato dai vari sensori.

Nell'immagine 19.17 a pagina 236 è possibile vedere lo smartphone come è stato utilizzato durante l'acquisizione dati.

Il velivolo sul quale è stata condotta la campagna di acquisizione dati è un Airbus A321 della compagnia EasyJet in servizio durante il volo del 20 Novembre 2016 in partenza da Amsterdam e in arrivo a Napoli.

Il sedile utilizzato per fissare lo smartphone si approssimativamente in una zona centrale della fusoliera, esattamente sulla fila 17 posto C.

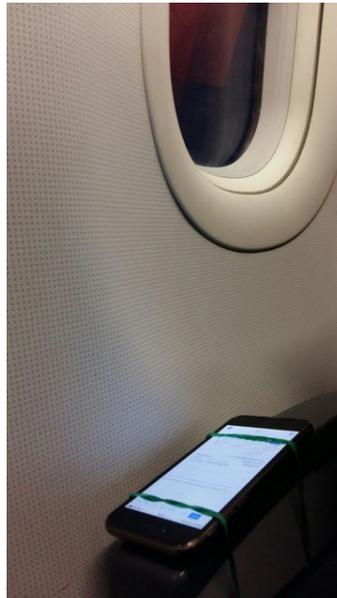


Figura 19.1 Posizionamento dello smartphone in fusoliera, in una zona a ridosso dell'ala

Lo strumento per com'è posizionato è stato considerato un corpo solidale al velivolo, quindi dotato di nessun moto relativo rispetto al velivolo.

Tra i molteplici parametri registrati dallo smartphone, quelli ritenuti di maggiore interesse per la nostra analisi dinamica sono:

- **loggingTime** e **loggingSample** due colonne contenenti data e ora di acquisizione e una colonna composta da numeri interi con ordine crescente (il **loggingSample** è stato utilizzato come indicatore dello step temporale);
- **locationLatitude**, **locationLongitude** e **locationAltitude** dati contenenti i valori di latitudine, longitudine e altitudine;
- **locationSpeed** contenente i dati di velocità;
- **accelerometerAccelerationXG**, **accelerometerAccelerationYG** e **accelerometerAccelerationZG** contenenti i valori delle componenti del vettore accelerazione;
- **motionYawrad**, **motionPitchrad** e **motionRollrad** contenenti i valori degli angoli in radianti di yaw, pitch e roll;
- **motionRotationRateXrads**, **motionRotationRateYrads** e **motionRotationRateZrads** contenenti i valori delle variazioni degli angoli ed espressi in rad/s;
- **motionQuaternionXR**, **motionQuaternionYR**, **motionQuaternionZR** e **motionQuaternionWR** contenenti i valori del quaternione dell'orientamento.

In figura 19.18 a pagina 237 sono stati registrati le coordinate di latitudine e longitudine dei punti in cui sono state condotte le 4 campagne di acquisizione dati.

19.2 Decollo

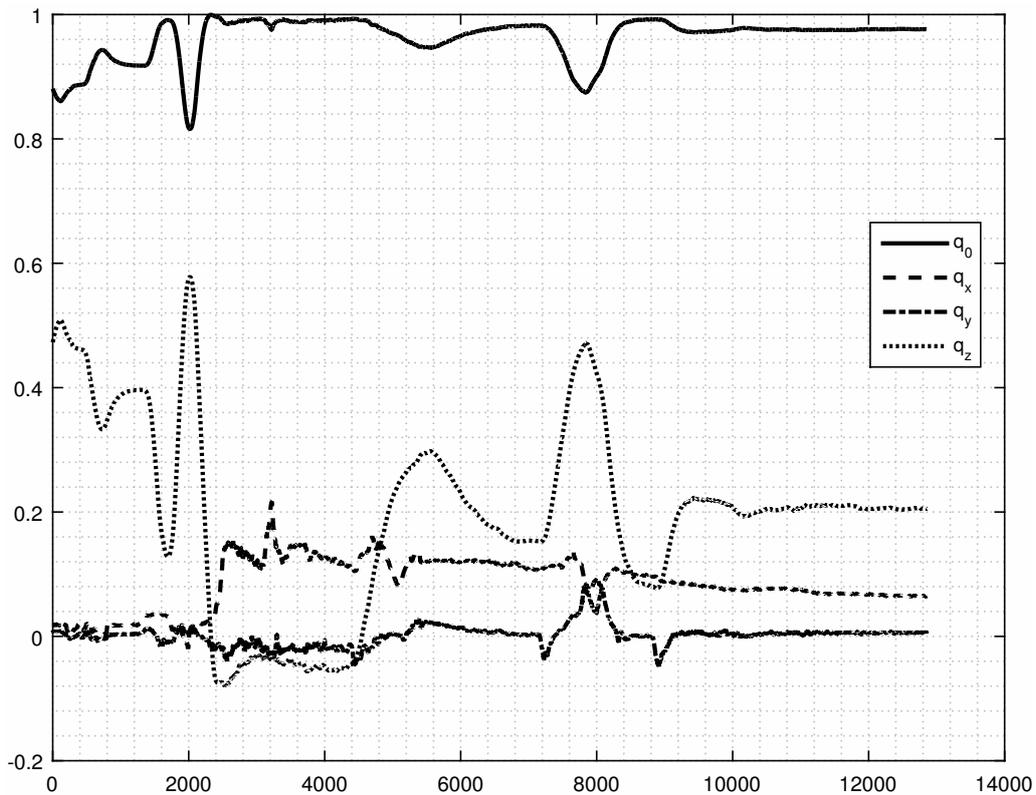


Figura 19.2 Time Histories delle variabili q_0 , q_x , q_y e q_z

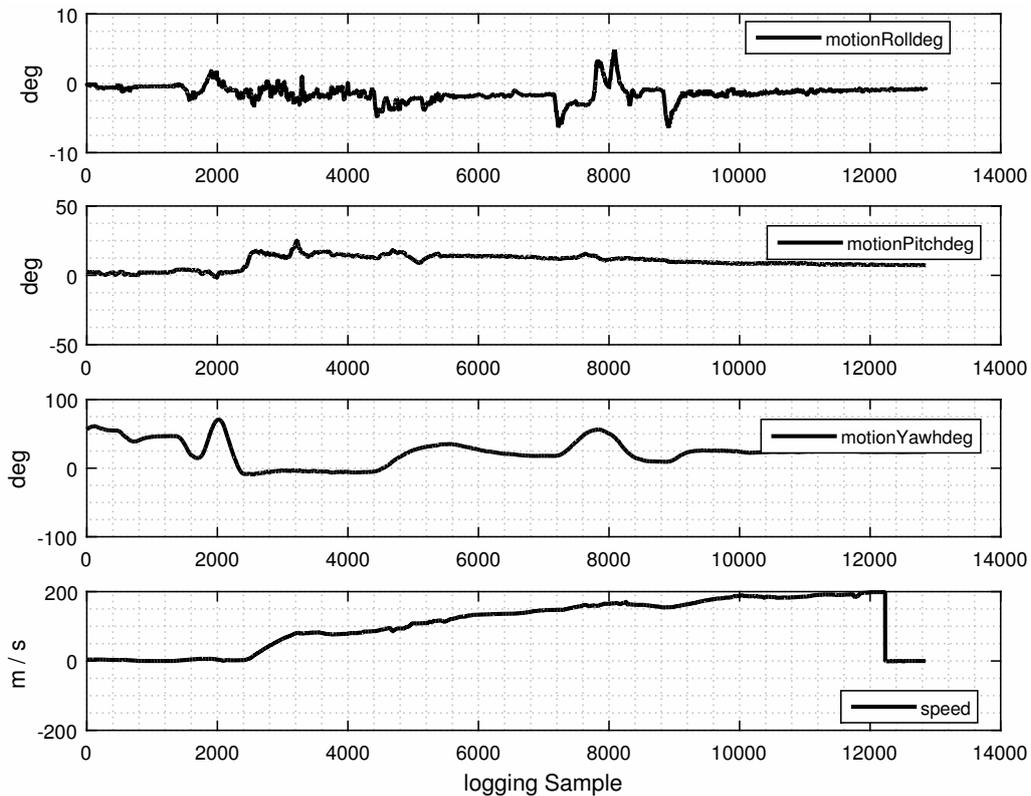


Figura 19.3 Time Histories delle variabili Yaw, Pitch, Roll e Speed.

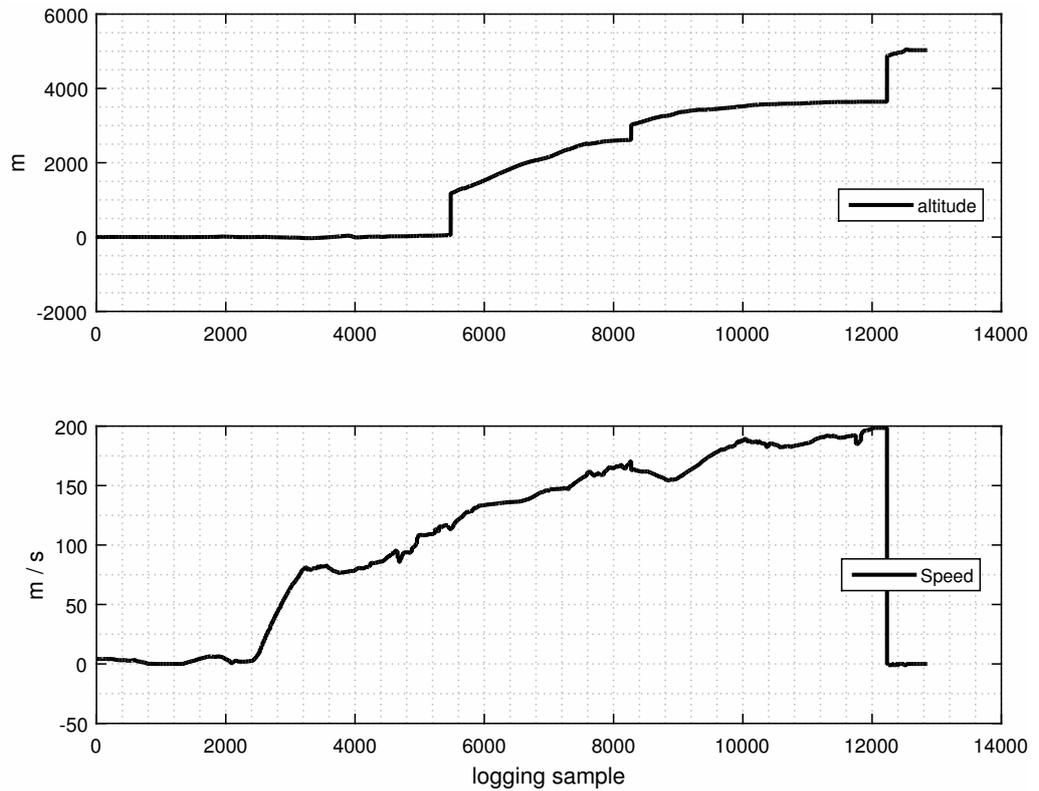


Figura 19.4 Time Histories: Altitude.

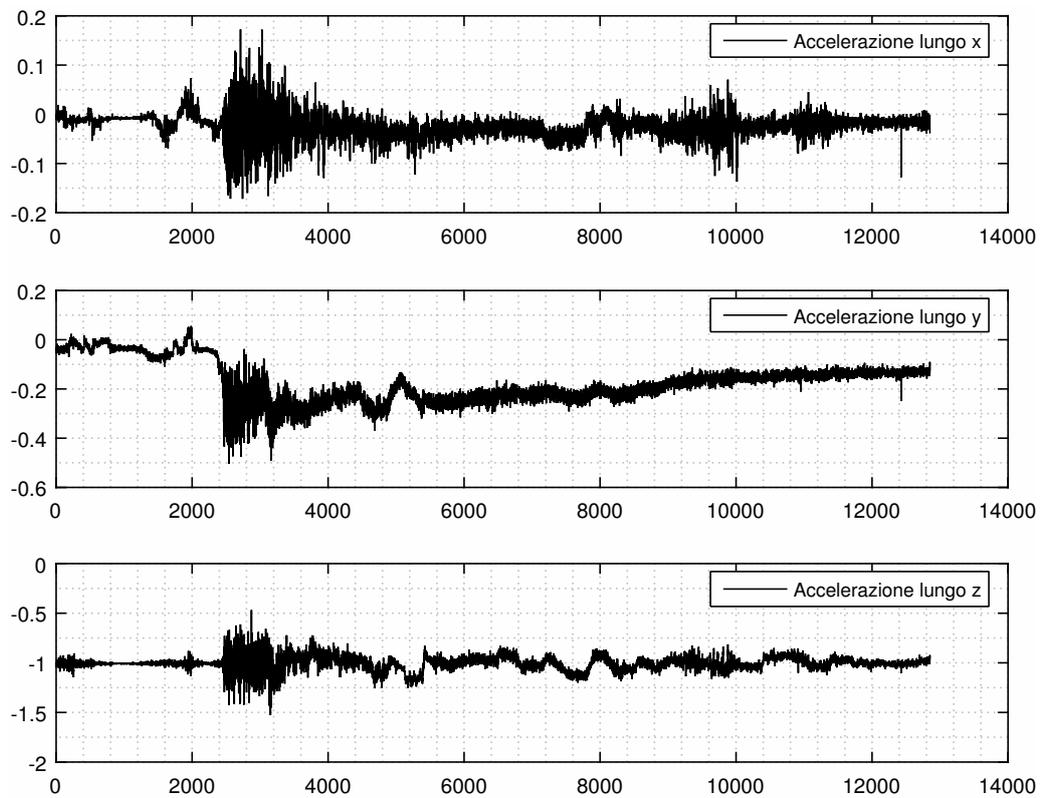


Figura 19.5 Time Histories: Accelerazioni lungo x, y e z, m/s^2 .

19.3 Crociera 1

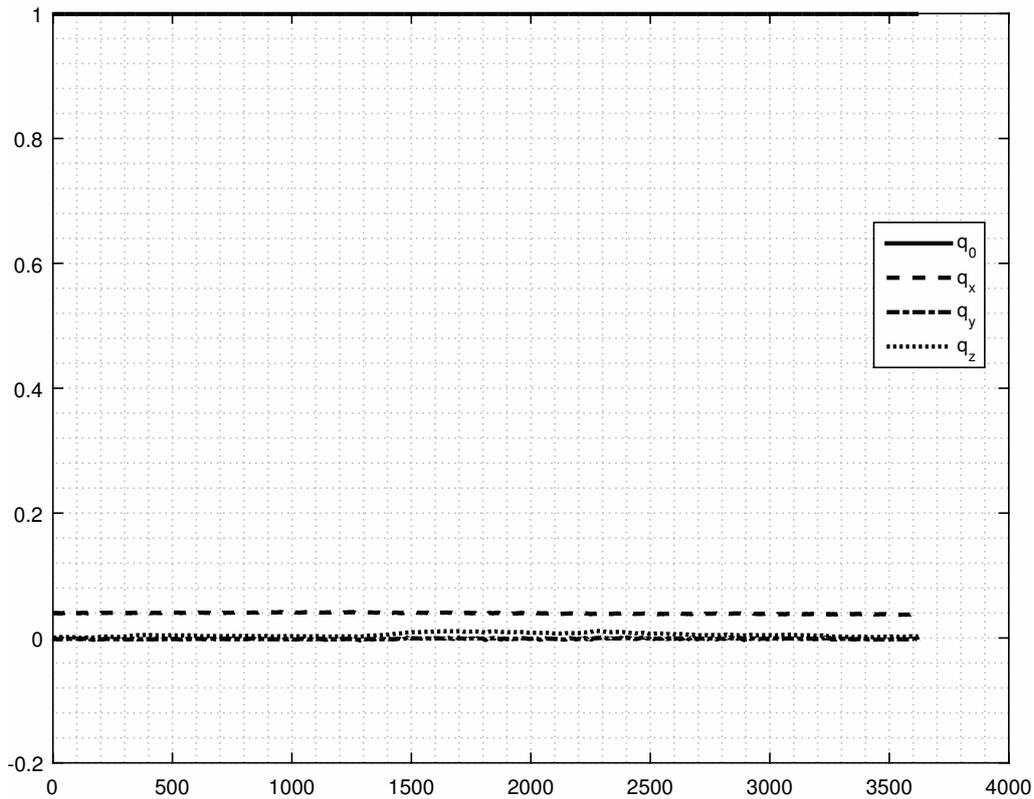


Figura 19.6 Time Histories delle variabili q_0 , q_x , q_y e q_z

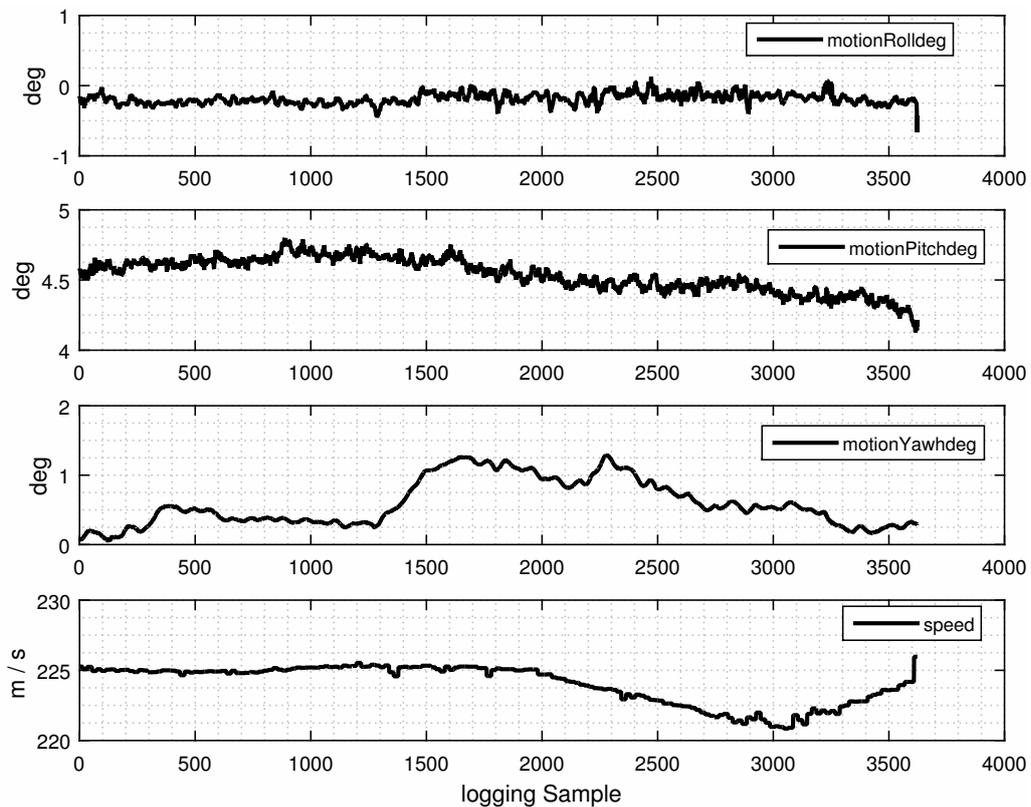


Figura 19.7 Time Histories delle variabili Yaw, Pitch, Roll e Speed.

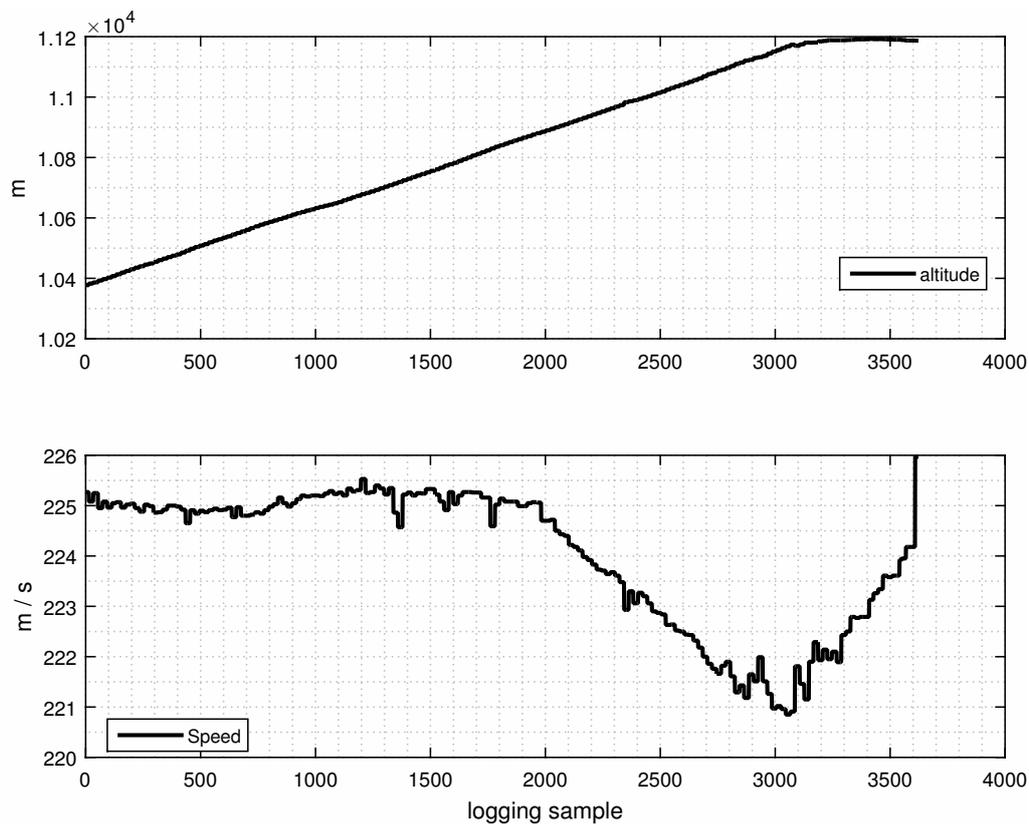


Figura 19.8 Time Histories: Altitude.

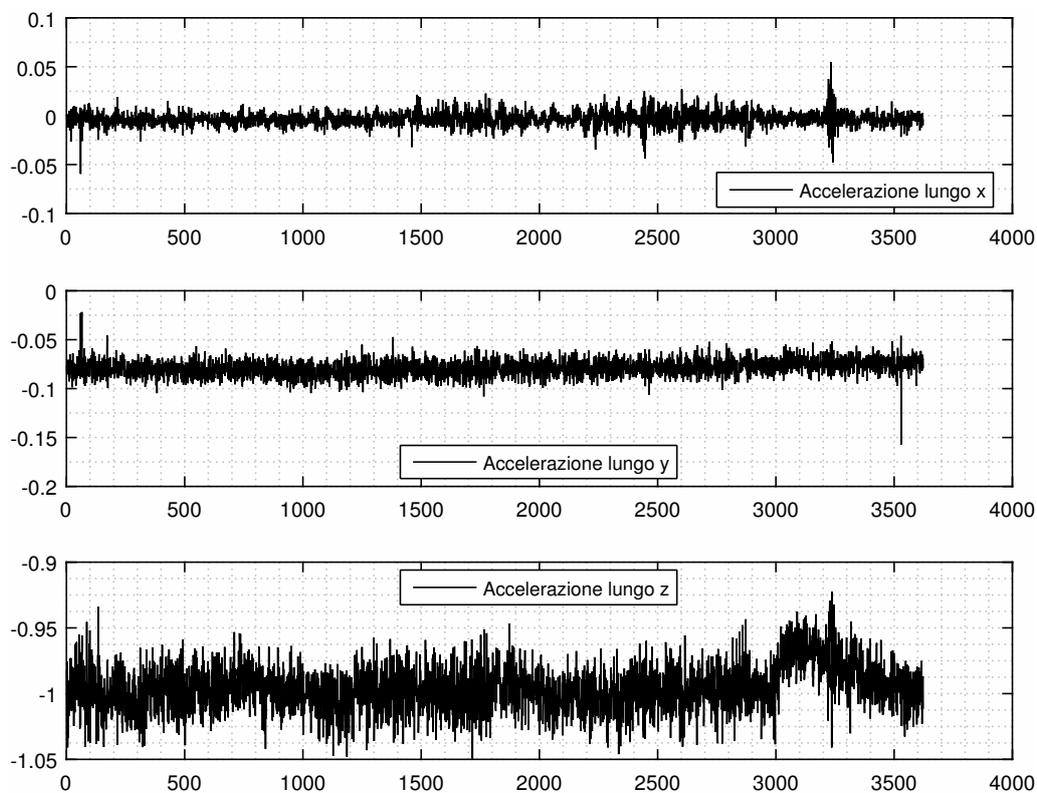


Figura 19.9 Time Histories: Accelerazioni lungo x, y e z, m/s^2 .

19.4 Crociera 2

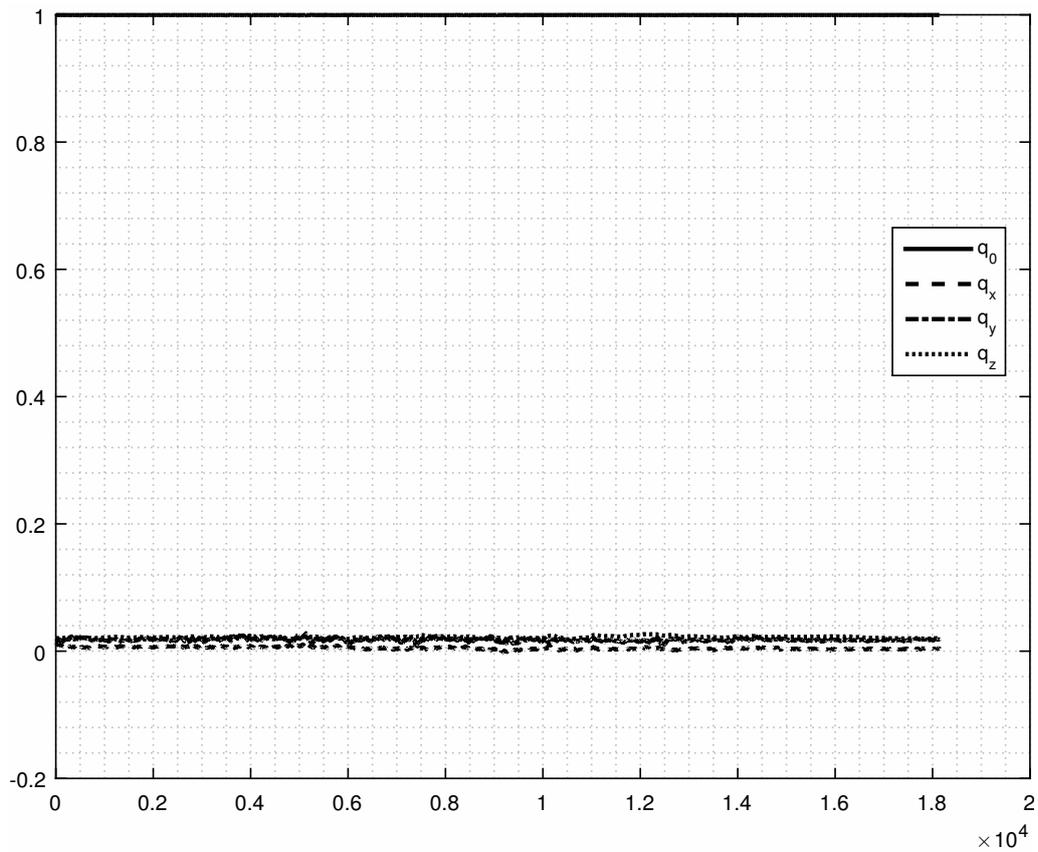


Figura 19.10 Time Histories delle variabili q_0 , q_x , q_y e q_z

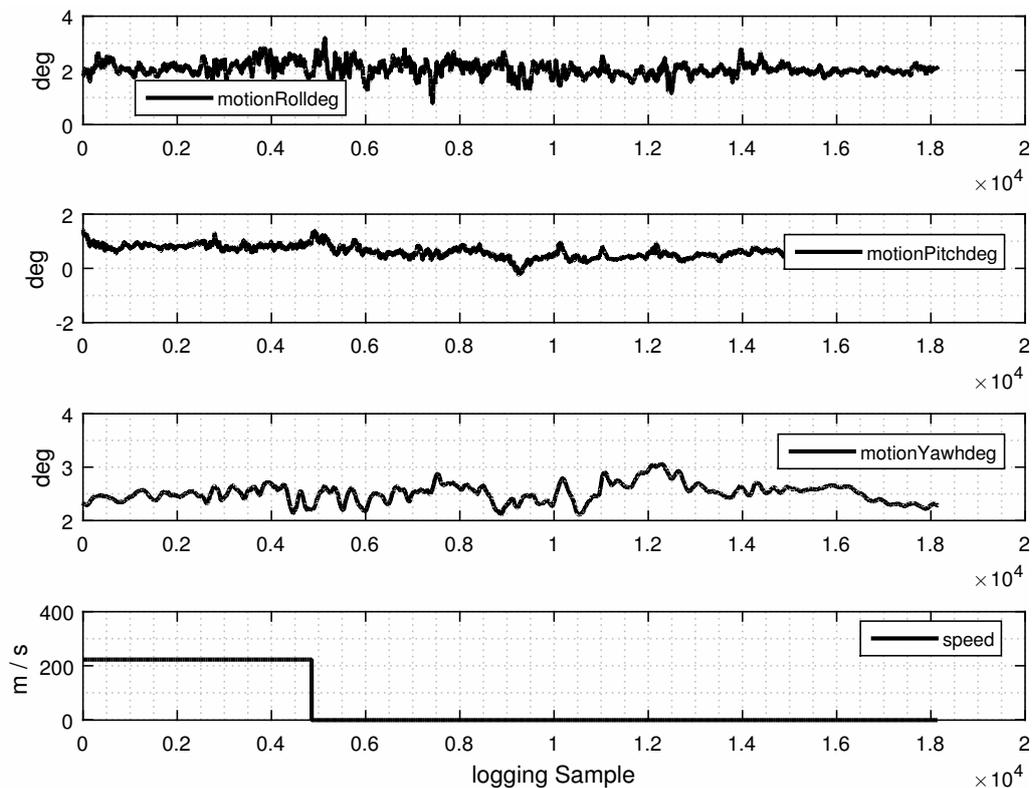


Figura 19.11 Time Histories delle variabili Yaw, Pitch, Roll e Speed.

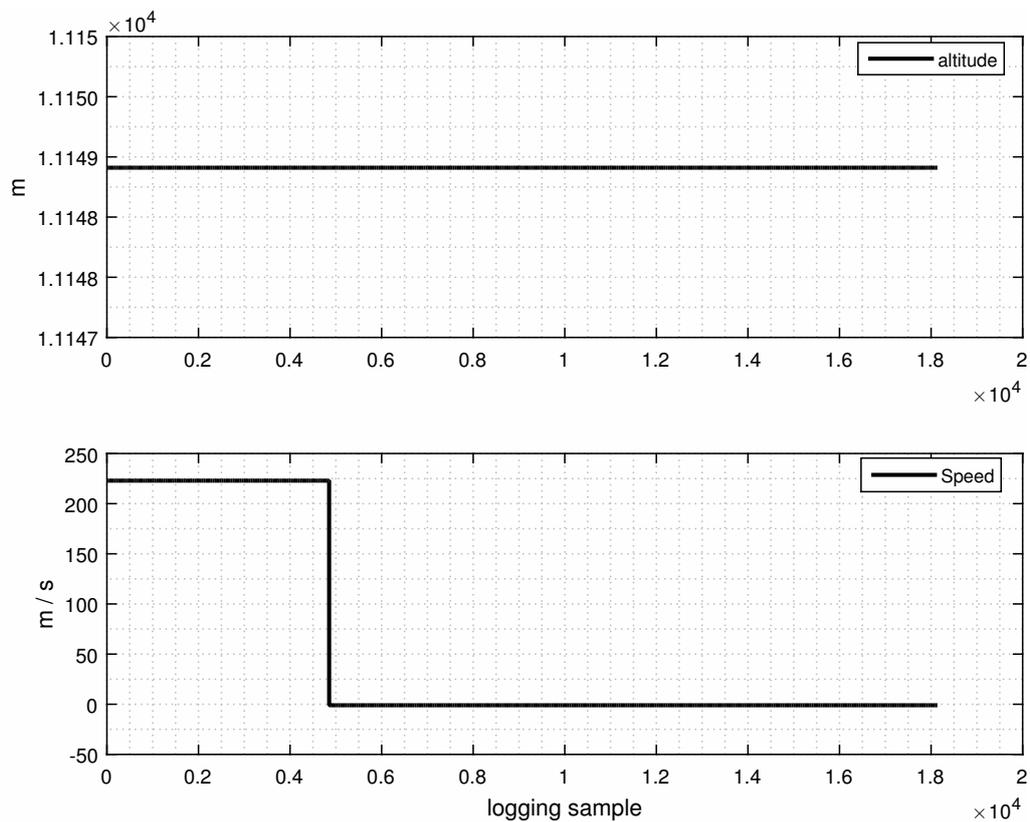
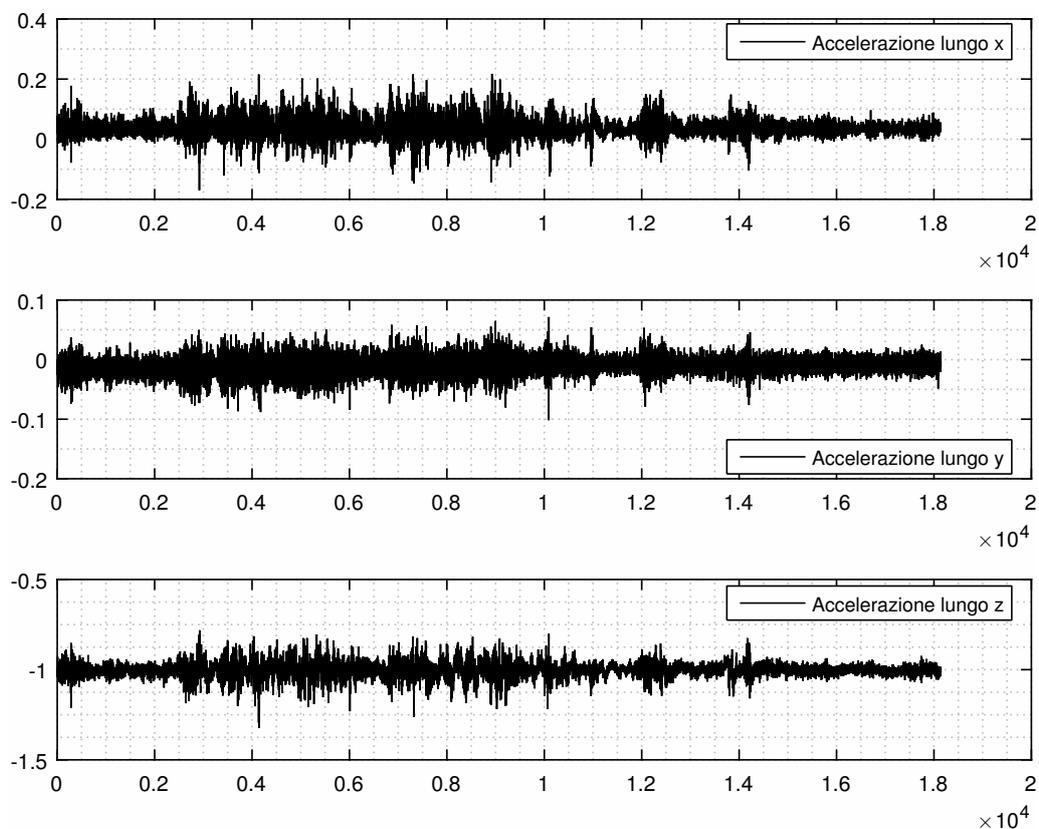


Figura 19.12 Time Histories: Altitude.

Figura 19.13 Time Histories: Accelerazioni lungo x, y e z, m/s^2 .

19.5 Atterraggio

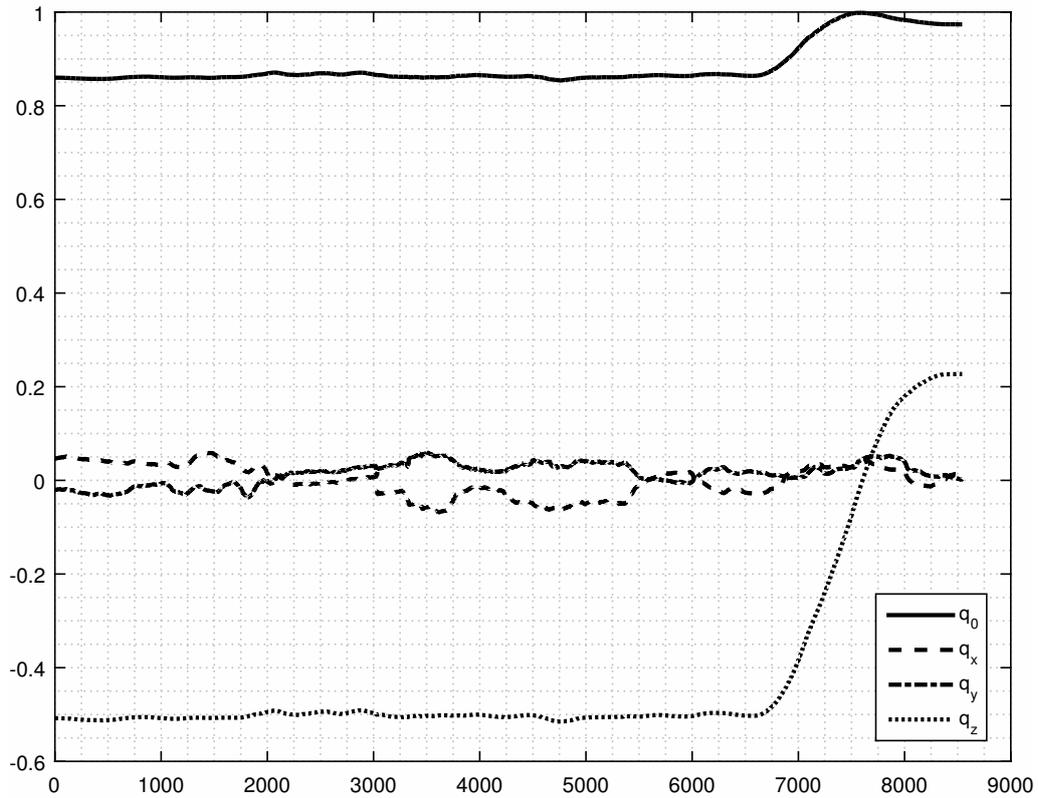


Figura 19.14 Time Histories delle variabili q_0 , q_x , q_y e q_z

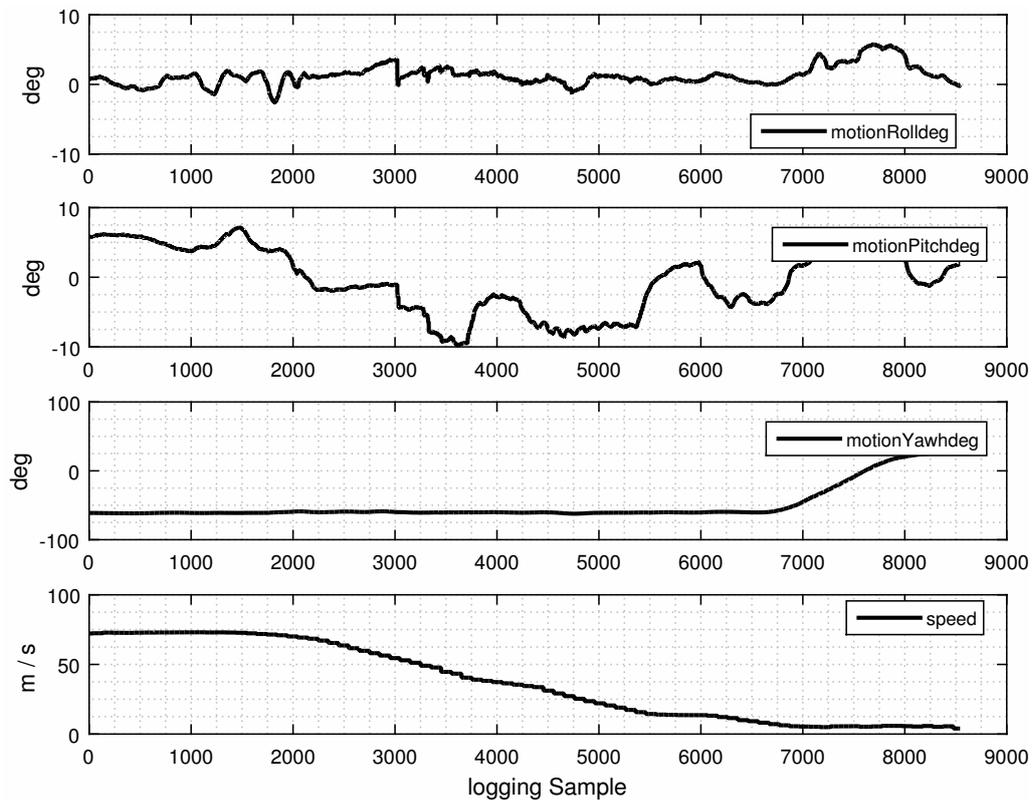


Figura 19.15 Time Histories delle variabili Yaw, Pitch, Roll e Speed.

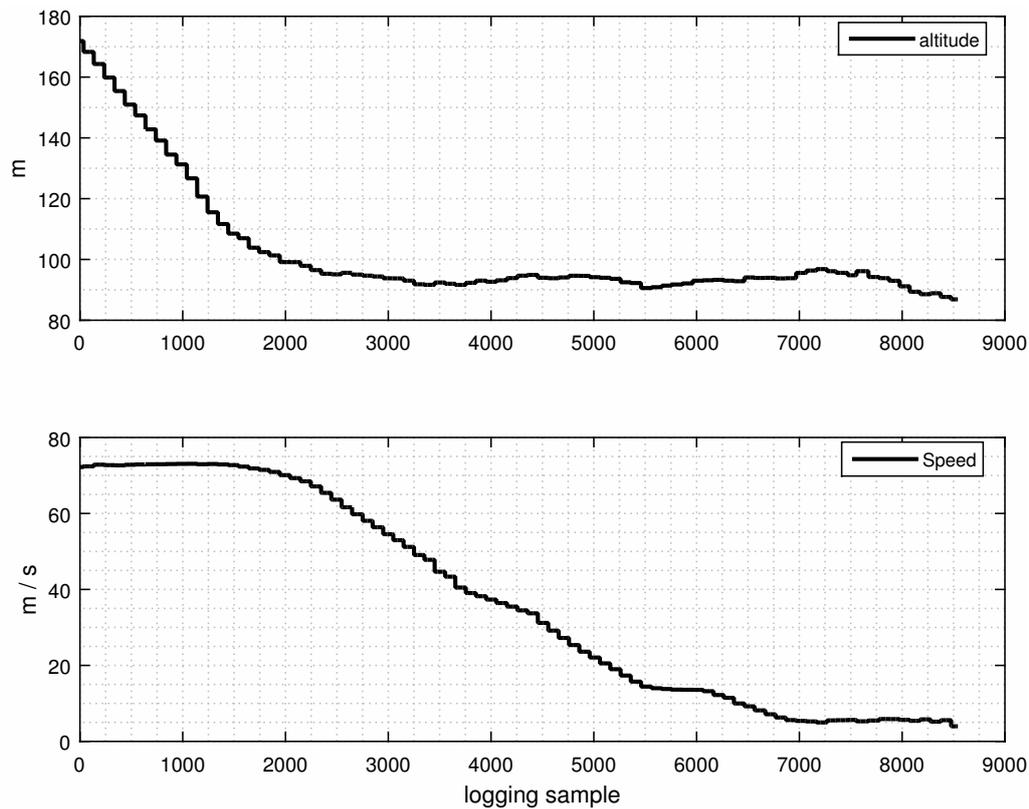


Figura 19.16 Time Histories: Altitude.

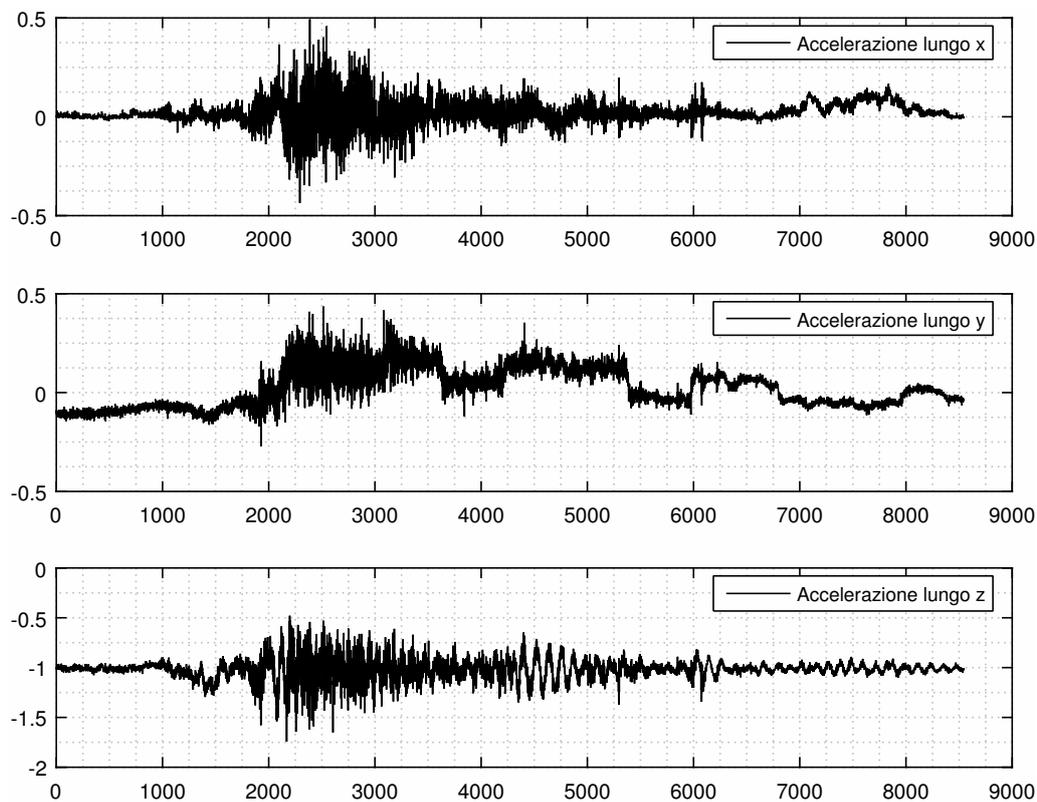


Figura 19.17 Time Histories: Accelerazioni lungo x, y e z, m/s^2 .



Figura 19.18 Punti in cui sono state fatte le acquisizioni dati.

