

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

- Polo di Forlì -

*CORSO DI LAUREA MAGISTRALE
IN INGEGNERIA MECCANICA
Classe 8202*

TESI DI LAUREA

in
Controlli automatici

**PROGETTO DI UNA LEGGE DI CONTROLLO PER LA
COMPENSAZIONE DEL VENTO IN UN QUADRICOTTERO**

CANDIDATO

Luca Fiumana

RELATORE:

Prof. Matteo Zanzi

Anno Accademico 2014/2015

Sessione III

Sommario

1. CENNI STORICI	7
2. CLASSIFICAZIONE DEI DRONI.....	9
3. STRUTTURA DEL QUADRICOTTERO.....	11
4. MECCANICA DI VOLO.....	17
5. ASSETTO DI UN CORPO RIGIDO	19
5.1. ANGOLI DI EULERO.....	22
5.2. VELOCITA' NEL SISTEMA BODY CON ANGOLI DI EULERO.....	24
5.3. EQUAZIONI STANDARD DEL VOLO.....	24
6. BARICENTRO E MATRICE DI INERZIA	28
6.1. CALCOLO BARICENTRO	29
6.2. CALCOLO MATRICE DI INERZIA.....	29
7. FORZE E MOMENTI APPLICATI AL MODELLO	31
7.1. VARIABILI DI STATO E INGRESSI DEL SISTEMA.....	33
7.2. SCOMPOSIZIONE DELLE FORZE E DEI MOMENTI.....	34
7.3. LEGAME TRA VELOCITÀ ANGOLARE E ROLL-PITCH-YAW	35
7.4. VELOCITA' ANGOLARE IN CASO DI HOVERING	38
8. LINEARIZZAZIONE	39
8.1. MODELLO LINEARIZZATO CASO HOVERING	41
8.2. MODELLO LINEARIZZATO IN FORMA MATRICIALE	42
9. CONTROLLO OTTIMO LINEARE QUADRATICO (LQ)	46
9.1. CONTROLLO OTTIMO LQ A TEMPO INFINITO CON SET-POINT DIVERSO DALL'ORIGINE	48
9.3. COMPILAZIONE MATRICI Q ED R.....	50
10. IMPLEMENTAZIONE DEL MODELLO IN MATLAB	52
10.1. SVILUPPO DEL MODELLO LINEARE	53
10.2. SVILUPPO DEL MODELLO REALE.....	60
10.3. TEST E VERIFICA DEL MODELLO.....	66
11. MODELLO LINEARIZZATO CASO MOTO RETTILINEO	72
11.1. CASO MOTO RETTILINEO LUNGO L'ASSE X.....	72
11.2. CASO MOTO RETTILINEO LUNGO L'ASSE Y.....	74
11.3. CASO MOTO RETTILINEO LUNGO L'ASSE Z.....	75
11.4. IMPLEMENTAZIONE CASO MOTO RETTILINEO.....	77
11.5. BLOCCO PER IL CALCOLO DELLE FORZE DOVUTE AL VENTO	79

11.6	BLOCCO PER IL CALCOLO DELLE COPPIE DOVUTE AL VENTO	80
11.7	BLOCCO PER IL CALCOLO DEL FEEDBACK	81
11.8	BLOCCO PER IL CALCOLO DELLA FORZA DI THRUST PER IL MOTO COSTANTE DEL DRONE	82
11.9	BLOCCO PER IL CALCOLO DELLE VELOCITÀ ANGOLARI PER IL MOTO UNIFORME	84
11.10	SCRIPT PER DEFINIRE LE CONDIZIONI INIZIALI O DI TRACKING	85
11.11	TEST DEL MODELLO REALE CON MOTO UNIFORME	86
11.12	COMPILAZIONE MATRICI R E Q.....	94
12.	METODO DELLE MATRICI ESTESE	95
12.1	IMPLEMENTAZIONE IN SIMULINK	97
12.2.	MODELLO NON COMPLETAMENTE CONTROLLABILE.....	99
12.3.	IMPLEMENTAZIONE IN MATLAB DEL MODELLO NON COMPLETAMENTE CONTROLLABILE	101
12.4	VERIFICA DELMODELLO NON COMPLETAMENTE CONTROLLABILE	103
12.5	RICHIAMI DI CONTROLLI AUTOMATICI, TIPO DI UN SISTEMA.....	105
12.6	ESPANSIONE DEL MODELLO NON COMPLETAMENTE CONTROLLABILE CON GLI INTEGRALI DELLE POSIZIONI	107
12.8	TEST E VERIFICA DEL MODELLO NON COMPLETAMENTE CONTROLLABILE	110
12.9	COMPILAZIONE MATRICI R E Q.....	113
13	IL MODELLO REALE COMPLETO.....	115
13.1	SCRITTURA DELLO SCRIPT PER LA COMPILAZIONE DELLA MATRICE K	115
13.2	IMPLEMENTAZIONE IN SIMULINK	119
13.3	TEST E VERIFICA DEL MODELLO COMPLETO	120
13.4	SVILUPPO DEL SISTEMA DI TRACKING	122
13.5	TEST E VERIFICA DEL MODELLO COMPLETO CON SISTEMA DI TRACKING	124
14.	SVILUPPO DELLE FORZE AERODINAMICHE.....	128
14.1	CREAZIONE DEL MODELLO IN SIMULINK.....	133
14.2	TEST E VERIFICA DEL MODELLO AERODINAMICO	135
15.	DISCRETIZZAZIONE DEL MODELLO REALE	138
15.1	DISPOSITIVI PER L'INTERFACCIA DIGITALE E ANALOGICA	140
15.2	VANTAGGI E SVANTAGGI DI UN CONTROLLO IN DIGITALE	141
15.3	IMPLEMENTAZIONE IN MATLAB.....	142
15.4	VERIFICA E TEST DEL MODELLO DISCRETIZZATO.....	145
16.	MODELLO INERZIALE	148
16.1	ESTENSIONE DEL MODELLO MATEMATICO.....	150
16.2	COMPILAZIONI MATRICI R E Q	153
16.3	IMPLEMENTAZIONE DEL MODELLO INERZIALE	154

16.4 TEST E VERIFICA DEL MODELLO INERZIALE	156
17. CONSIDERAZIONI CONCLUSIONI E SVILUPPI FUTURI	158
18. BIBLIOGRAFIA	160
19. SITOGRAFIA.....	161

1. CENNI STORICI

“Trovo, se questo strumento a vite sarà ben fatto, cioè fatto di tela lina, stopata i suoi pori con amido, e svoltata con prestezza, che detta vite si fa la femmina nell’aria e monterà in alto”

Leonardo Da Vinci, Manoscritto B, foglio 83 v., 1483 – 1486

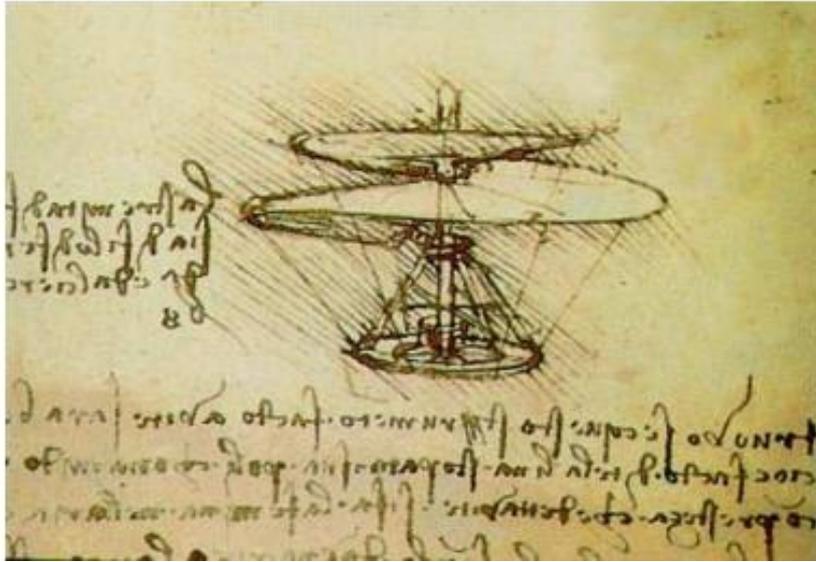


Figura 1.1 progetto elicottero Leonardo Da Vinci

Il primo concetto di macchina volante risale a Leonardo Da Vinci, con il suo elicottero a vite, ma dovettero trascorrere secoli prima che si potesse passare da questo disegno un po' goffo, pesante e poco funzionante ad immaginare una macchina tecnologicamente avanzata, compatta e priva di conducente, capace di solcare i cieli con agilità e sicurezza.

Il drone, noto con l'acronimo APR (aeromobile a pilotaggio remoto) è un velivolo caratterizzato dall'assenza di pilota a bordo. La sua prima comparsa avvenne nel 1849 quando gli Austriaci attaccarono la città di Venezia legando cariche esplosive a palloni aerostatici, i quali, oltre a danneggiare le linee nemiche, hanno compromesso, a causa di un colpo di vento, anche le proprie truppe.

Il primo prototipo di APR a radio controllo venne prodotto durante la prima guerra mondiale nel 1916 noto con il nome di Aerial Target, questo fu la prima dimostrazione di pilotaggio a distanza tramite una piattaforma che montava un sistema di giroscopi.

Nel corso della seconda guerra mondiale, negli USA, i droni vennero prodotti in larga scala (circa 15.000 modelli), mentre è nel periodo della guerra fredda che le dimensioni iniziarono a ridursi con l'uso di tecnologie mano a mano sempre più spinte.

Per quanto riguarda l'Italia, i droni fecero comparsa durante gli anni sessanta, utilizzati dall'esercito e prodotti dalla Canadair.

2. CLASSIFICAZIONE DEI DRONI

nella tabella di figura 2.1 sono classificati droni secondo le dimensioni, da cui poi si estrapolano i relativi dati operativi di utilizzo.

Categoria	Acronimo	Raggio operativo [km]	Quota di volo [m]	Durata del volo [h]	MTOW [kg]	Operativo
Tactical UAV						
Nano	η	< 1	100	< 1	< 0,0250	Sì
Micro	μ	< 10	250	1	< 5	Sì
Mini	Mini	< 10	150 - 300	< 2	< 30	Sì
Close Range	CR	10 - 30	3 000	2 - 4	150	Sì
Short Range	SR	30 - 70	3 000	3 - 6	200	Sì
Medium Range	MR	70 - 200	5 000	6 - 10	1 250	Sì
Medium Range Endurance	MRE	> 500	8 000	10 - 18	1 250	Sì
Low Altitude Deep Penetration	LADP	> 250	50 - 9 000	0,5 - 1	350	Sì
Low Altitude Long Endurance	LALE	> 500	3 000	> 24	< 30	Sì
Medium Altitude Long Endurance	MALE	> 500	14 000	24 - 48	1500	Sì
Strategic UAV						
High Altitude Long Endurance	HALE	> 2 000	20 000	24 - 48	12 000	Sì
Special purpose UAV						
Unmanned combat aerial vehicle	UCAV	1 500	10 000	2	10 000	Sì
Lethal	LETH	300	4 000	3 - 4	250	Sì
Decoy	DEC	0 - 500	5 000	< 4	250	Sì
Stratospheric	STRATO	> 2 000	> 20 000 & < 30 000	> 48	Da definire	No
Exo - stratospheric	EXO	Da definire	< 30 000	Da definire	Da definire	No
Space	SPACE	Da definire	Da definire	Da definire	Da definire	No

Figura 2.1

Il modello da noi in oggetto di studio è un quadricottero con un raggio operativo attorno ai 5 km, con una quota di volo sui 200 m, una durata di 15' ed un payload di 2 kg, tutte queste peculiarità fanno rientrare questo drone nella categoria dei Mini.



Figura 2.2

Per definizione un quadricottero (dal greco ptéron: ala, “quattro ali”) è un aerogiro sollevato e spinto da quattro rotori, quindi la presenza di più propulsori include questo modello fra i multirotori, famiglia che comprende anche i tricottero, gli esacotteri e gli octocotteri.

Sempre nella categoria dei droni si distinguono quelli ad ala fissa, qui rientrano quei velivoli dove la portanza è generata da un profilo alare solidale con la struttura (figura 2.3) e quelli ad ala rotante, dove la portanza è generata da una o più eliche poste in rotazione rispetto alla struttura.



Figura 2.3

Altra distinzione si deve al metodo adottato per generare l'incremento di portanza di un rotore, si hanno così eliche a passo fisso, ovvero con angolo d'attacco costante ma velocità di rotazione controllata e a passo variabile cioè il caso opposto al precedente.

3. STRUTTURA DEL QUADRICOTTERO

Il drone come ogni essere o macchinario che esegue ordini o copie lavoro è composto da una struttura (telaio) che regge tutto il sistema, un cuore (batteria) che fornisce e distribuisce energia, un cervello (controller) che elabora i dati ricevuti dai sensori (piattaforma IMU), una rete nervosa (regolatori) per la distribuzione degli impulsi elettrici ai muscoli (rotori) che eseguono fisicamente le operazioni calcolate.

Qui di seguito si riporta una miniatura del nostro quadricottero privo della scocca di protezione per evidenziare le parti principali che lo compongono.

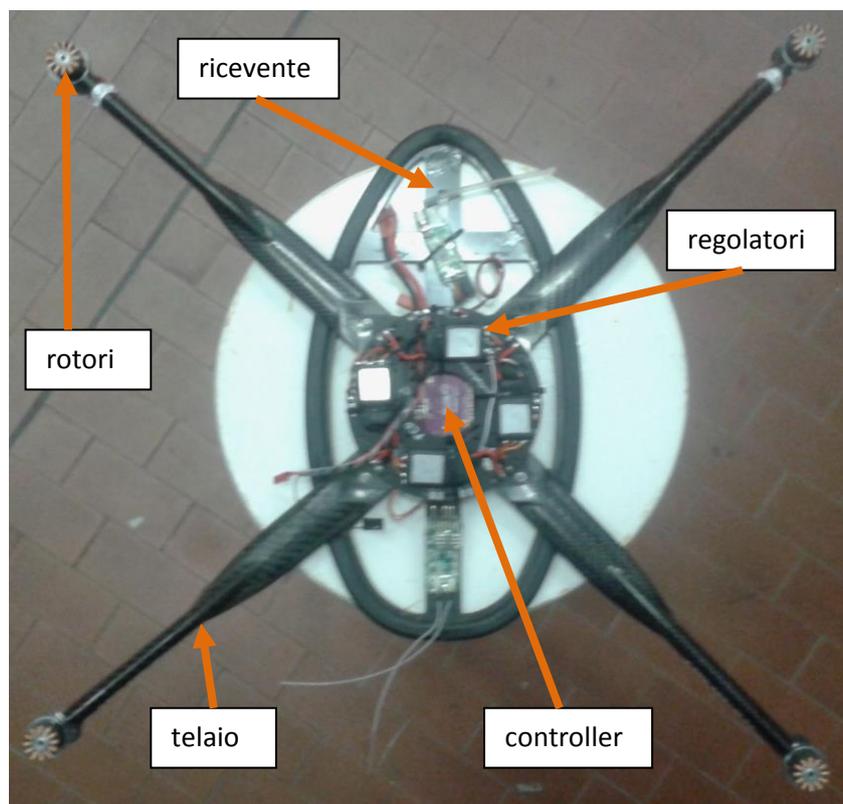


Figura 3.1

Facciamo ora una breve analisi dei componenti evidenziati:

- Telaio:

questo elemento è lo scheletro del drone, può presentarsi in forme e dimensioni diverse, in base al numero di rotori che deve supportare e in base al payload che deve gestire.

Quello di figura 3.1 è il telaio di un quadricottero di medie dimensioni con payload pari a circa 1.5 kg. Questo componente meccanico è composto da un corpo centrale e da quattro braccetti generalmente perpendicolari fra di loro.

È facile intuire che le caratteristiche principali che lo caratterizzano sono il peso e la rigidità, infatti l'obiettivo che si prefigge ogni progettista è di produrre una struttura estremamente leggera ma infinitamente rigida.

Uno scheletro pesante riduce il payload ovvero il carico che il velivolo può sopportare, quindi un telaio più leggero comporta di trasportare più peso oppure a parità di carico consente di avere un risparmio energetico delle batterie maggiore implicando un tempo di volo più lungo o di consentire una qualità del volo migliore cioè una migliore agilità e manovrabilità del mezzo.

La rigidità invece interviene sulla stabilità del drone.

Se immaginiamo il quadricottero vincolato nel centro, i rotori generano una portanza che altro non è che una forza che produce taglio e flessione dei braccetti. Si ha quindi una deformazione della struttura, che porta alle propulsioni generate dai singoli rotori, a non essere più perpendicolari al piano che prima li conteneva.

ne consegue che piccoli errori di montaggio, inevitabili, si amplificano e producono instabilità nella fase di hovering o di traiettorie scorrette nelle fasi di volo.

È fondamentale allora la ricerca di un materiale che comprenda entrambe queste peculiarità, leggerezza e poca flessibilità.

I materiali preferenziali ora utilizzati nel mercato sono dunque le leghe di alluminio o i materiali compositi.

Il quadricottero oggetto di questa tesi è per l'appunto costituito da fibra di carbonio con spessore di 0.75 mm.

- Regolatori:



Figura 3.2

Di questi componenti elettronici ne sussistono uno per ogni rotore, ed hanno il compito di gestire i dati elaborati dal controller e tramutarli in un corretto voltaggio elettrico da fornire ai rotori.

- Rotori:



Figura 3.3

La categoria di motori che possono essere utilizzati da questo genere di mezzi si restringe principalmente a motori brushless.

Questo perché quello che principalmente è richiesto è alta efficienza, alta velocità di rotazione e poco peso.

Motori trifase infatti non sono adatti perché di norma sono molto massicci e pesanti, mentre i classici motori a spazzola hanno problemi di scintillio alle alte velocità.

I brushless invece si prestano benissimo per questa categoria di velivoli in quanto sprigionano la portanza desiderata con un'elevata dinamica e buona efficienza energetica.

- Ricevitore:



Figura 3.4

È il dispositivo che permette di comunicare con la trasmittente, riceve in input i segnali sotto forma di onde radio e li trasduce in impulsi elettrici per il controller.

- Controller:

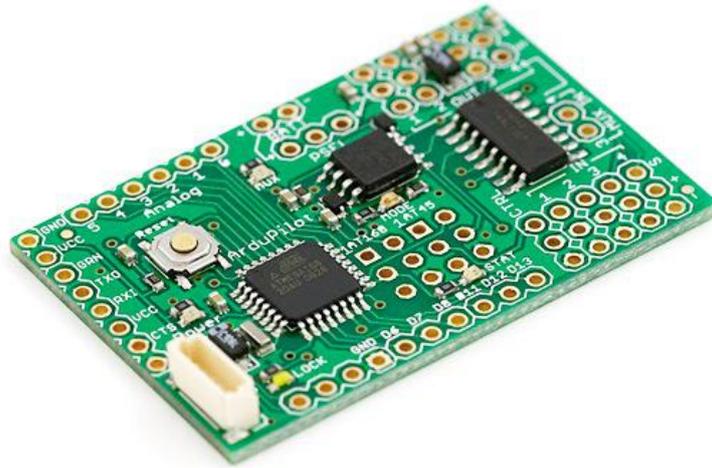


Figura 3.5

Il tipo di controller che monta il nostro quadricottero è un Ardupilot arduino.

Questo elemento altro non è che una scheda che monta un processore per l'analisi dei dati prodotti dalla piattaforma IMU, dal magnetometro, un altimetro e dal GPS sempre collocati su questa.

Lo schema è molto semplice, la piattaforma IMU è un insieme di giroscopi ed accelerometri per il calcolo delle accelerazioni rettilinee e rotazionali, il magnetometro altro non è che una bussola che rileva la posizione del nord e quindi l'orientamento del drone, l'altimetro misura la sua posizione in quota e in fine il GPS fornisce indicazioni su velocità e ubicazione del velivolo nello spazio.

Il processore invece assimila questi dati e tramite un algoritmo di base implementato al suo interno (software) trasforma questi input in segnali forniti ai regolatori.

Obiettivo di questo studio è proprio la formulazione del software da implementare nel controllore per la gestione di un drone "anti-vento".

- Batteria:



Figura 3.6

Rappresentano uno dei grandi limiti di questi mezzi che purtroppo nel mercato odierno hanno il difetto di essere troppo pesanti e di immagazzinare poca energia.

Ad ogni modo sono elementi indispensabili e di norma forniscono un'autonomia di volo di circa quindici minuti prima che venga segnalata l'emergenza dal radiocomando (figura3.7).



Figura 3.7

- Eliche:



Figura 3.7

Possono essere costituite da i materiali più disparati, ma per questa tesi saranno considerate eliche di polimero plastico della Apc 10x4.7 inc.

Il produttore mette a disposizione per ogni tipologia una scheda (disponibile nel sito : <http://www.apcprop.com/v/index.html>) che fornisce le caratteristiche delle varie dinamiche di volo, ovvero coefficienti, portanza e momenti ai vari regimi di rotazione.

4. MECCANICA DI VOLO

Sempre facendo riferimento alla categoria dei quadricotteri ad ala rotante con passo fisso, iniziamo ad analizzare le tecniche utilizzate per movimentare il drone.

Come prima analisi valutiamo i momenti generati dalla rotazione delle eliche, difatti come conseguenza dell'attrito si ha che un rotore in movimento produce un'azione contraria sul sostegno.

Un esempio pratico è incentrato nell'argomento qui trattato sono gli elicotteri, infatti l'elica mossa da una turbina a gas produce a sua volta una coppia che metterebbe in rotazione l'intero velivolo, per ovviare quindi a questo problema si monta un'elica che ruota su un'asse trasversale capace di equilibrare i momenti in gioco e tenere l'elicottero stabile e fisso in quota.

Lo stratagemma usato per questo tipo di droni è di assegnare ai rotori diametralmente opposti lo stesso verso di rotazione e contrario per i restanti, ciò produce sui due assi una coppia di momenti che, a pari velocità di giri di tutti i propulsori, si eguagliano e quindi permettono al quadricottero di non ruotare attorno a nessun asse e senza l'uso di rotori extra.

Ecco in figura 3.1 una possibile configurazione di volo funzionante.

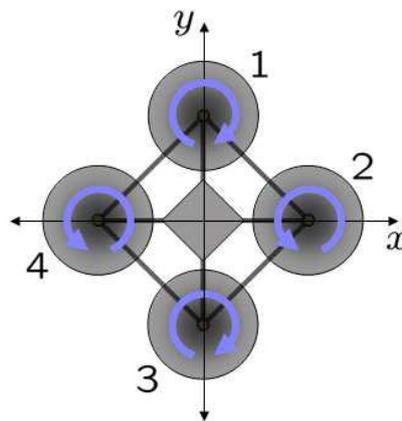


Figura 4.1

È facile intuire a questo punto che un incremento (o decremento) di potenza in un singolo rotore o ad una coppia di questi provoca uno sbilancio di forze o momenti tale da provocare il movimento del velivolo, stiliamo qui di seguito le combinazioni utilizzate:

- incremento di potenza in tutti e quattro i rotori : non provoca uno sbilancio di coppia ma un incremento eguale di portanza su ogni propulsore, provocando una salita verticale in quota del drone (figura 4.2a);
- incremento di potenza di un rotore e decremento di potenza di quello opposto : tale combinazione crea un aumento di portanza su un rotore e un decremento in quello opposto generando una rotazione attorno all'asse che monta i propulsori in cui non è avvenuta

variazione di rotazione. Questa manovra induce il drone a ruotare secondo il pitch e il roll ma mantenendo costante la rotazione attorno all'asse verticale di yaw in quanto si mantengono bilanciati i momenti totali sulle coppie dei bracci (figura 4.2b-c);

- incremento pari di potenza di due motori opposti e decremento pari dei restanti due : questa combinazione si bilancia nella portanza totale, mantenendo il velivolo ad una quota costante ma genera uno sbilanciamento tra le coppie degli assi portando il drone a ruotare rispetto all'asse verticale di yaw (figura 4.2d).

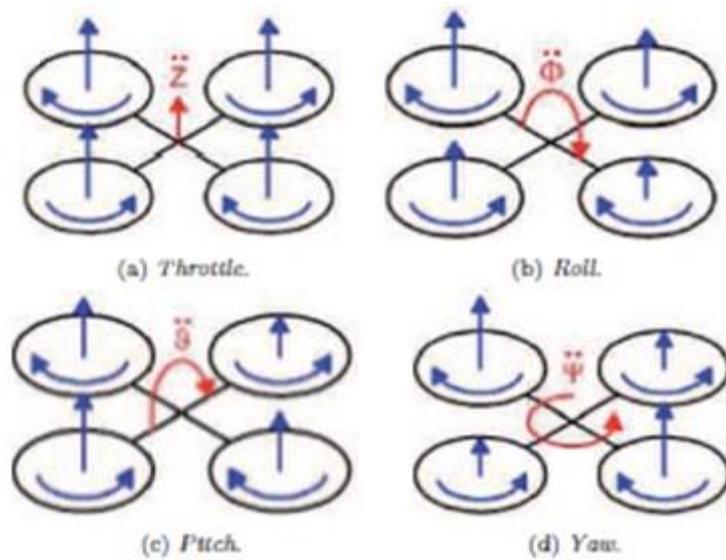


Figura 4.2

5. ASSETTO DI UN CORPO RIGIDO

Prima di passare ad una descrizione dettagliata del metodo utilizzato per la movimentazione della macchina, bisogna definire il modello e quale tipo di elica monta, assumiamo per semplicità un quadricottero con telaio a croce che monta eliche a passo fisso.

Tale scelta si basa sul modello che mi è stato offerto in studio dall'azienda Italdron SRL e di cui viene riportata una miniatura a seguito.



Figura 5.1

Assunto il modello si deve definire il sistema di riferimento e come l'oggetto in esame si orienta al suo interno, si introducono a tal proposito il roll, il pitch e lo yaw e la relativa posizione nello spazio.

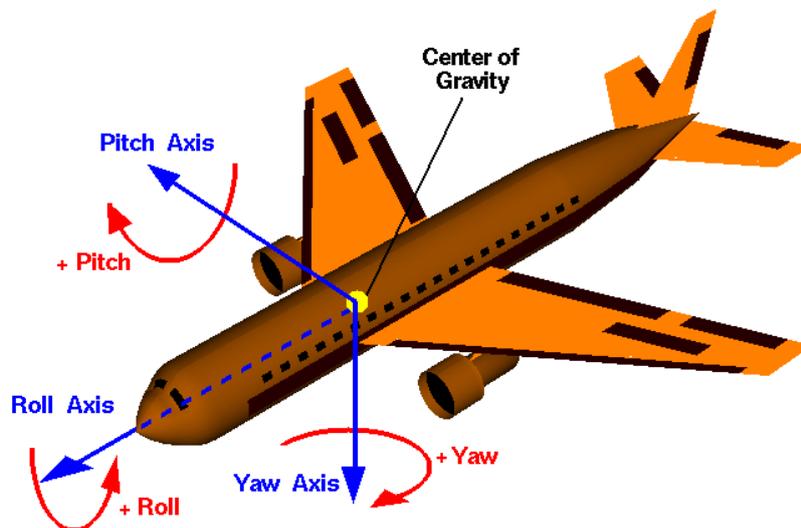


Figura 5.2

Come si evince dalla figura sopra riportata questi rappresentano le rotazioni attorno agli assi principali, avremo perciò sei gradi di libertà totali per il mezzo, gli elenchiamo qui di seguito:

- X : movimento frontale del velivolo;
- Y : movimento laterale del velivolo;
- Z : movimento verticale del velivolo;

- Beccheggio o pitch : oscillazione attorno ad un asse trasversale (pitchaxis);
- Rollio o Roll : oscillazione attorno ad un asse longitudinale (rollaxis);
- Imbardata o yaw : oscillazione attorno ad un asse verticale (yawaxis).

Accordati i gradi di libertà con gli opportuni versi, definendo così i segni positivi e negativi, lo step successivo è di stabilire i sistemi di riferimento nello spazio.

A tal proposito si genera una terna cartesiana nominata Sistema ECEF (Earth Centered Earth Fixed) ossia un sistema cartesiano ortogonale OXYZ, fisso rispetto alla Terra e con origine nel suo centro e con :

- asse Z coincidente con l'asse di rotazione media della Terra;
- assi X e Y giacenti sul piano equatoriale terrestre, con asse X passante per l'intersezione tra equatore e meridiano di Greenwich;
- asse Y perpendicolare agli altri due in modo da formare una terna destrorsa.

Solo questo però non è sufficiente per orientare il drone nello spazio, si ricorre allora al Sistema NED (North East Down) un sistema mobile definito in riferimento al piano tangente alla superficie terrestre in un qualsiasi punto.

Tale sistema di riferimento si colloca nel baricentro del modello, ma in un generico punto P, fornendo così un sistema solidale su cui riferire il drone, avremo quindi nello specifico:

- assi N e E giacenti sul piano orizzontale, cioè quel piano parallelo al piano tangente alla Terra e passante per la proiezione P' di P sulla superficie terrestre;
- asse N parallelo alla tangente al meridiano locale, cioè quel meridiano passante per la proiezione P' di P sulla superficie terrestre. Il verso di N punta al Polo Nord geografico;
- asse E parallelo alla tangente al parallelo locale, cioè quel *parallelo* passante per la proiezione P' di P sulla superficie terrestre. Il verso di E a Est;
- asse D perpendicolare agli altri due assi N e E e rivolto verso il basso. L'asse D è diretto lungo la verticale locale, cioè la retta passante per il punto considerato P e la proiezione P' di tale punto sulla superficie terrestre, perpendicolare al piano orizzontale;
- L'azimuth o prua o *heading* è l'angolo definito tra l'asse N e la proiezione dell'asse longitudinale dell'aereo sul piano N-E.

In fine necessitiamo di definire un ultimo riferimento che mi espliciti le rotazioni che il modello possiede rispetto agli assi fissi del NED, otteniamo il Sistema Assi Corpo (Body) ovvero un sistema di riferimento solidale al velivolo, con origine nel suo baricentro, e tale che:

- asse 1 (Rollaxis) lungo l'asse longitudinale del velivolo;
- asse 3 (yawaxis) giace sul piano di simmetria verticale del velivolo ed è perpendicolare all'asse 1 e

punta verso il basso;

- asse 2 (pitch axis) è quello trasversale al velivolo (è diretto verso l'ala destra, se si guarda il velivolo dall'alto), perpendicolarmente agli assi 1 e 2, in modo da ottenere una terna destrorsa.

Riportiamo qui di seguito un'illustrazione per comprendere meglio le terne cartesiane appena elencate.

5.1. ANGOLI DI EULERO

Sono tre angoli che si definiscono quando il sistema di riferimento body coincide solo con l'origine con il sistema di riferimento NED ma senza sovrapposizione degli assi, si generano così tre sfasamenti angolari che analizzeremo nel dettaglio. Per facilitarne la rappresentazione, tramite la figura 5.3 procederemo step by step per creare questi angoli a partire dalla situazione iniziale in cui i due sistemi di riferimento coincidono perfettamente.

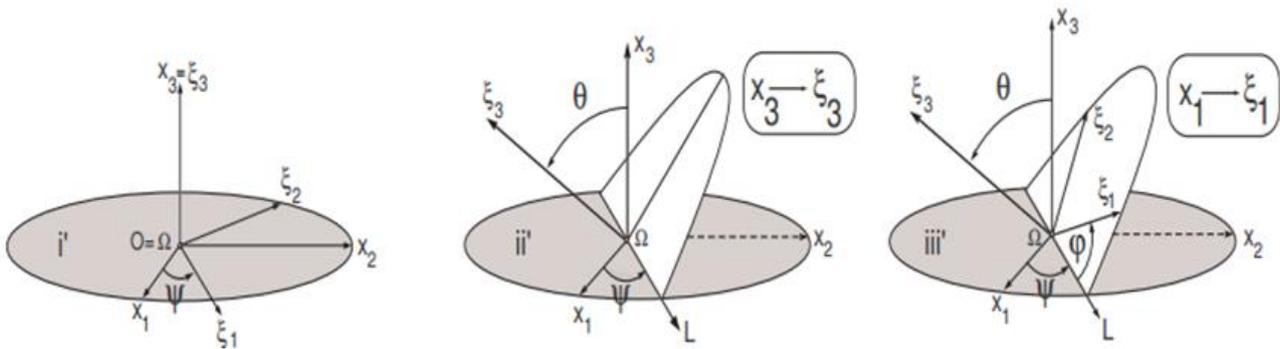


Figura 5.3 roll pitch yaw

Si definiscono:

- ψ angolo di imbardata o di azimuth (yaw): si genera mettendo in rotazione il sistema di riferimento body ($\xi_1\xi_2\xi_3$) attorno all'asse X_3 ;
- θ : angolo di beccheggio o di elevazione (pitch): per generare questo angolo si mantiene fisso l'asse ξ_1 e si ruota il sistema body attorno a questo;
- φ : angolo di rollio o di inclinazione (roll): in fine per realizzare l'ultima rotazione si mantiene costante l'asse ξ_3 e si ruota il sistema body attorno a questo.

Si nota che le rotazioni positive sono in senso antiorario e il dominio dei tre angoli sono:

$$\varphi \in]-180^\circ, 180^\circ]$$

$$\theta \in]-90^\circ, 90^\circ[$$

$$\psi \in]-180^\circ, 180^\circ]$$

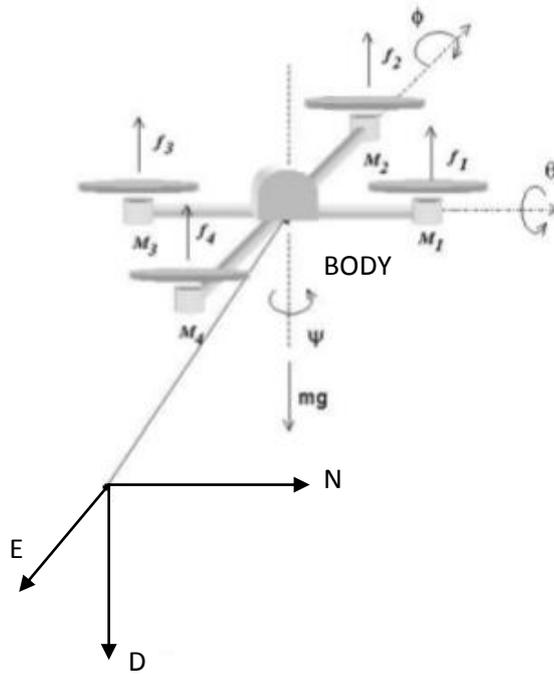


Figura 5.4

Come si nota dalla figura 5.4, per trasformare ora le coordinate dal sistema mobile Body a quello fisso NED, necessita di tre matrici di rotazione per ogni angolo, ovvero per il roll, pitch e yaw. Avremo perciò :

$$R_{\phi} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{pmatrix}$$

$$R_{\theta} = \begin{pmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{pmatrix}$$

$$R_{\psi} = \begin{pmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Moltiplicando fra loro ottengo la matrice totale di rotazione **DCM** (direction cosine matrix):

$$\mathbf{DCM} = \begin{pmatrix} c\vartheta c\psi & c\vartheta s\psi & -s\vartheta \\ s\phi s\vartheta c\psi - c\phi s\psi & s\phi s\vartheta s\psi + c\phi c\psi & s\phi c\vartheta \\ s\phi c\vartheta c\psi + s\phi s\psi & s\phi c\vartheta s\psi - s\phi c\psi & c\phi c\vartheta \end{pmatrix}$$

5.2. VELOCITA' NEL SISTEMA BODY CON ANGOLI DI EULERO

In questo capitolo si analizza come sintetizzare la posizione e il movimento del quadricottero nello spazio sotto forma di equazioni matriciali, ovvero si procede con la modellizzazione vera e propria del caso reale.

Nella teoria del controllo le dinamiche del sistema modellato sono espresse tipicamente attraverso lo stato del sistema che nel caso del quadricottero corrispondono a 12 equazioni che ne descrivono l'assetto e la posizione a sei gradi di libertà.

Quindi si evince immediatamente che 6 gradi di libertà appartengono al sistema di riferimento NED e i restanti 6 al sistema di riferimento Body.

Come primo step identifichiamo tutte le velocità di traslazione e di rotazione del drone, ovvero le componenti nel sistema Body, avremo quindi un vettore per le velocità di traslazione (V_B) e uno per le velocità angolari (ω_B):

$$V_B = [U \ V \ W]^T \ ; \ \omega_B = [P \ Q \ R]^T$$

5.3. EQUAZIONI STANDARD DEL VOLO

Per il sistema NED o termini lineare invece avremo :

$$\Gamma_{NED} = [N \ E \ D]^T \ ; \ \Theta_{NED} = [\phi \ \theta \ \psi]^T$$

Quindi per i termini lineari avremo :

$$\dot{\Gamma}_{NED} = \mathbf{DCM} \cdot V_B$$

E analogamente per le rotazioni varrà :

$$\dot{\Theta}_{NED} = \mathbf{T} \cdot \omega_B$$

Dove la matrice \mathbf{T} analogamente alla \mathbf{R} è una matrice per le trasformazioni angolari :

$$\mathbf{T} = \begin{pmatrix} 1 & s\phi tg\theta & c\phi tg\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{pmatrix}$$

E quindi esplicitando le equazioni matriciali delle traslazioni e delle rotazioni otteniamo i seguenti sistemi :

$$\begin{cases} \dot{\phi} = P + Q \cdot tg\theta \cdot \sin\phi + R \cdot tg\theta \cdot \cos\phi \\ \dot{\theta} = Q \cdot \cos\theta - R \cdot \sin\phi \\ \dot{\psi} = \frac{\sin\phi}{\cos\theta} Q + \frac{\cos\phi}{\cos\theta} R \end{cases}$$

$$\begin{cases} \dot{N} = U \cdot \cos\theta \cdot \cos\Psi + V(\sin\theta \cdot \sin\phi \cdot \cos\Psi - \cos\phi \cdot \sin\Psi) + W(\sin\theta \cdot \cos\phi \cdot \cos\Psi + \sin\phi \cdot \sin\Psi) \\ \dot{E} = U \cdot \cos\theta \cdot \sin\Psi + V(\sin\theta \cdot \sin\phi \cdot \sin\Psi + \cos\phi \cdot \cos\Psi) + W(\sin\theta \cdot \cos\phi \cdot \sin\Psi - \sin\phi \cdot \cos\Psi) \\ \dot{D} = -U \cdot \sin\theta + V \cdot \cos\theta \cdot \sin\phi + W \cdot \cos\theta \cdot \cos\phi \end{cases}$$

È possibile ora, tramite le matrici di trasformazione delle coordinate, applicare la seconda legge di Newton avvalendosi delle seguente ipotesi :

- L'origine della terna BODY coincide con quella NED;
- Il centro delle terna BODY coincide con gli assi principali di inerzia.

Queste ipotesi permettono di scrivere un tensore di inerzia diagonale e di annullare i prodotti centrifughi e ottenere quindi un tensore diagonale. Se applichiamo quindi le leggi cardinali della dinamica si ha:

$$m \cdot \ddot{\Gamma}_{NED} = \sum_{i=0}^n F_i$$

$$I \cdot \ddot{\Theta}_{NED} = \sum_{i=0}^n \tau_i$$

E ricordando le derivate prime dei termini di traslazione e di rotazione abbiamo sostituendo :

$$m \cdot \ddot{\Gamma}_{NED} = m \cdot \frac{d}{dt}(\mathbf{DCM} \cdot V_B) = \mathbf{DCM} \cdot F_B$$

E quindi :

$$m \cdot (\mathbf{D}\dot{\mathbf{C}}\mathbf{M} \cdot V_B + \dot{V}_B \cdot \mathbf{DCM}) = \mathbf{DCM} \cdot F_B$$

E ricordando che la velocità di un terna si può esprimere come velocità (lineare) della terna stessa più la sua rotazione si ha :

$$m \cdot \mathbf{DCM} \cdot (\omega_B \times V_B + \dot{V}_B) = \mathbf{DCM} \cdot F_B$$

Da cui si ricava in fine semplificando :

$$m \cdot (\omega_B \times V_B + \dot{V}_B) = F_B$$

Ora per quanto riguarda i termini angolari :

$$I \cdot \ddot{\Theta}_{NED} = I \cdot \frac{d}{dt}(\mathbf{T} \cdot \omega_B) = \mathbf{T} \cdot \tau_B$$

Che svolgendo la derivata :

$$I \cdot \mathbf{T} \cdot \dot{\omega}_B + \omega_B \cdot \mathbf{T} \times (I \cdot \omega_B) = \mathbf{T} \cdot \tau_B$$

E in fine si ottiene :

$$I \cdot \dot{\omega}_B + \omega_B \times (I \cdot \omega_B) = \tau_B$$

Non resta che sviluppare i prodotti vettoriali e si ottengono le restanti equazioni vettoriali in forma matriciale :

$$\begin{pmatrix} F_X \\ F_Y \\ F_Z \end{pmatrix} = m \begin{pmatrix} \dot{U} & Q \cdot W & -R \cdot V \\ \dot{V} & R \cdot U & -P \cdot W \\ \dot{W} & P \cdot V & -Q \cdot U \end{pmatrix}$$

$$\begin{pmatrix} M_X \\ M_Y \\ M_Z \end{pmatrix} = m \begin{pmatrix} \dot{P} \cdot I_X & Q \cdot R \cdot (I_Z - I_Y) \\ \dot{Q} \cdot I_Y & P \cdot R \cdot (I_X - I_Z) \\ \dot{R} \cdot I_Z & P \cdot Q \cdot (I_Y - I_X) \end{pmatrix}$$

6. BARICENTRO E MATRICE DI INERZIA

Possiamo partire stilizzando il quadricottero in un modello più facile da commutare in seguito nella matrice d'inerzia.

A tale scopo è doveroso creare un modello virtuale il più possibile realistico, ovvero porre le masse laddove i pesi sono concentrati, avremo così la schematizzazione seguente :

- 4 masse concentrate m_1 agli estremi per rappresentare i rotori;
- 4 masse distribuite m_1' per rappresentare i bracci in carbonio;
- una massa M centrale che include batteria, corpo in carbonio ed elettronica all'interno di un volume a forma di elissoide;
- una massa m_2 distribuita in una sfera per rappresentare la telecamera e il peso del gimbal.

Si nota che per alcuni casi si sono usate masse concentrate e per altri masse distribuite, questo perché nella rotazione attorno all'asse verticale, se si utilizzavano masse concentrate per il corpo centrale e il gimbal non avremmo avuto effetti inerziali, invece approssimare i rotori con masse concentrate non riporta grandi variazioni di calcolo in quanto i rotori hanno piccole dimensioni e i termini quadratici quindi risultano trascurabili.

Ecco di seguito un possibile schema del drone stilizzato :

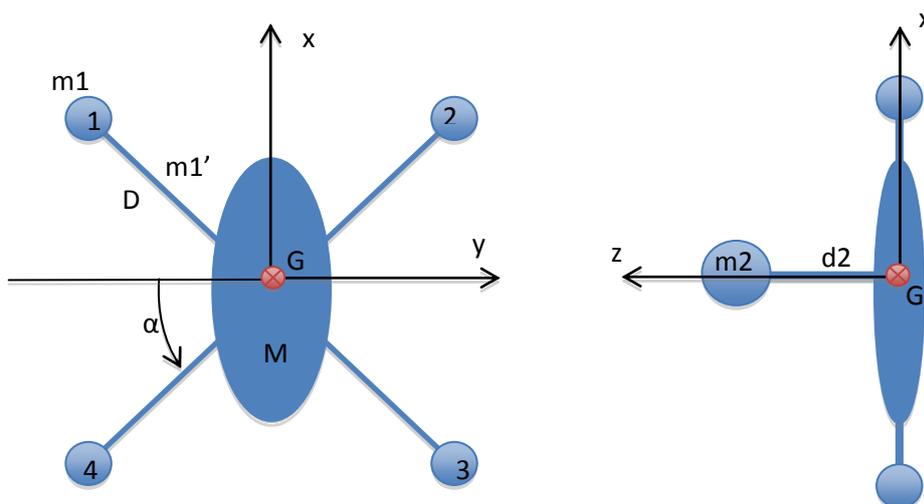


Figura 6.1

Dati :

- $m_1=67$ gr
- $D=33.5$ cm
- $m_1'=30$ gr
- $m_2=260$ gr
- $d_2=11.5$ cm
- $M=659$ gr
- $\alpha=45^\circ$
- raggio sfera corpo m_2 : $r=5$ cm
- semiassi ellissoide: $a=17$ cm ; $b=9$ cm ; $c=4.5$ cm.
- LunghezzaXpasso elica: 10×4.7 inc

6.1. CALCOLO BARICENTRO

essendo simmetrico rispetto alla verticale avremo $X_g = Y_g = 0$;

rispetto all'asse z invece applicando la formula risuta :

$$Z_g = \frac{\sum m_i \cdot Z_i}{\sum m_i} = \frac{m_2 \cdot d_2}{4m_1 + 4m_1' + M + m_2} = 2.3 \text{ cm}$$

6.2. CALCOLO MATRICE DI INERZIA

Ora componiamo la matrice d'inerzia :

$$I = \begin{pmatrix} A & -C' & -B' \\ -C' & B & -A' \\ -B' & -A' & C \end{pmatrix}$$

A:

$$I_{xx} = I_x = 4 \left(m_1 d_1^2 \cos^2 \alpha + \frac{1}{3} m_1' d_1^2 \cos^2 \alpha \right) + \frac{M}{5} (b^2 + c^2) + [4(m_1 + m_1') + M] Z_g^2 + \frac{2}{5} m_2 r^2 + m_2 (d_2 - Z_g)^2 = 0.0216316 \text{ kgm}^2$$

B:

$$I_{yy} = I_y = 4 \left(m_1 d_1^2 \sin^2 \alpha + \frac{1}{3} m_1' d_1^2 \sin^2 \alpha \right) + \frac{M}{5} (a^2 + c^2) + [4(m_1 + m_1') + M] Z_g^2 + \frac{2}{5} m_2 r^2 + m_2 (d_2 - Z_g)^2 = 0.0243731 \text{ kg m}^2$$

C:

$$I_{zz} = I_z = 4 \left(m_1 d_1^2 + \frac{1}{3} m_1' d_1^2 \right) + \frac{M}{5} (a^2 + b^2) + \frac{2}{5} m_2 r^2 = 0.0397019 \text{ kg m}^2$$

$A' = B' = C' = 0$; siccome il baricentro è centrato rispetto ai due piani verticali.

Si evince dunque che la matrice I è diagonale.

7. FORZE E MOMENTI APPLICATI AL MODELLO

In questo capitolo vediamo come si modificano le equazioni costituenti del moto con la presenza di forze agenti sul velivolo.

Prima di passare ad una formulazione prettamente matematica cerchiamo di comprendere intuitivamente come si scompongono le forze, di che natura sono e dove agiscono.

Per semplificare enormemente il modello faremo riferimento ad un sistema baricentrico, ovvero il sistema di riferimento BODY è coincidente con il baricentro del velivolo che è anche centro degli assi principali di inerzia (causando come visto $A' = B' = C' = 0$) e punto di applicazione di tutte le forze e i momenti.

Come prima analisi complessiva è facile immaginare che sussisterà una forza gravitazionale dovuta proprio alla massa che possiede il mezzo, e sarà sempre verticale e diretta verso il basso, al contrario la forza dovuta alla propulsione dei rotori sarà sempre perpendicolare al piano che contiene il corpo principale del drone e di verso opposto a quello gravitazionale, vedi figura 7.1.

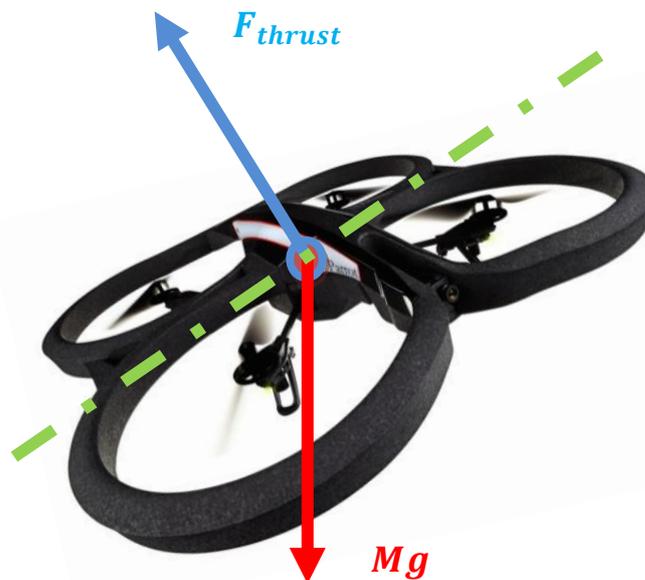


Figura 7.1

Quindi la forza di spinta, essendo il sistema calcolato sul BODY avrà componente solo e soltanto lungo l'asse Z.

I momenti invece saranno presenti in tutte e tre le componenti della terna di riferimento e si comporranno in base a come sono ripartite le velocità dei rotori, ma di questo si parlerà in seguito.

Ora analizziamo le componenti dovute dal vento.

Questa entità, che poi saranno gli ingressi non controllati del nostro modello, ovvero i disturbi, non sono altro che reazioni dovute all'aerodinamicità del velivolo e quindi matematicamente troppo complesse da formulare analiticamente.

Una strategia che useremo sarà valutare gli effetti dovuti alla presenza di questi attriti superficiali, ovvero i segnali che generano i sensori montati sul drone, quindi accelerazioni, posizioni e velocità.

Al contrario della forza di spinta, quella dovuta al vento, si ripercuote su tutti gli assi della terna BODY e genera momenti che generano rotazione attorno a tutti gli elementi della terna.

Non resta altro che aggiungere queste nuove entità ai sistemi che contengono le equazioni sviluppate dal secondo principio della dinamica di Newton.

Si ottengono dunque per le traslazioni e le rotazioni:

$$\left\{ \begin{array}{l} m(\dot{U} + Q \cdot W - R \cdot V) = -m \cdot g \cdot \sin \theta + F_{wx} \\ m(\dot{V} - P \cdot W + R \cdot U) = +m \cdot g \cdot \cos \theta \cdot \sin \phi + F_{wy} \\ m(\dot{W} - U \cdot Q + P \cdot V) = +m \cdot g \cdot \cos \theta \cdot \cos \phi - F_{tz} + F_{wz} \end{array} \right.$$

$$\left\{ \begin{array}{l} I_x \cdot \dot{P} + I_z \cdot R \cdot Q - I_y \cdot R \cdot Q = M_{tx} + M_{wx} \\ I_y \cdot \dot{Q} + I_x \cdot P \cdot R - I_z \cdot P \cdot R = M_{ty} + M_{wy} \\ I_x \cdot \dot{R} + I_y \cdot P \cdot Q - I_x \cdot P \cdot Q = M_{tz} + M_{wz} \end{array} \right.$$

Dove il pedice t indica thrust mentre w rappresenta wind.

Si nota difatti che la forza di spinta delle pale compare solo sull'asse Z, al contrario di quella dovuta dalle azioni aerodinamiche del vento, in quanto le eliche sono fisse e quindi la forza generata è sempre perpendicolare al piano X,Y.

7.1. VARIABILI DI STATO E INGRESSI DEL SISTEMA

Non rimane che stabilire quali siano le mie variabili stato che posso per comodità compattare all'interno di un vettore denominato vettore di stato :

$$\mathbf{x} = [\Phi \ \theta \ \Psi \ P \ Q \ R \ U \ V \ W \ N \ E \ D]^T$$

Mentre gli input del mio sistema saranno:

$$\mathbf{u} = [F_{tz} \ F_{wx} \ F_{wy} \ F_{wz} \ M_{tx} \ M_{ty} \ M_{tz} \ M_{wx} \ M_{wy} \ M_{wz}]^T$$

Riscriviamo ora tutte le equazioni che governano il mio sistema in funzione delle variabili sopra riportate.

Equazioni delle forze assi corpo:

$$\begin{cases} \dot{U} = -Q \cdot W + R \cdot V - g \cdot \sin \theta + F_{wx}/m \\ \dot{V} = P \cdot W - R \cdot U + g \cdot \cos \theta \cdot \sin \phi + F_{wy}/m \\ \dot{W} = Q \cdot U - P \cdot V + g \cdot \cos \theta \cdot \cos \phi + (-F_{tz} + F_{wz})/m \end{cases}$$

Equazioni dei momenti assi corpo:

$$\begin{cases} \dot{P} = \frac{I_y - I_z}{I_x} R \cdot Q + \frac{(M_{tx} + M_{wx})}{I_x} \\ \dot{Q} = \frac{I_z - I_x}{I_y} P \cdot R + \frac{(M_{ty} + M_{wy})}{I_y} \\ \dot{R} = \frac{I_y - I_x}{I_z} P \cdot Q + \frac{(M_{tz} + M_{wz})}{I_z} \end{cases}$$

Equazioni di Eulero per l'assetto:

$$\begin{cases} \dot{\phi} = P + Q \cdot \operatorname{tg} \theta \cdot \sin \phi + R \cdot \operatorname{tg} \theta \cdot \cos \phi \\ \dot{\theta} = Q \cdot \cos \theta - R \cdot \sin \theta \\ \dot{\psi} = \frac{\sin \phi}{\cos \theta} Q + \frac{\cos \phi}{\cos \theta} R \end{cases}$$

Equazioni che esprimono le velocità lineari in assi terra a partire da quelle assi corpo:

$$\left\{ \begin{array}{l} \dot{N} = U \cdot \cos\theta \cdot \cos\Psi + V(\sin\theta \cdot \sin\phi \cdot \cos\Psi - \cos\phi \cdot \sin\Psi) + W(\sin\theta \cdot \cos\phi \cdot \cos\Psi + \sin\phi \cdot \sin\Psi) \\ \dot{E} = U \cdot \cos\theta \cdot \sin\Psi + V(\sin\theta \cdot \sin\phi \cdot \sin\Psi - \cos\phi \cdot \cos\Psi) + W(\sin\theta \cdot \cos\phi \cdot \sin\Psi + \sin\phi \cdot \cos\Psi) \\ \dot{D} = -U \cdot \sin\theta + V \cdot \cos\theta \cdot \sin\phi + W \cdot \cos\theta \cdot \cos\phi \end{array} \right.$$

7.2. SCOMPOSIZIONE DELLE FORZE E DEI MOMENTI

La forza di spinta e il momento a sua volta sono entità dovute alle pale, le si possono scomporre tramite le formulazioni di Renard, per la forza traente avremo :

$$Ft = K_s' \cdot n^2$$

I cui termini indicano :

- n : numero di giri dell'elica;
- $K_s' = L^4 \cdot \rho \cdot Cs + K_i$.

avendo:

- L : lunghezza elica;
- ρ : densità aria;
- Cs : coefficiente di spinta ;
- K_i : coefficiente del momento angolare di inerzia delle singole eliche.

Per quanto concerne i momenti dovuti alla portanza generata dalla rotazione delle eliche avremo che il modulo sarà equivalente a:

$$M_{tx, y} = n^2 \cdot K_s' \cdot D$$

$$M_{tz} = n^2 \cdot K_D'$$

Dove:

- D : distanza fra il centro di rotazione dell'elica e il baricentro;
- $K_D' = L^5 \cdot \rho \cdot Cp + K_i$
- Cp : coefficiente di resistenza.

Se riconsidero tutto in base alle velocità angolari :

$$Ft = K_s \cdot \omega^2$$

$$M_{tx,y} = \omega^2 \cdot K_s \cdot D$$

$$M_{tz} = \omega^2 \cdot K_D$$

Con :

- $K_s = K'_s \cdot (30/\pi)^2$;
- $K_D = K'_D \cdot (30/\pi)^2$.

7.3. LEGAME TRA VELOCITÀ ANGOLARE E ROLL-PITCH-YAW

Prima di tutto si evidenzia il fatto che lo scrabble , al contrario di molti quadricotteri, non presenta la direzione di movimentazione coincidente con l'asse di un braccetto, ma è ruotato di 45° , quindi il suo assetto è quello riportato qui in figura 7.2.

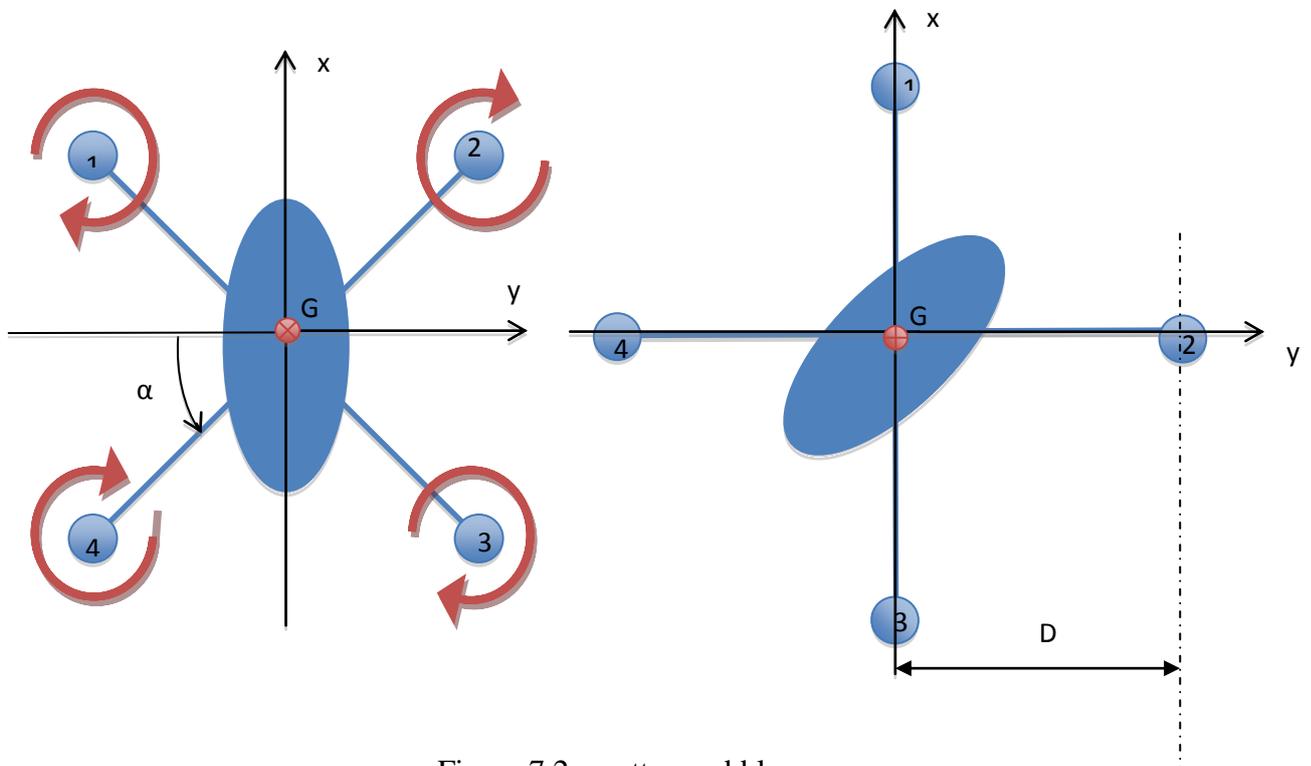


Figura 7.2 assetto scrabble

Come ultima fase si cerca ora di formulare un'espressione algebrica universale in grado di adattarsi a tutti i sistemi di riferimento desiderati.

È necessario dunque definire l'angolo di rotazione rispetto ad il sistema di riferimento fisso, avremo per ciò come evidenziato in figura una rotazione del sistema body pari ad α .

Si assumono poi arbitrariamente le coppie di rotazione dei propulsori, che come illustrato nella miniatura, saranno orario per i rotori 1-3 e antiorario per i rotori 2-4.

In figura 7.3 è riportata una sezione della pala e le relative forze generate dalla rotazione e dalla presenza dell'aria.

Come già riassunto avremo che la reazione dovuta al movimento delle eliche genera due forze principali, una di propulsione verso l'alto (lift) e una di resistenza aerodinamica opposta al movimento dell'elica (drag).

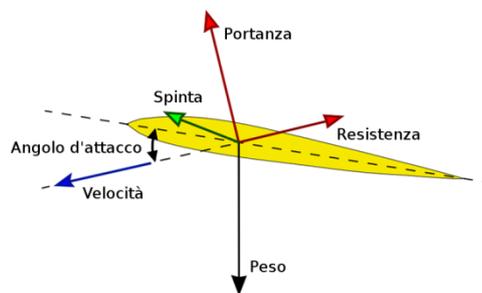


Figura 7.3

Questa affermazione non è del tutto veritiera, in quanto come si comprende da figura 7.3, proiettando le nostre forze lungo gli assi BODY avremo nelle componenti orizzontali (per lo yaw) principalmente dovute al drag, come già detto, ma anche in piccola parte dovute al lift.

Viceversa anche per le rotazioni roll e pitch, dove interagiscono solo le forze verticali, abbiamo una piccola parte di drag e la maggior parte di lift.

Per facilitare i calcoli assumiamo trascurabili questi valori che comunque influenzano il modello.

Per calcolare le coppie sarà sufficiente moltiplicare le relative forze per le relative distanze ottenendo lo schema di figura 7.4.

Il coefficiente K_i presentato nella pagina precedente esprime la reazione generata dal campo elettrico che si oppone a quella di inerzia.

Il rotore in uno stato di quiete o di moto di rivoluzione costante, soggetto ad un campo magnetico crescente, riceve una coppia che lo impone ad accelerare. Possedendo però questo una massa e quindi un inerzia crea una reazione opposta, a quella impartita dalle bobine, ovvero un momento sul sistema su cui è montato lo statore. Tale entità è molto inferiore a quella impartita dal drag

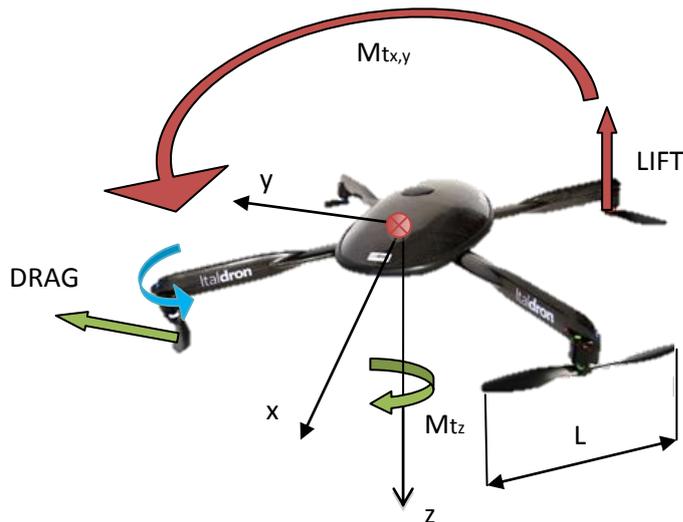


Figura 7.4 coppie generate dalle forze di lift e drag

Attorno agli assi X e Y si utilizzeranno le forze generate dalla portanze delle eliche, che generano i momenti evidenziati in rosso, mentre per la coppia attorno all'asse Z useremo le forze dovute all'attrito delle pale che costituiranno il momento evidenziato in verde.

Aggiornando tutte la 12 equazioni avremo in fine:

$$\begin{cases} \dot{U} = -Q \cdot W + R \cdot V - g \cdot \sin \theta + F_{wx} / m \\ \dot{V} = P \cdot W - R \cdot U + g \cdot \cos \theta \cdot \sin \phi + F_{wy} / m \\ \dot{W} = Q \cdot U - P \cdot V + g \cdot \cos \theta \cdot \cos \phi + (-\sum_i K_s \cdot \omega_i^2 + F_{wz}) / m \end{cases}$$

$$\begin{cases} \dot{P} = \frac{I_y - I_z}{I_x} R \cdot Q + \frac{(((\omega_1^2 - \omega_3^2) \cdot \sin \alpha + (\omega_4^2 - \omega_2^2) \cdot \cos \alpha) \cdot D \cdot K_s + M_{wx})}{I_x} \\ \dot{Q} = \frac{I_z - I_x}{I_y} P \cdot R + \frac{(((\omega_1^2 - \omega_3^2) \cdot \cos \alpha + (\omega_2^2 - \omega_4^2) \cdot \sin \alpha) \cdot D \cdot K_s + M_{wy})}{I_y} \\ \dot{R} = \frac{I_x - I_y}{I_z} P \cdot Q + \frac{((\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2) \cdot K_D + M_{wz})}{I_z} \end{cases}$$

$$\begin{cases} \dot{\phi} = P + Q \cdot \tan \theta \cdot \sin \phi + R \cdot \tan \theta \\ \dot{\theta} = Q \cdot \cos \theta - R \cdot \sin \theta \\ \dot{\psi} = \frac{\sin \phi}{\cos \theta} Q + \frac{\cos \phi}{\cos \theta} R \end{cases}$$

$$\left\{ \begin{array}{l} \dot{N} = U \cdot \cos\theta \cdot \cos\Psi + V(\sin\theta \cdot \sin\phi \cdot \cos\Psi - \cos\phi \cdot \sin\Psi) + W(\sin\theta \cdot \cos\phi \cdot \cos\Psi + \sin\phi \cdot \sin\Psi) \\ \dot{E} = U \cdot \cos\theta \cdot \sin\Psi + V(\sin\theta \cdot \sin\phi \cdot \sin\Psi - \cos\phi \cdot \cos\Psi) + W(\sin\theta \cdot \cos\phi \cdot \sin\Psi + \sin\phi \cdot \cos\Psi) \\ \dot{D} = -U \cdot \sin\theta + V \cdot \cos\theta \cdot \sin\phi + W \cdot \cos\theta \cdot \cos\phi \end{array} \right.$$

7.4. VELOCITA' ANGOLARE IN CASO DI HOVERING

Come si evince dalle 12 equazioni ricavate, ogni variabile presenta al suo interno delle funzioni trigonometriche, quindi non sono lineari.

Sotto un aspetto puramente fisico l'intera formulazione è completamente risolta ma in fini matematici questo problema non ha soluzione proprio a causa della non linearità delle equazioni.

Si mette in evidenza il caso in cui il drone assume una posizione fissa nell'aria, ovvero :

$\Phi = \Theta = \Psi = U = V = W = P = Q = R = 0$; si ottiene:

$$\omega_1 = \omega_2 = \omega_3 = \omega_4 = \sqrt{\frac{g \cdot m}{4 \cdot K_S}} = \omega_0$$

Ossia la velocità angolare che deve conseguire ogni rotore affinché generino una portanza in grado da equivalere la forza peso del drone e senza generare squilibri nei momenti.

Ora abbiamo calcolato il valore di ω che mantiene il quadricottero nella posizione di hovering, questo calcolo si rivela necessario per la trattazione che faremo qui di seguito per lo studio della linearizzazione.

8. LINEARIZZAZIONE

Il modello non lineare precedentemente introdotto non è utile ai fini della stesura di una logica di controllo lineare, e quindi delucidato questo presupposto è stato linearizzato attorno ad un punto di equilibrio.

Questo punto per semplicità verrà assunto nel caso di volo livellato ad una quota diversa da zero, con rotazioni e velocità nulle, tale caso viene chiamato hovering.

Come è noto la procedura di linearizzazione prevede prima la ricerca di un punto di equilibrio, che sarà dunque l'hovering, e con tale valore in ingresso avremo come soluzione del sistema algebrico:

$$f(\mathbf{x} \mathbf{u} t) = 0$$



Funzione non lineare

Essendo la funzione f non lineare la soluzione del sistema non è detto che esista e nel caso esista non è detto che sia unica o determinabile. In particolare, per il sistema in esame, la soluzione è praticamente impossibile trovarla essendo presenti funzioni trigonometriche legate tra di loro in modo non elementare. Si utilizza per ovviare a questo problema il teorema delle piccole oscillazioni ovvero si considerano rotazioni infinitesimali attorno agli assi principali tali da approssimare le funzioni trigonometriche con variabili lineari rendendo così i sistemi tutti risolvibili.

Avremo in sostanza per $\xi \rightarrow 0$:

- $\sin \xi \rightarrow \xi$
- $\cos \xi \rightarrow 1$
- $\text{tg } \xi \rightarrow \xi$

Per quanto riguarda le variabili principali le si può immaginare come un valore che oscilla attorno alla posizione di equilibrio, per notazione assumeremo:

- Lettere maiuscole senza pedice: per indicare valore istantaneo;
- Lettere minuscole: per indicare la variazione del valore attorno alla posizione di equilibrio;
- Lettere maiuscole con pedice zero: per indicare il valore di equilibrio;

Prefisso "d" : indica la variazione della forza rispetto alla sua posizione di equilibrio.

Avremo quindi per le 12 variabili in trattazione:

$$\Phi = \varphi + \Phi_0$$

$$\Theta = \theta + \Theta_0$$

$$\Psi = \psi + \Psi_0$$

$$P = p + P_0$$

$$Q = q + Q_0$$

$$R = r + R_0$$

$$U = u + U_0$$

$$V = v + V_0$$

$$W = w + W_0$$

$$N = n + N_0$$

$$E = e + E_0$$

$$D = d + D_0$$

Mentre per le forze e i momenti :

$$F_t = df_t + F_{t0}$$

$$F_w = df_w + F_{w0}$$

$$M_t = dm_t + M_{t0}$$

$$M_w = dm_w + M_{w0}$$

Quindi derivando ogni equazione otteniamo per i sistemi di forze e momenti :

$$\left\{ \begin{array}{l} \dot{u} = -Q_0 \cdot w - q \cdot W_0 + R_0 \cdot v + r \cdot V_0 - g \cdot \vartheta \cdot \cos\theta_0 + \frac{df_{wx}}{m} \\ \dot{v} = P_0 \cdot w + p \cdot W_0 - R_0 \cdot u - r \cdot U_0 + g \cdot \phi \cdot \cos\Phi_0 \cdot \cos\theta_0 - g \cdot \vartheta \cdot \sin\Phi_0 \cdot \sin\theta_0 + \frac{df_{wy}}{m} \\ \dot{w} = Q_0 \cdot u + q \cdot U_0 - P_0 \cdot v - p \cdot V_0 - g \cdot \phi \cdot \cos\theta_0 \cdot \sin\Phi_0 - g \cdot \vartheta \cdot \sin\theta_0 \cdot \cos\Phi_0 + \frac{-(\sum_i df_{tzi}) + df_{wz}}{m} \end{array} \right.$$

$$\left\{ \begin{array}{l} \dot{p} = \frac{I_y - I_z}{I_x} \cdot (R_0 \cdot q + r \cdot Q_0) + \frac{dm_{tx} + dm_{wx}}{I_x} \\ \dot{q} = \frac{I_z - I_x}{I_y} \cdot (P_0 \cdot r + p \cdot R_0) + \frac{dm_{ty} + dm_{wy}}{I_y} \\ \dot{r} = \frac{I_x - I_y}{I_z} \cdot (P_0 \cdot q + p \cdot Q_0) + \frac{dm_{tz} + dm_{wz}}{I_z} \end{array} \right.$$

8.1. MODELLO LINEARIZZATO CASO HOVERING

Si cerca ora di correlare la variazione delle forze di propulsione con la velocità angolare, in quanto è il valore che avremo in input:

$$F_t = K_s \cdot \omega^2$$

Derivando rispetto ad ω in hovering:

$$df_{tz} = 2 \cdot K_s \cdot \omega_0 \cdot d\omega$$

Mentre per i momenti avremo :

$$dm_{tx,y} = 2 \cdot K_s \cdot D \cdot \omega_0 \cdot d\omega$$

$$dm_{tz} = 2 \cdot K_D \cdot \omega_0 \cdot d\omega$$

Sostituendo nelle equazioni e ricordando che nella posizione di hovering si ha :

$$\Phi_0 = \theta_0 = \Psi_0 = P_0 = Q_0 = R_0 = U_0 = V_0 = W_0 = 0$$

$$\left\{ \begin{array}{l} \dot{u} = -g \cdot \vartheta + \frac{df_{wx}}{m} \\ \dot{v} = g \cdot \phi + \frac{df_{wy}}{m} \\ \dot{w} = \frac{-2 \cdot K_s \cdot \omega_0 \cdot (d\omega_1 + d\omega_2 + d\omega_3 + d\omega_4) + df_{wz}}{m} \end{array} \right.$$

$$\left\{ \begin{array}{l} \dot{p} = \frac{2 \cdot K_s \cdot D \cdot \omega_0 \cdot [(d\omega_1 - d\omega_3) \cdot \sin\alpha + (d\omega_4 - d\omega_2) \cdot \cos\alpha] + dm_{wx}}{I_x} \\ \dot{q} = \frac{2 \cdot K_s \cdot D \cdot \omega_0 \cdot [(d\omega_1 - d\omega_3) \cdot \cos\alpha + (d\omega_2 - d\omega_4) \cdot \sin\alpha] + dm_{wy}}{I_y} \\ \dot{r} = \frac{-2 \cdot K_D \cdot \omega_0 \cdot (d\omega_4 + d\omega_2 - d\omega_1 - d\omega_3) + dm_{wz}}{I_z} \end{array} \right.$$

$$\begin{cases} \dot{\phi} = p \\ \dot{\theta} = q \\ \dot{\psi} = r \end{cases}$$

$$\begin{cases} \dot{n} = u \\ \dot{e} = v \\ \dot{d} = w \end{cases}$$

8.2 MODELLO LINEARIZZATO IN FORMA MATRICIALE

Si nota che i sistemi appena ricavati si possono raggruppare in un'equazione matriciale del tipo:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

In cui si ricordano le componenti del vettore \mathbf{x} e del vettore \mathbf{u} già stabilite precedentemente:

$$\mathbf{x} = [\phi \ \theta \ \psi \ p \ q \ r \ u \ v \ w \ n \ e \ d]^T$$

$$\mathbf{u} = [df_{tz} \ df_{wx} \ df_{wy} \ df_{wz} \ dm_{tx} \ dm_{wx} \ dm_{ty} \ dm_{wy} \ dm_{tz} \ dm_{wz}]^T$$

Troviamo le matrici derivando la nostra $f(\mathbf{x} \ \mathbf{u} \ t)$, non lineare, per tutti questi valori e ponendo poi le condizioni di hovering.

$$\mathbf{A} = \frac{\delta f}{\delta x} \Big|_{x_0, u_n}$$

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\mathbf{B} = \frac{\delta f}{\delta u} \Big|_{x_0, u_n}$$

$$\mathbf{B} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/l_x & 1/l_x & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/l_y & 1/l_y & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/l_z & 1/l_z \\ 0 & 1/m & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/m & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/m & 0 & 0 & 1/m & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Che separando le variabili disturbo da quelle aleatorie si ottiene :

$$\dot{x} = \mathbf{A}x + \bar{\mathbf{B}}\bar{u} + \mathbf{D}u_d$$

$$\bar{\mathbf{u}} = [df_{tz} \quad dm_{tx} \quad dm_{ty} \quad dm_{tz}]^T$$

$$\mathbf{u}_d = [df_{wx} \quad df_{wy} \quad df_{wz} \quad dm_{wx} \quad dm_{wy} \quad dm_{wz}]^T$$

Ossia :

$$\bar{\mathbf{B}} = \frac{\delta f}{\delta \bar{\mathbf{u}}} |_{x_0, \bar{\mathbf{u}}_n}$$

$$\bar{\mathbf{B}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1/I_x & 0 & 0 \\ 0 & 0 & 1/I_y & 0 \\ 0 & 0 & 0 & 1/I_z \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1/m & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{D} = \frac{\delta f}{\delta \mathbf{u}_d} |_{x_0, \mathbf{u}_{dn}}$$

$$\mathbf{D} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/I_x & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/I_y & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/I_z \\ 1/m & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/m & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/m & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Che difatti se sommate vanno a ricostituire una 12x10.

Ultimo passaggio è scindere il vettore \bar{u} in modo tale che compaia la nostra vera variabile aleatoria ω , avremo dopo pochi semplici passaggi :

$$\mathbf{G} = \frac{\delta \bar{u}}{\delta \omega} \Big|_{x_0, \omega_0}$$

$$\mathbf{G} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 \cdot K_S \cdot \omega_0 \cdot D \cdot \sin \alpha & -2 \cdot K_S \cdot \omega_0 \cdot D \cdot \cos \alpha & -2 \cdot K_S \cdot \omega_0 \cdot D \cdot \sin \alpha & 2 \cdot K_S \cdot \omega_0 \cdot D \cdot \cos \alpha \\ 2 \cdot K_S \cdot \omega_0 \cdot D \cdot \cos \alpha & 2 \cdot K_S \cdot \omega_0 \cdot D \cdot \sin \alpha & -2 \cdot K_S \cdot \omega_0 \cdot D \cdot \cos \alpha & -2 \cdot K_S \cdot \omega_0 \cdot D \cdot \sin \alpha \\ -2 \cdot K_D \cdot \omega_0 & 2 \cdot K_D \cdot \omega_0 & -2 \cdot K_D \cdot \omega_0 & 2 \cdot K_D \cdot \omega_0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -2 \cdot K_S \cdot \omega_0 & -2 \cdot K_S \cdot \omega_0 & -2 \cdot K_S \cdot \omega_0 & -2 \cdot K_S \cdot \omega_0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Ossia compattato nella forma matriciale :

$$\dot{x} = \mathbf{A}x + \tilde{\mathbf{B}}d\omega + \mathbf{D}u_d$$

E il vettore $d\omega$:

$$d\omega = [d\omega_1 \ d\omega_2 \ d\omega_3 \ d\omega_4]^T$$

E con:

- $\bar{u} = \mathbf{G}d\omega$
- $\tilde{\mathbf{B}} = \bar{\mathbf{B}}\mathbf{G}$

9. CONTROLLO OTTIMO LINEARE QUADRATICO (LQ)

Obiettivo del controllo ottimo è determinare i segnali di controllo tali per cui il sistema da controllare soddisfi determinati vincoli fisici e allo stesso tempo renda minimo o massimo un qualche criterio scelto per misurarne le performance.

Il nostro schema di riferimento sarà un sistema MIMO così strutturato:

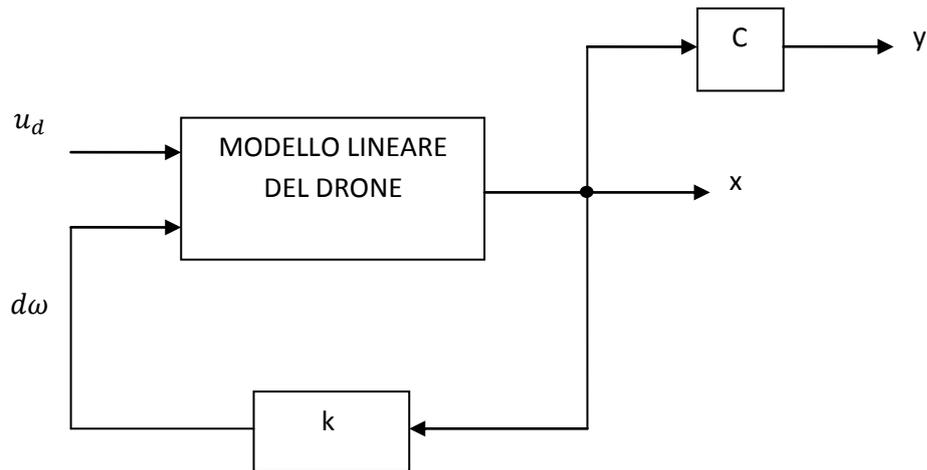


Figura 9.1 modello linearizzato in retroazione

La formulazione di un controllo ottimo richiede :

- Un modello che descriva al meglio il comportamento del sistema dinamico da controllare;
- Un indice di comportamento J (o criterio di ottimalità o indice di costo) che tenga conto delle specifiche desiderate e dalle esigenze del progettista;
- La dichiarazione delle condizioni al contorno e dei vincoli fisici sugli stati.

Questo ultimo non potrebbe essere presente.

Sia dato il sistema dinamico

$$\dot{x} = f(x(t), u(t), t);$$

Con x ed u vettori di dimensioni:

- $x_{n \times 1}$;
- $u_{r \times 1}$.

Si definisce così l'indice di comportamento

$$J = \underbrace{S(x(t_f), t_f)}_{\text{Peso sullo stato finale}} + \underbrace{\int_{t_0}^{t_f} f_0(x(t), u(t), t) dt}_{\text{Peso sull'evoluzione } [t_0, t_f]}$$

con le condizioni al contorno

$$x(t_0) = x_0, \text{ mentre sia } x(t_f), \text{ che } t_f \text{ sono liberi}$$

Si definisce quindi “Problema di controllo ottimo” la determinazione della funzione di :

$$\tilde{u}(t), \forall t \in [t_0, t_f]$$

in modo da minimizzare J.

9.1. CONTROLLO OTTIMO LQ A TEMPO INFINITO CON SET-POINT DIVERSO DALL'ORIGINE

Nello sviluppo delle equazioni generali, considerando un tempo finale t_f tendente all'infinito e con un set-point diverso dall'origine, si assume a tal fine:

Sistema lineare a tempo invariante:

$$\dot{x} = Ax + Gd\omega + Du_d$$

$$y = Cx$$

Una coppia di vettori ($d\omega_p; y_p$) che soddisfa le relazioni :

$$0 = Ax_p + \tilde{B}d\omega_p + Du_d$$

$$y_p = Cx_p$$

L'indice di comportamento assume perciò la forma :

$$J = \int_{t_0}^{\infty} \left\{ [d\omega(t) - d\omega_p]^T \cdot R \cdot [d\omega(t) - d\omega_p] + [x(t) - x_p]^T \cdot Q \cdot [x(t) - x_p] \right\} dt$$

Identificati:

R =costo degli attuatori ($R = R^T \geq 0; R \in R^{m \times m}$);

Q =costo dello stato ($Q = Q^T > 0; Q \in R^{r \times r}$).

Ponendo:

- $x_s = X(t) - X_p$;
- $d\omega_s = \omega(t) - \omega_p$;
- $y_s = Y(t) - Y_p$.

Si ottiene un problema a tempo infinito di questo tipo:

$$\dot{x}_s(t) = \mathbf{A}x_s(t) + \mathbf{G}d\omega_s(t) + \mathbf{D}u_d(t)$$

$$y_s = \mathbf{C}x_s(t)$$

E in fine l'indice di comportamento si evolve :

$$J = \int_{t_0}^{\infty} \{d\omega_s^T(t) \cdot \mathbf{R} \cdot d\omega_s(t) + x_s^T(t) \cdot \mathbf{Q} \cdot x_s(t)\} dt$$

Lo schema a blocchi risultante è :

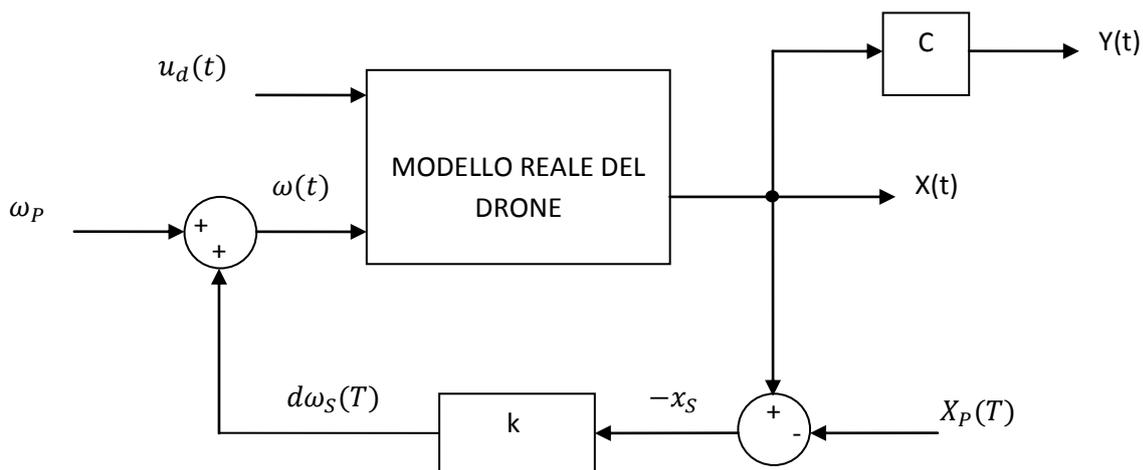


Figura 9.2 modello reale con set-points diversi dall'origine

Come visto, la soluzione del problema di controllo ottimo appena definito risulta :

$$d\tilde{\omega}(t) = -\mathbf{K}x(t)$$

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{S}$$

$$\mathbf{S}\mathbf{A} + \mathbf{A}^T\mathbf{S} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S} + \mathbf{C}^T\mathbf{Q}\mathbf{C} = 0 ; \quad \mathbf{S} > 0$$

Queste equazioni matriciali si risolvono con il metodo Riccati, che viene svolto direttamente dal software tramite la funzione LQR.

$$K = LQR(A, B, Q, R)$$

Un particolare importante che si apprende dallo schema riguarda proprio il regolatore K, infatti questo viene realizzato sul modello linearizzato e di seguito implementato su quello reale.

L'espedito che rende funzionante questo metodo è aggiungere le condizioni iniziali $(X_p, d\omega_p)$ all'uscita ed all'ingresso del regolatore per avere le stesse condizioni di input ed output dal regolatore identiche a quelle nel sistema linearizzato.

Quindi basterà creare il regolatore una sola volta per poi poterlo usare in ambo i casi.

Vediamo ora come sono strutturate le matrici Q e R in modo tale da poterle compilare.

9.3. COMPILAZIONE MATRICI Q ED R

Per quanto concerne la matrice R questa è correlata alle variabili aleatorie, ovvero quelle su cui noi possiamo intervenire, ovvero le velocità angolari dei rotori, quindi dovrà moltiplicare prima un vettore 1x4 e poi uno 4x1.

La sua dimensione quindi risulta essere una 4x4 e siccome il nostro interesse è di ottenere un sistema lineare, ovvero un coefficiente che moltiplica ogni singola variabile risulterà essere anche diagonale. Stesse considerazioni valgono per la matrice Q ma al contrario avendo come elemento della moltiplicazione un vettore di dimensione 12 risulterà essere una matrice quadrata diagonale 12x12.

Ecco riportati di seguito i due sistemi matriciali esplicitati :

$$\begin{pmatrix} d\omega_1 \\ d\omega_2 \\ d\omega_3 \\ d\omega_4 \end{pmatrix} \begin{pmatrix} r_1 & 0 & 0 & 0 \\ 0 & r_2 & 0 & 0 \\ 0 & 0 & r_3 & 0 \\ 0 & 0 & 0 & r_4 \end{pmatrix} \begin{pmatrix} d\omega_1 & d\omega_2 & d\omega_3 & d\omega_4 \end{pmatrix}$$

$$\begin{pmatrix} \varphi \\ \theta \\ \psi \\ p \\ q \\ r \\ u \\ v \\ w \\ n \\ e \\ d \end{pmatrix} \begin{pmatrix} q_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & q_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & q_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & q_6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & q_7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_{10} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_{11} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_{12} \end{pmatrix} \begin{pmatrix} \varphi \\ \theta \\ \psi \\ p \\ q \\ r \\ u \\ v \\ w \\ n \\ e \\ d \end{pmatrix}$$

Non resta che determinare i singoli coefficienti delle matrici.

I valori da attribuire sono del tutto arbitrari, quindi di regola si compilano le matrici inverse inserendo i valori minimi che si desiderano.

Per esempio se l'obiettivo è ottenere una posizione finale il più prossima a quella iniziale, dell'ordine di qualche millimetro, allora nei valori corrispondenti di ned (*coefficienti* q_7, q_8, q_9) inseriremo 0.001 m.

Una volta scelti tutti i valori si mette in esecuzione il programma ed in base ai risultati estrapolati si modificano finché non rientrano nelle tolleranze desiderate.

I corrispondenti valori ottenuti dopo i molteplici test sono :

$$\left\{ \begin{array}{l} q_1, q_2, q_3 = 0.01 \\ q_4, q_5, q_6 = 1 \\ q_7, q_8, q_9 = 1 \\ q_{10}, q_{11}, q_{12} = 0.001 \\ r_1, r_2, r_3, r_4 = 0.01 \end{array} \right.$$

10. IMPLEMENTAZIONE DEL MODELLO IN MATLAB

Arrivati a questo punto abbiamo tutti gli elementi per iniziare a costruire il modello da implementare poi nell'hardware del drone.

Il software utilizzato sarà Simulink, un sistema di programmazione incluso nell'ambiente Matlab.

Si utilizza questo programma per due principali ragioni, facile ed intuitivo da usare e tramite un'opportuna operazione di trascrizione, trasforma lo script formulato in un testo di codice leggibile dall'hardware Harduicopter.

I casi sviluppati saranno due:

- Linearizzato: utile per le stime attorno al caso di hovering e permette di sviluppare il modello del regolatore da utilizzare poi nell'altro modello;
- Reale: funzionale per i tutti i casi, quindi fornisce informazioni e stime anche fuori dal caso delle piccole oscillazioni e per i casi di moto rettilineo che vedremo più avanti.

Una volta realizzati i modelli per valutarne la validità basterà fare dei test e verificare se i risultati sono veritieri o meno.

Una volta che il modello nel caso di hovering è corretto si procederà poi ad estenderlo per tutti i casi interessati.

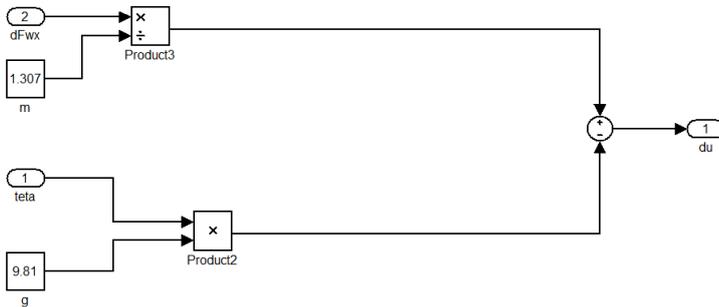
10.1. SVILUPPO DEL MODELLO LINEARE

Come primo step si costruiscono le equazioni principali del moto.

Basterà costruire un blocco per ogni singolo sistema e poi collegarli correttamente fra loro per svolgere l'intero pacchetto di equazioni ed ottenere i risultati desiderati.

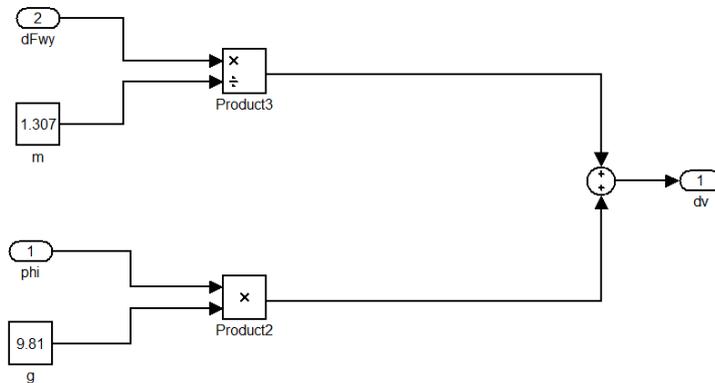
I sistemi studiati fin'ora si trasformano quindi:

- Equazioni delle forze assi corpo:



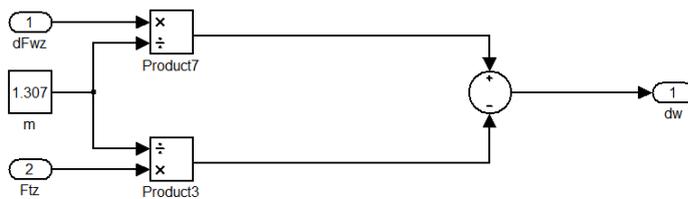
$$\dot{u} = -g \cdot \vartheta + \frac{df_{wx}}{m}$$

Figura 10.1 equazione di \dot{u}



$$\dot{v} = g \cdot \phi + \frac{df_{wy}}{m}$$

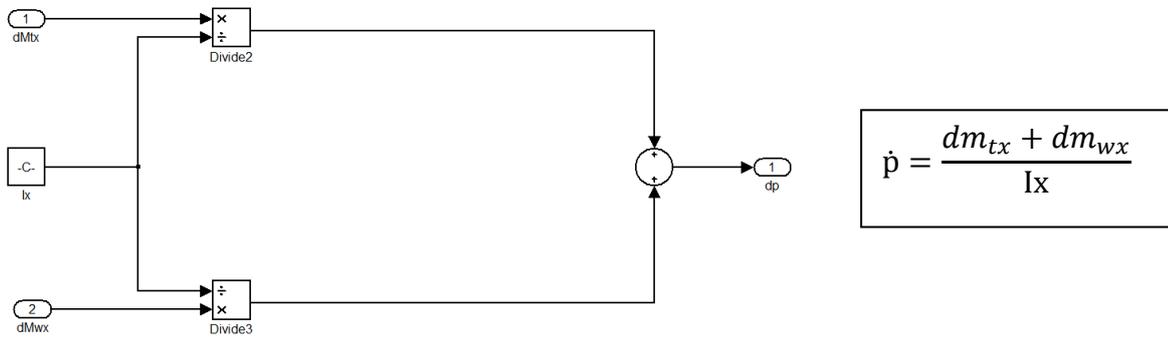
Figura 10.2 equazione di \dot{v}



$$\dot{w} = \frac{-df_{wz} + df_{wz}}{m}$$

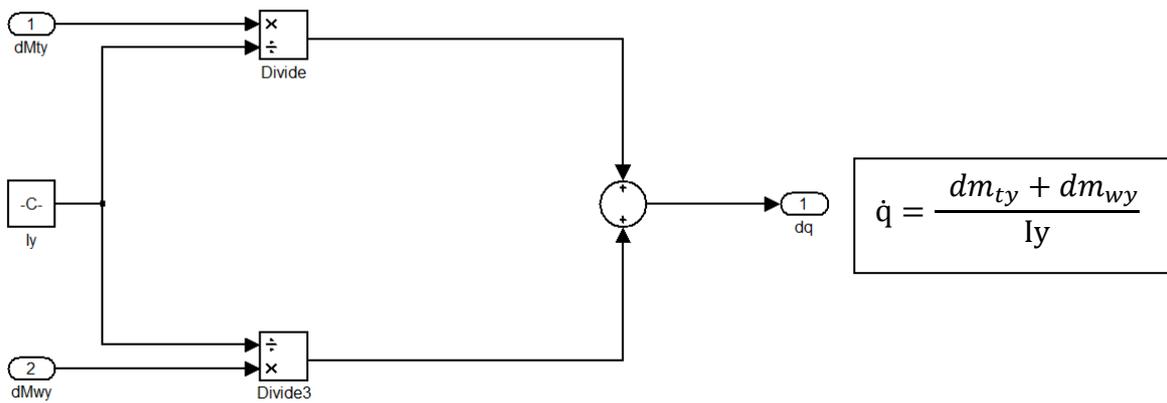
Figura 10.3 equazione di \dot{w}

- Equazioni delle momenti assi corpo:



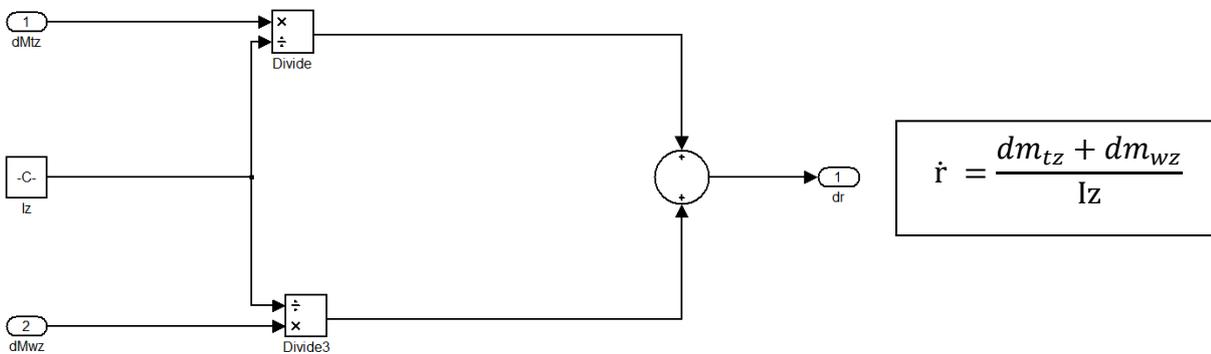
$$\dot{p} = \frac{dm_{tx} + dm_{wx}}{I_x}$$

Figura 10.4 equazione di \dot{p}



$$\dot{q} = \frac{dm_{ty} + dm_{wy}}{I_y}$$

Figura 10.5 equazione di \dot{q}



$$\dot{r} = \frac{dm_{tz} + dm_{wz}}{I_z}$$

Figura 10.6 equazione di \dot{r}

E per ricavare le restanti equazioni nel caso lineare basterà integrare le equazioni appena citate.

Ecco di seguito il sistema completo delle equazioni del moto linearizzate:

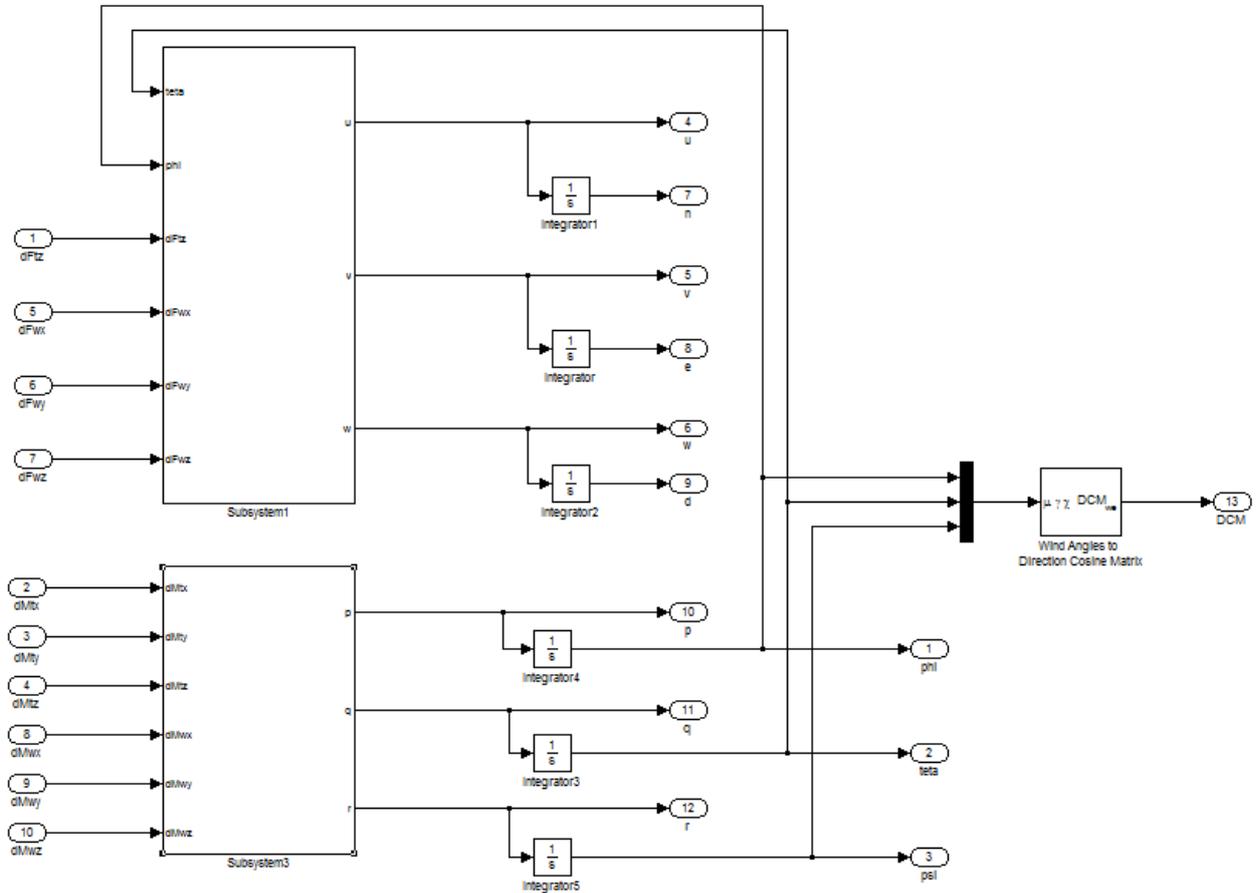


Figura 10.7 sistema a blocchi per il calcolo delle equazioni del moto linearizzate

Ora non rimane che sviluppare un blocco per gestire le forze e i momenti sviluppati dalle forze di propulsione delle eliche.

In ingresso possiamo immaginare di assumere le velocità di rotazione, mentre in uscita avremo i valori desiderati di: df_{tz} ; dm_{tx} ; dm_{ty} ; dm_{tz} .

Siccome la velocità di rotazione dei rotori non sarà costante, ma dipenderà dalla massa del velivolo per ogni caso di hovering, sarà necessario sviluppare una funzione che prende in ingresso la velocità angolare ω e restituisce i valori di forza e momenti per ogni elica.

Tale funzione è rappresentata dai blocchi Matlab Fcn.

Per il calcolo dei momenti che producono rotazioni lungo pitch e roll invece si usa il blocco Subsystem dove sono implementati i calcoli base.

Otteniamo quindi il seguente schema a blocchi:

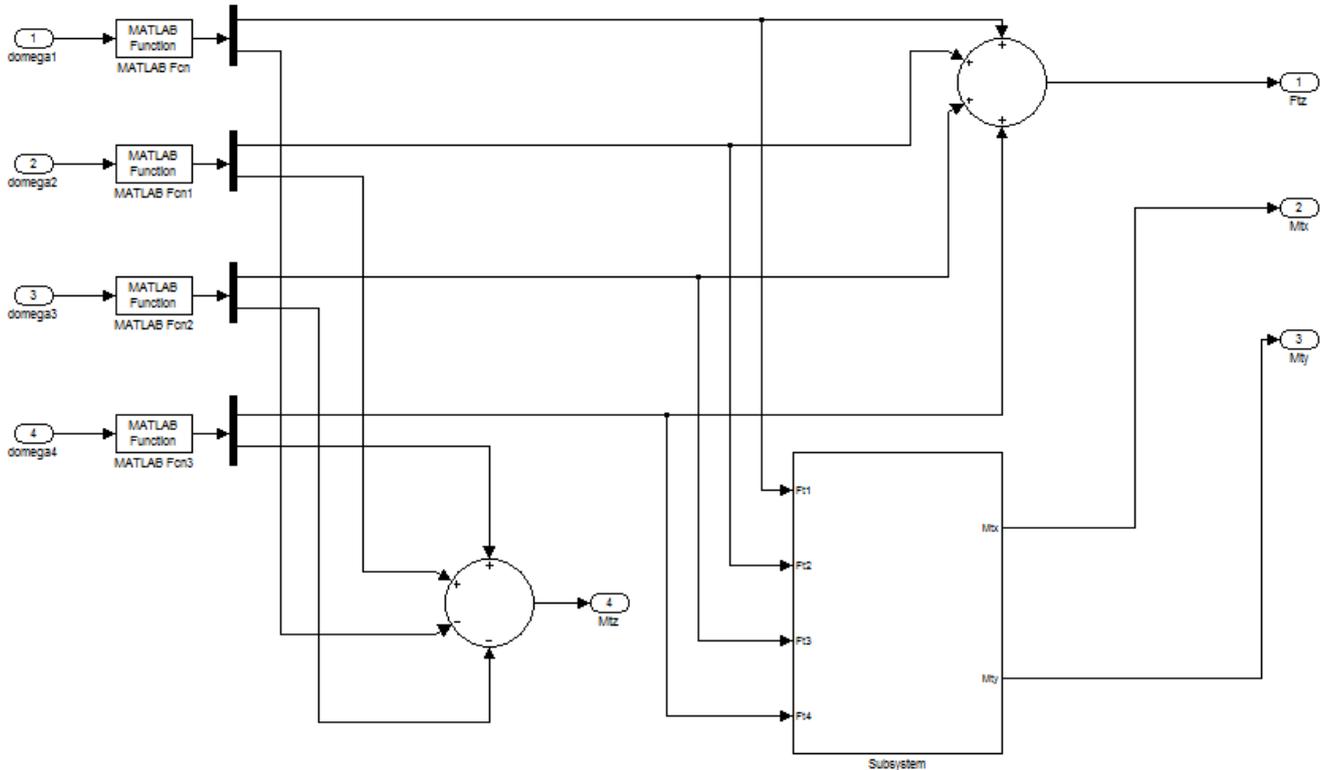


Figura 10.8 sistema a blocchi per il calcolo delle propulsioni e dei momenti generato dalle eliche

Quindi vediamo ora la struttura del blocco Matlab Fcn per il calcolo delle portanze e dei momenti.

```
function y = interp_lin( dom )

N=2000:1000:20000;

t=[0.156,0.343,0.601,0.930,1.341,1.851,2.433,3.073,3.77,4.528,5.406,6.425,7.502,8.607,9.905,11.295,12.8,14.259,16.53];
m=[0.162,0.354,0.608,0.913,1.281,1.793,2.262,2.851,3.499,4.21,5.027,5.98,7.001,8.071,9.344,10.688,12.299,13.95,16.398];

%converti unità di misura

Ti=t.*4.45;
Mi=m.*4.45*2.54/100;

T0=1.307*9.81/4;
no=interp1(Ti,N,T0);
n=no+dom*30/pi;
T=interp1(N,Ti,n);
M=interp1(N,Mi,n);
dT=T-T0;

if dT<10^-5
    dT=0;
end

y = [ dT,M ];
```

I valori datati all'interno dei vettori "t" ed "m" sono estrapolati da una tabella (datasheet) fornita dal produttore delle eliche (APC propellers : <http://www.apcprop.com/v/index.html>).

Quindi una volta nota la marca ed il modello dell'elica montata sul quadricottero, si estrapolano i dati desiderati dalla relativa tabella in base al passo ed alla lunghezza del tipo scelto (10x4.7 inc).

Tali entità sono dati per un dato giro al minuto (rpm) in un range che va da 2000 a 20000 rpm.

I valori ottenuti però sono in unità di misura non convenzionali, si trasducono in N e Nm le forze ed i momenti tramite le corrispettive conversioni.

Non resta che convertire la forza peso del drone (T0), ripartita equamente su ogni braccio, in una velocità in rpm per ogni elica per avere overing tramite la funzione Matlab "interp1" che realizza un interpolazione dei valori richiesti.

Ora si convertono il delta delle velocità angolari in ingresso alla function in rpm e si sommano quelle in caso di overing ottenendo così una velocità che ci fornisce la spinta e le coppie nell'assetto di funzionamento di tale istante.

Sempre tramite la funzione "interp1" si ottengono i relativi valori di delta delle propulsioni e delle coppie che vengono restituite all'output.

Si sottraggono i valori di thrust in caso di volo nell'istante di tempo in esame e quelli di overing e si ottengono così i delta di propulsione che avevamo come obiettivo.

In fine non rimane che creare il regolatore che sarà il cuore dello script e che verrà poi utilizzato per retroazionare il modello reale.

Tale blocco dovrà prendere in ingresso le variabili di stato e restituire i valori di velocità angolare tali da correggere le deviazioni di traiettoria dalla posizione di hovering.

Per convenire a questo obiettivo è necessario sviluppare una funzione in Matlab che svolga l'equazione di Riccati e restituisca la matrice K tale che :

$$u = -K \cdot x$$

Si mostra quindi nella pagina seguente lo script di Matlab per il calcolo di tale matrice.

Siccome i risultati saranno calcolati con decimali prossimi a 10^{-15} , per velocizzare il tempo di compilazione si approssimano i risultati ad un valore di 10^{-5} tramite le funzioni clean.

```

function [ K ] = riccati( al )

M=1.307;
Ix=0.0216316;
Iy=0.0243731;
Iz=0.0397019;
D=0.335;

N=2000:1000:20000;

t=[0.156,0.343,0.601,0.930,1.341,1.851,2.433,3.073,3.77,4.528,5.406,6.425,7.502,8.607,9.905,11.295,12.8,14.259,16.53];
m=[0.162,0.354,0.608,0.913,1.281,1.793,2.262,2.851,3.499,4.21,5.027,5.98,7.001,8.071,9.344,10.688,12.299,13.95,16.398];

Ti=t.*4.45;
Mi=m.*4.45*2.54/100;

T0=M*9.81/4;
no=interp1(Ti,N,T0);
om0=no*pi/30;
Ks=T0/(om0^2);
M0=interp1(N,Mi,no);
Kd=M0/(om0^2);

A=[0 0 0 1 0 0 0 0 0 0 0 0;
    0 0 0 0 1 0 0 0 0 0 0 0;
    0 0 0 0 0 1 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0;
    0 -9.81 0 0 0 0 0 0 0 0 0 0;
    9.81 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 1 0 0 0 0 0;
    0 0 0 0 0 0 0 1 0 0 0 0;
    0 0 0 0 0 0 0 0 1 0 0 0;
    0 0 0 0 0 0 0 0 0 1 0 0];

B=[0 0 0 0;
    0 0 0 0;
    0 0 0 0;
    2*Ks*om0*D*sin(al)/Ix -2*Ks*om0*D*cos(al)/Ix -2*Ks*om0*D*sin(al)/Ix 2*Ks*om0*D*cos(al)/Ix;
    2*Ks*om0*D*cos(al)/Iy 2*Ks*om0*D*sin(al)/Iy -2*Ks*om0*D*cos(al)/Iy -2*Ks*om0*D*sin(al)/Iy;
    -2*Kd*om0/Iz 2*Kd*om0/Iz -2*Kd*om0/Iz 2*Kd*om0/Iz;
    0 0 0 0;
    0 0 0 0;
    -2*Ks*om0/M -2*Ks*om0/M -2*Ks*om0/M -2*Ks*om0/M;
    0 0 0 0;
    0 0 0 0;
    0 0 0 0];

r=[0.01 0 0 0;
    0 0.01 0 0;
    0 0 0.01 0;
    0 0 0 0.01];

q=[0.01 0 0 0 0 0 0 0 0 0 0 0;
    0 0.01 0 0 0 0 0 0 0 0 0 0;
    0 0 0.01 0 0 0 0 0 0 0 0 0;
    0 0 0 1 0 0 0 0 0 0 0 0;
    0 0 0 0 1 0 0 0 0 0 0 0;
    0 0 0 0 0 1 0 0 0 0 0 0;
    0 0 0 0 0 0 1 0 0 0 0 0;
    0 0 0 0 0 0 0 1 0 0 0 0;
    0 0 0 0 0 0 0 0 1 0 0 0;
    0 0 0 0 0 0 0 0 0 0.001 0 0;
    0 0 0 0 0 0 0 0 0 0 0.001 0;
    0 0 0 0 0 0 0 0 0 0 0 0.001];

Q=inv(q);
R=inv(r);

[K] = LQR(A,B,Q,R);

end

```

Ed il modello a blocchi per risolvere l'equazione matriciale ed ottenere le velocità angolari desiderate è:

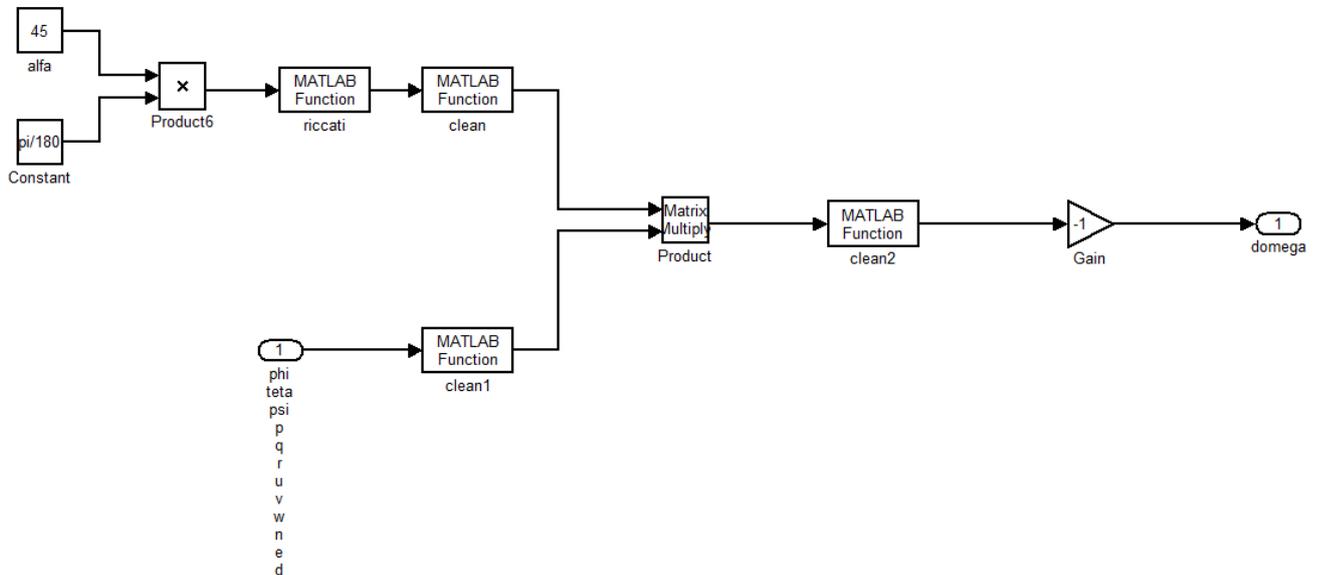


Figura 10.9 sistema a blocchi per rappresentare $u = -K \cdot x$

Ora non rimane che collegare il modello linearizzato del drone in retroazione con il regolatore ed otteniamo lo schema mostrato al capitolo 9.

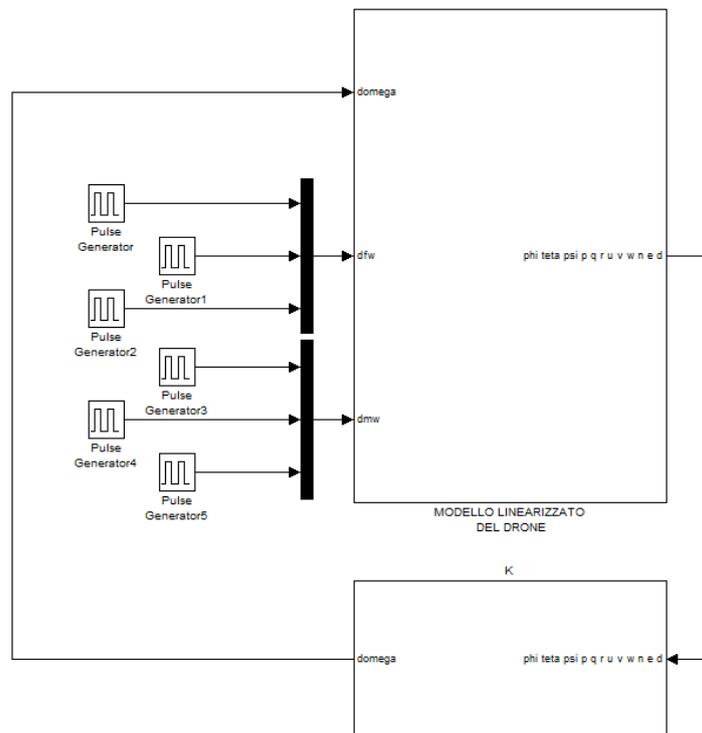


Figura 10.10 sistema a blocchi in feedback del modello linearizzato

10.2 SVILUPPO DEL MODELLO REALE

Per questo tipo di modellazione, essendo un caso reale, è necessario usare le equazioni originali del moto, quindi senza linearizzazione e senza imporre condizioni iniziali.

Riportiamo ora di seguito le equazioni caratteristiche del moto.

- Equazioni delle forze assi corpo:

$$\dot{U} = -Q \cdot W + R \cdot V - g \cdot \sin \theta + F_{wx} / m$$

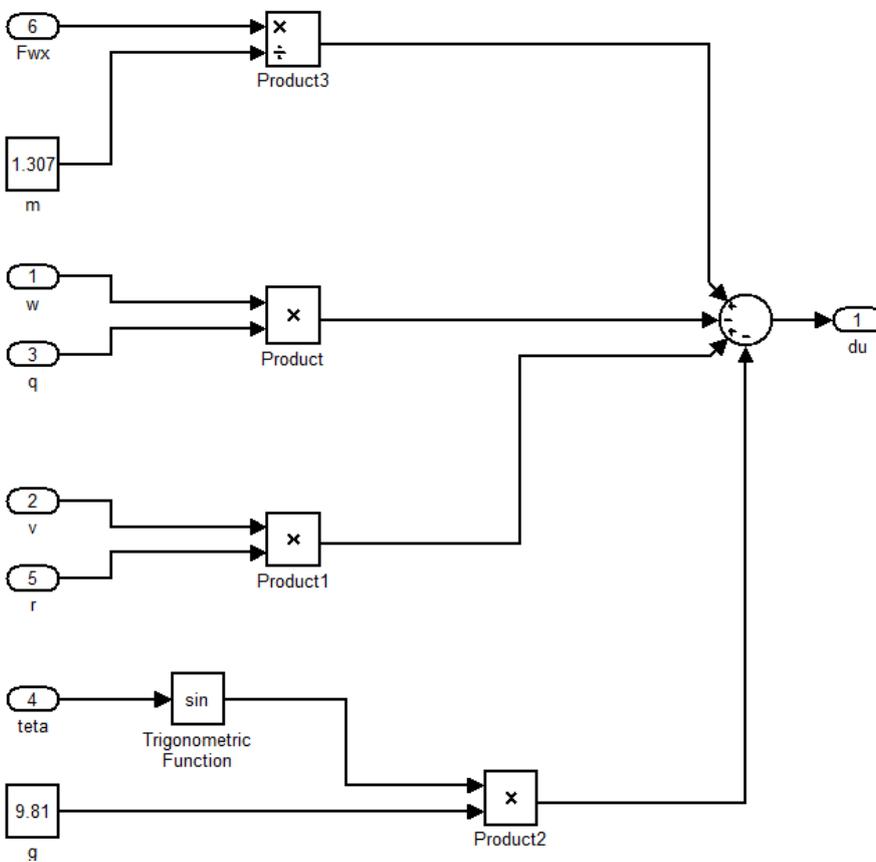


Figura 10.11 equazione di \dot{U}

$$\dot{V} = P \cdot W - R \cdot U + g \cdot \cos \theta \cdot \sin \phi + F_{wy}/m$$

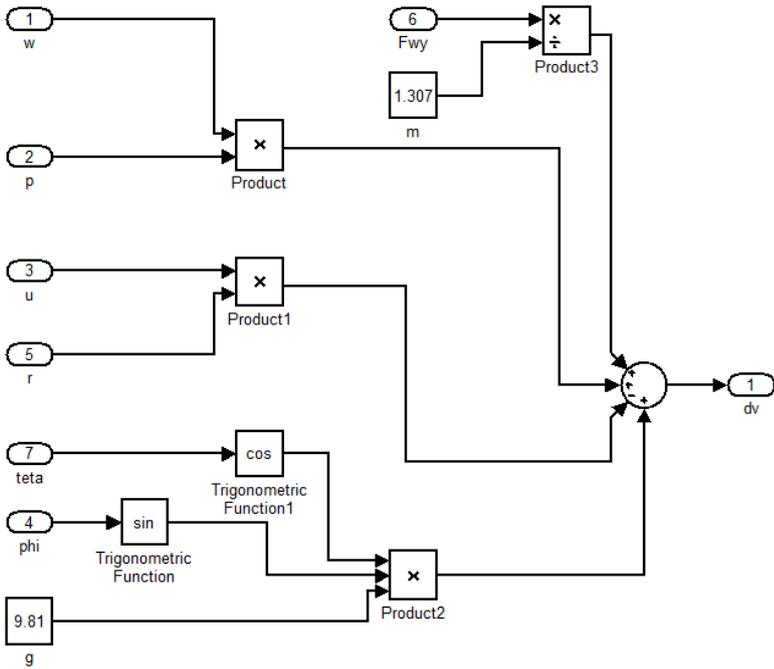


Figura 10.12 equazione di \dot{V}

$$\dot{W} = Q \cdot U - P \cdot V + g \cdot \cos \theta \cdot \cos \phi + (-F_{tz} + F_{wx})/m$$

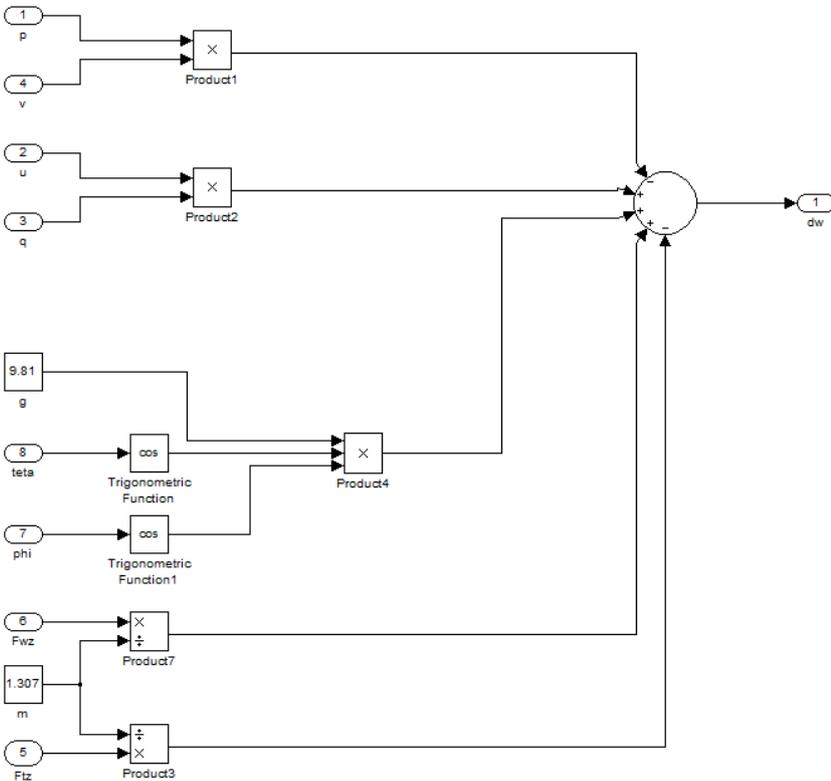


Figura 10.13 equazione di \dot{W}

- Equazioni dei momenti assi corpo:

$$\dot{P} = \frac{I_y - I_z}{I_x} R \cdot Q + \frac{(M_{tx} + M_{wx})}{I_x}$$

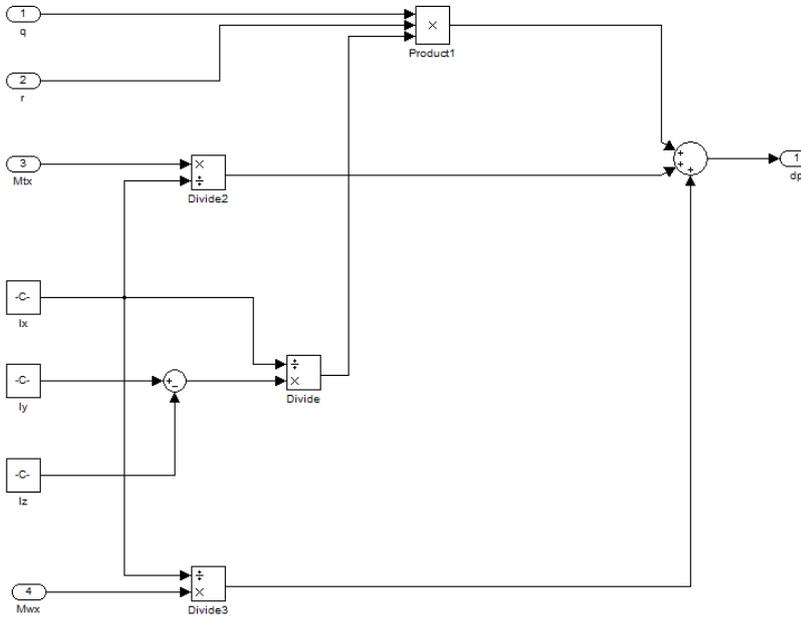


Figura 10.14 equazione di \dot{P}

$$\dot{Q} = \frac{I_z - I_x}{I_y} P \cdot R + \frac{(M_{ty} + M_{wy})}{I_y}$$

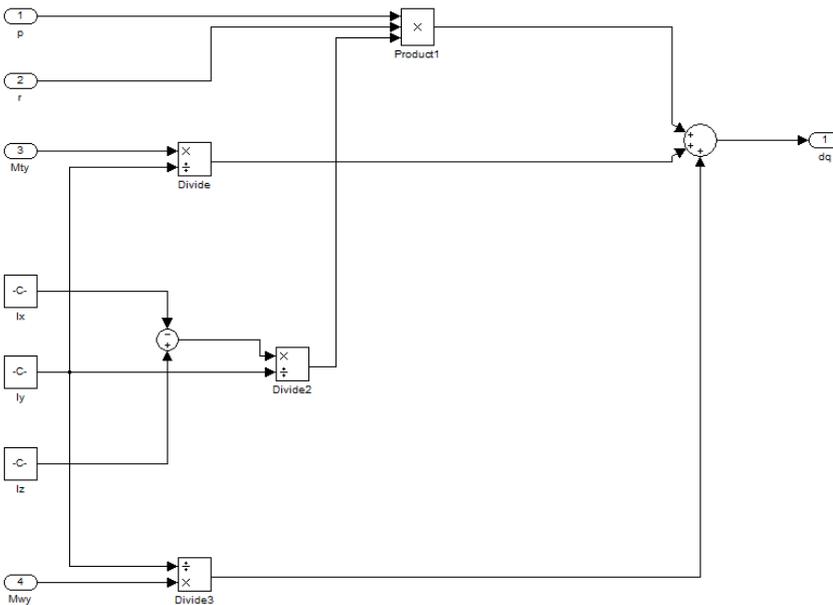


Figura 10.15 equazione di \dot{Q}

$$\dot{R} = \frac{I_y - I_x}{I_z} P \cdot Q + \frac{(M_{ty} + M_{wz})}{I_z}$$

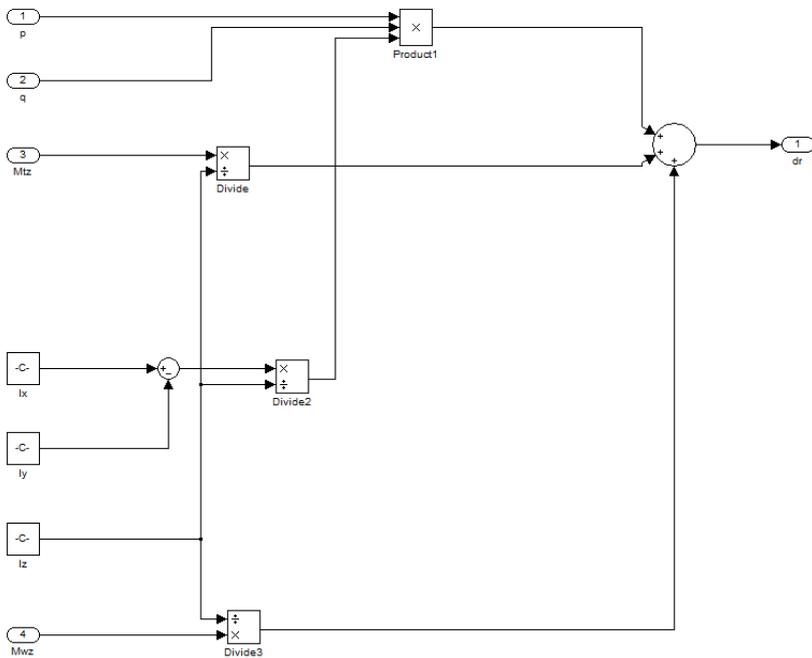


Figura 10.16 equazione di \dot{R}

- Equazioni di Eulero per l'assetto:

$$\dot{\phi} = P + Q \cdot \tan\theta \cdot \sin\phi + R \cdot \tan\theta \cdot \cos\phi$$

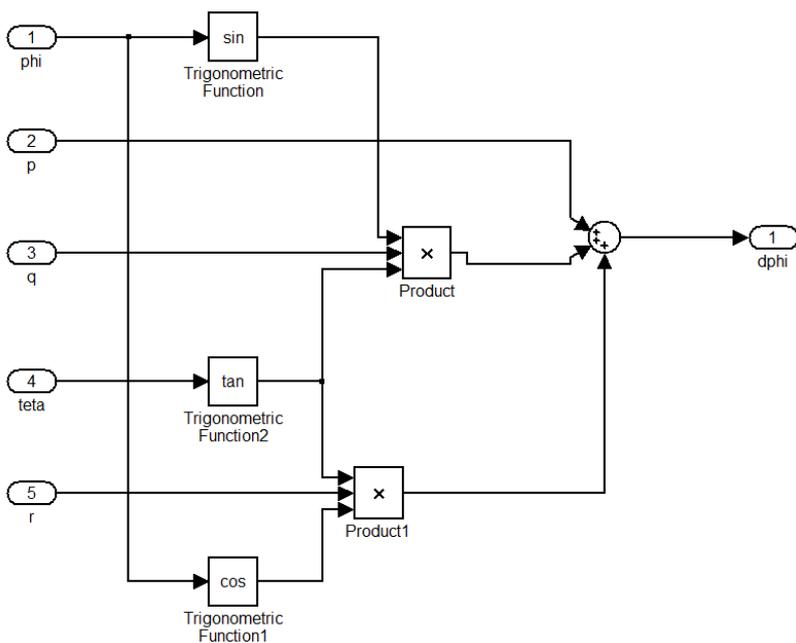


Figura 10.17 equazione di $\dot{\phi}$

$$\dot{\theta} = Q \cdot \cos\theta - R \cdot \sin\phi$$

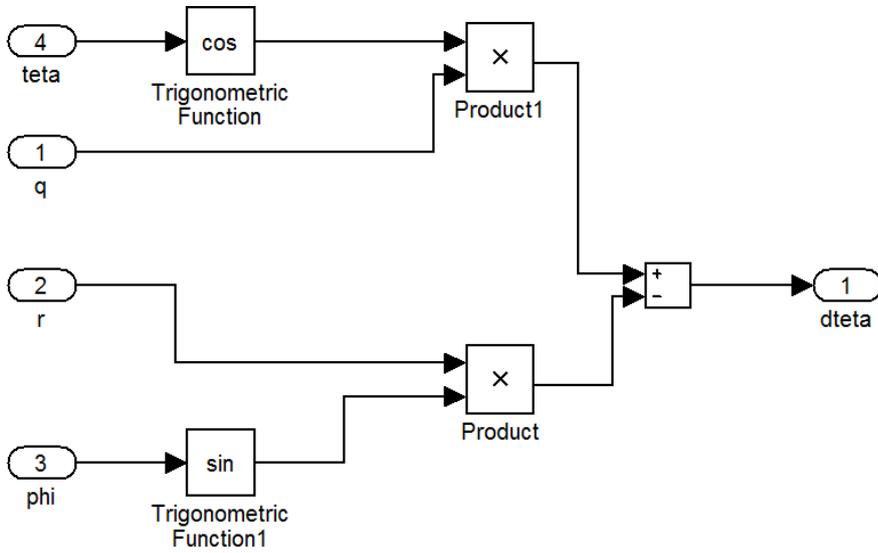


Figura 10.18 equazione di $\dot{\theta}$

$$\dot{\psi} = \frac{\sin\phi}{\cos\theta} Q + \frac{\cos\phi}{\cos\theta} R$$

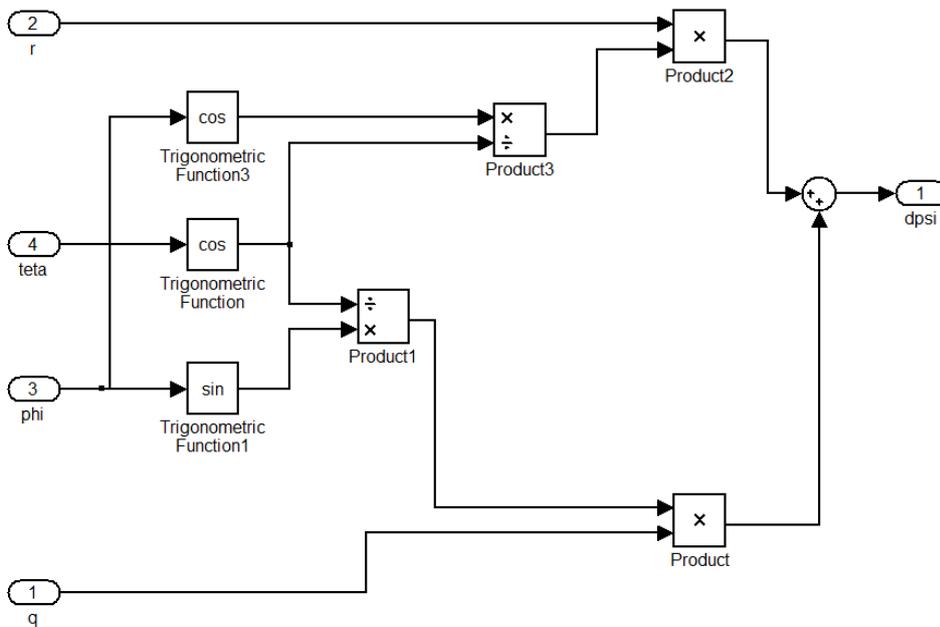


Figura 10.19 equazione di $\dot{\psi}$

- Equazioni che esprimono le velocità lineari in assi terra a partire da quelle assi corpo:

Per queste tipologie di equazioni era più conveniente sviluppare delle funzioni Matlab che svolgessero immediatamente in una riga di codice i valori di N,E e D.

Il modello finale risultante, eguale a quello del capitolo 9.1, sarà in Simulink:

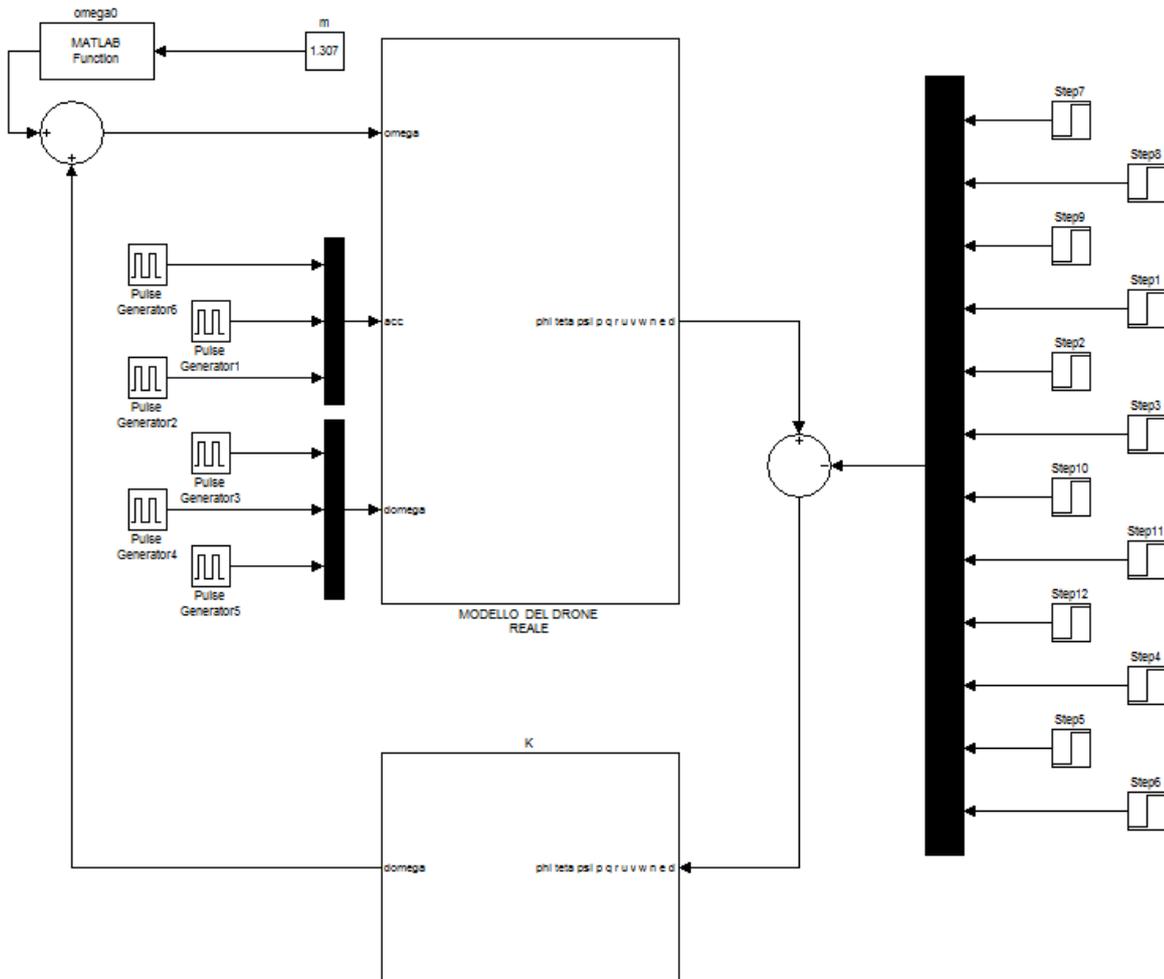


Figura 10.20 sistema a blocchi in feedback del modello reale

Come si può notare a sinistra abbiamo le sei tipologie di impulso di disturbo dovute dal vento (per la terna di assi NED), mentre a destra abbiamo le condizioni iniziali desiderate (per il tracking).

Omega0 è un blocco che prende in ingresso la massa del drone e fornisce la velocità angolare di ciascun rotore affinché si abbia overing.

Non è altro che un blocco che contiene i valori di thrust per data velocità periferica e tramite un interpolazione restituisce, per una certa massa, il corrispettivo valore di velocità angolare per ottenere overing.

Si ricorda inoltre che il blocco K è il medesimo del caso linea rizzato.

10.3. TEST E VERIFICA DEL MODELLO

Applicando una forza di 1 N lungo l'asse X per un periodo pari all'1% di quello della simulazione (20s) si ottiene il seguente risultato :

Φ, Θ, Ψ :

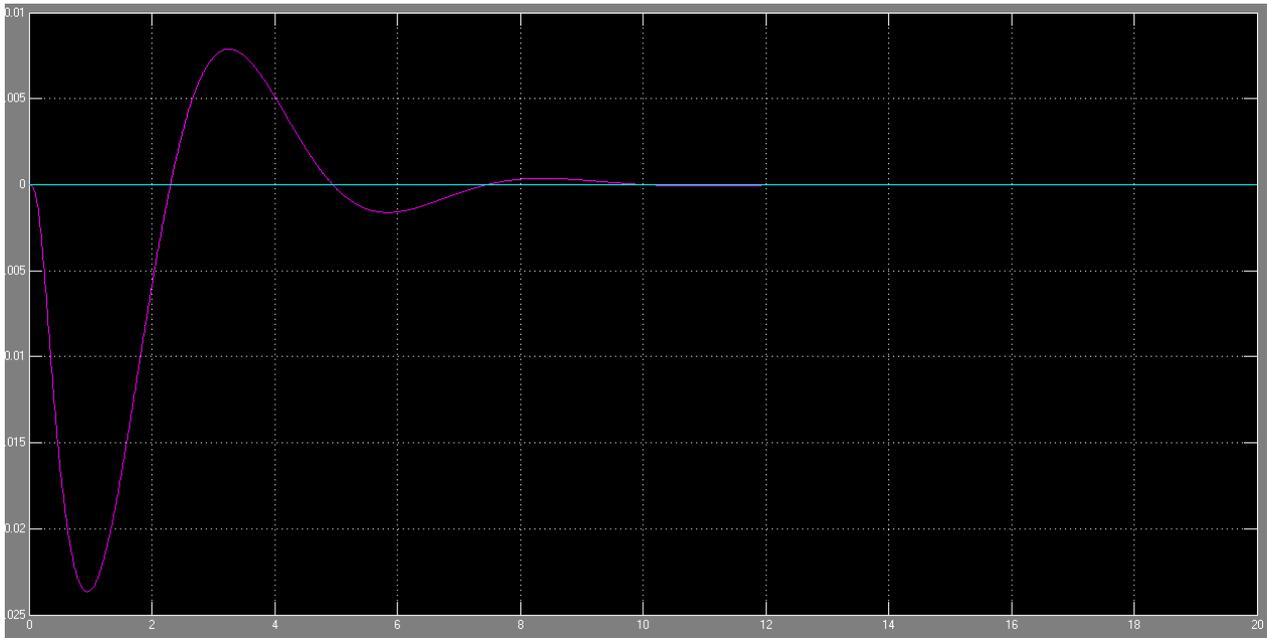


Figura 10.21 grafico di risposta all'impulso in Φ, Θ, Ψ

P, Q, R :

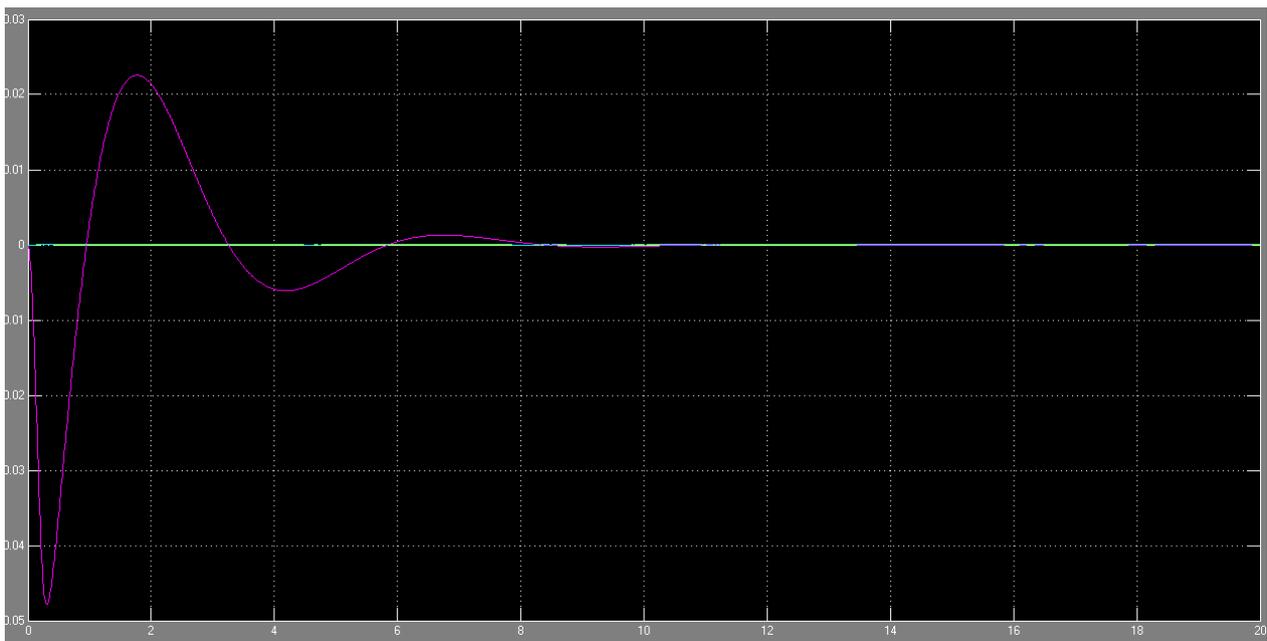


Figura 10.22 grafico di risposta all'impulso in P, Q, R

U,V,W :

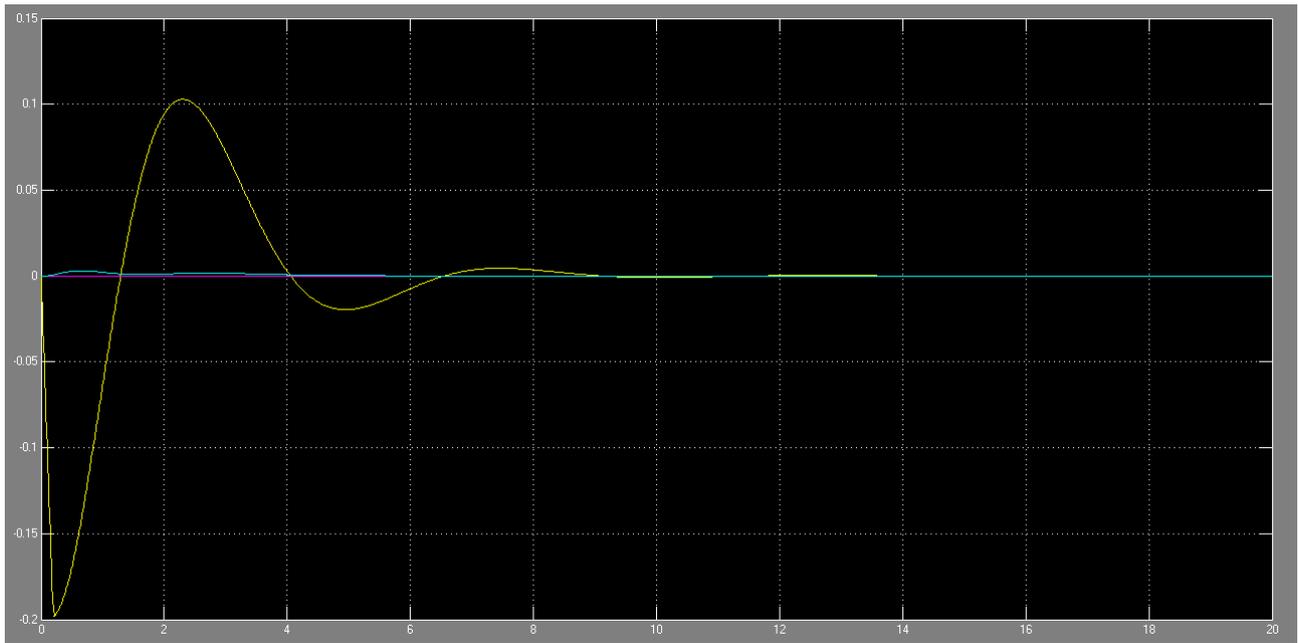


Figura 10.23 grafico di risposta all'impulso in U,V,W

N,E,D :

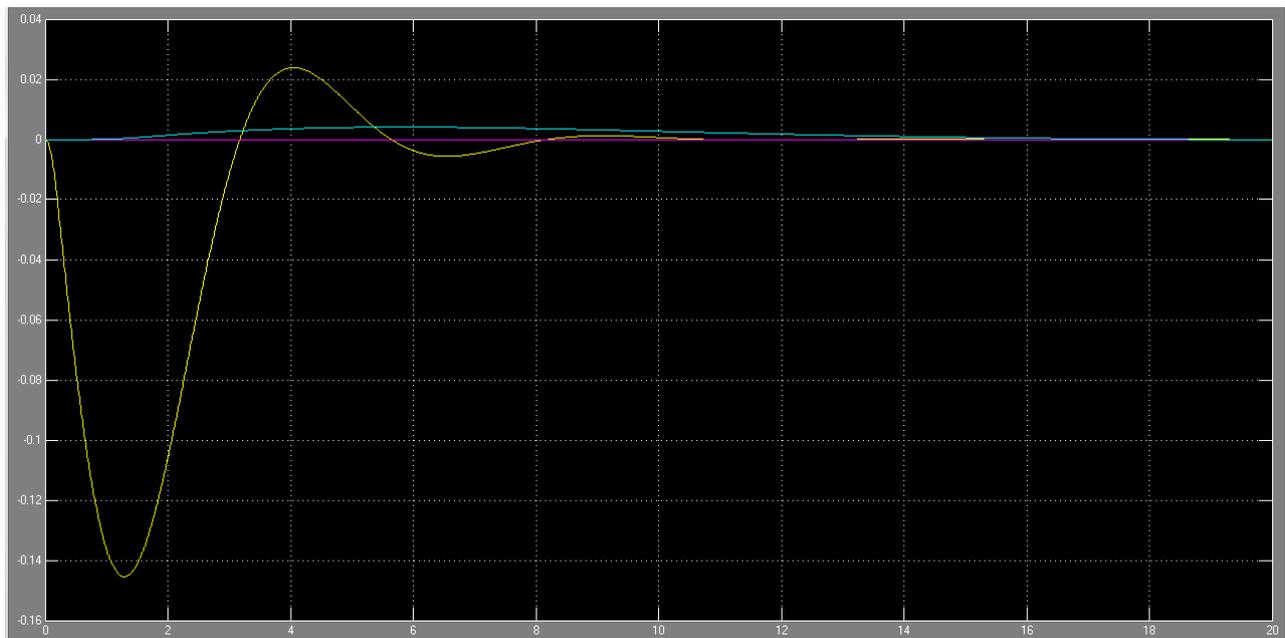


Figura 10.24 grafico di risposta all'impulso in N,E,D

I grafici mostrano chiaramente una stabilizzazione asintotica verso il valore di hovering.

Lo stesso risultato si ottiene per tutti i tipi di forze e momenti applicati, valutiamo anche una combinazione di entità, per esempio forza lungo X e Y (1N) e coppia lungo l'asse di roll (1 Nm) otteniamo :

Φ, Θ, Ψ :

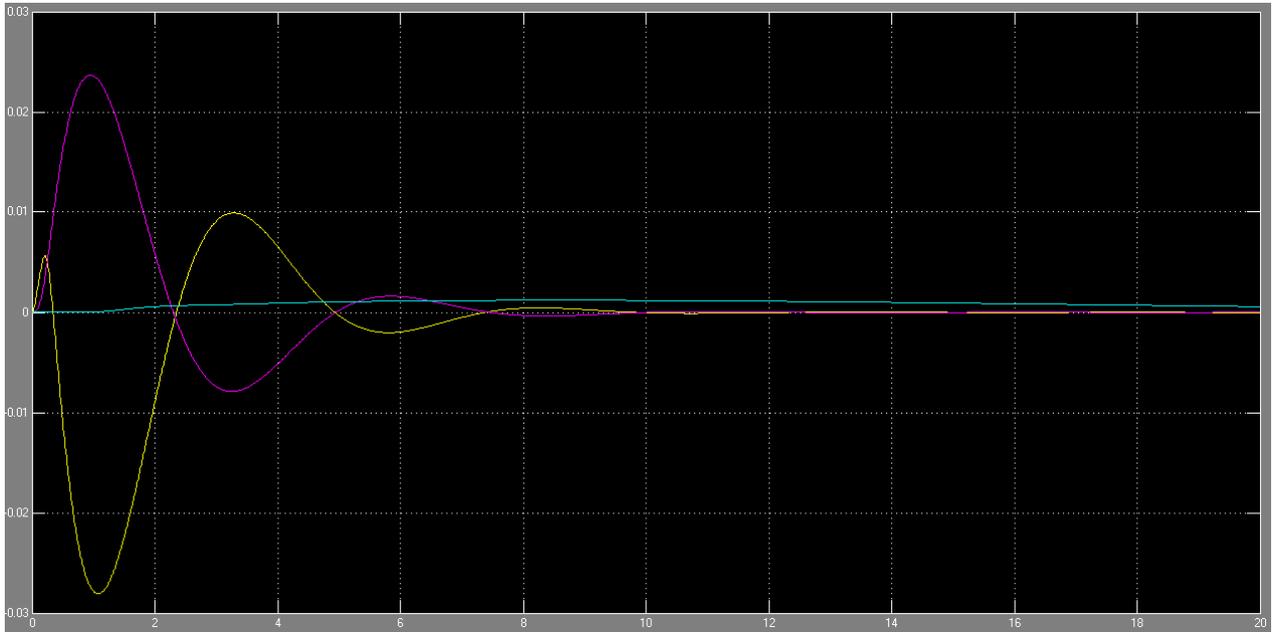


Figura 10.25 grafico di risposta all'impulso di coppia in Φ, Θ, Ψ

P, Q, R :

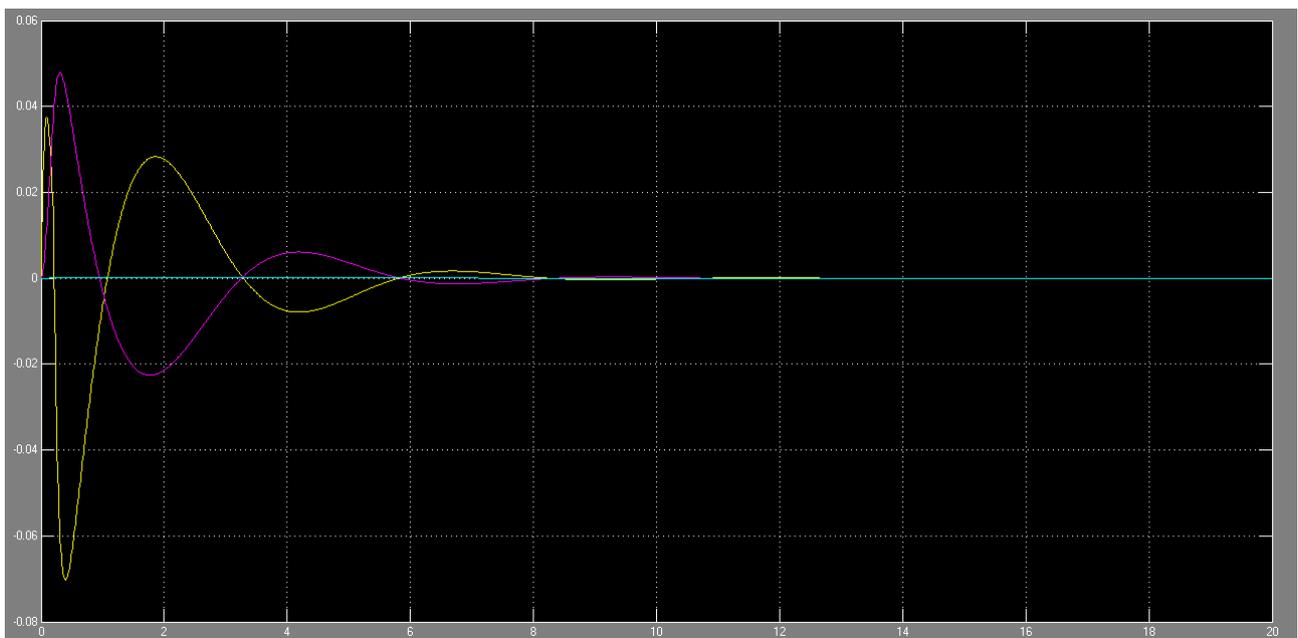


Figura 10.26 grafico di risposta all'impulso di coppia in in P, Q, R

U,V,W :

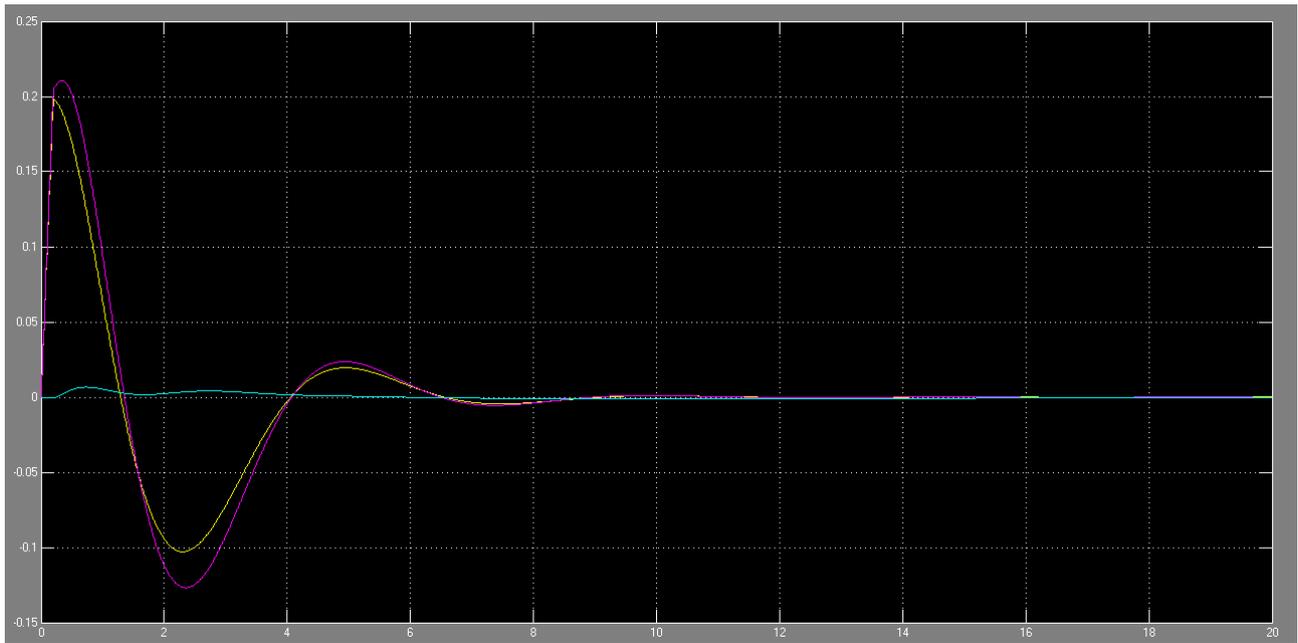


Figura 10.27 grafico di risposta all'impulso di coppia in U,V,W

N,E,D :

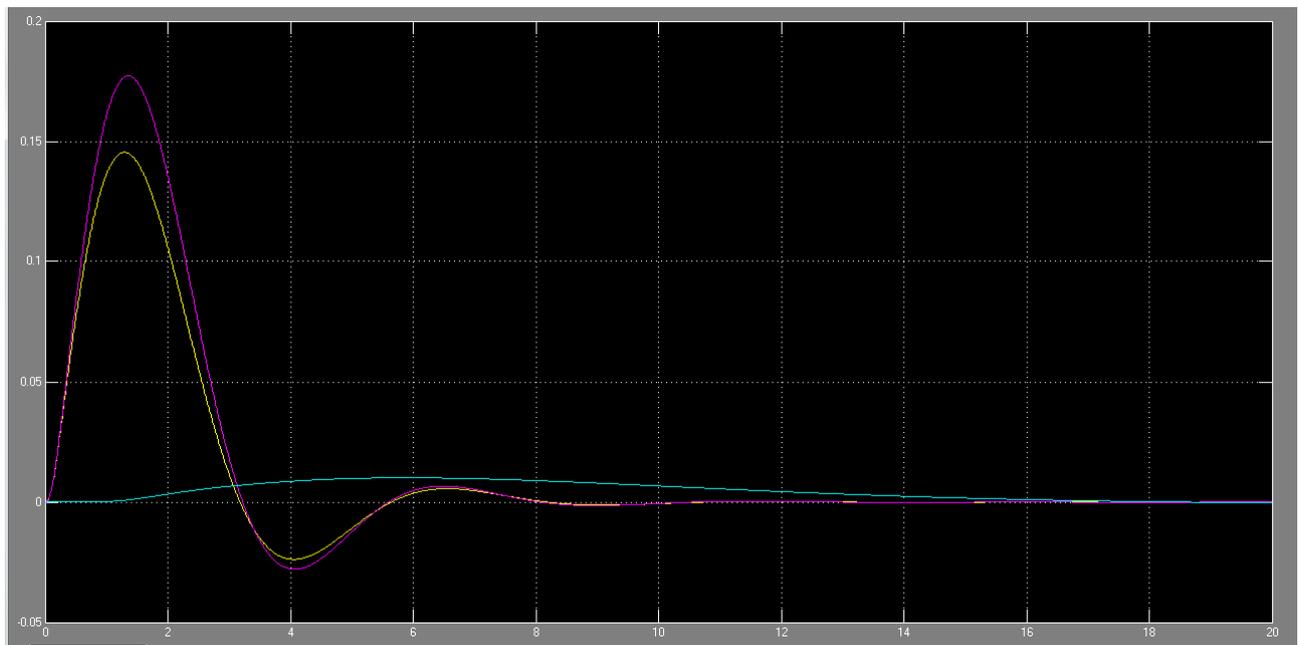


Figura 10.28 grafico di risposta all'impulso di coppia in N,E,D

Risultando ancora asintoticamente stabile.

Ora saranno mostrati i limiti di questo regolatore, ovvero il caso in cui la forza non sarà un impulso ma bensì un'entità costante nel tempo, ne fa esempio un vento lungo l'asse X di valore 1N per una tempo di test pari a 20s.

Φ, Θ, Ψ :

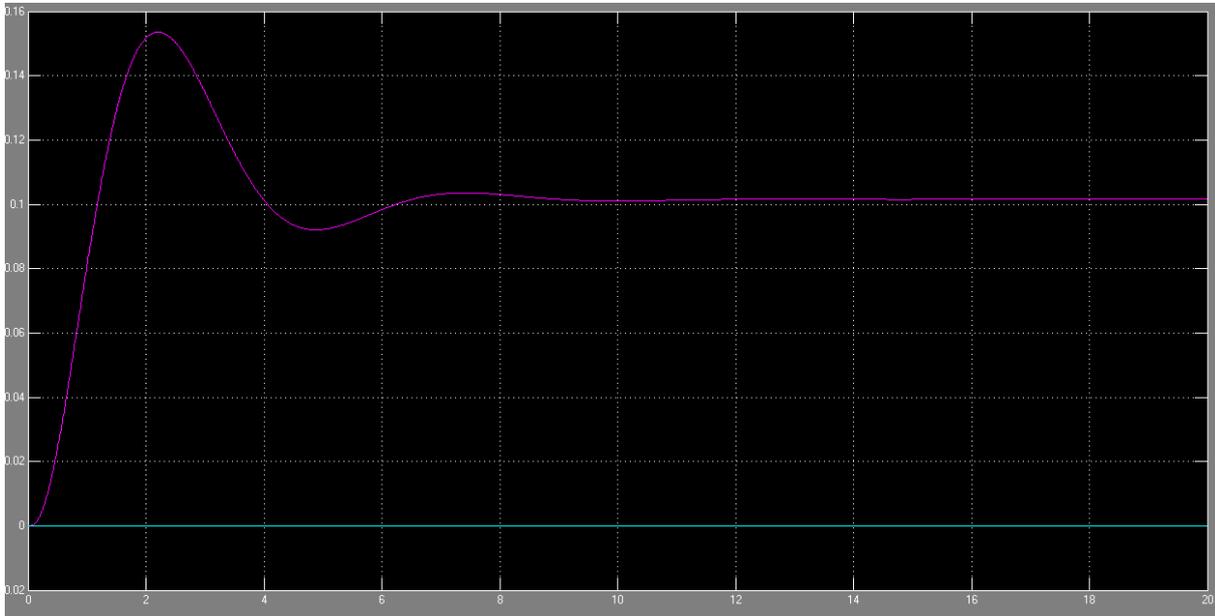


Figura 10.29 grafico di risposta alla forza costante in Φ, Θ, Ψ

P, Q, R :

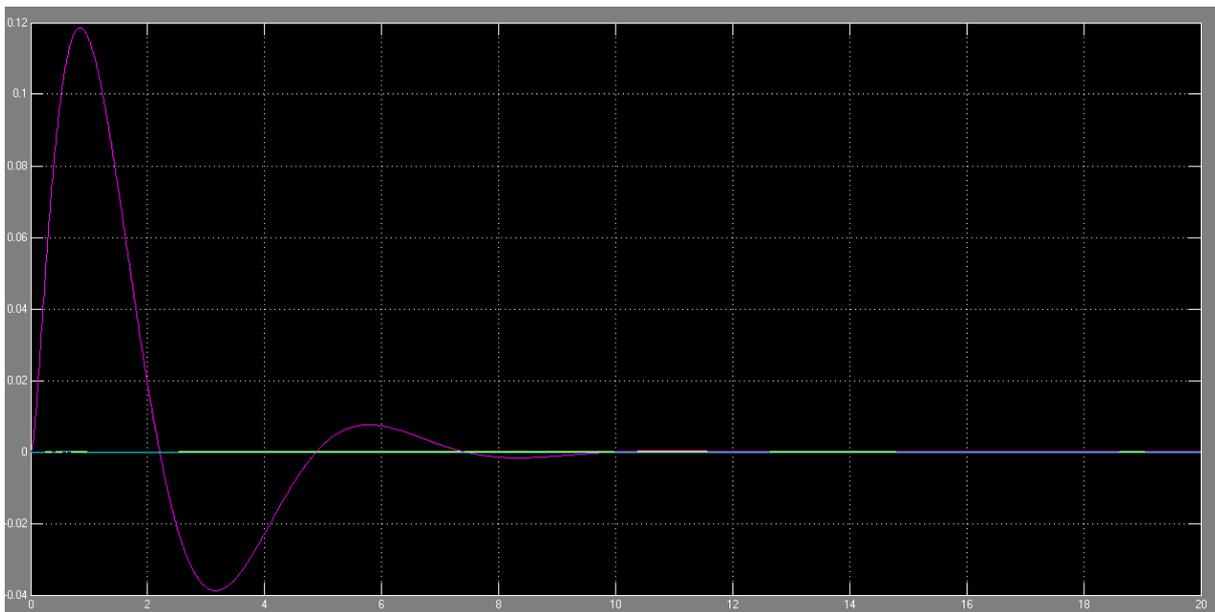


Figura 10.29 grafico di risposta alla forza costante IN P, Q, R

U,V,W :

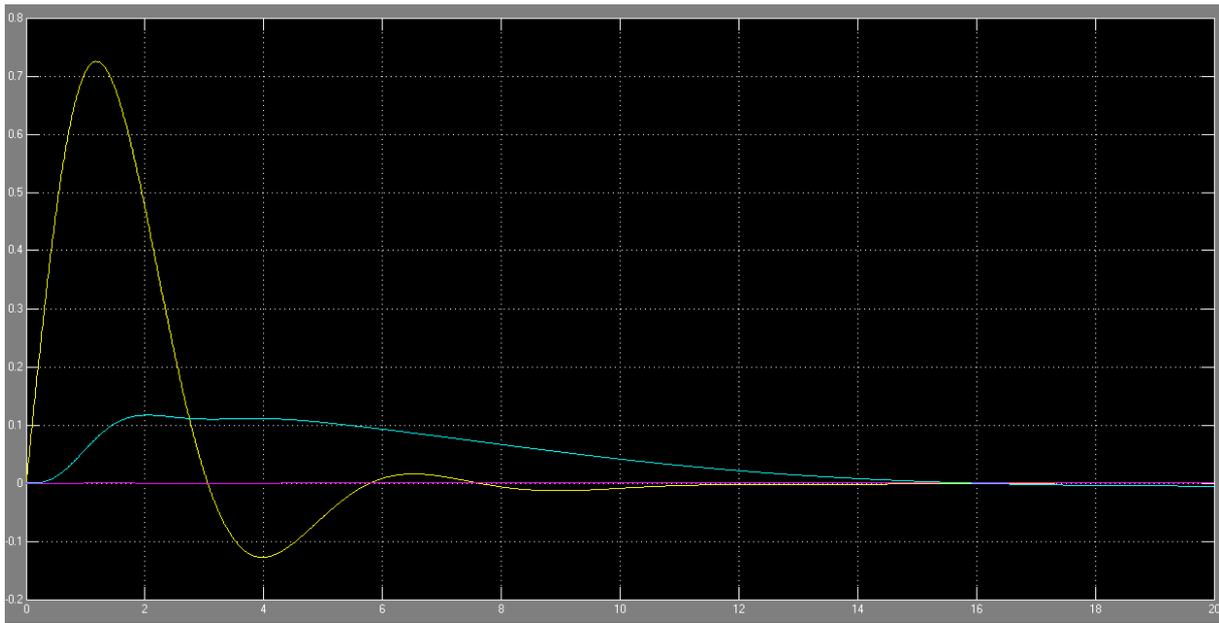


Figura 10.30 grafico di risposta alla forza costante IN U,V,W

N,E,D :

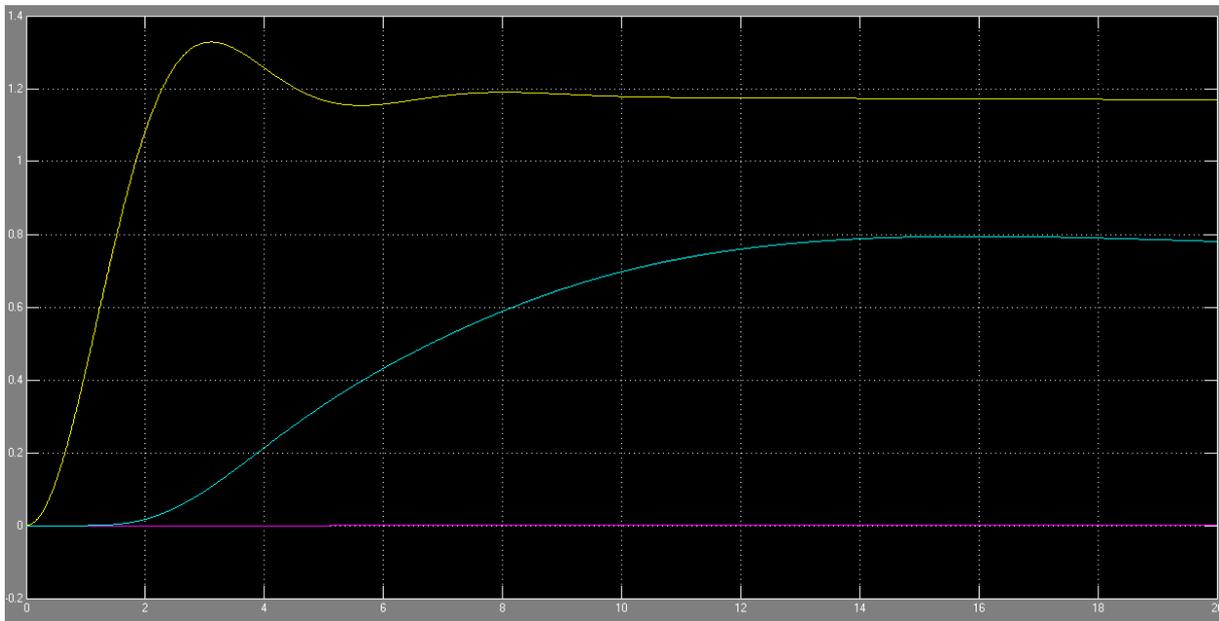


Figura 10.31 grafico di risposta alla forza costante in N,E,D

Si evince dal primo e dall'ultimo grafico che la posizione finale tende ad un valore diverso da quello iniziale di hovering, in particolare una rotazione attorno al pitch di 0.1 rad, lungo l'asse n abbiamo uno scostamento di circa 1.2 m e lungo l'asse d di circa 0.8 m, valori al quanto eccessivi.

11. MODELLO LINEARIZZATO CASO MOTO RETTILINEO

Questa analisi ha lo scopo di verificare le condizioni di stabilità del drone quando sono presenti disturbi durante le condizioni di non stazionarietà.

Prima di cominciare ad approcciarsi in maniera analitica al problema, è opportuno formulare alcune considerazioni riguardanti le tipologie di moto per capire in quali casi non è necessario intraprendere lo studio .

Le tipologie di moto sono correlate ai vincoli nello spazio, difatti essendoci sei gradi di libertà avremo come casi totali da analizzare tre traslazioni lungo gli assi principali e tre rotazioni attorno ad essi.

Ai fini pratici però le rotazioni non sono dannose, in quanto il vento produce principalmente traslazioni del velivolo e in seconda analisi le rotazioni vengono collimate dal gimbal.

Saranno oggetto di questa trattazione solo i rimanenti tre casi di moto uniforme lungo gli assi principali

Il punto di partenza saranno sempre le nostre equazioni del moto classiche, unica differenza riguarderanno le condizioni iniziali che vedranno, per ogni tipologia di moto, valori diversi da sostituire nel sistema di equazioni da svolgere.

11.1. CASO MOTO RETTILINEO LUNGO L'ASSE X

condizioni iniziali :

- $U_0 \neq 0$;
- $\Theta_0 \neq 0$;

Le equazioni svolte con questi postulati diventano :

$$\left\{ \begin{array}{l} \dot{u} = -g \cdot \vartheta \cdot \cos\Theta_0 + \frac{df_{wx}}{m} \\ \dot{v} = -r \cdot U_0 + g \cdot \varphi \cdot \cos\Theta_0 + \frac{df_{wy}}{m} \\ \dot{w} = +q \cdot U_0 - g \cdot \vartheta \cdot \sin\Theta_0 + \frac{df_{tz} + df_{wz}}{m} \end{array} \right.$$

$$\left\{ \begin{array}{l} \dot{p} = \frac{dm_{tx} + dm_{wx}}{I_x} \\ \dot{q} = \frac{dm_{ty} + dm_{wy}}{I_y} \\ \dot{r} = \frac{dm_{tz} + dm_{wz}}{I_z} \end{array} \right.$$

$$\left\{ \begin{array}{l} \dot{\varphi} = p + r \cdot \operatorname{tg}\theta_0 \\ \dot{\vartheta} = q \\ \dot{\psi} = \frac{r}{\cos\theta_0} \end{array} \right.$$

$$\left\{ \begin{array}{l} \dot{n} = u \cdot \cos\theta_0 + w \cdot \sin\theta_0 - \vartheta \cdot U_0 \cdot \sin\theta_0 \\ \dot{e} = v \cdot \cos\theta_0 + \psi \cdot U_0 \cdot \cos\theta_0 \\ \dot{d} = -u \cdot \sin\theta_0 + w \cdot \cos\theta_0 - \vartheta \cdot U_0 \cdot \cos\theta_0 \end{array} \right.$$

A cui ovviamente corrisponde la nuova matrice \mathbf{A} per il moto rettilineo lungo X :

$$\mathbf{A}_x = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & \operatorname{tg}\theta_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\cos\theta_0} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -g \cdot \cos\theta_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g \cdot \cos\theta_0 & 0 & 0 & 0 & 0 & -U_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -g \cdot \sin\theta_0 & 0 & 0 & U_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -U_0 \cdot \sin\theta_0 & 0 & 0 & 0 & 0 & \cos\theta_0 & 0 & \sin\theta_0 & 0 & 0 & 0 \\ 0 & 0 & U_0 \cdot \cos\theta_0 & 0 & 0 & 0 & 0 & \cos\theta_0 & 0 & 0 & 0 & 0 \\ 0 & -U_0 \cdot \cos\theta_0 & 0 & 0 & 0 & 0 & -\sin\theta_0 & 0 & \cos\theta_0 & 0 & 0 & 0 \end{pmatrix}$$

11.2. CASO MOTO RETTILINEO LUNGO L'ASSE Y

condizioni iniziali :

- $V_0 \neq 0$;
- $\Phi_0 \neq 0$;

Le equazioni svolte con questi postulati diventano :

$$\left\{ \begin{array}{l} \dot{u} = V_0 \cdot r + \frac{df_{wx}}{m} \\ \dot{v} = g \cdot \varphi \cdot \cos\Phi_0 + \frac{df_{wy}}{m} \\ \dot{w} = V_0 \cdot p - g \cdot \varphi \cdot \sin\Phi_0 + \frac{df_{tz} + df_{wz}}{m} \end{array} \right.$$

$$\left\{ \begin{array}{l} \dot{p} = \frac{dm_{tx} + dm_{wx}}{I_x} \\ \dot{q} = \frac{dm_{ty} + dm_{wy}}{I_y} \\ \dot{r} = \frac{dm_{tz} + dm_{wz}}{I_z} \end{array} \right.$$

$$\left\{ \begin{array}{l} \dot{\varphi} = p \\ \dot{\vartheta} = q \cdot \cos\Phi_0 - r \cdot \sin\Phi_0 \\ \dot{\psi} = q \cdot \sin\Phi_0 + r \cdot \cos\Phi_0 \end{array} \right.$$

$$\left\{ \begin{array}{l} \dot{n} = u + \vartheta \cdot V_0 \cdot \sin\Phi_0 + \vartheta \cdot W_0 \cdot \cos\Phi_0 - \psi \cdot V_0 \cdot \cos\Phi_0 + \psi \cdot W_0 \cdot \sin\Phi_0 \\ \dot{e} = v \cdot \cos\Phi_0 - w \cdot \sin\Phi_0 - \varphi \cdot V_0 \cdot \sin\Phi_0 - \varphi \cdot W_0 \cdot \cos\Phi_0 + \psi \cdot U_0 \cdot \cos\Phi_0 \\ \dot{d} = u \cdot \sin\Phi_0 + w \cdot \cos\Phi_0 - \vartheta \cdot U_0 + \varphi \cdot V_0 \cdot \cos\Phi_0 - \varphi \cdot W_0 \cdot \sin\Phi_0 \end{array} \right.$$

E la nuova matrice \mathbf{A} per il moto rettilineo lungo Y :

$$\mathbf{A}_y = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos\Phi_0 & -\sin\Phi_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sin\Phi_0 & \cos\Phi_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g \cdot \cos\Phi_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -g \cdot \sin\Phi_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -V_0 \cdot \sin\Phi_0 - W_0 \cdot \cos\Phi_0 & V_0 \cdot \sin\Phi_0 + W_0 \cdot \cos\Phi_0 & -V_0 \cdot \cos\Phi_0 + W_0 \cdot \sin\Phi_0 & 0 & 0 & 0 & 1 & 0 & \sin\theta_0 & 0 & 0 & 0 & 0 \\ V_0 \cdot \cos\Phi_0 - W_0 \cdot \sin\Phi_0 & U_0 & 0 & 0 & 0 & 0 & 0 & \cos\theta_0 & -\sin\Phi_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sin\Phi_0 & 0 & \cos\Phi_0 & 0 & 0 \end{pmatrix}$$

11.3. CASO MOTO RETTILINEO LUNGO L'ASSE Z

condizioni iniziali :

- $W_0 \neq 0$;

Le equazioni svolte con questi postulati diventano :

$$\left\{ \begin{array}{l} \dot{u} = -q \cdot W_0 - g \cdot \theta + \frac{df_{wx}}{m} \\ \dot{v} = -p \cdot W_0 + g \cdot \varphi + \frac{df_{wy}}{m} \\ \dot{w} = \frac{df_{tz} + df_{wz}}{m} \end{array} \right.$$

$$\left\{ \begin{array}{l} \dot{p} = \frac{dm_{tx} + dm_{wx}}{I_x} \\ \dot{q} = \frac{dm_{ty} + dm_{wy}}{I_y} \\ \dot{r} = \frac{dm_{tz} + dm_{wz}}{I_z} \end{array} \right.$$

$$\begin{cases} \dot{\phi} = p \\ \dot{\vartheta} = q \\ \dot{\psi} = r \end{cases}$$

$$\begin{cases} \dot{n} = u + W_0 \cdot \vartheta \\ \dot{e} = v + W_0 \cdot \vartheta - W_0 \cdot \phi \\ \dot{d} = w \end{cases}$$

A cui corrisponde la matrice \mathbf{A} per il moto rettilineo lungo Z :

$$\mathbf{A}_z = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -g & 0 & 0 & -W_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g & 0 & 0 & -W_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & W_0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -W_0 & W_0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

11.4 IMPLEMENTAZIONE CASO MOTO RETTILINEO

Per implementare questa nuova condizione nel software è necessario dunque rielaborare la matrice A per darla nuovamente in “pasto” alla funzione di Riccati.

Le condizioni di moto dunque sono le traslazioni lungo gli assi principali o una combinazione lineare di queste e quindi si ritiene necessario compilare la matrice A con le 12 equazioni del moto reali nelle 12 variabili di stato.

Si ricava in tal modo la possibilità di avere un modello flessibile che, preso in input la condizione iniziale desiderata, automaticamente fornisca i sistemi di equazioni opportuni da produrre la legge del moto voluta.

Si ottiene la seguente struttura in codice Matlab:

```
xu=cos(teta)*cos(psi);
xv=sin(phi)*sin(teta)*cos(psi)-cos(teta)*sin(psi);
xw=cos(phi)*sin(teta)*cos(psi)+sin(phi)*sin(psi);
xt=-u*sin(teta)*cos(psi)+v*sin(phi)*cos(teta)*cos(psi)+w*cos(phi)*cos(teta)*cos(psi);
xphi=(v*cos(phi)-w*sin(phi))*sin(teta)*cos(psi)+(v*sin(phi)+w*cos(phi))*sin(psi);
xpsi=-u*cos(teta)*sin(psi)-v*sin(phi)*sin(teta)*sin(psi)-v*cos(phi)*cos(psi)-
w*cos(phi)*sin(teta)*sin(psi)+w*sin(phi)*cos(psi);
yu=cos(teta)*sin(psi);
yv=sin(phi)*sin(teta)*sin(psi)+cos(teta)*cos(psi);
yw=cos(phi)*sin(teta)*sin(psi)-sin(phi)*cos(psi);
yt=-u*sin(teta)*sin(psi)+v*sin(phi)*cos(teta)*sin(psi)+w*cos(phi)*cos(teta)*sin(psi);
yphi=v*cos(phi)*sin(teta)*sin(psi)-v*sin(phi)*cos(psi)-w*sin(phi)*sin(teta)*sin(psi)-
w*cos(phi)*cos(psi);
ypsi=u*cos(teta)*cos(psi)+v*sin(phi)*sin(teta)*cos(psi)-
v*cos(phi)*sin(psi)+w*cos(phi)*sin(teta)*cos(psi)+w*sin(phi)*sin(psi);
zu=-sin(teta);
zv=sin(phi)*cos(teta);
zw=cos(phi)*cos(teta);
zt=-u*cos(teta)-v*sin(phi)*sin(teta)-w*cos(phi)*sin(teta);
zphi=v*cos(phi)*cos(teta)-w*sin(phi)*cos(teta);

A=[0 0 0 1 sin(phi)*tan(teta) cos(phi)*tan(teta) 0 0 0 0 0 0;
  0 0 0 0 cos(phi) -sin(phi) 0 0 0 0 0 0;
  0 0 0 0 sin(phi)/cos(teta) cos(phi)/cos(teta) 0 0 0 0 0 0;
  0 0 0 0 ix*r ix*q 0 0 0 0 0 0;
  0 0 0 iy*r 0 iy*p 0 0 0 0 0 0;
  0 0 0 iz*q iz*p 0 0 0 0 0 0;
  0 -9.81*cos(teta) 0 0 -w v 0 r -q 0 0 0;
  9.81*cos(teta)*cos(phi) -9.81*sin(teta)*sin(phi) 0 w 0 -u -r 0 p 0 0 0;
  -9.81*cos(teta)*sin(phi) -9.81*sin(teta)*cos(phi) 0 -v u 0 q -p 0 0 0 0;
  xphi xt xpsi 0 0 0 xu xv xw 0 0 0;
  yphi yt ypsi 0 0 0 yu yv yw 0 0 0;
  zphi zt 0 0 0 0 zu zv zw 0 0 0];
```

il resto dello script rimane immutato e quindi strutturato come visto per il caso di overing. Al contrario bisogna modificare lo schema a blocchi in Simulink, per eseguire il tracking delle condizioni desiderate, in figura 11.1 è riportato il modello per il caso di moto rettilineo (o tracking delle velocità).

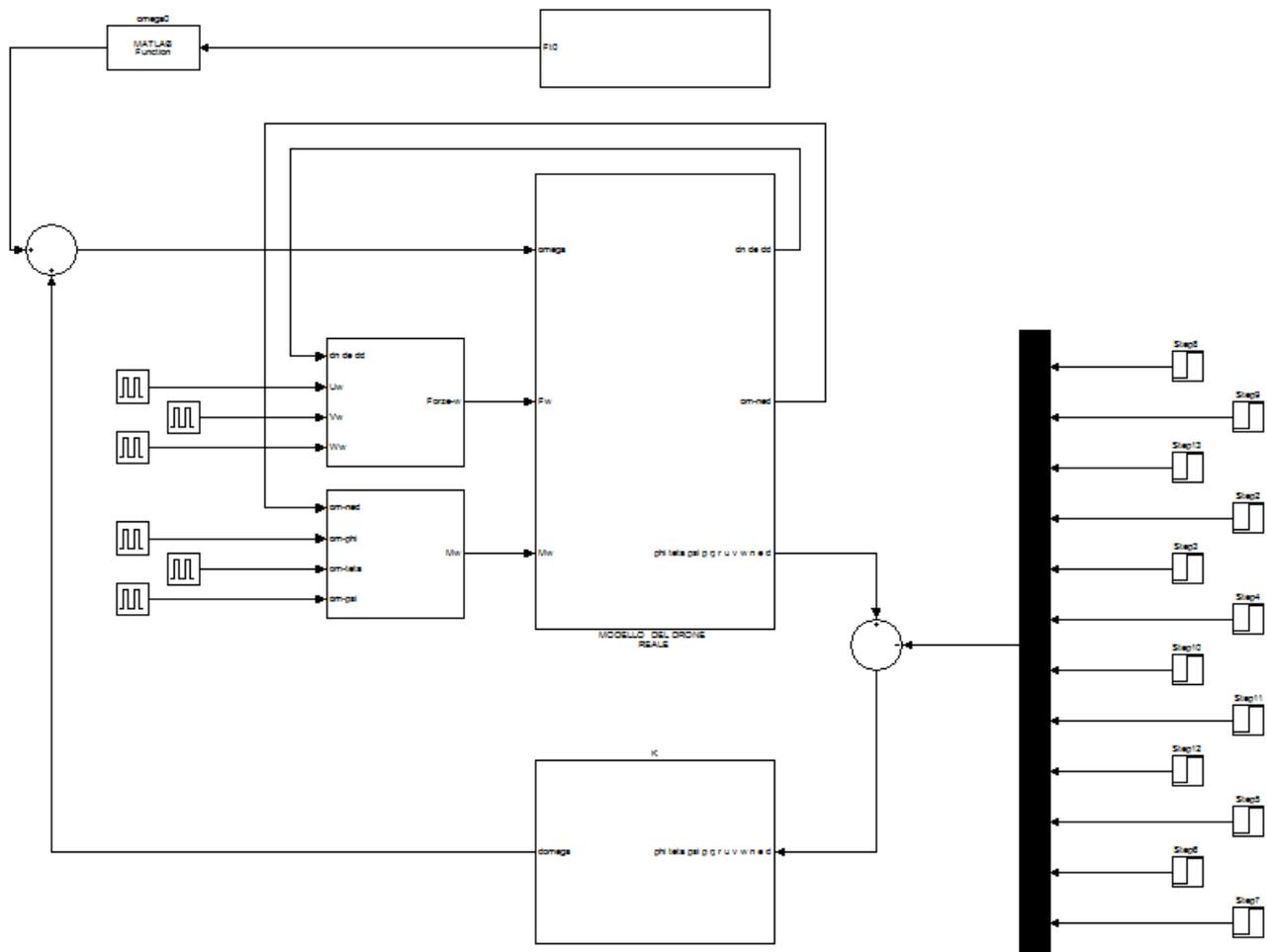


Figura 11.1 sistema a blocchi in feedback del modello reale in condizione di tracking

Rispetto al modello precedente si notano varie modifiche nel layout del sistema in Simulink, in particolare la presenza di nuovi blocchi per la gestione delle forze di disturbo e per il calcolo della velocità di tracking.

Ovviamente anche lo schema del blocco “K” è modificato in quanto anche la matrice A ha subito una ricompilazione come abbiamo già esplicitato nella pagine precedente.

Al contrario il blocco che descrive il modello reale del drone è rimasto immutato in quanto le equazioni che descrivono il moto sono ovviamente sempre le stesse.

Analizziamo ogni singolo nuovo elemento.

11.5 BLOCCO PER IL CALCOLO DELLE FORZE DOVUTE AL VENTO

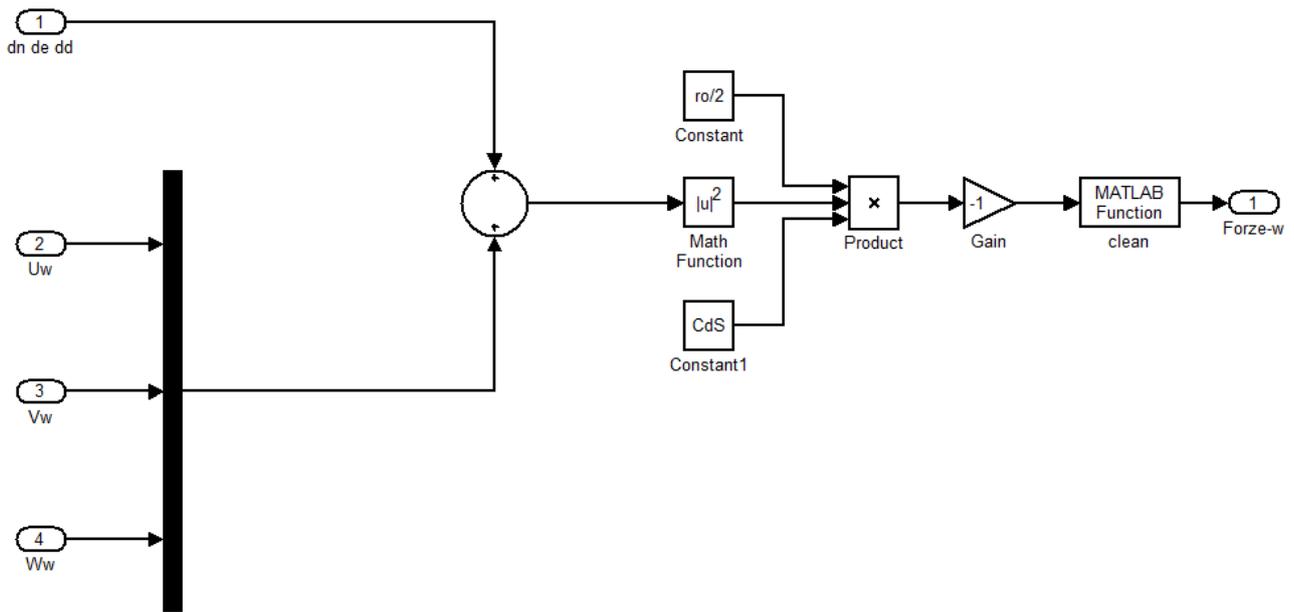


Figura 11.2 schema per il calcolo delle forze dovute al vento

Tramite questo schema è possibile tradurre un velocità in una forza di disturbo per il quadricottero. In input abbiamo in alto le tre componenti del vento dovute al moto del drone rispetto alla terna fissa NED, mentre in basso abbiamo le tre componenti del vento lungo la stessa terna di assi. Essendo il vento, tramite le convenzioni da noi assunte nelle dodici equazioni del corpo rigido, disturbi che si oppongono alla direzione del moto stesso, si vanno a sommare a quelle generate dal movimento del quadricottero rispetto al sistema di riferimento solidale. Non rimane che convertire la velocità (v) in forza (F) e lo si fa tramite l'espressione che correla queste due entità tramite l'attrito fluidodinamico:

$$F = \rho \cdot C_D \cdot S \cdot \frac{v^2}{2}$$

Con:

- ρ : densità standard dell'aria (1.225 kg/m^3);
- S : sezione resistente del mezzo perpendicolare al moto di avanzamento;
- C_D : coefficiente di attrito fluidodinamico in funzione della velocità e superficie del mezzo.

11.6 BLOCCO PER IL CALCOLO DELLE COPPIE DOVUTE AL VENTO

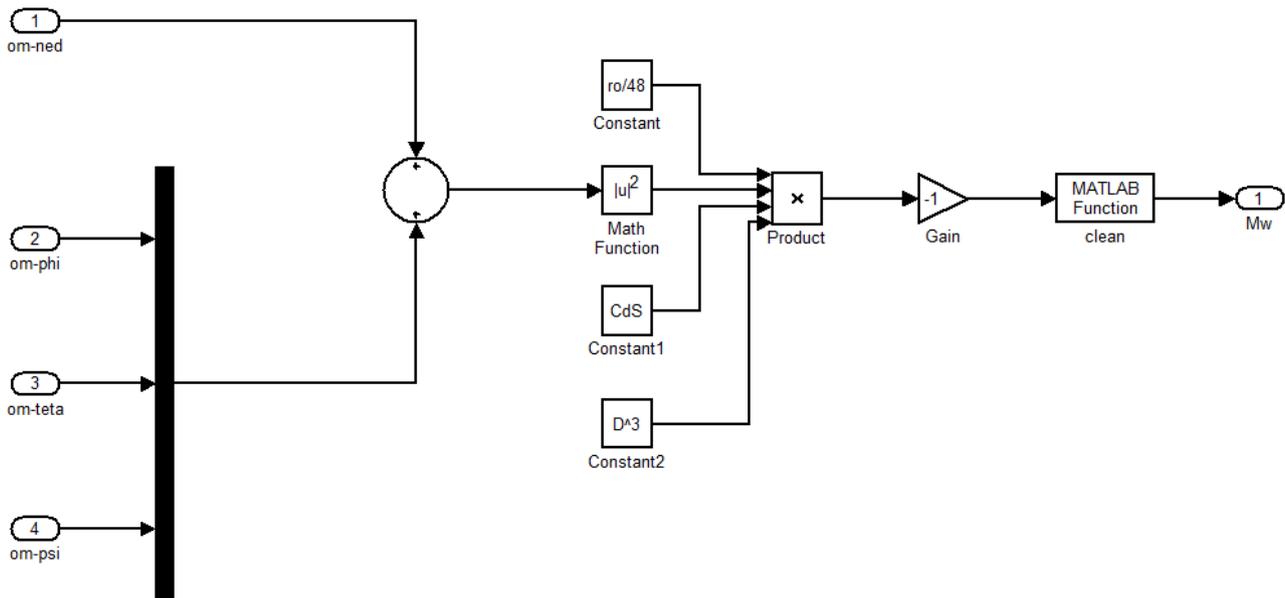


Figura 11.3 schema per il calcolo delle coppie dovute al vento

Questo schema è del tutto analogo a quello appena descritto nella pagina precedente, cambiano i valori di input, che saranno velocità angolari, e di output, che saranno momenti.

La formula per il calcolo della coppia deriva da un integrale, analizziamo in pochi passaggi la sua dimostrazione.

In figura 11.4 si evince come la velocità cresce linearmente quando un corpo ruota attorno ad un punto fisso 0.

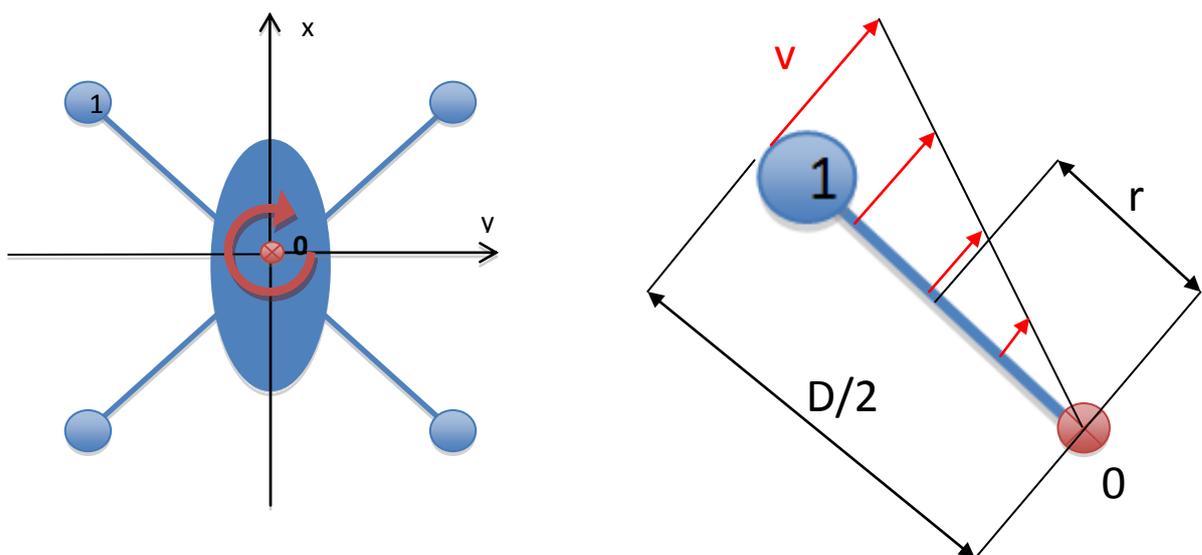


Figura 11.4 esempio di rotazione in yaw e velocità periferica per il braccio 1

Si ha quindi:

$$v = \omega \cdot r$$

Con r che indica il valore generico di distanza dal centro 0.

Sostituendo questa nella formula delle forze si ottiene:

$$F = \rho \cdot C_D \cdot S \cdot \frac{(\omega \cdot r)^2}{2}$$

Per il calcolo del momento quindi basterà moltiplicare la relazione appena trovata per l'incremento di distanza (dr) e quindi integrare il tutto tra 0 e D/2:

$$M = \int_0^{D/2} \rho \cdot C_D \cdot S \cdot \frac{(\omega \cdot r)^2}{2} \cdot dr = \rho \cdot C_D \cdot S \cdot \omega^2 \cdot \frac{D^3}{48}$$

11.7 BLOCCO PER IL CALCOLO DEL FEEDBACK

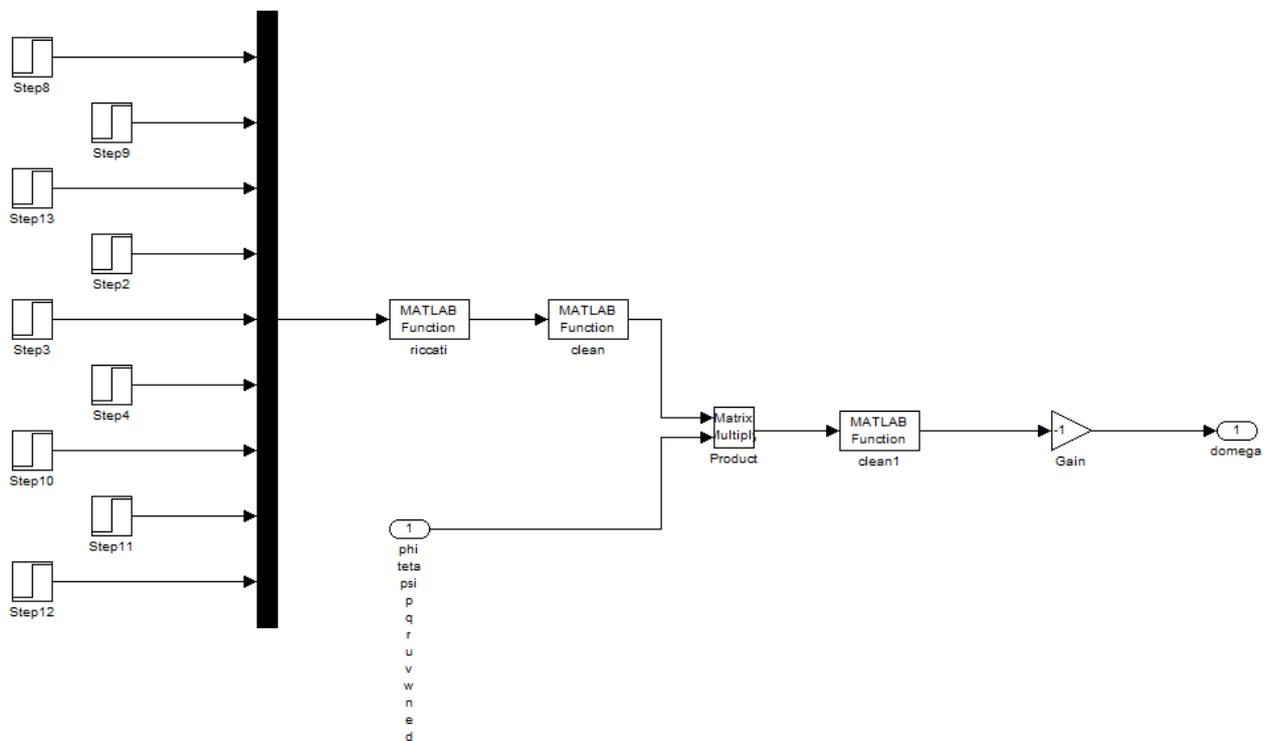


Figura 11.4 struttura del regolatore

Come già preannunciato l'unica variazione concerne la struttura della matrice A che dovrà prendere in ingresso i valori desiderati per il tracking, rappresentata dai blocchi in cascata sulla sinistra.

11.8 BLOCCO PER IL CALCOLO DELLA FORZA DI THRUST PER IL MOTO COSTANTE DEL DRONE

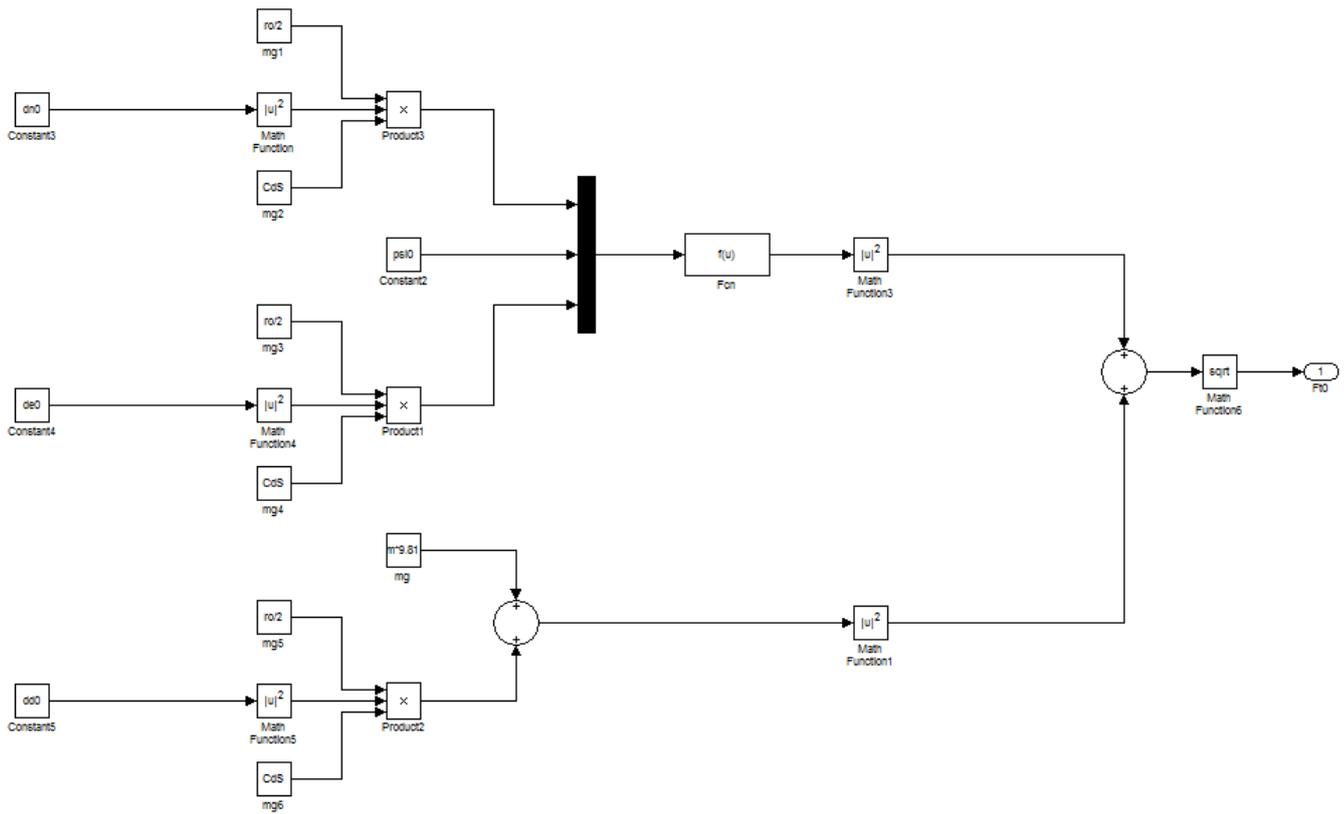


Figura 11.5 struttura del blocco Ft0

Lo schema qui riportato è la rappresentazione della situazione illustrata in figura 11.6.

L'obiettivo è il calcolo della forza di thrust (F_{t0}) tale da permettere la traslazione costante lungo gli assi principali o lungo una combinazione lineare di questi.

Esaminiamo come esempio il moto costante lungo N.

È facile immaginare che se il velivolo compie un moto perfettamente orizzontale, questo dovrà avere una certa inclinazione Θ rispetto all'asse Y.

Infatti in questo assetto la forza di thrust non sarà più verticale ma ruotata di Θ , si scompone così in una componente verticale che dovrà eguagliare la forza peso (per mantenere la quota lungo D) e una orizzontale (F_{tn}) capace di contrastare la componente di forza del vento ed eguagliarlo producendo così moto costante.

Un escamotage adottato quindi potrebbe essere di calcolare le forze di attrito dovuto al contrasto aerodinamico col vento e quindi trovare la risultante con la forza peso.

Quindi nello schema di figura 11.5 prima si fa uso del teorema di Carnot per il calcolo della risultante tra le due entità orizzontali usando come angolo quello di yaw.

Successivamente si sommano la forza di thrust per la componente di forza peso e quella per il volo lungo l'asse Z e come ultima fase tramite il teorema di Pitagora si ottiene la forza risultante di Thrust per il moto continuo lungo un asse principale.

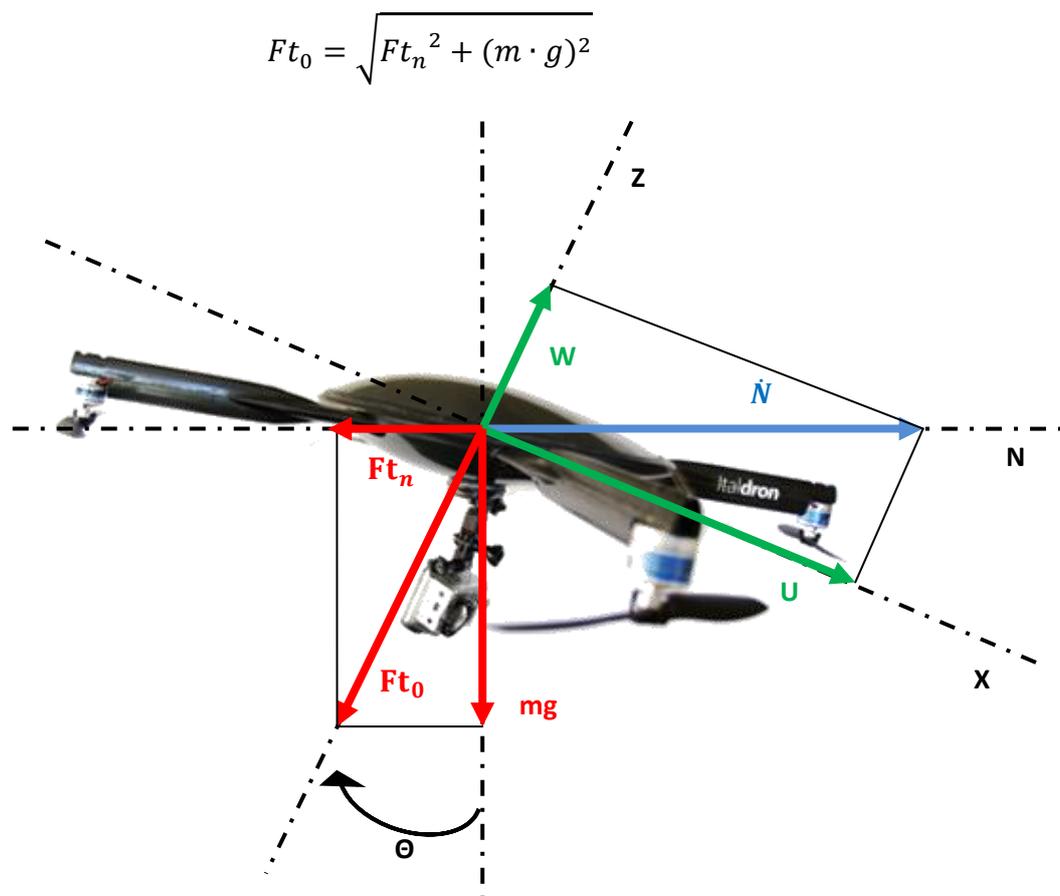


Figura 11.6 schema per il calcolo di Ft_0

Si necessita quindi di introdurre un nuovo sistema di riferimento da utilizzare per plottare il moto del drone, il sistema $\dot{N}, \dot{E}, \dot{D}$.

Difatti con questa nuova terna avremo a disposizione le velocità del velivolo nel sistema di riferimento assoluto.

Questo perché, essendo il mezzo inclinato di un certo angolo, la velocità assoluta \dot{N} si scompone in due componenti (mostrate dai vettori verdi in figura 11.6) lungo gli assi del sistema BODY.

11.9 BLOCCO PER IL CALCOLO DELLE VELOCITÀ ANGOLARI PER IL MOTO UNIFORME

Questo elemento non fa altro che prendere i valori di thrust dal blocco appena esplicitato () e restituisce i valori di ω per i singoli rotori.

Lo script di Matlab è così strutturato:

```
function [om0]=omega0_rett(Ft0)

N=2000:1000:20000;

t=[0.156,0.343,0.601,0.930,1.341,1.851,2.433,3.073,3.77,4.528,5.406,6.425,7.502,
8.607,9.905,11.295,12.8,14.259,16.53];

%converti unità di misura

Ti=t.*4.45;

Ft0i=Ft0/4;

if Ft0i>Ti(1,19)
    om0=20000*pi/30*[1;1;1;1];

else if Ft0i<Ti(1,1)
    om0=2000*pi/30*[1;1;1;1];
    else
        no=interp1(Ti,N,Ft0i);
        om0=no*pi/30*[1;1;1;1];
    end
end
end
```

Il ciclo “if” serve per saturare il sistema, infatti i rotori hanno un range di funzionamento che vanno dai 2000 rpm ai 20000 rpm, quindi se in input entra una forza che richiede velocità superiori o inferiori a quelle limite, il ciclo impostato le satura ai corrispettivi valore massimo o minimo ammessi dai rotori.

11.10 SCRIPT PER DEFINIRE LE CONDIZIONI INIZIALI O DI TRACKING

Come ultimo non resta che uno script da “lanciare” ad ogni inizio simulazione che definisce i valori iniziali desiderati dalla prova.

È sufficiente quindi sostituire i valori voluti in questo elenco per ottenere in tutto le fasi di calcolo i medesimi postulati scelti.

Si riporta di seguito il seguente foglio di calcolo:

```
%inizializzazione trim

m=1.307;
D=0.335;
ro=1.225;

% C.I. assetto

phi0=0;
teta0=-0.096;
psi0=0;
p0=0;
q0=0;
r0=0;
dn0=1;
de0=0;
dd0=0;
n0=0;
e0=0;
d0=0;
DCM=[cos(teta0)*cos(psi0) cos(teta0)*sin(psi0) -sin(teta0);
      sin(teta0)*sin(phi0)*cos(psi0)-cos(phi0)*sin(psi0)
      sin(teta0)*sin(phi0)*sin(psi0)+cos(phi0)*cos(psi0) sin(phi0)*cos(teta0);
      sin(teta0)*cos(phi0)*cos(psi0)+sin(phi0)*sin(psi0)
      sin(teta0)*cos(phi0)*sin(psi0)-sin(phi0)*cos(psi0) cos(phi0)*cos(teta0)];

VETned=[dn0 de0 dd0]';

VETuvw=DCM*VETned;
u0=VETuvw(1,1);
v0=VETuvw(2,1);
w0=VETuvw(3,1);

% C.I. calcolo drag

CdS=0.5;
```

I valori di U_0, V_0, W_0 si ricavano semplicemente moltiplicando la DCM per la terna delle velocità $\dot{N}_0, \dot{E}_0, \dot{D}_0$.

11.11 TEST DEL MODELLO REALE CON MOTO UNIFORME

Si giunge quindi alla verifica del programma tramite dei test mirati per poi valutare la risposta che elabora il calcolatore.

Come prima fase si devono impostare le condizioni iniziali desiderate.

Si decide la velocità di traslazione e si ricava dalle equazioni del moto il corrispondente valore dell'angolo, in quanto ad ogni assetto (dovuto all'attrito ed alla velocità di crociera) si hanno configurazioni specifiche.

Partiamo con il primo test di volo imponendo una velocità di traslazione solo l'ungo l'asse N con velocità pari ad 1 m/s:

$$\begin{cases} \dot{N}_0 = 1 \text{ m/s} \\ \dot{E}_0 = 0 \text{ m/s} \\ \dot{D}_0 = 0 \text{ m/s} \end{cases}$$

Utilizzando la **DCM** ricavo U_0, V_0, W_0 :

$$\begin{pmatrix} U_0 \\ V_0 \\ W_0 \end{pmatrix} = \begin{pmatrix} c\vartheta c\psi & c\vartheta s\psi & -s\vartheta \\ s\phi s\vartheta c\psi - c\phi s\psi & s\phi s\vartheta s\psi + c\phi c\psi & s\phi c\vartheta \\ c\phi s\vartheta c\psi + s\phi s\psi & s\phi c\vartheta s\psi - s\phi c\psi & c\phi c\vartheta \end{pmatrix} \times \begin{pmatrix} \dot{N}_0 \\ \dot{E}_0 \\ \dot{D}_0 \end{pmatrix}$$

Quindi:

$$\begin{cases} U_0 = c\vartheta \cdot \dot{N}_0 = c\vartheta \\ V_0 = 0 \\ W_0 = s\vartheta \cdot \dot{N}_0 = s\vartheta \end{cases}$$

Si ricava sostituendo ogni valore nell'equazioni dl moto:

$$\begin{cases} 0 = -0 + 0 - g \cdot s\vartheta - F_{wx}/m \\ 0 = 0 - 0 + 0 + 0 \\ 0 = 0 - 0 + g \cdot c\vartheta - F_{tz}/m - F_{wz}/m \end{cases}$$

Dalla prima si ricava:

$$\vartheta_0 = -\arctg \frac{F_{tn}}{m \cdot g}$$

Con:

$$F_{wx} = \rho \cdot C_D \cdot S \cdot \frac{\dot{N}_0^2}{2} \cdot c\vartheta$$

$$F_{tn} = F_{wx}/c\vartheta$$

E si ricava:

$$\vartheta = -0.048$$

$$U_0 = 1$$

$$W_0 = -0.048$$

Ed impostandole nello script per le condizioni iniziali si ottengono i seguenti plot:

$\dot{N}, \dot{E}, \dot{D}$

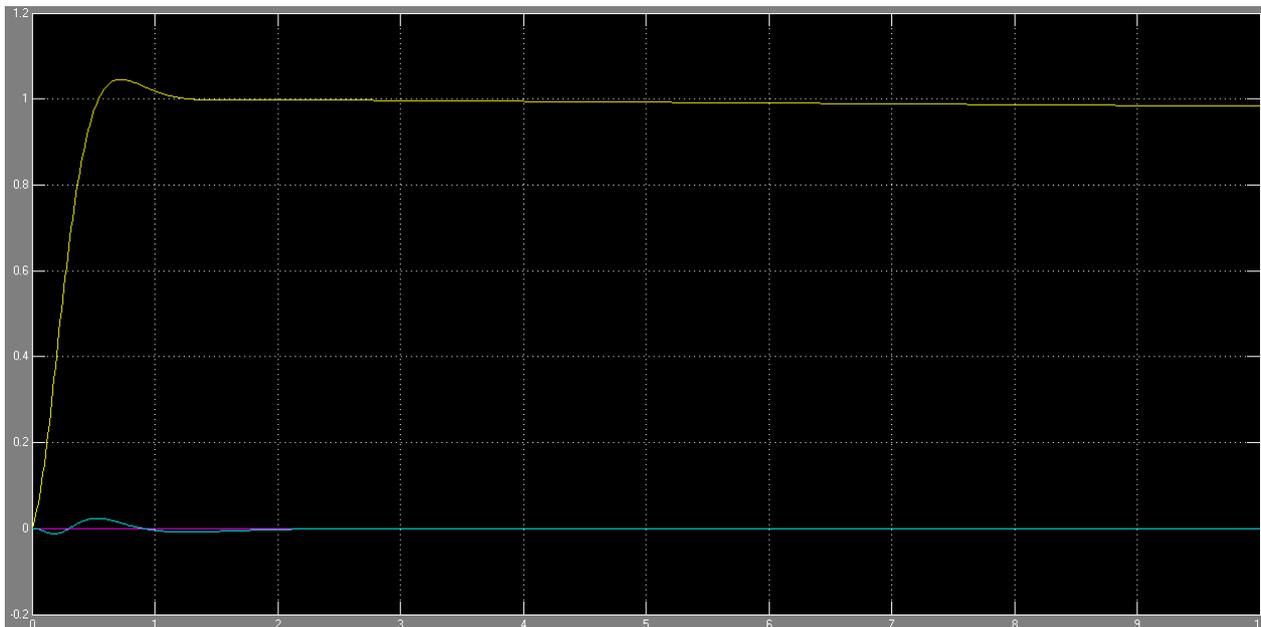


Figura 11.7 grafico del moto costante lungo N in $\dot{N}, \dot{E}, \dot{D}$

U,V,W

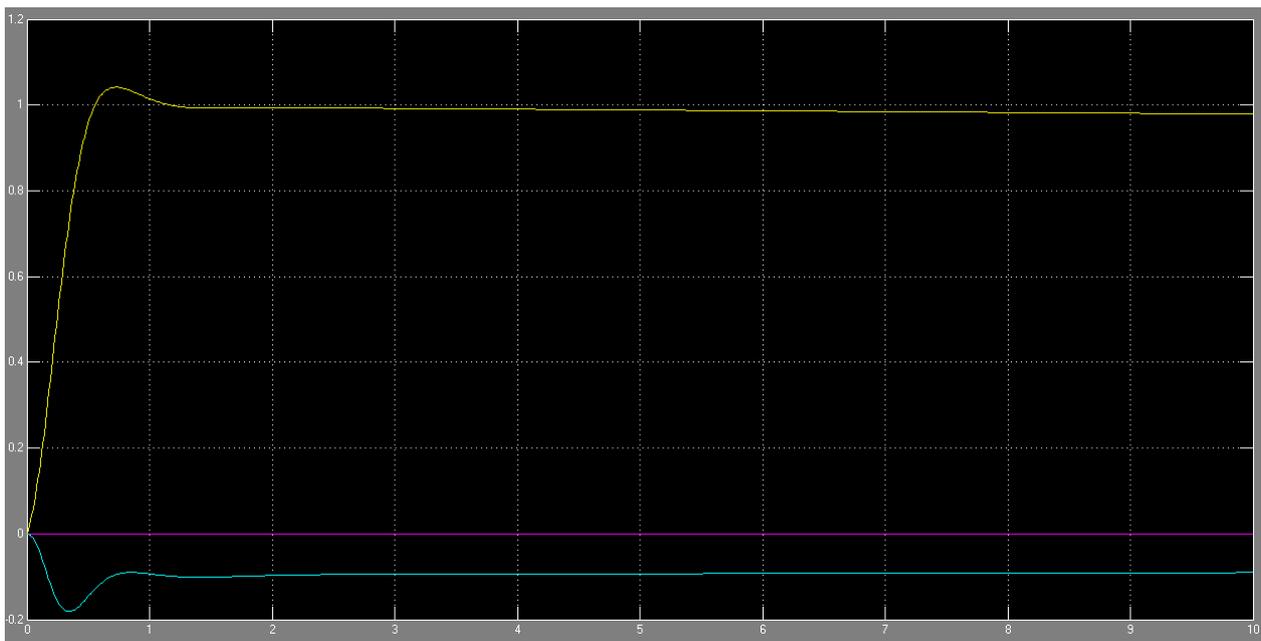


Figura 11.8 grafico del moto costante lungo N in U,V,W

N,E,D

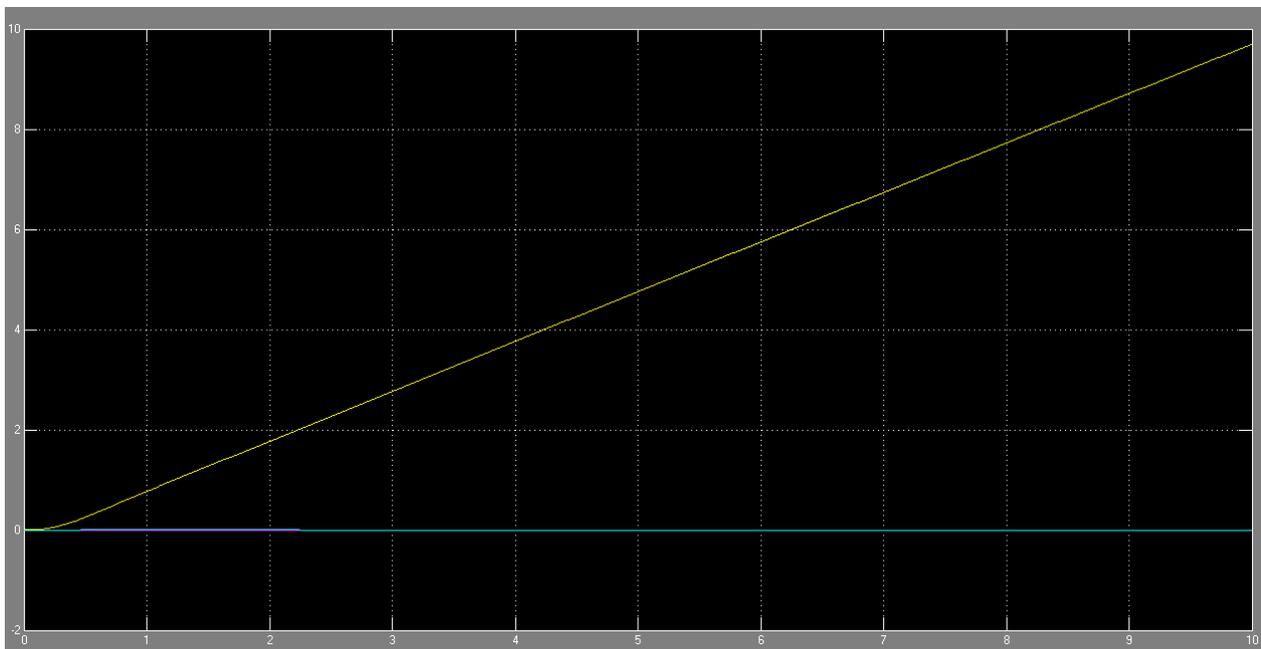


Figura 11.8 grafico del moto costante lungo N in N,E,D

La risposta è valida e quindi si ritiene il test passato con successo

La fase successiva è introdurre un impulso mentre il mezzo si muove in due direzioni contemporaneamente.

Ipotizziamo quindi:

$$\left\{ \begin{array}{l} \dot{N}_0 = 1 \text{ m/s} \\ \dot{E}_0 = 1 \text{ m/s} \\ \dot{D}_0 = 0 \text{ m/s} \end{array} \right.$$

Ed impulso di vento lungo l'asse E dopo 3 s con un valore di 4 m/s.

Si ottiene:

$$\begin{aligned} \varphi &= 0.048 \\ \vartheta &= -0.048 \\ U_0 &= 1 \\ V_0 &= 1 \\ W_0 &= -0.096 \end{aligned}$$

$\dot{N}, \dot{E}, \dot{D}$

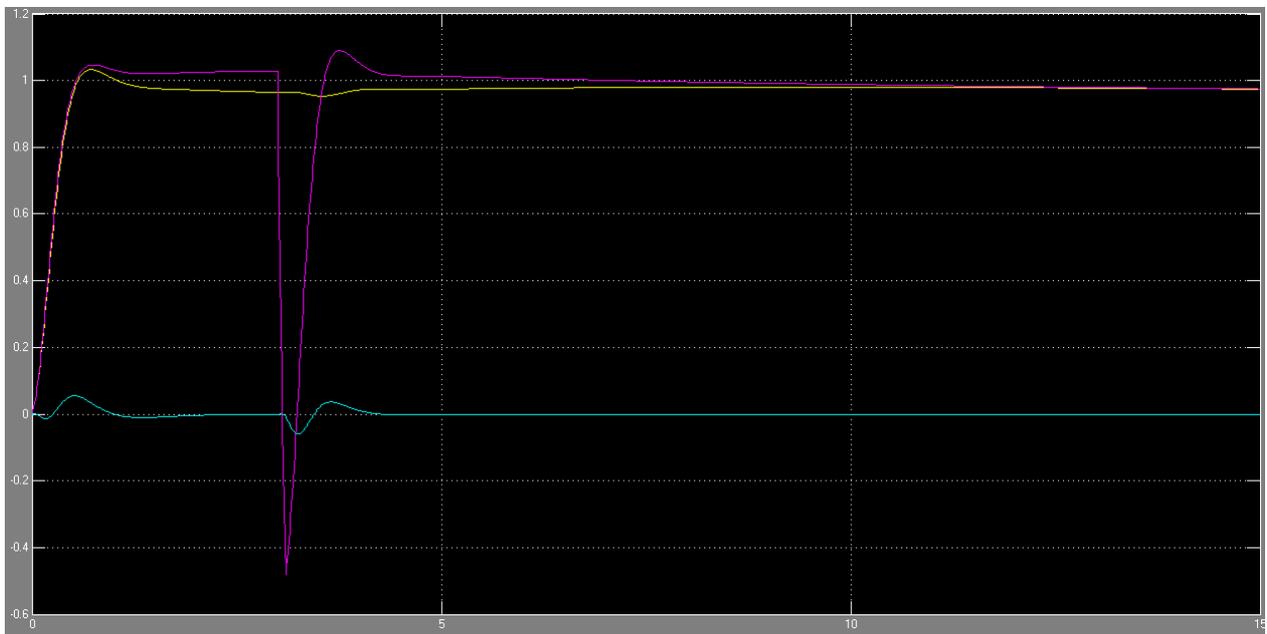


Figura 11.9 grafico del moto costante lungo N e E ed impulso lungo E in $\dot{N}, \dot{E}, \dot{D}$

U,V,W

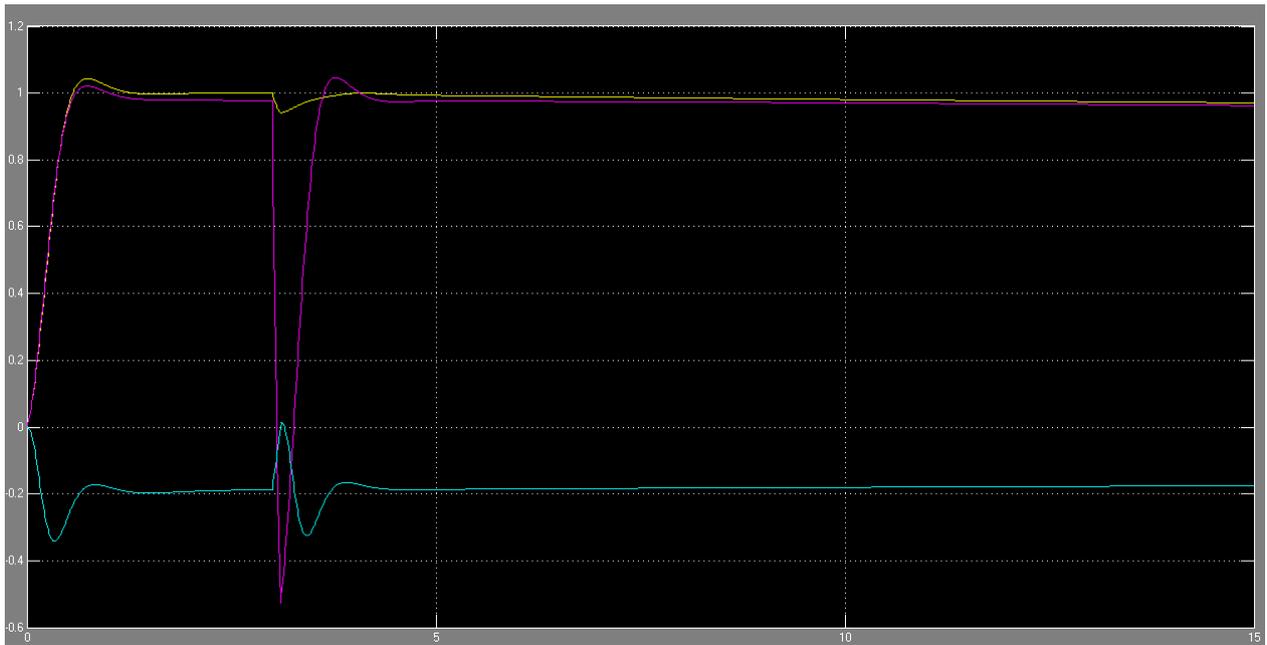


Figura 11.10 grafico del moto costante lungo N e E ed impulso lungo E in U,V,W

N,E,D

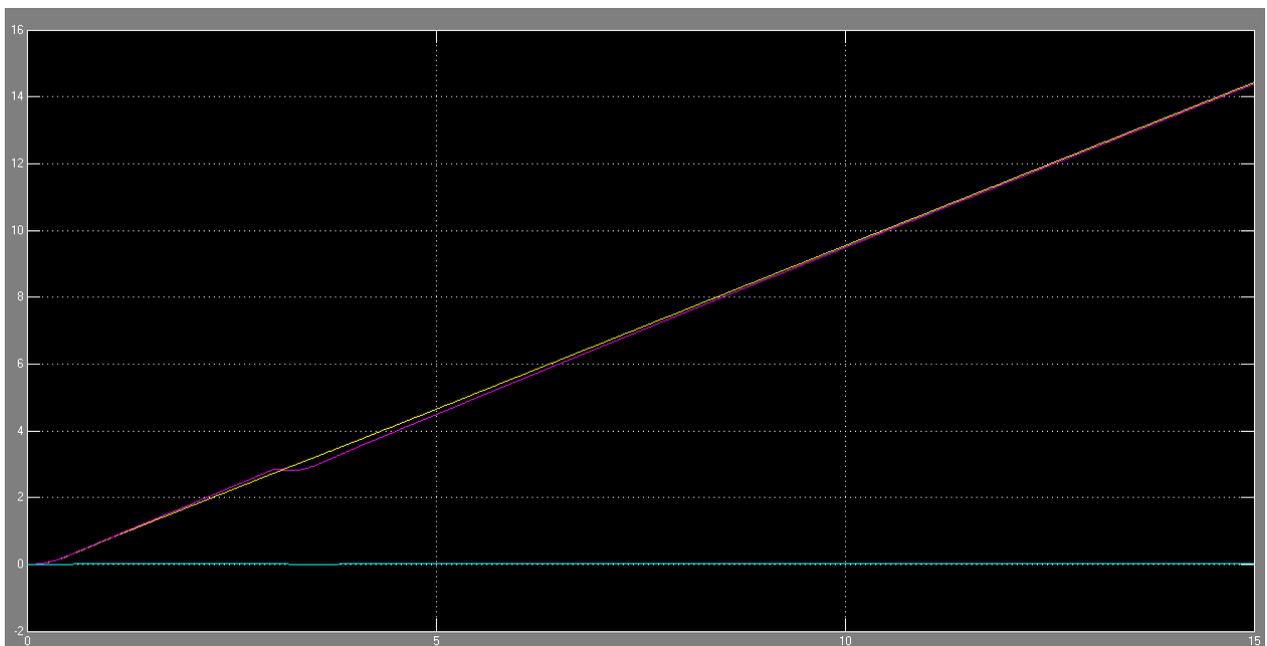


Figura 11.11 grafico del moto costante lungo N e E ed impulso lungo E in N,E,D

Anche in questo caso la risposta del sistema si ritiene molto soddisfacente in quanto il sistema è sia robusto che pronto ai disturbi esterni.

In fine si testa il software ad un moto lungo N e un forza contraria a questa direzione e costante, le condizioni iniziali risultano:

$$\begin{cases} \dot{N}_0 = 1 \text{ m/s} \\ \dot{E}_0 = 0 \text{ m/s} \\ \dot{D}_0 = 0 \text{ m/s} \end{cases}$$

Ed una forza costante lungo N che appare a 3 s dall'inizio della prova di entità pari a -2 m/s

Si ottengono :

$$\begin{aligned} \varphi &= 0 \\ \vartheta &= -0.048 \\ U_0 &= 1 \\ V_0 &= 0 \\ W_0 &= -0.048 \end{aligned}$$

$\dot{N}, \dot{E}, \dot{D}$

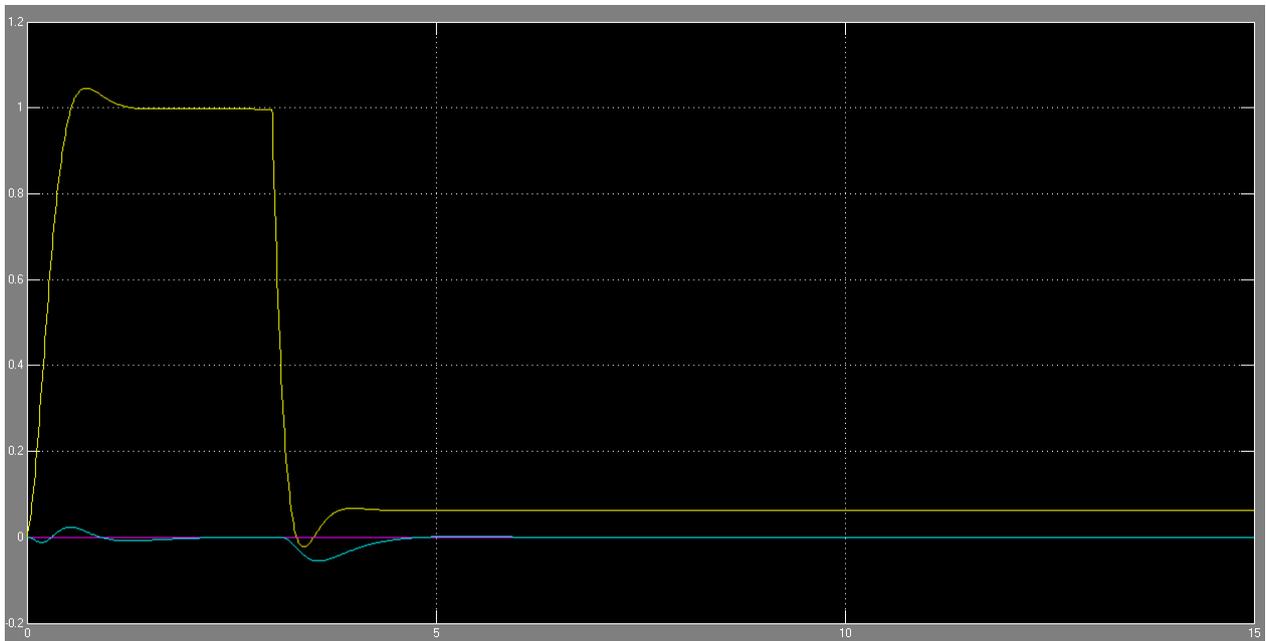


Figura 11.12 grafico del moto costante lungo N e forza costante lungo N in $\dot{N}, \dot{E}, \dot{D}$

U,V,W

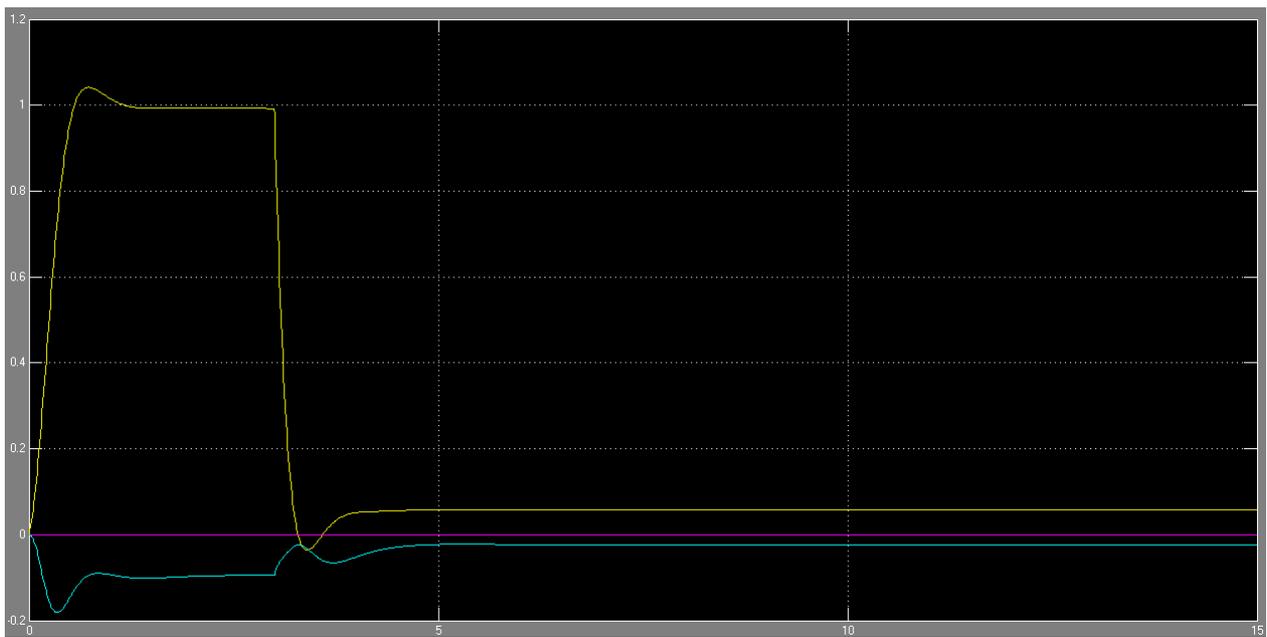


Figura 11.13 grafico del moto costante lungo N e forza costante lungo N in U,V,W

N,E,D

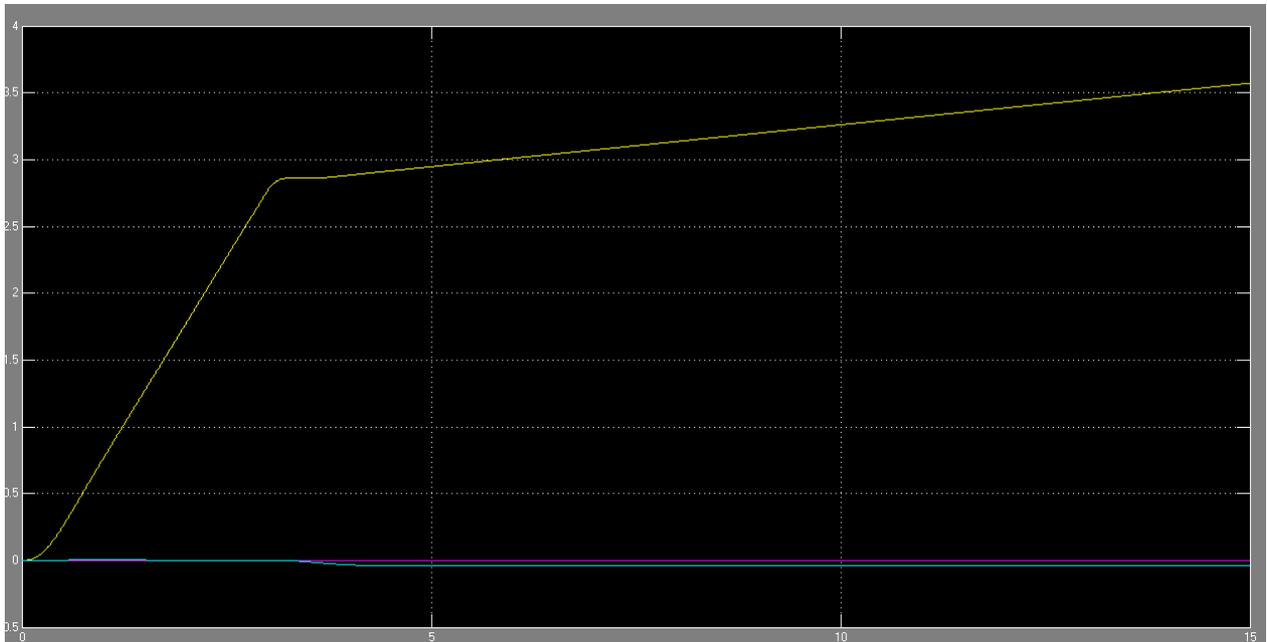


Figura 11.14 grafico del moto costante lungo N e forza costante lungo N in N,E,D

Come è mostrato dai grafici, questo sistema non è efficace per le forze costanti, ma la risposta ne risente in modo consistente.

Sia nel plot in $\dot{N}, \dot{E}, \dot{D}$ e sia in quello in U, V, W si nota come, all'apparire del vento, cali drasticamente la velocità di crociera.

Tutto sommato la velocità assoluta \dot{D} ritorna a zero, quindi il vento non ha effetti sulla quota.

Infatti se si analizza il grafico N, E, D si impara che la posizione lungo N con la presenza del vento muta, difatti varia l'inclinazione della retta, mentre la quota subisce uno spostamento del tutto trascurabile.

La presenza di un disturbo costante crea danni alla risposta del sistema che il controllore non è in grado di collimare e correggere, quindi si necessita un ulteriore sviluppo.

Si necessita costruire un controllore che valuti anche i disturbi del vento e li annulli.

11.12 COMPILAZIONE MATRICI R E Q

Vediamo che valori sono stati assegnati alle matrici di costo per il calcolo dell'indice di costo J .
come prima considerazione in questa tipologia di controllo ciò che noi desideriamo è ottimizzare la velocità in tutte le direzioni e la quota in D, mentre essendo il vento in direzione verticale solitamente di piccole entità (di solito è una componente dovuto alla termica, ovvero alle ascensioni di vento dovute alla convezione naturale dell'aria creata dalla differenza di temperatura tra il suolo e l'atmosfera).

Quindi si penalizza il feedback in direzione verticale in quanto il disturbo sarà di piccola entità mentre per le restanti direzioni la retroazione sarà più robusta e pronta.

In figura 11.15 si mostra come un moto lungo N con vento contrario nella stessa direzione provochi uno scostamento in quota se il relativo coefficiente viene penalizzato e posto uguale a 1000.

N,E,D

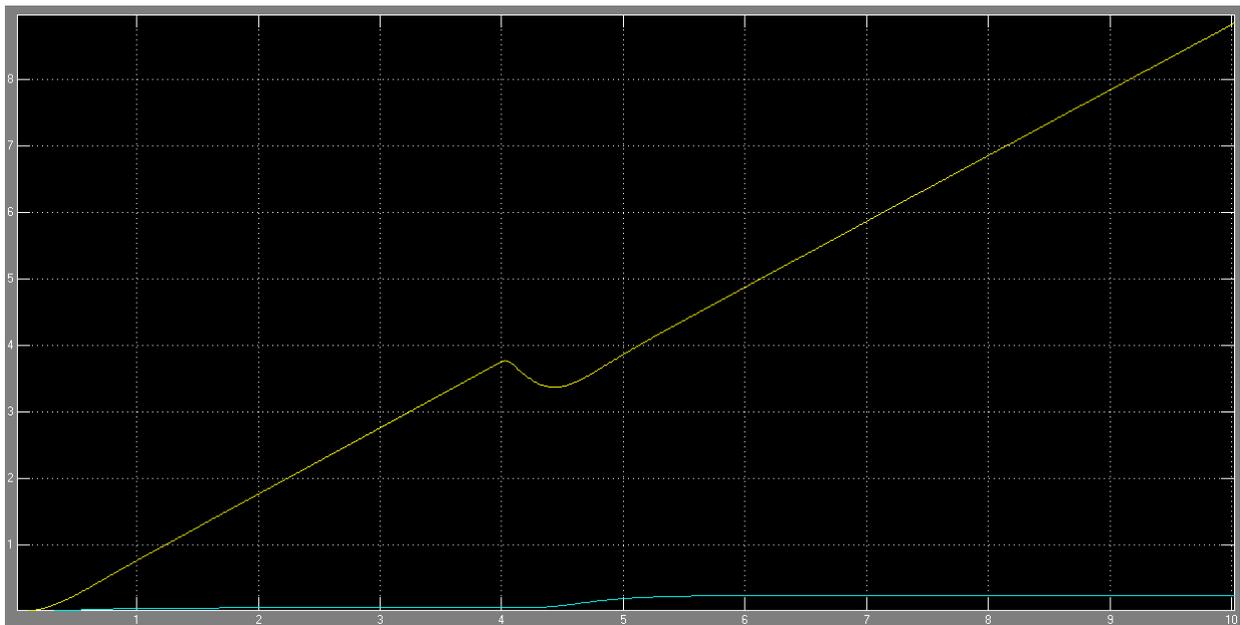


Figura 11.15 grafico del moto costante lungo N e forza costante lungo N in N,E,D

I valori ottimali trovati dopo prove empiriche risultano in fine :

$$\left\{ \begin{array}{l} q_1, q_2, q_3 = 1000 \\ q_4, q_5, q_6 = 1 \\ q_7, q_8, q_9, q_{12} = 0.001 \\ q_{10}, q_{11} = 1000 \\ r_1, r_2, r_3, r_4 = 100 \end{array} \right.$$

12. METODO DELLE MATRICI ESTESE

Dato il sistema originale :

$$\dot{x} = \mathbf{A}x + \mathbf{G}d\omega + \mathbf{D}u_d$$

$$y = \mathbf{C}x$$

È possibile riscriverlo in una nuova forma tale da annullare l'effetto del vento anche se questo è un'entità costante nel tempo, difatti essendo immutabile significa che la sua derivata è nulla.

Avremo dopo tale considerazione l'aggiunta di una nuova equazione:

$$\dot{u}_d = 0$$

Sotto questa nuova ipotesi è lecito introdurre un nuovo vettore delle variabili x_e detto appunto esteso :

$$\dot{x}_e = \begin{pmatrix} \dot{x} \\ \dot{u}_d \end{pmatrix}$$

Si può procedere ora con un nuovo sistema matriciale così formulato :

$$\dot{x}_e = \bar{\mathbf{A}}x_e + \bar{\mathbf{G}}d\omega$$

$$y_e = \bar{\mathbf{C}}x_e$$

Dove :

$$\bar{\mathbf{A}} = \begin{pmatrix} \mathbf{A} & \mathbf{D} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} = \text{matrice estesa di } \mathbf{A}$$

$$\bar{\mathbf{G}} = \begin{pmatrix} \mathbf{G} \\ \mathbf{0} \end{pmatrix} = \text{matrice estesa di } \mathbf{G}$$

$$\bar{\mathbf{C}} = \begin{pmatrix} \mathbf{C} \\ \mathbf{0} \end{pmatrix} = \text{matrice estesa di } \mathbf{C}$$

Ora studiamo la matrice $\bar{\mathbf{A}}$ per capirne la dimensione.

Essendo composta da una matrice 12x12 (\mathbf{A}) e da una matrice 6x6 (\mathbf{D}) otterremo con un semplice ragionamento una matrice 18x18.

Applicando questa logica alle altre avremo che la $\bar{\mathbf{G}}$ sarà una 18x4 e la $\bar{\mathbf{C}}$ una 18x18.

Non resta che riformulare l'indice di comportamento che a sua volta è riscritto in forma estesa:

$$J_e = \int_{t_0}^{\infty} \{d\omega^T(t) \cdot \bar{\mathbf{R}} \cdot d\omega(t) + x_e^T(t) \cdot \bar{\mathbf{Q}} \cdot x_e(t)\} dt$$

Con:

$$\bar{\mathbf{R}} = \mathbf{R}$$

$$\bar{\mathbf{Q}} = \mathbf{Q}$$

Definiamo ora la legge che descrive u_d , composto da forze e momenti.

Secondo lo studio appena stilato, necessitiamo di un valore la cui derivata si annulla, ed essendo il vento una quantità che prima è considerata nulla ed ad un certo istante cambia fino ad un certo valore possiamo considerarlo come un disturbo a forma di gradino, quindi:

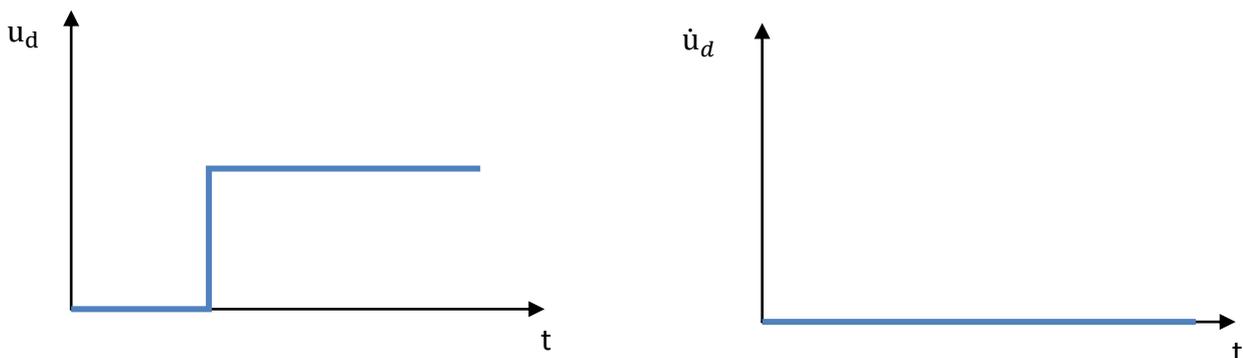


Figura12.1 vento di tipo a gradino e la sua derivata

Quindi il sistema appena studiato è in grado di vincere forze costanti, che fino ad ora, quando affliggevano il modello, producevano una risposta inaccettabile.

12.1 IMPLEMENTAZIONE IN SIMULINK

Per trascrivere tutte le ipotesi trattate nel capitolo precedente in linguaggio Matlab bisogna rivedere tutta la struttura delle equazioni di stato per il calcolo della matrice **k**.

Come già enunciato bisogna estendere le matrici con i disturbi al loro interno, otteniamo così:

```
A=[0 0 0 1 0 0 0 0 0 0 0 0 0;
    0 0 0 0 1 0 0 0 0 0 0 0 0;
    0 0 0 0 0 1 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 -9.81 0 0 0 0 0 0 0 0 0 0 0;
    9.81 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 1 0 0 0 0 0 0;
    0 0 0 0 0 0 0 1 0 0 0 0 0;
    0 0 0 0 0 0 0 0 1 0 0 0 0];

D=[0 0 0 0 0 0;
    0 0 0 0 0 0;
    0 0 0 0 0 0;
    0 0 0 1/Ix 0 0;
    0 0 0 0 1/Iy 0;
    0 0 0 0 0 1/Iz;
    1/m 0 0 0 0 0;
    0 1/m 0 0 0 0;
    0 0 1/m 0 0 0;
    0 0 0 0 0 0;
    0 0 0 0 0 0;
    0 0 0 0 0 0];

G=[0 0 0 0;
    0 0 0 0;
    0 0 0 0;
    2*Ks*om0*d*sin(al)/Ix -2*Ks*om0*d*cos(al)/Ix -2*Ks*om0*d*sin(al)/Ix 2*Ks*om0*d*cos(al)/Ix;
    2*Ks*om0*d*cos(al)/Iy 2*Ks*om0*d*sin(al)/Iy -2*Ks*om0*d*cos(al)/Iy -2*Ks*om0*d*sin(al)/Iy;
    -2*Kd*om0/Iz 2*Kd*om0/Iz -2*Kd*om0/Iz 2*Kd*om0/Iz;
    0 0 0 0;
    0 0 0 0;
    -2*Ks*om0/m -2*Ks*om0/m -2*Ks*om0/m -2*Ks*om0/m;
    0 0 0 0;
    0 0 0 0;
    0 0 0 0];

r=[1 0 0 0;
    0 1 0 0;
    0 0 1 0;
    0 0 0 1];

q=[1 0 0 0 0 0 0 0 0 0 0 0 0;
    0 1 0 0 0 0 0 0 0 0 0 0 0;
    0 0 1 0 0 0 0 0 0 0 0 0 0;
    0 0 0 1 0 0 0 0 0 0 0 0 0;
    0 0 0 0 1 0 0 0 0 0 0 0 0;
    0 0 0 0 0 1 0 0 0 0 0 0 0;
    0 0 0 0 0 0 1 0 0 0 0 0 0;
    0 0 0 0 0 0 0 1 0 0 0 0 0;
    0 0 0 0 0 0 0 0 1 0 0 0 0;
    0 0 0 0 0 0 0 0 0 1 0 0 0;
    0 0 0 0 0 0 0 0 0 0 1 0 0;
    0 0 0 0 0 0 0 0 0 0 0 1 0];

C=eye(12);

R=inv(r);

Aex=[A,D;zeros(6,18)];
```

```
Gex=[G;zeros(6,4)];  
Cex=[C;zeros(6,12)];  
  
[K] = LQR(Aex,Gex,Cex'/q*Cex,R);
```

Questa scrittura però è incorretta, difatti Matlab non produce alcuna compilazione e restituisce un errore nella dimensione di \mathbf{Q} che essendo una 12×12 non è compatibile con la $\bar{\mathbf{A}}$ di dimensioni 18×18 .

È necessario quindi utilizzare uno “svincolo matematico” per risolvere questa apparente situazione di stallo.

La soluzione è suddividere il sistema in due sotto sistemi, uno che chiameremo controllabile e l'altro non controllabile.

Nel primo ne fanno parte tutte le variabili di stato su cui noi possiamo agire attivamente e controllare il suo sviluppo istante per istante, mentre nel secondo rientrano i disturbi, ovvero il vento che è un entità a se stante quindi un effetto imposto da condizioni esterne a noi ignote.

Vediamo nel capitolo che segue lo sviluppo di questa idea di base.

I valori di \mathbf{R} e \mathbf{Q} sono tutti pari all'unità in quanto questo è un test preliminare di validità.

12.2. MODELLO NON COMPLETAMENTE CONTROLLABILE

Tutte le ipotesi enunciate nel capitolo precedente quindi erano postulati per la regolazione di un sistema completamente controllabile, ovvero anche la matrice **D** conteneva disturbi che erano controllabili dal sistema stesso ed in particolare la matrice **Q** sarebbe risultata una 18x18.

Tale affermazione non descrive il nostro caso dove i disturbi non sono altro che forze e momenti causate dal vento e quindi impossibili da controllare e retroazionare.

È necessario, come già preannunciato, scomporre le matrici in due parti, avremo quindi utilizzando le medesime annotazioni:

$$\dot{x}_e = \bar{\mathbf{A}}x_e + \bar{\mathbf{G}}d\omega$$

$$y_e = \bar{\mathbf{C}}x_e$$

$$\bar{\mathbf{A}} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{D} \\ \mathbf{0} & \mathbf{A}_{22} \end{pmatrix} = \text{matrice estesa di } \mathbf{A}$$

$$\bar{\mathbf{G}} = \begin{pmatrix} \mathbf{G} \\ \mathbf{0} \end{pmatrix} = \text{matrice estesa di } \mathbf{G}$$

$$\bar{\mathbf{C}} = \begin{pmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{pmatrix} = \text{matrice estesa di } \mathbf{C}$$

Con $(\bar{\mathbf{A}}, \bar{\mathbf{G}})$ non completamente controllabili, ma identifichiamo:

A, **G** = completamente controllabili

D = non completamente controllabile

È possibile dunque scomporre l'equazione di Riccati nel seguente modo:

$$\mathbf{S}_{11} \cdot \mathbf{A}_{11} + \mathbf{A}_{11}^T \cdot \mathbf{S}_{11} - (\mathbf{G}^T \cdot \mathbf{S}_{11})^T \cdot \mathbf{R}^{-1} \cdot (\mathbf{G}^T \cdot \mathbf{S}_{11}) + \mathbf{C}_1^T \cdot \mathbf{Q} \cdot \mathbf{C}_1 = \mathbf{0}$$

$$[\mathbf{A}_{11}^T - (\mathbf{S}_{11} \cdot \mathbf{G})] \cdot \mathbf{S}_{12} + \mathbf{S}_{12} \cdot \mathbf{A}_{22} + \mathbf{S}_{11} \cdot \mathbf{A}_{12} + \mathbf{C}_1^T \cdot \mathbf{Q} \cdot \mathbf{C}_2 = \mathbf{0}$$

$$\mathbf{A}_{22}^T \cdot \mathbf{S}_{22} + \mathbf{S}_{22} \cdot \mathbf{A}_{22} + \mathbf{S}_{12}^T \cdot \mathbf{A}_{22} + \mathbf{A}_{12}^T \cdot \mathbf{S}_{12} - \mathbf{S}_{12}^T \cdot \mathbf{G} \cdot \mathbf{R}^{-1} \cdot \mathbf{G}^T \cdot \mathbf{S}_{12} + \mathbf{C}_2^T \cdot \mathbf{Q} \cdot \mathbf{C}_2 = \mathbf{0}$$

Essendo:

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{21} & \mathbf{S}_{22} \end{pmatrix}$$
$$\mathbf{C}_2 = \mathbf{0}$$

La prima funzione è una classica funzione di Riccati, mentre le altre due sono funzioni di Lyapunov, la prima in \mathbf{S}_{12} mentre la seconda in \mathbf{S}_{22} .

E quindi per determinare la legge di controllo ottima:

$$u(t) = -\mathbf{R}^{-1} \cdot \mathbf{G}^T \cdot \mathbf{S} \cdot x(t)$$

Ovvero:

$$-\mathbf{R}^{-1} \cdot \mathbf{G}^T \cdot \mathbf{S} \cdot x(t) = \mathbf{R}^{-1} \cdot [\mathbf{G}^T \cdot \mathbf{S}_{11} ; \mathbf{G}^T \cdot \mathbf{S}_{12}]$$

Non è nemmeno necessario ricavare la \mathbf{S}_{22} dalla terza equazione.

12.3. IMPLEMENTAZIONE IN MATLAB DEL MODELLO NON COMPLETAMENTE CONTROLLABILE

Se applichiamo tutti gli assiomi del capitolo appena trattato, otteniamo una struttura in linguaggio Matlab così composta:

```

A11=[0 0 0 1 0 0 0 0 0 0 0 0;
      0 0 0 0 1 0 0 0 0 0 0 0;
      0 0 0 0 0 1 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0;
      0 -9.81 0 0 0 0 0 0 0 0 0 0;
      9.81 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 1 0 0 0 0;
      0 0 0 0 0 0 0 0 1 0 0 0;
      0 0 0 0 0 0 0 0 0 1 0 0];

D=[0 0 0 0 0 0;
   0 0 0 0 0 0;
   0 0 0 0 0 0;
   0 0 0 1/Ix 0 0;
   0 0 0 0 1/Iy 0;
   0 0 0 0 0 1/Iz;
   1/M 0 0 0 0 0;
   0 1/M 0 0 0 0;
   0 0 1/M 0 0 0;
   0 0 0 0 0 0;
   0 0 0 0 0 0;
   0 0 0 0 0 0];

A22=zeros(6,6);

G=[0 0 0 0;
   0 0 0 0;
   0 0 0 0;
   2*Ks*om0*d*sin(al)/Ix -2*Ks*om0*d*cos(al)/Ix -2*Ks*om0*d*sin(al)/Ix 2*Ks*om0*d*cos(al)/Ix;
   2*Ks*om0*d*cos(al)/Iy 2*Ks*om0*d*sin(al)/Iy -2*Ks*om0*d*cos(al)/Iy -2*Ks*om0*d*sin(al)/Iy;
   -2*Kd*om0/Iz 2*Kd*om0/Iz -2*Kd*om0/Iz 2*Kd*om0/Iz;
   0 0 0 0;
   0 0 0 0;
   -2*Ks*om0/M -2*Ks*om0/M -2*Ks*om0/M -2*Ks*om0/M;
   0 0 0 0;
   0 0 0 0;
   0 0 0 0];

r=[100 0 0 0;
   0 100 0 0;
   0 0 100 0;
   0 0 0 100];

q=[1 0 0 0 0 0 0 0 0 0 0 0;
   0 1 0 0 0 0 0 0 0 0 0 0;
   0 0 1 0 0 0 0 0 0 0 0 0;
   0 0 0 1 0 0 0 0 0 0 0 0;
   0 0 0 0 1 0 0 0 0 0 0 0;
   0 0 0 0 0 1 0 0 0 0 0 0;
   0 0 0 0 0 0 1 0 0 0 0 0;
   0 0 0 0 0 0 0 1 0 0 0 0;
   0 0 0 0 0 0 0 0 1 0 0 0;
   0 0 0 0 0 0 0 0 0 1 0 0;
   0 0 0 0 0 0 0 0 0 0 0.01 0;
   0 0 0 0 0 0 0 0 0 0 0 0.01;
   0 0 0 0 0 0 0 0 0 0 0 0.01];

C1=eye(15);

R=inv(r);

```

```

[k,S11] = LQR(A11,G,C1'/q*C1,R);
S12=lyap(A11'-S11*G/R*B1',D,S11*A12);
K=r*[G'*S11,G'*S12];
end

```

La struttura del sistema a blocchi retroazionato viene mostrato in figura 12.1.

Le condizioni iniziali (che coincide col tracking in questo caso) essendo pari a zero le possiamo omettere per focalizzare i reali interventi “strutturali” che ha subito lo schema.

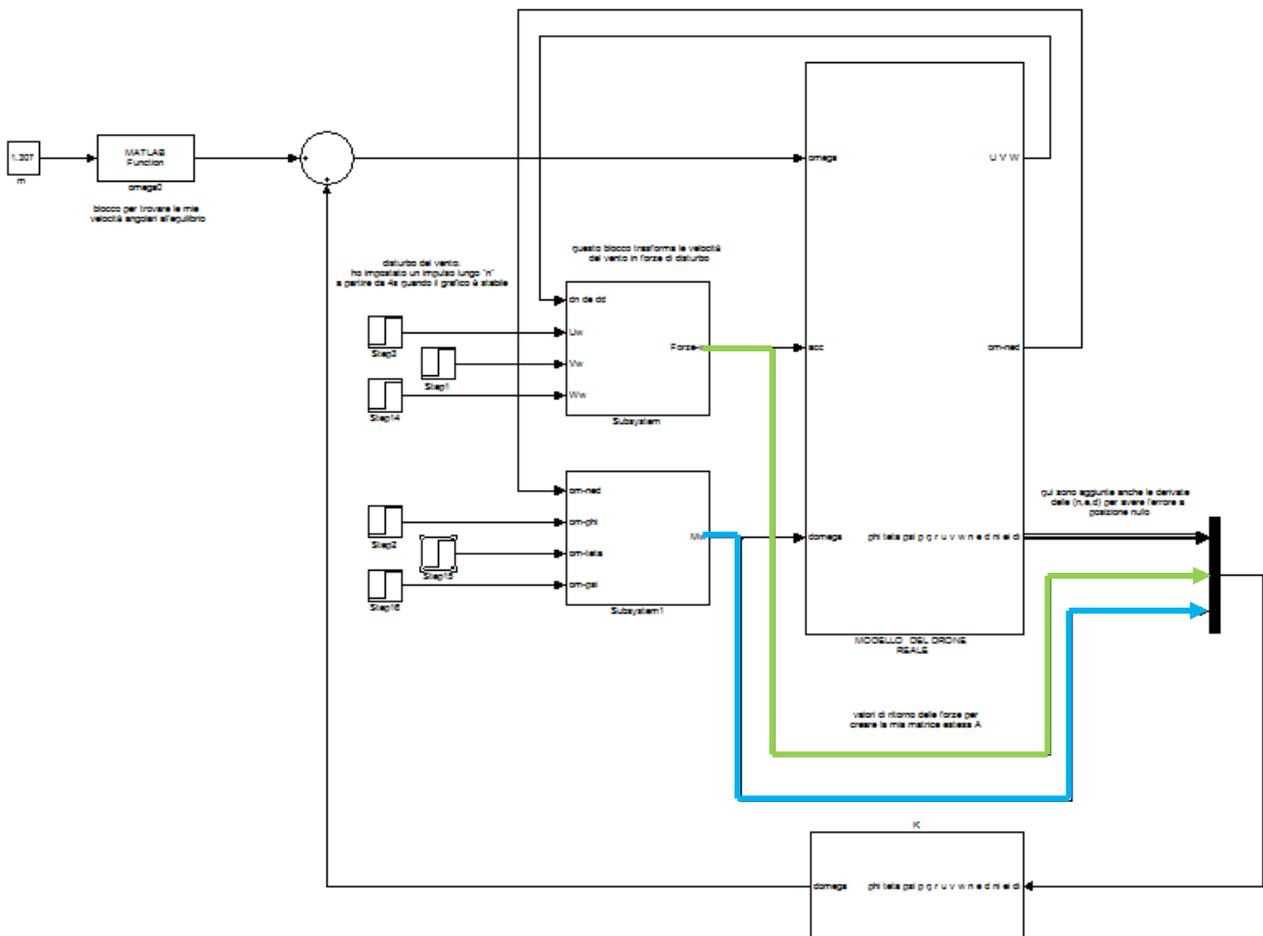


Figura 12.2 schema del sistema a blocchi nel caso di matrici estese

Come menzionato precedentemente, il regolatore prende in ingresso oltre che le variabili di stato, anche le forze e i momenti del vento (indicate in verde e blu).

12.4 VERIFICA DEL MODELLO NON COMPLETAMENTE CONTROLLABILE

Vediamo di affliggere il velivolo in condizioni di hovering ad un vento costante e di direzione lungo N di entità pari a 2 m/s (opportunamente trasformato poi in forza) che appare ad 1 s di inizio della prova.

I plot risultanti sono:

N,E,D

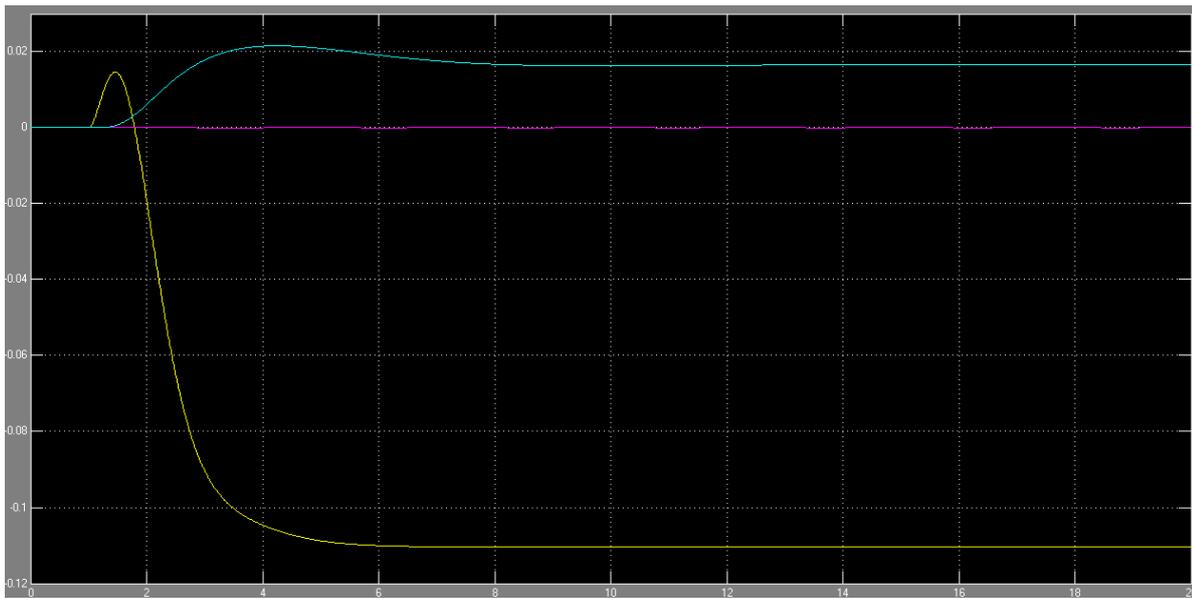


Figura 12.3 vento costante nel sistema esteso in N,E,D

U,V,W

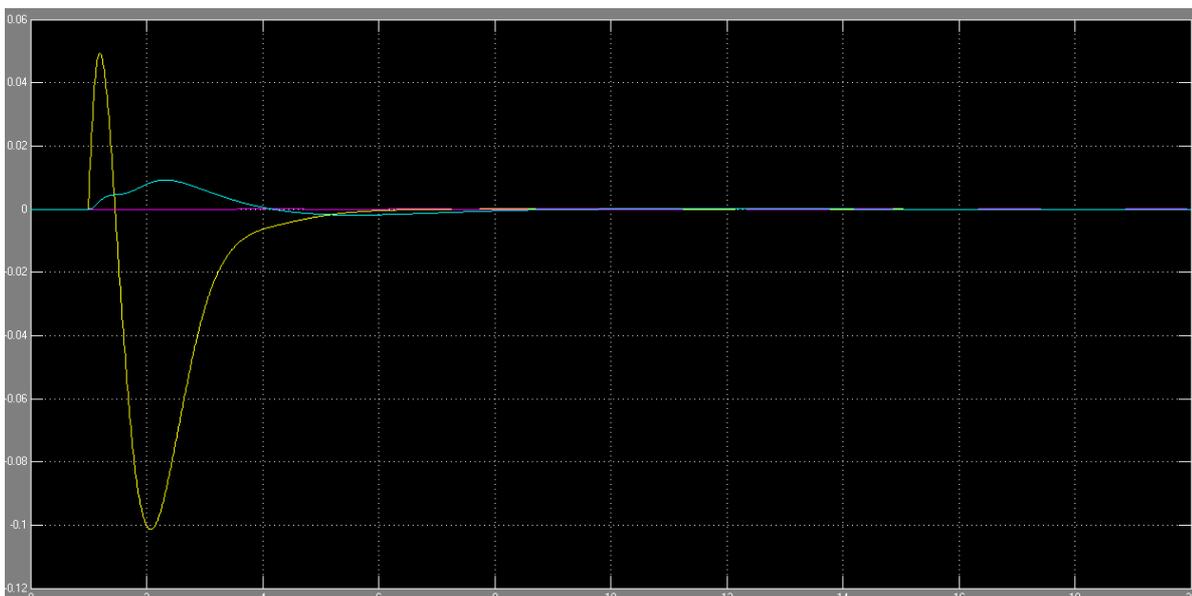


Figura 12.4 vento costante nel sistema esteso in U,V,W

Si nota immediatamente che nella terna N,E,D il vento produce ancora uno scostamento, seppure molto piccolo e trascurabile ($\approx 0.1\text{m}$ lungo N), mentre le velocità si annullano circa a 4 s dall'inizio della comparsa del vento.

Sebbene lo scostamento sia piccolo è anche vero che la velocità impostata del vento è piccola, quindi per venti consistenti avremmo divergenze dalle posizioni di hovering non più accettabili.

È necessario adottare un altro espediente.

Come soluzione si potrebbe pensare di aumentare il tipo del sistema, difatti se la velocità mostra un errore nullo asintotico per una retroazione sulle posizioni, basterà allora applicare il feedback all'integrale delle posizioni, così facendo aumenta il tipo del sistema e l'errore asintotico per le posizioni tenderà a zero.

Vedremo nel prossimo capitolo come si sviluppa questa teoria.

12.5 RICHIAMI DI CONTROLLI AUTOMATICI, TIPO DI UN SISTEMA

Consideriamo un sistema ad anello così ipotizzato:



Figura 12.5 esempio di sistema ad anello aperto

Si definisce quindi la funzione di trasferimento $G(S)$ sotto forma fattorizzata:

$$G(S) = K \cdot \frac{\prod_{k=1}^m (s + z_k)}{\prod_{i=1}^n (s + p_i)}$$

Con:

- K : guadagno dell'anello;
- z_k : zeri della funzione;
- p_i : poli della funzione.

La prima operazione è evidenziare i poli della funzione situati nell'origine.

Nell'ipotesi che $G(S)$ abbia un numero di poli nell'origine pari a τ , si ha:

$$G(S) = K \cdot \frac{\prod_{k=1}^m (s + z_k)}{s^\tau \cdot \prod_{i=1}^{n-\tau} (s + p_i)}$$

Come seconda operazione si può moltiplicare e dividere il sistema per $z_1 \dots z_m$ e per $p_1 \dots p_{n-\tau}$, si hanno i seguenti sviluppi matematici :

$$G(S) = K \cdot \frac{z_1 \dots z_m}{p_1 \dots p_{n-\tau}} \cdot \frac{\frac{s+z_1}{z_1} \cdot \frac{s+z_2}{z_2} \cdot \dots \cdot \frac{s+z_m}{z_m}}{s^\tau \cdot \left[\frac{s+p_1}{p_1} \cdot \frac{s+p_2}{p_2} \cdot \dots \cdot \frac{s+p_{n-\tau}}{p_{n-\tau}} \right]}$$

$$G(S) = \frac{K \cdot (z_1 \dots z_m)}{s^\tau \cdot (p_1 \dots p_{n-\tau})} \cdot \frac{[\left(\frac{s}{z_1} + 1\right) \cdot \left(\frac{s}{z_2} + 1\right) \cdot \dots \cdot \left(\frac{s}{z_m} + 1\right)]}{[\left(\frac{s}{p_1} + 1\right) \cdot \left(\frac{s}{p_2} + 1\right) \cdot \dots \cdot \left(\frac{s}{p_{n-\tau}} + 1\right)]}$$

E quindi:

$$G(S) = K' \cdot \frac{\prod_{k=1}^m (1 + s \cdot T_k)}{s^\tau \cdot \prod_{i=1}^{n-\tau} (1 + s \cdot T_i)}$$

Dove nel termine $K' = \frac{K \cdot (z_1 \dots z_m)}{(p_1 \dots p_{n-\tau})}$ sono stati inglobati il guadagno, gli m zeri e gli n-τ poli della funzione e dove le T_k si identificano come costanti di tempo (misurate in secondi).

Si definisce quindi il tipo di un sistema il numero τ di poli che il sistema presenta nell'origine.

Quindi l'errore a regime si esprime come:

$$e_\infty = \lim_{t \rightarrow \infty} S \cdot E(S) = \lim_{S \rightarrow 0} S \cdot \frac{R(S)}{1 + G(S)}$$

Secondo la teoria esposta quindi per avere a regime un errore nullo bisogna aumentare il tipo del sistema.

Va da se che la trasformata di Laplace di una funzione integrale $g(t) = \int_0^t f(t) dt$ da come risultato:

$$L[g(t)] = \frac{L[f(t)]}{S} = \frac{F(S)}{S}$$

Quindi integrando se la nostra funzione in ingresso r(t) è l'integrale di una certa funzione avremo che il sistema aumenta di tipo e va a zero l'errore a regime se prima era delimitato ad un certo valore diverso da zero.

12.6 ESPANSIONE DEL MODELLO NON COMPLETAMENTE CONTROLLABILE CON GLI INTEGRALI DELLE POSIZIONI

Per ridurre l'errore a regime in un sistema, secondo la teoria dei controlli automatici, è necessario aumentarne il tipo ovvero integrare una volta l'intera funzione di trasferimento.

Essendo la retroazione effettuata sulle posizioni implica che è sufficiente inserire nell'equazione delle variabili di stato l'integrale delle posizioni.

Così facendo la risposta tenderà a zero per il tempo che tenderà all'infinito.

È sufficiente allora inserire queste nuove variabili di stato nel sistema trattato fino ad ora.

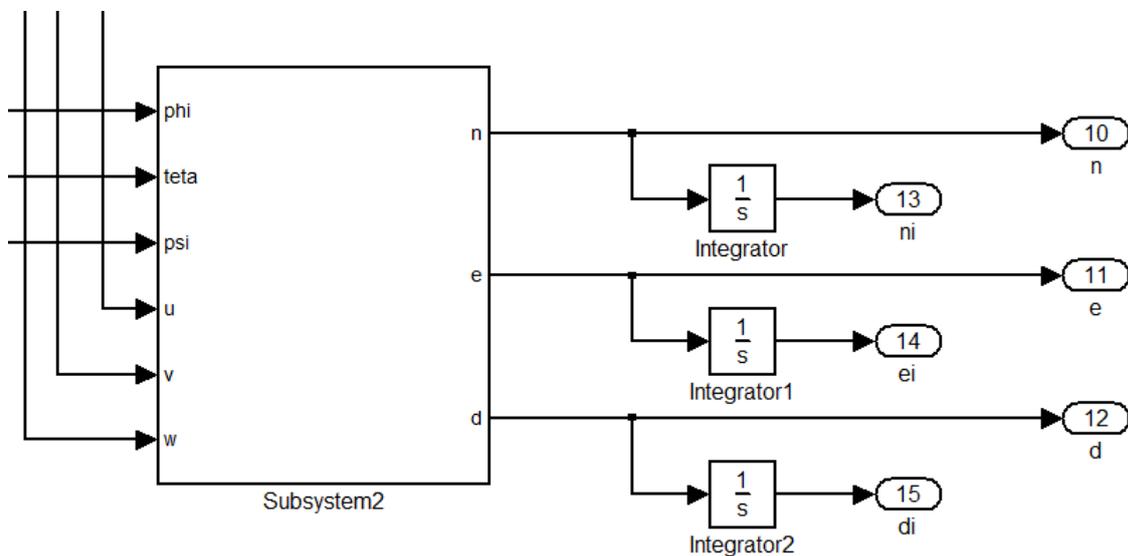


Figura 12.6 aggiunta dell'integrale delle posizioni assolute

Come si mostra in figura 12.6 sono state aggiunte nuove variabili di stato al sistema semplicemente integrando le posizioni elaborate dal blocco composto con le equazioni canoniche del moto.

Ovviamente l'aggiunta di queste variabili va a modificare il sistema a blocchi come visto ma anche la struttura dello script per il calcolo della matrice K del regolatore.

Si riporta di seguito la nuova forma dello script per il calcolo della matrice K per zero errore a regime.

```

A11=[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 -9.81 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      9.81 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0];

```

} Aggiunta delle nuove variabili di stato

```

D=[0 0 0 0 0 0;
   0 0 0 0 0 0;
   0 0 0 0 0 0;
   0 0 0 1/Ix 0 0;
   0 0 0 0 1/Iy 0;
   0 0 0 0 0 1/Iz;
   1/M 0 0 0 0 0;
   0 1/M 0 0 0 0;
   0 0 1/M 0 0 0;
   0 0 0 0 0 0;
   0 0 0 0 0 0;
   0 0 0 0 0 0;
   0 0 0 0 0 0;
   0 0 0 0 0 0;
   0 0 0 0 0 0;
   0 0 0 0 0 0];

```

```
A22=zeros(6,6);
```

```

G=[0 0 0 0;
   0 0 0 0;
   0 0 0 0;
   2*Ks*om0*D*sin(al)/Ix -2*Ks*om0*D*cos(al)/Ix -2*Ks*om0*D*sin(al)/Ix 2*Ks*om0*D*cos(al)/Ix;
   2*Ks*om0*D*cos(al)/Iy 2*Ks*om0*D*sin(al)/Iy -2*Ks*om0*D*cos(al)/Iy -2*Ks*om0*D*sin(al)/Iy;
   -2*Kd*om0/Iz 2*Kd*om0/Iz -2*Kd*om0/Iz 2*Kd*om0/Iz;
   0 0 0 0;
   0 0 0 0;
   -2*Ks*om0/M -2*Ks*om0/M -2*Ks*om0/M -2*Ks*om0/M;
   0 0 0 0;
   0 0 0 0;
   0 0 0 0;
   0 0 0 0;
   0 0 0 0;
   0 0 0 0];

```

```

r=[1000 0 0 0;
   0 1000 0 0;
   0 0 1000 0;
   0 0 0 1000];

```

```

q=[100 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 100 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 100 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 100 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 100 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 100 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 100 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 100 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0.001 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0.001 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0.001 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0.001 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0.001 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.001 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.001];

```

} Feedback effettuato sulle nuove variabili di stato

```

C1=eye(15);
R=inv(r);
[S11] = LQR(A11,G,C1'/q*C1,R);
S12=lyap(A11'-S11*G/R*G',A22,S11*D);

K=r*[G'*S11,G'*D];

end

```

Come detto alla matrice A si aggiungono gli integrali delle posizioni ampliando la sua dimensione da una 12x12 ad una 15x15, così come la matrice Q.

Il sistema aggiunto è il seguente:

$$\left\{ \begin{array}{l} \dot{N}_i = N \\ \dot{E}_i = E \\ \dot{D}_i = D \end{array} \right.$$

Dove:

$$\left\{ \begin{array}{l} N_i = \int N \\ E_i = \int E \\ D_i = \int D \end{array} \right.$$

12.8 TEST E VERIFICA DEL MODELLO NON COMPLETAMENTE CONTROLLABILE

Non rimane che testare il programma con i valori della prova precedente e verificare che a regime gli errori abbiano un valore asintotico pari a zero.

Si re impongono quindi gli stessi valori dei disturbi assunti nel capitolo precedente ovvero vento costante lungo N di entità pari a 2 m/s applicato ad 1 s dall'inizio del test.

N,E,D

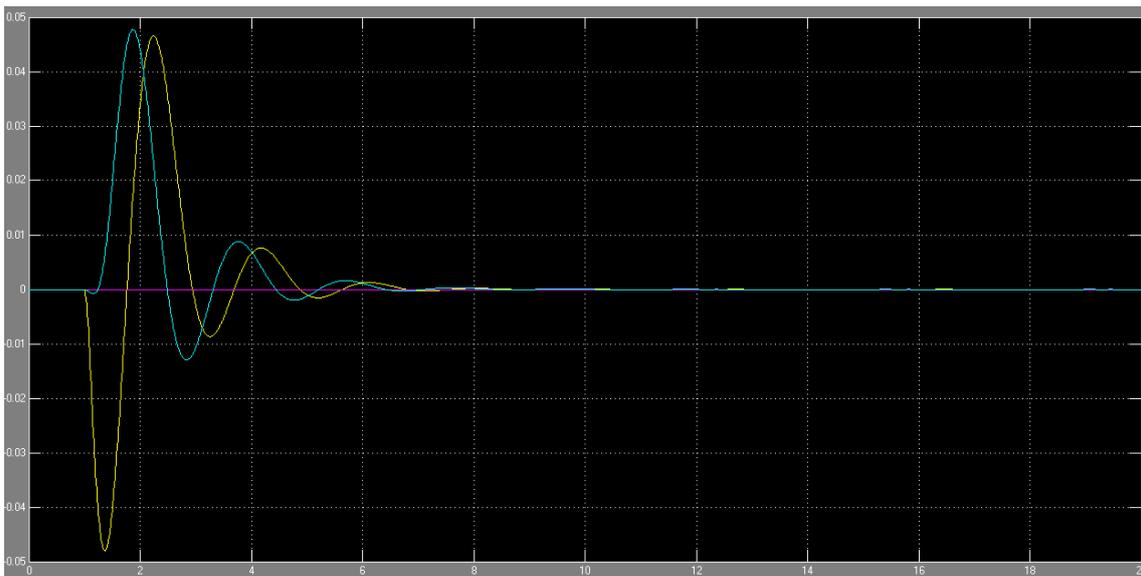


Figura 12.7 vento costante nel sistema esteso e con integrali delle posizioni in N,E,D

U,V,W

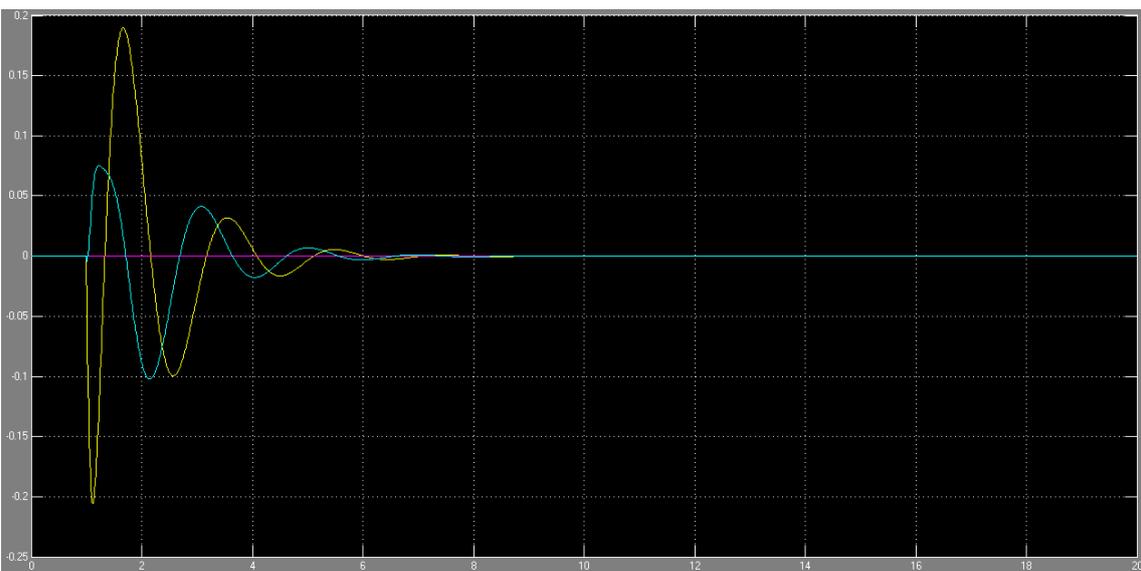


Figura 12.8 vento costante nel sistema esteso e con integrali delle posizioni in U,V,W

La retroazione in questo modo risulta ottima e più che valida portando l'errore a regime nullo.
Vediamo ora un caso più complicato, ovvero il caso precedente sommato ad un vento lungo E di entità pari a 2 m/s ed una torsione dovuto ad un vortice lungo l'asse yaw di velocità pari a 0.1 rad/s.

N,E,D

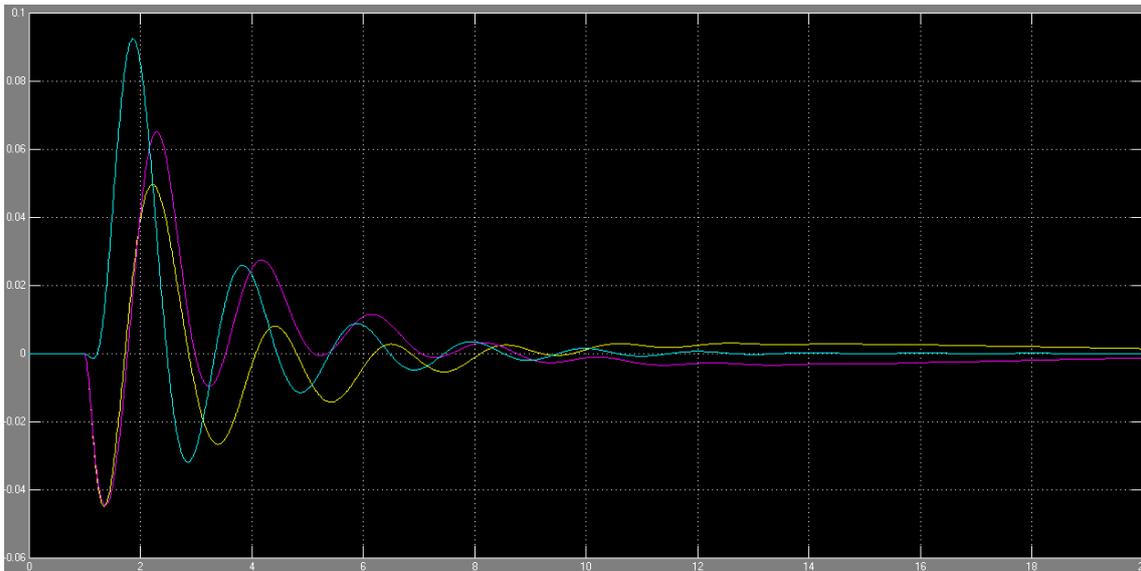


Figura 12.9 vento costante nel sistema esteso e con integrali delle posizioni in N,E,D

U,V,W

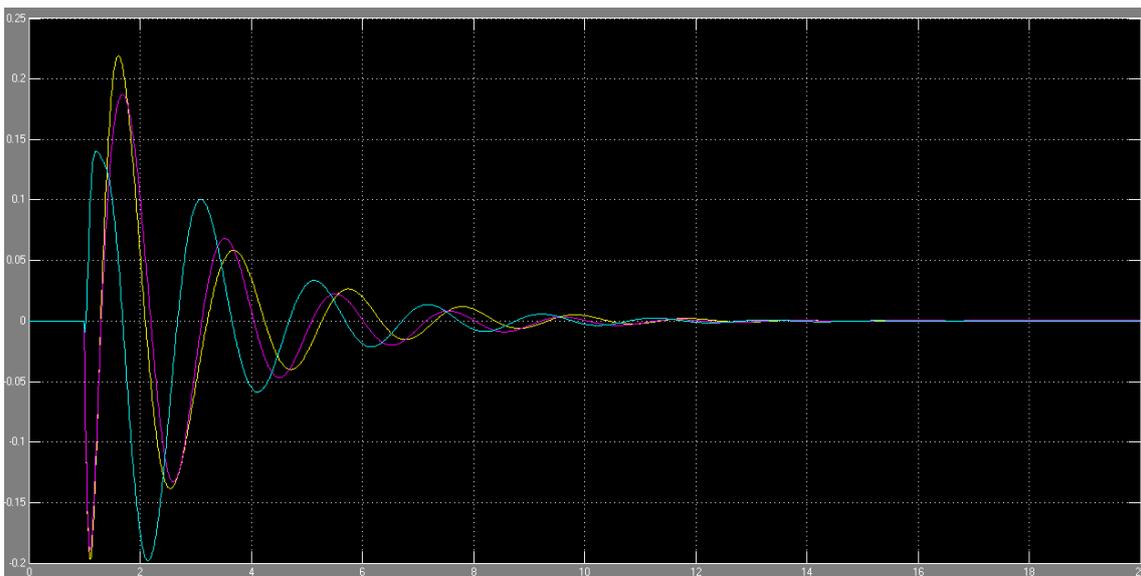


Figura 12.10 vento costante nel sistema esteso e con integrali delle posizioni in U,V,W

Φ, Θ, Ψ

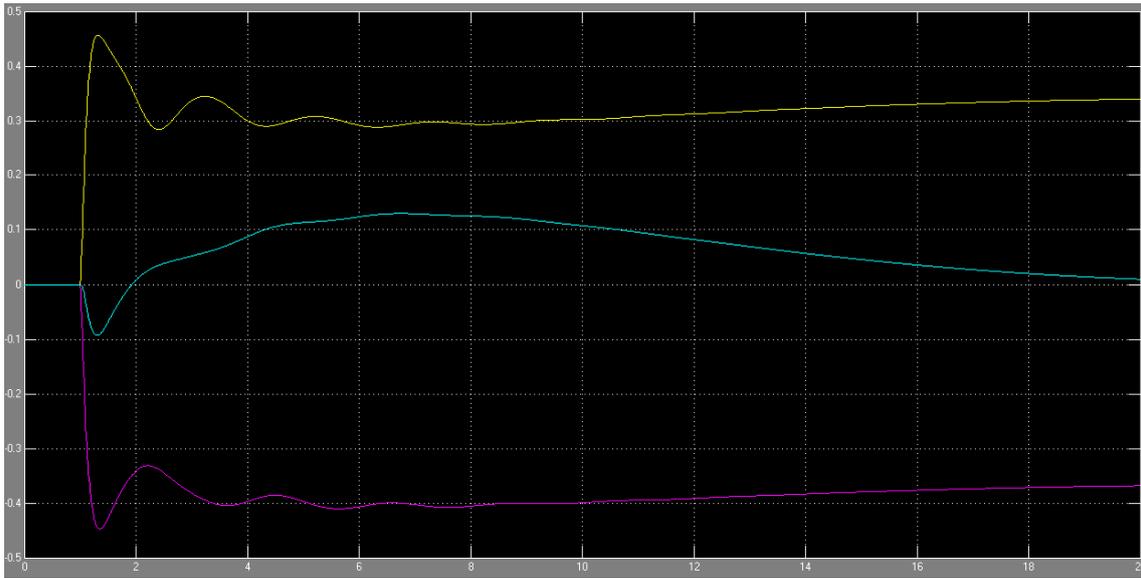


Figura 12.11 vento costante nel sistema esteso e con integrali delle posizioni in Φ, Θ, Ψ

P, Q, R

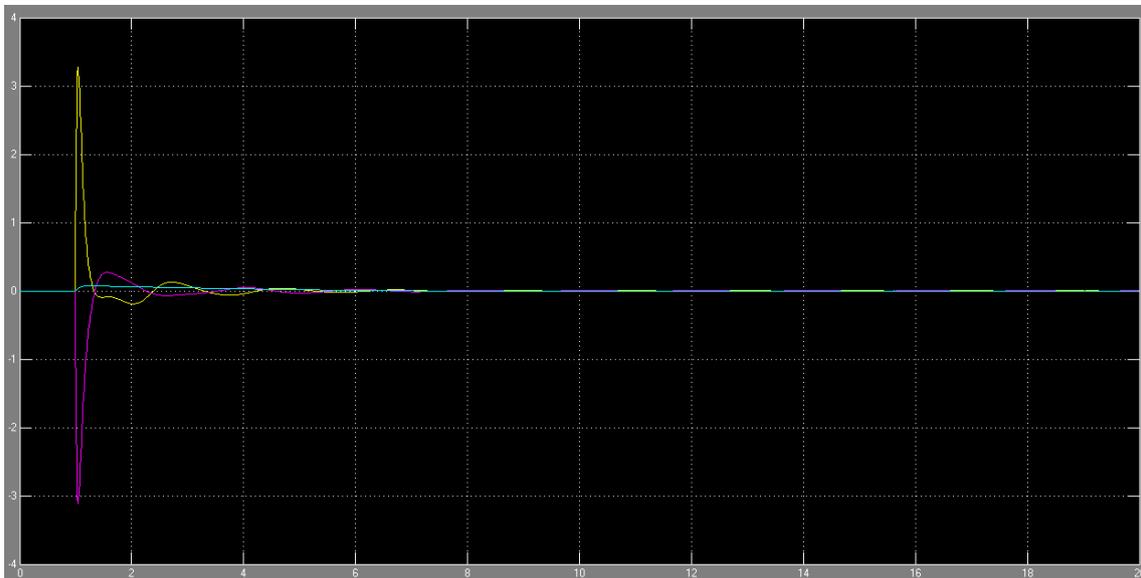


Figura 12.12 vento costante nel sistema esteso e con integrali delle posizioni in P, Q, R

E anche in questo test il software risulta più che valido e corretto, sia per gli spostamenti, sia che per le rotazioni.

12.9 COMPILAZIONE MATRICI R E Q

Anche per questo caso è opportuno scovare i coefficienti con cui compilare le matrici dei costi in modo adeguato e tali da retroazionare il modello nel moto migliore possibile

Valori errati pesano male il modello e restituiscono un feedback non appropriato e desiderato del sistema.

Per esempio come primo approccio si è pensato di valorizzare solo i coefficienti che pesano gli integrali delle posizioni, si era deciso per un primo test:

$$\left\{ \begin{array}{l} q_1, q_2, q_3 = 1 \\ q_4, q_5, q_6 = 1 \\ q_7, q_8, q_9 = 1 \\ q_{10}, q_{11}, q_{12} = 1 \\ q_{13}, q_{14}, q_{15} = 0.01 \\ r_1, r_2, r_3, r_4 = 100 \end{array} \right.$$

Si ottiene, per un disturbo come nel primo caso analizzato:

N,E,D

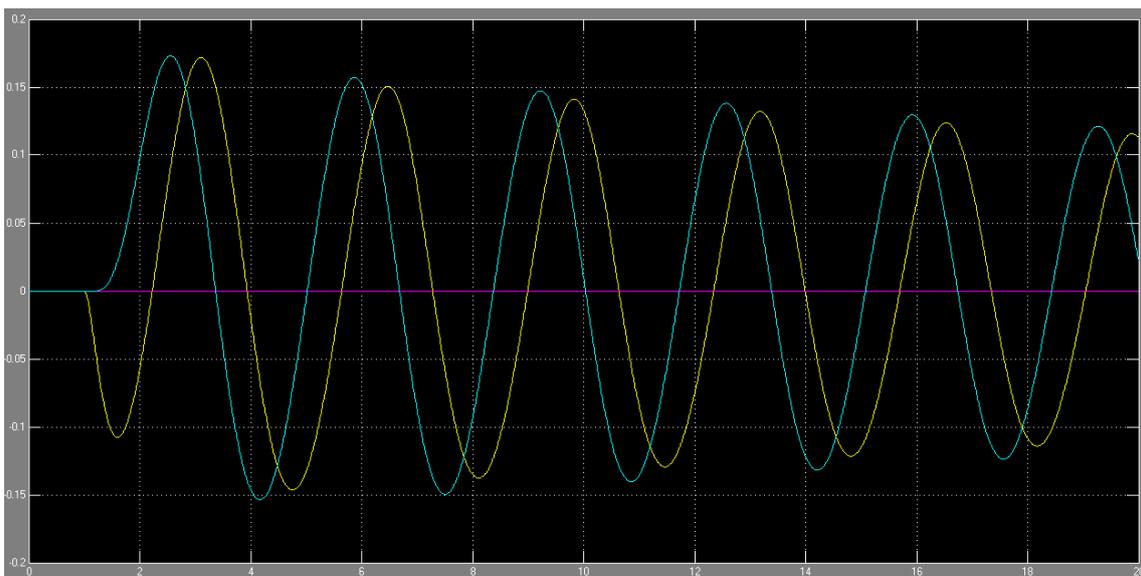


Figura 12.13 vento costante nel sistema esteso e con integrali delle posizioni in N,E,D

U,V,W

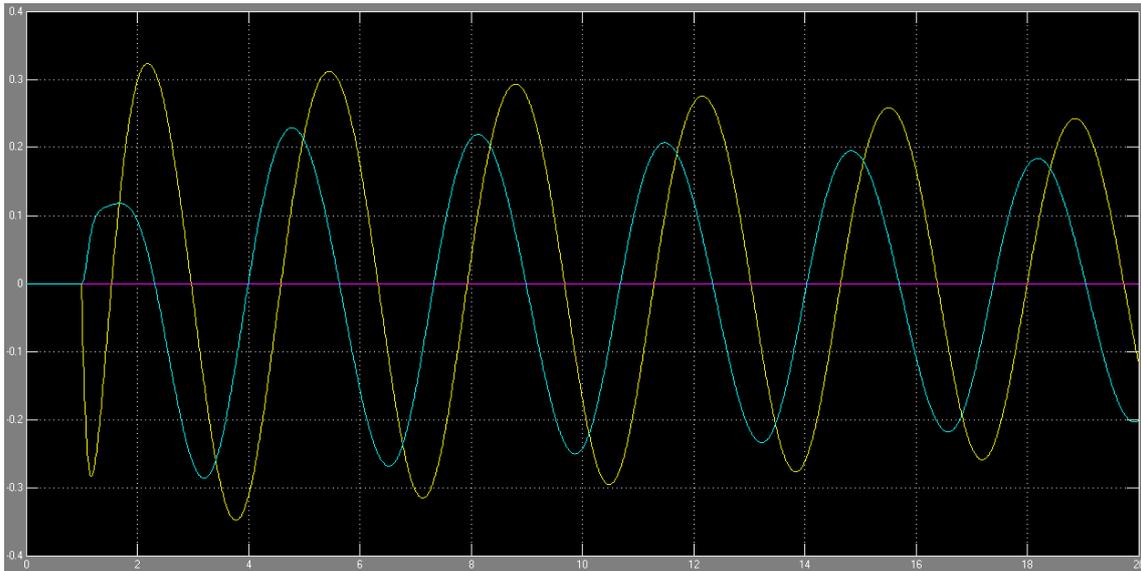


Figura 12.14 vento costante nel sistema esteso e con integrali delle posizioni in U,V,W

Sebbene le oscillazioni siano minime (± 0.15 m) si evince che la retroazione interviene ma smorza le oscillazioni molto lentamente creando un'oscillazione non trascurabile.

Dopo svariati tentativi si trovano, in via empirica, i coefficienti che danno un feedback valido visto nelle prove precedenti.

$$\left\{ \begin{array}{l} q_1, q_2, q_3 = 100 \\ q_4, q_5, q_6 = 100 \\ q_7, q_8, q_9 = 100 \\ q_{10}, q_{11}, q_{12} = 0.001 \\ q_{13}, q_{14}, q_{15} = 0.001 \\ r_1, r_2, r_3, r_4 = 100 \end{array} \right.$$

Penalizzando tutti i valori tranne le posizioni ed i relativi integrali è come valorizzare questi, che a loro volta sono pesati tramite numeri compresi tra zero ed uno.

13 IL MODELLO REALE COMPLETO

Ottenuti due modelli validi che stabilizzano il drone nei casi più importanti in cui ci si può imbattere durante una fase di volo, non rimane che farli convergere fino ad ottenere un unico programma che risolva entrambe le situazioni con un unico regolatore che gestisce il feedback.

Il principi di funzionamento dei due schemi analizzati rimarranno immutati, verranno semplicemente elaborati per farli collaborare insieme ed ottenere una risposta valida e affidabile.

Partiamo a costruire il concetto di base che darà origine al programma.

Per quanto riguarda il modello esteso si ricorda che la retroazione avviene sugli integrali delle posizioni del sistema NED, mentre al contrario, nel modello rettilineo il feedback si applica sulle velocità.

13.1 SCRITTURA DELLO SCRIPT PER LA COMPILAZIONE DELLA MATRICE K

Si evince immediatamente che susseguirà una retroazione assolutamente errata se si utilizza la stessa matrice A per entrambi i casi, questo perché i poli, dovuti agli integrali delle posizioni, portano a convergere a zero l'errore asintotico delle posizioni, che in un sistema in hovering a vento costante è corretto, ma per un sistema a moto uniforme dove la posizione cresce linearmente produce un sistema che andrà anch'esso a stabilizzarsi sulle posizioni iniziali.

Si può pensare allora di agire nel modo seguente, separare nello script di Matlab il caso di hovering da quello di moto uniforme tramite una funzione IF.

Si associa allora ad una variabile (indicata con j) la somma di tutte le variabili iniziali a meno delle posizioni NED e se tale risulta uguale a zero all'ora saremo in presenza di hovering, nel caso contrario invece in caso di moto.

Assunta tale ipotesi come corretta ora si necessita di incorporare il modello di moto uniforme con quello delle matrici estese per ottenere un feedback corretto anche per venti costanti.

Tale sviluppo è al quanto facile, in quanto è sufficiente sostituire nella matrice \bar{A} la matrice con i valori diversi dall'hovering (12x12) alla A_{11} (15x15) quindi previa dell'integrali di posizione.

Riassumendo lo script sarà diviso in due parti, dove nella prima (caso di hovering) la mia matrice A_{11} sarà compilata anche con l'integrale delle posizioni, come già visto nel caso esteso, mentre nella seconda (caso di moto) la matrice A_{11} coinciderà con quella del moto uniforme, quindi esente da quelle variabili di stato.

Si riporta nella pagina che segue il nuovo script che apporta le modifiche menzionate.

```

al=45*pi/180;
M=1.307;
Ix=0.0216316;
Iy=0.0243731;
Iz=0.0397019;
D=0.335;
ix=(Iy-Iz)/Ix;
iy=(Iz-Ix)/Iy;
iz=(Ix-Iy)/Iz;

N=2000:1000:20000;

t=[0.156,0.343,0.601,0.930,1.341,1.851,2.433,3.073,3.77,4.528,5.406,6.425,7.502,8.607,9.905,11.295,12.8,14.259,16.53];
m=[0.162,0.354,0.608,0.913,1.281,1.793,2.262,2.851,3.499,4.21,5.027,5.98,7.001,8.071,9.344,10.688,12.299,13.95,16.398];
Ti=t.*4.45;
Mi=m.*4.45*2.54/100;
T0=M*9.81/4;
no=interp1(Ti,N,T0);
om0=no*pi/30;
Ks=T0/(om0^2);
M0=interp1(N,Mi,no);
Kd=M0/(om0^2);

j=phi+teta+psi+p+q+r+u+v+w;

if j==0;

A11=[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 -9.81 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      9.81 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0];

A12=[0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 1/Ix 0 0;
      0 0 0 0 1/Iy 0;
      0 0 0 0 0 1/Iz;
      1/M 0 0 0 0 0;
      0 1/M 0 0 0 0;
      0 0 1/M 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0];

A22=zeros(6,6);

B1=[0 0 0 0;
     0 0 0 0;
     0 0 0 0;
     2*Ks*om0*D*sin(al)/Ix -2*Ks*om0*D*cos(al)/Ix -2*Ks*om0*D*sin(al)/Ix 2*Ks*om0*D*cos(al)/Ix;
     2*Ks*om0*D*cos(al)/Iy 2*Ks*om0*D*sin(al)/Iy -2*Ks*om0*D*cos(al)/Iy -2*Ks*om0*D*sin(al)/Iy;
     -2*Kd*om0/Iz 2*Kd*om0/Iz -2*Kd*om0/Iz 2*Kd*om0/Iz;
     0 0 0 0;
     0 0 0 0;
     -2*Ks*om0/M -2*Ks*om0/M -2*Ks*om0/M -2*Ks*om0/M;
     0 0 0 0;
     0 0 0 0;
     0 0 0 0;
     0 0 0 0;
     0 0 0 0;
     0 0 0 0];

```

```

r=[1000 0 0 0;
   0 1000 0 0;
   0 0 1000 0;
   0 0 0 1000];

q=[100 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 100 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 100 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 100 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 100 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 100 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 100 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 100 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 100 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0.001 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0.001 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0.001 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0.001 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0.001 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.001];

C1=eye(15);

R=inv(r);

[kk,S11] = LQR(A11,B1,C1'/q*C1,R);

S12=lyap(A11'-S11*B1/R*B1',A22,S11*A12);

K=r*[B1'*S11,B1'*S12];

else

xu=cos(teta)*cos(psi);
xv=sin(phi)*sin(teta)*cos(psi)-cos(teta)*sin(psi);
xw=cos(phi)*sin(teta)*cos(psi)+sin(phi)*sin(psi);
xt=-u*sin(teta)*cos(psi)+v*sin(phi)*cos(teta)*cos(psi)+w*cos(phi)*cos(teta)*cos(psi);
xphi=(v*cos(phi)-w*sin(phi))*sin(teta)*cos(psi)+(v*sin(phi)+w*cos(phi))*sin(psi);
xpsi=-u*cos(teta)*sin(psi)-v*sin(phi)*sin(teta)*sin(psi)-v*cos(phi)*cos(psi)-
w*cos(phi)*sin(teta)*sin(psi)+w*sin(phi)*cos(psi);
yu=cos(teta)*sin(psi);
yv=sin(phi)*sin(teta)*sin(psi)+cos(teta)*cos(psi);
yw=cos(phi)*sin(teta)*sin(psi)-sin(phi)*cos(psi);
yt=-u*sin(teta)*sin(psi)+v*sin(phi)*cos(teta)*sin(psi)+w*cos(phi)*cos(teta)*sin(psi);
yphi=v*cos(phi)*sin(teta)*sin(psi)-v*sin(phi)*cos(psi)-w*sin(phi)*sin(teta)*sin(psi)-
w*cos(phi)*cos(psi);
ypsi=u*cos(teta)*cos(psi)+v*sin(phi)*sin(teta)*cos(psi)-
v*cos(phi)*sin(psi)+w*cos(phi)*sin(teta)*cos(psi)+w*sin(phi)*sin(psi);
zu=-sin(teta);
zv=sin(phi)*cos(teta);
zw=cos(phi)*cos(teta);
zt=-u*cos(teta)-v*sin(phi)*sin(teta)-w*cos(phi)*sin(teta);
zphi=v*cos(phi)*cos(teta)-w*sin(phi)*cos(teta);

A11=[0 0 0 1 sin(phi)*tan(teta) cos(phi)*tan(teta) 0 0 0 0 0 0;
     0 0 0 0 cos(phi) -sin(phi) 0 0 0 0 0 0;
     0 0 0 0 sin(phi)/cos(teta) cos(phi)/cos(teta) 0 0 0 0 0 0;
     0 0 0 0 ix*r ix*q 0 0 0 0 0 0;
     0 0 0 iy*r 0 iy*p 0 0 0 0 0 0;
     0 0 0 iz*q iz*p 0 0 0 0 0 0;
     0 -9.81*cos(teta) 0 0 -w v 0 r -q 0 0 0;
     9.81*cos(teta)*cos(phi) -9.81*sin(teta)*sin(phi) 0 w 0 -u -r 0 p 0 0 0;
     -9.81*cos(teta)*sin(phi) -9.81*sin(teta)*cos(phi) 0 -v u 0 q -p 0 0 0 0;
     xphi xt xpsi 0 0 0 xu xv xw 0 0 0;
     yphi yt ypsi 0 0 0 yu yv yw 0 0 0;
     zphi zt 0 0 0 0 zu zv zw 0 0 0];

```

```

A12=[0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 1/Ix 0 0;
      0 0 0 0 1/Iy 0;
      0 0 0 0 0 1/Iz;
      1/M 0 0 0 0 0;
      0 1/M 0 0 0 0;
      0 0 1/M 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0];

A22=zeros(6,6);

B1=[0 0 0 0;
     0 0 0 0;
     0 0 0 0;
     2*Ks*om0*D*sin(al)/Ix -2*Ks*om0*D*cos(al)/Ix -2*Ks*om0*D*sin(al)/Ix 2*Ks*om0*D*cos(al)/Ix;
     2*Ks*om0*D*cos(al)/Iy 2*Ks*om0*D*sin(al)/Iy -2*Ks*om0*D*cos(al)/Iy -2*Ks*om0*D*sin(al)/Iy;
     -2*Kd*om0/Iz 2*Kd*om0/Iz -2*Kd*om0/Iz 2*Kd*om0/Iz;
     0 0 0 0;
     0 0 0 0;
     -2*Ks*om0/M -2*Ks*om0/M -2*Ks*om0/M -2*Ks*om0/M;
     0 0 0 0;
     0 0 0 0;
     0 0 0 0];

r=[1000 0 0 0;
   0 1000 0 0;
   0 0 1000 0;
   0 0 0 1000];

q=[1000 0 0 0 0 0 0 0 0 0 0 0;
   0 1000 0 0 0 0 0 0 0 0 0 0;
   0 0 1000 0 0 0 0 0 0 0 0 0;
   0 0 0 1000 0 0 0 0 0 0 0 0;
   0 0 0 0 1000 0 0 0 0 0 0 0;
   0 0 0 0 0 1000 0 0 0 0 0 0;
   0 0 0 0 0 0 0.01 0 0 0 0 0;
   0 0 0 0 0 0 0 0.01 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0.01 0 0;
   0 0 0 0 0 0 0 0 0 0 0.001 0;
   0 0 0 0 0 0 0 0 0 0 0 0.001;
   0 0 0 0 0 0 0 0 0 0 0 0.001];

C1=eye(12);

R=inv(r);

[kk,S11] = LQR(A11,B1,C1'/q*C1,R);

S12=lyap(A11'-S11*B1/R*B1',A22,S11*A12);

k=r*[B1'*S11,B1'*S12];

K=[k(:,1:12),zeros(4,3),k(:,13:18)];

end

```

Le matrici **Q** e **R** sono le stesse identiche già studiate nei casi precedenti.

13.2 IMPLEMENTAZIONE IN SIMULINK

Ora necessitiamo solo di creare uno schema a blocchi che possa eseguire il programma appena stilato e lo inizializzi nel modo corretto.

Tale software dovrà essere strutturato con un layout che richiami entrambi i casi, quindi composto da un blocco che fornisca l'integrale delle posizioni (A), da uno che calcoli i thrust necessari per l'hovering e il moto costante (B), due per il calcolo delle forze e delle coppie (C,D), un regolatore che legga lo script della pagina precedente (E) e un sistema che introduca le condizioni iniziali o di tracking (F).

In figura 13.1 viene illustrato la prima bozza dello schema.

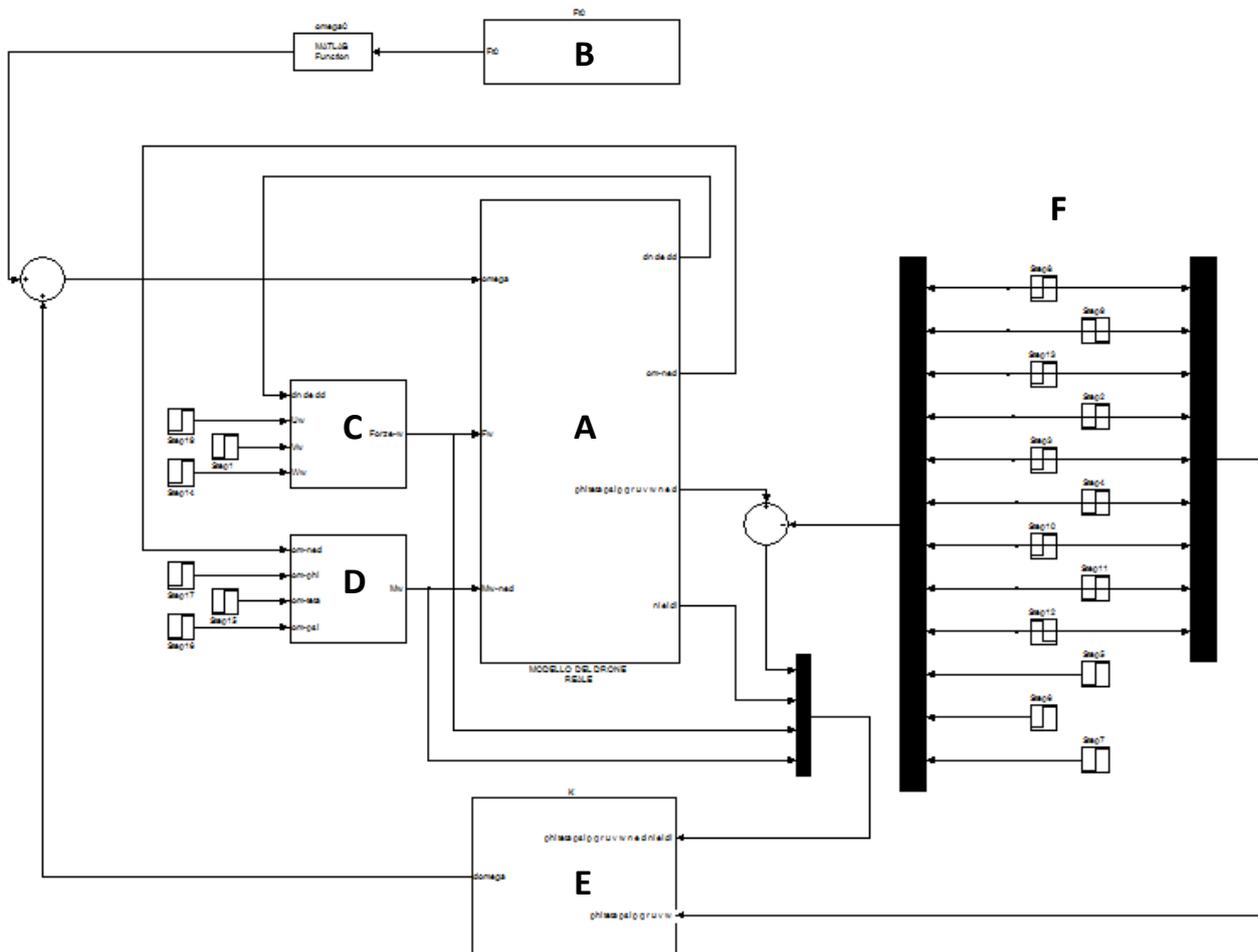


Figura 13.1 layout dello schema a blocchi del modello completo

13.3 TEST E VERIFICA DEL MODELLO COMPLETO

Non resta che provare come reagisce il programma ai disturbi e alle condizioni di moto imposte. La prima verifica sarà incentrata sul caso di vento costante e hovering, si impone quindi un disturbo continuo lungo l'asse N di entità 2 m/s inflitto ad 1 s dall'inizio del test.

$\dot{N}, \dot{E}, \dot{D}$

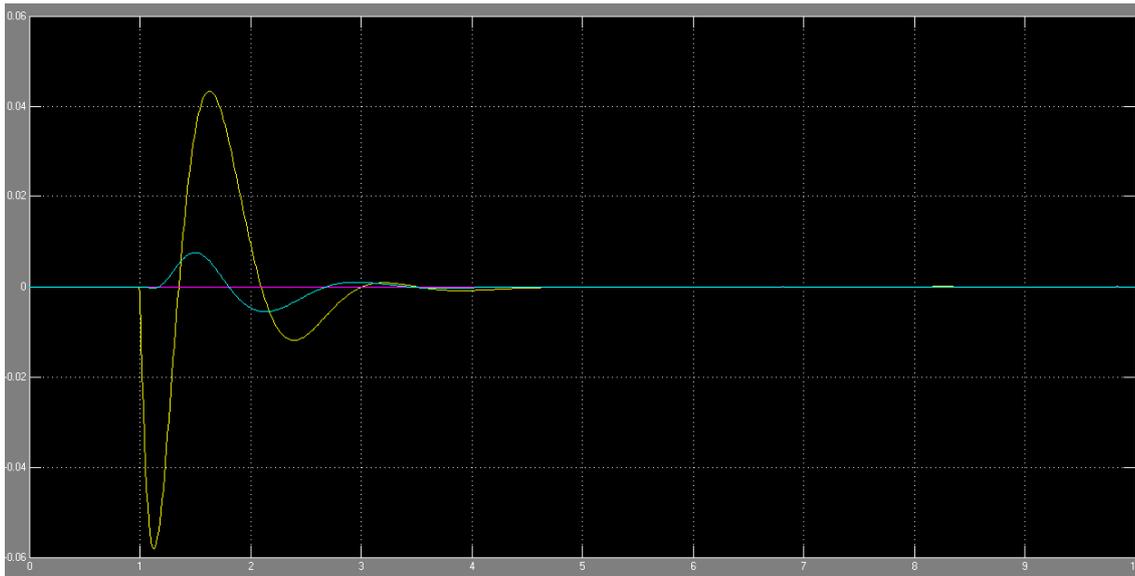


Figura 13.2 hovering e vento costante lungo N in $\dot{N}, \dot{E}, \dot{D}$

N, E, D

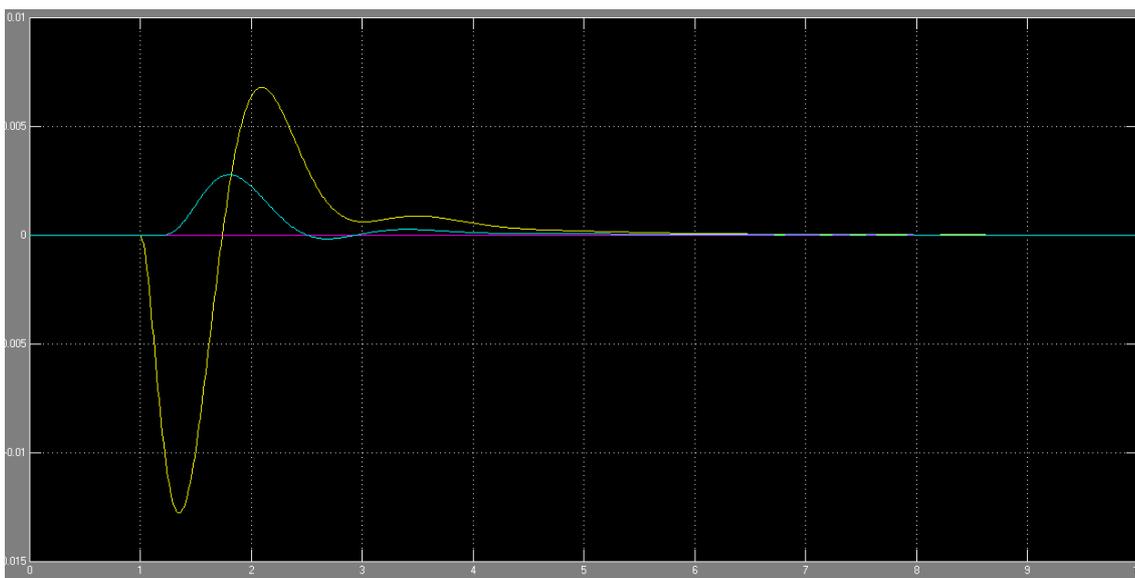


Figura 13.3 hovering e vento costante lungo N in N, E, D

La risposta è assolutamente ottima e accettabile, non solo il sistema è pronto ma anche molto robusto.

Testiamo ora sempre lo stesso modello ad un caso di solo moto costante e rettilineo lungo N senza la presenza di alcun disturbo.

$\dot{N}, \dot{E}, \dot{D}$

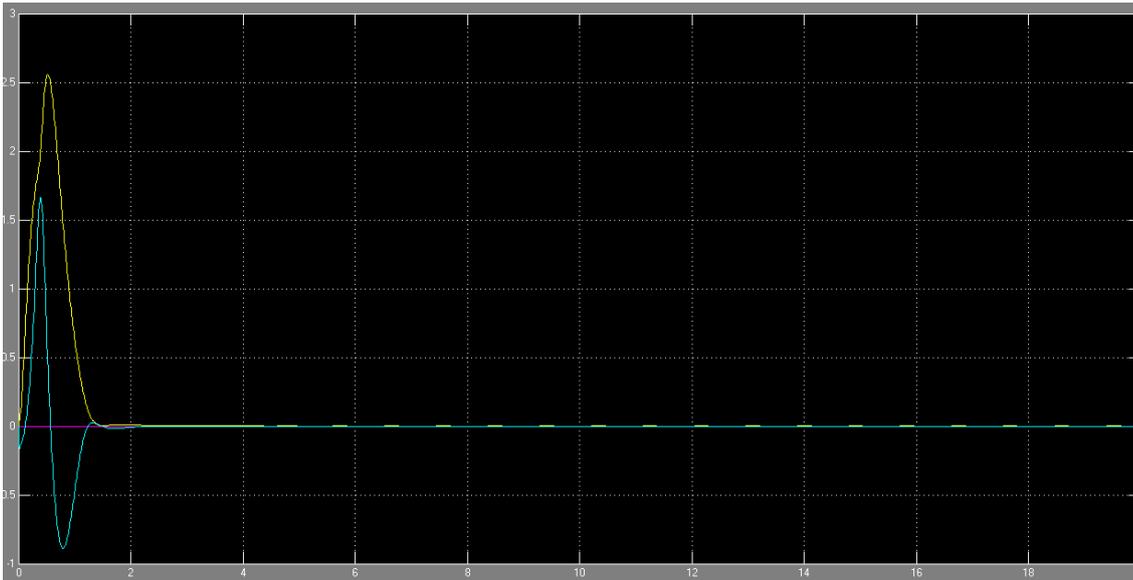


Figura 13.4 moto costante lungo N in $\dot{N}, \dot{E}, \dot{D}$

N, E, D

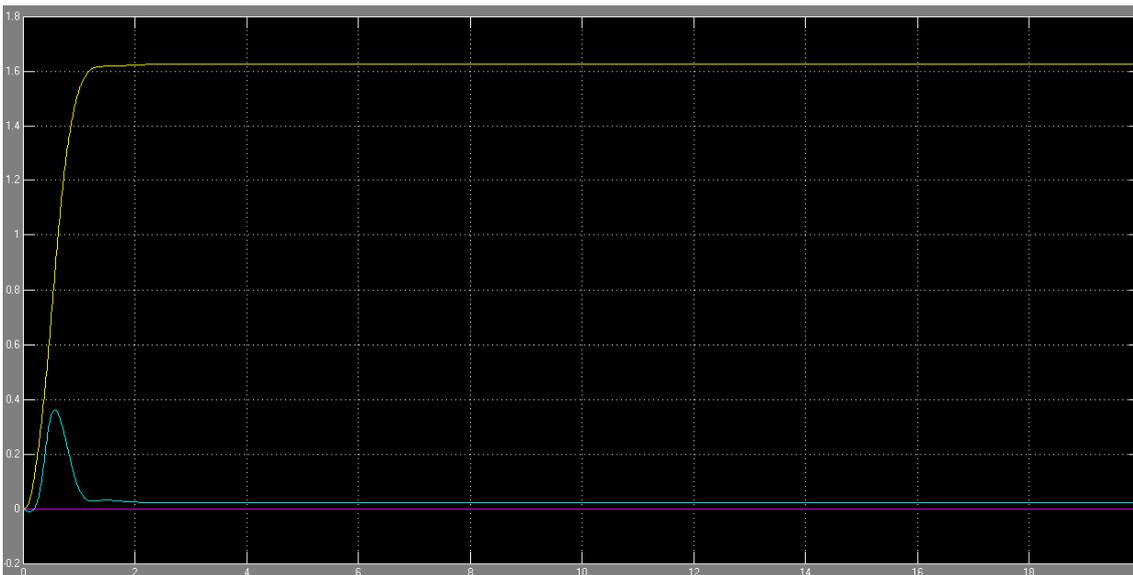


Figura 13.5 moto costante lungo N in N, E, D

E risulta inaccettabile in quanto la velocità anzi che tendere al valore di tracking 1 m/s assume all'infinito valore nullo, mentre la posizione in N converge ad un valore costante anzi che aumentare in modo lineare.

Quindi la situazione risultante descrive un corpo rigido che si sposta di una certa quantità e poi si ferma, non quindi il caso di moto rettilineo.

Il seguente modello non risulta adeguato a soddisfare i casi di studio richiesti.

13.4 SVILUPPO DEL SISTEMA DI TRACKING

È necessario sviluppare una logica che incrementino le posizioni nel tempo.

Questo perché le posizioni una volta stabilite nelle condizioni iniziali non variano nel tempo, quindi essendo la retroazione eseguita su queste, il modello andrà sempre a cercare di tornare sui punti fissati.

L'idea allora è di realizzare un sistema di riferimento che incrementino in modo lineare la sua posizione, in modo tale che il modello virtuale cerchi continuamente di "rincorrerlo" nello spazio.

Si potrebbe attuare quest'idea pensando di sommare progressivamente alla posizione iniziale il tempo che incrementa linearmente, ma tale dispositivo è molto oneroso nel costruirlo rispetto all'uso di un integratore.

Difatti stabilita la condizione iniziale (ovvero la posizione di partenza) il blocco integratore in modo autonomo calcola la sua crescita nel tempo.

In figura 13.6 è mostrato come è strutturato tale sistema di calcolo.

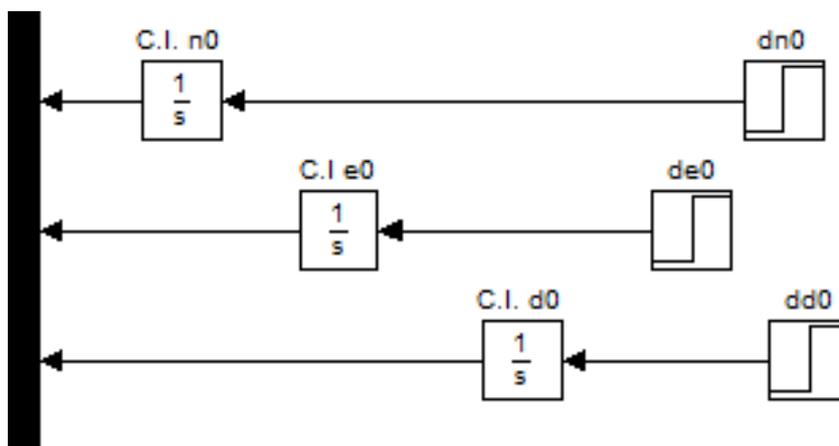


Figura 13.6 sistema a blocchi per il calcolo della posizione istantanea

Nel rettangolo marcato di blu, di figura 13.7, si evince dove è stato introdotto il sistema di tracking appena illustrato nel modello in Simulink.

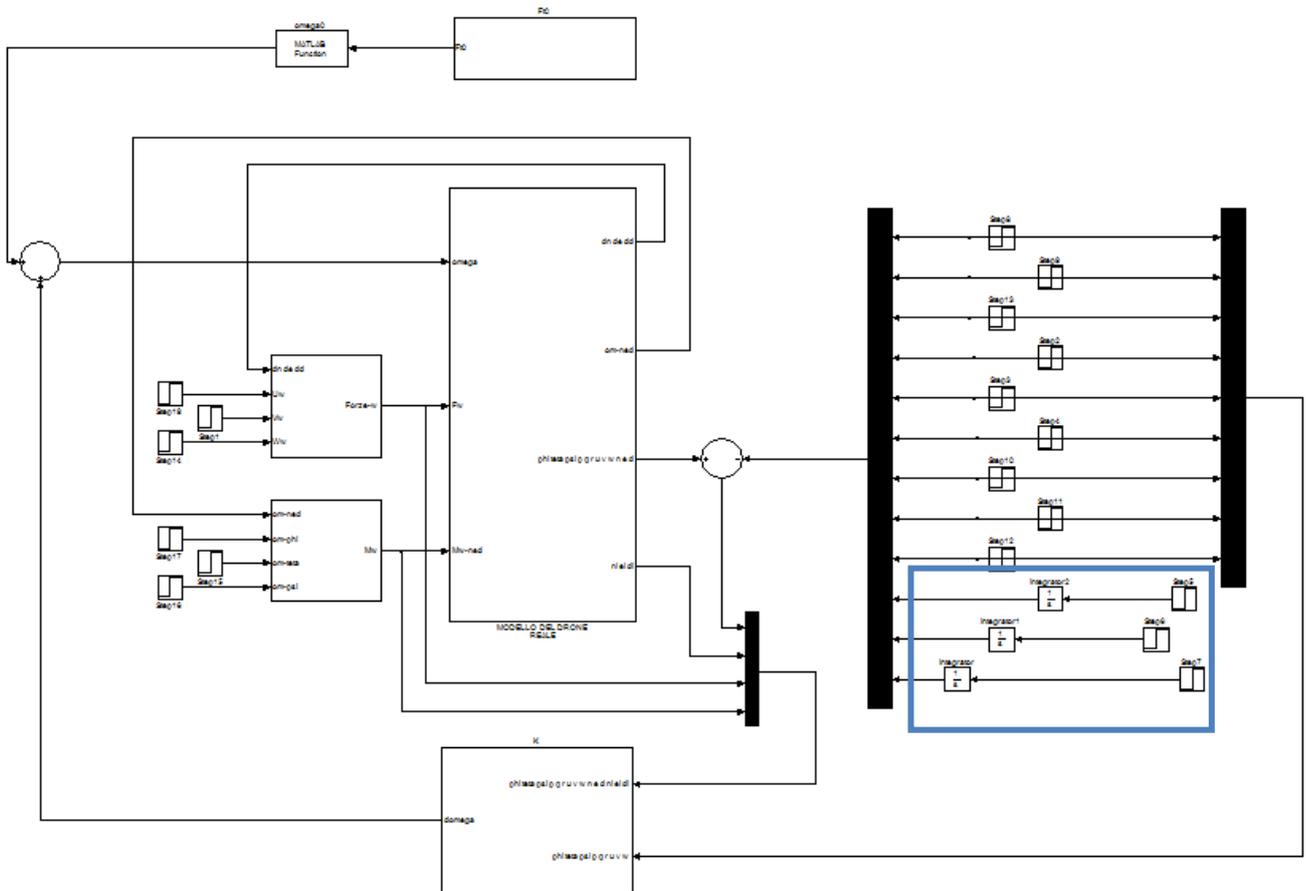


Figura 13.7 layout dello schema a blocchi del modello completo con integrali nelle posizioni

Quindi nel blocco a gradino si inizializzano le condizioni iniziali di velocità assoluta $\dot{N}_0, \dot{E}_0, \dot{D}_0$ poi tali valori vengono elaborati dalla funzione integrale che li trasforma in posizione assoluta istantanee N,E,D e con costante iniziale di integrazione pari a N_0, E_0, D_0 .

In figura 13.8 si mostra un esempio di come la posizione lungo N cresce nel tempo con $\dot{N}_0 = 2 \frac{m}{s}$ e $N_0 = 5 m$.

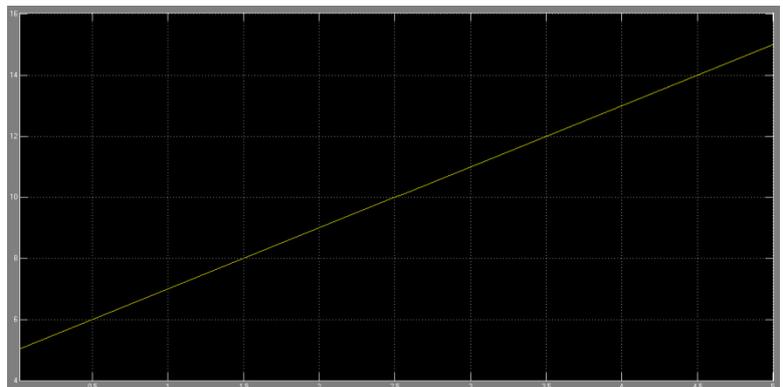


Figura 13.8 esempio di incremento di posizione lungo N

13.5 TEST E VERIFICA DEL MODELLO COMPLETO CON SISTEMA DI TRACKING

Non rimane che reimpostare gli stessi dati di prima e verificare se la risposta del sistema è valida, quindi il caso di solo moto costante e rettilineo lungo N senza la presenza di alcun disturbo.

$\dot{N}, \dot{E}, \dot{D}$

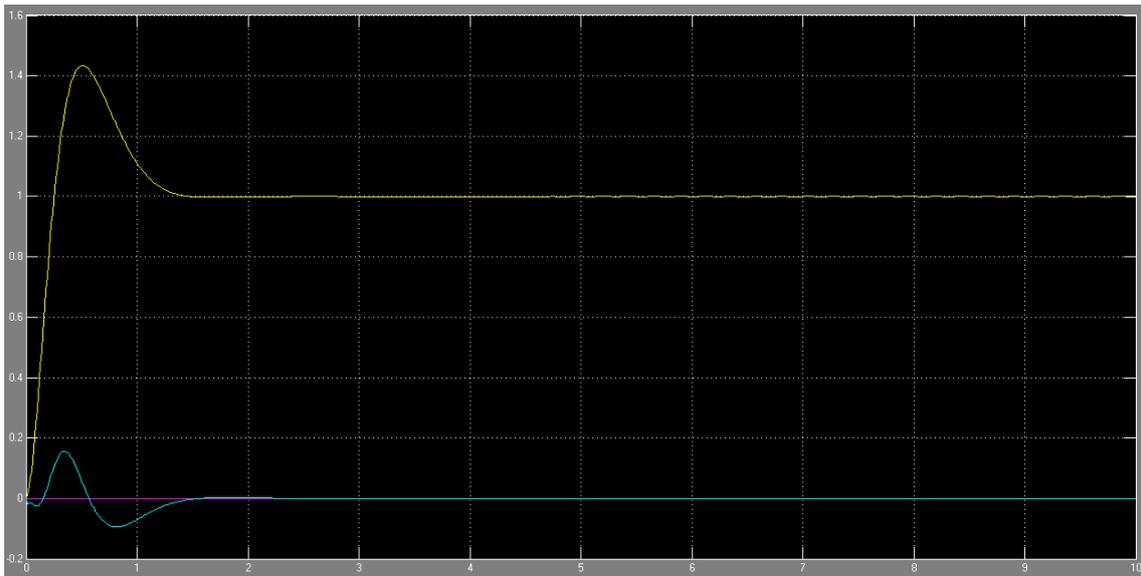


Figura 13.9 moto costante lungo N in $\dot{N}, \dot{E}, \dot{D}$

N, E, D

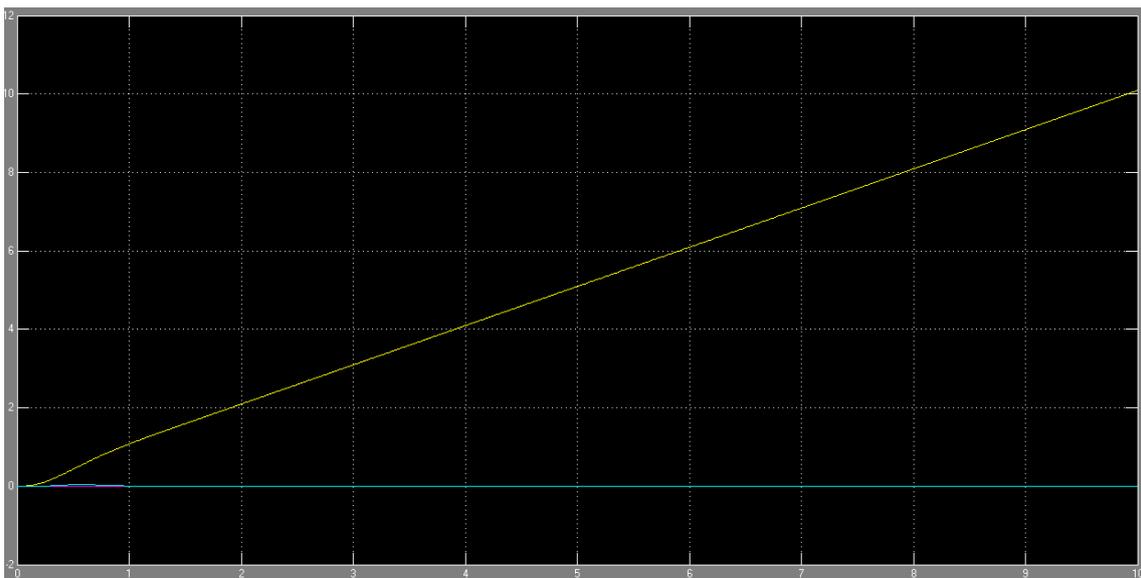


Figura 13.10 moto costante lungo N in N, E, D

La risposta del sistema è più che valida e stabile, quindi complichiamo la situazione aggiungendo un vento costante lungo E pari a 2 m/s ad 1 s dall'istante iniziale e un vortice lungo lo yaw di 0.1 rad/s a 4 s dall'inizio della prova.

$\dot{N}, \dot{E}, \dot{D}$

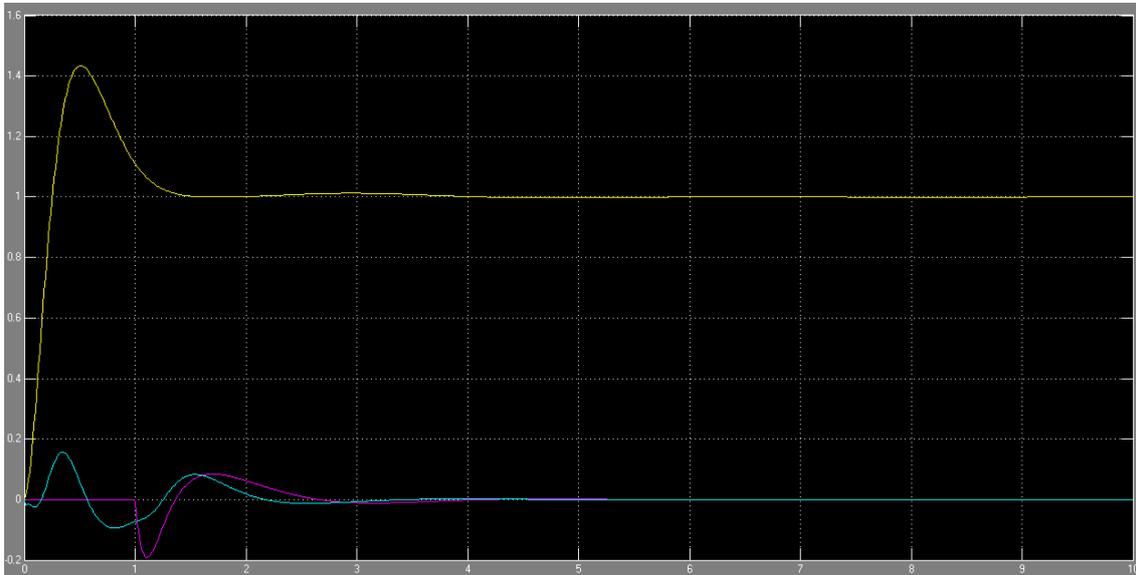


Figura 13.11 moto costante lungo N in $\dot{N}, \dot{E}, \dot{D}$

N, E, D

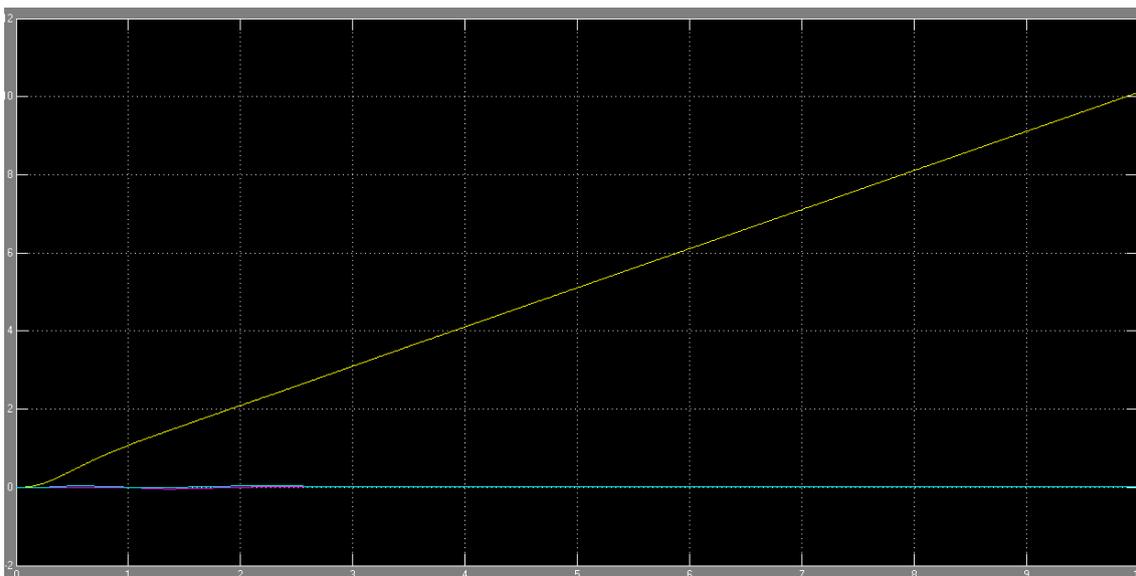


Figura 13.12 moto costante lungo N in N, E, D

Siamo arrivati dunque ad ottenere un sistema stabile e pronto ai disturbi di qualsiasi natura.

Il caso di accelerazione del vento è stato omesso in quanto abbiamo considerato entità molto piccole (1 o 2 m/s), se invece volessimo analizzare venti di entità più consistenti, è necessario considerare una variazione lineare.

Questa semplificazione in realtà è molto valida in quanto nella realtà una raffica di vento non sarebbe mai gestibile dal velivolo, mentre un vento che varia in modo costante può essere elaborato dal suo hardware montato sulla sua piattaforma.

Mettiamo ora sotto test il modello con un vento lungo l'asse N di entità variabile in modo casuale, ovvero usiamo un blocco random, mentre ha un moto costante di 1 m/s lungo il medesimo asse.

In figura 13.13 è plottato in m/s la variazione del disturbo.

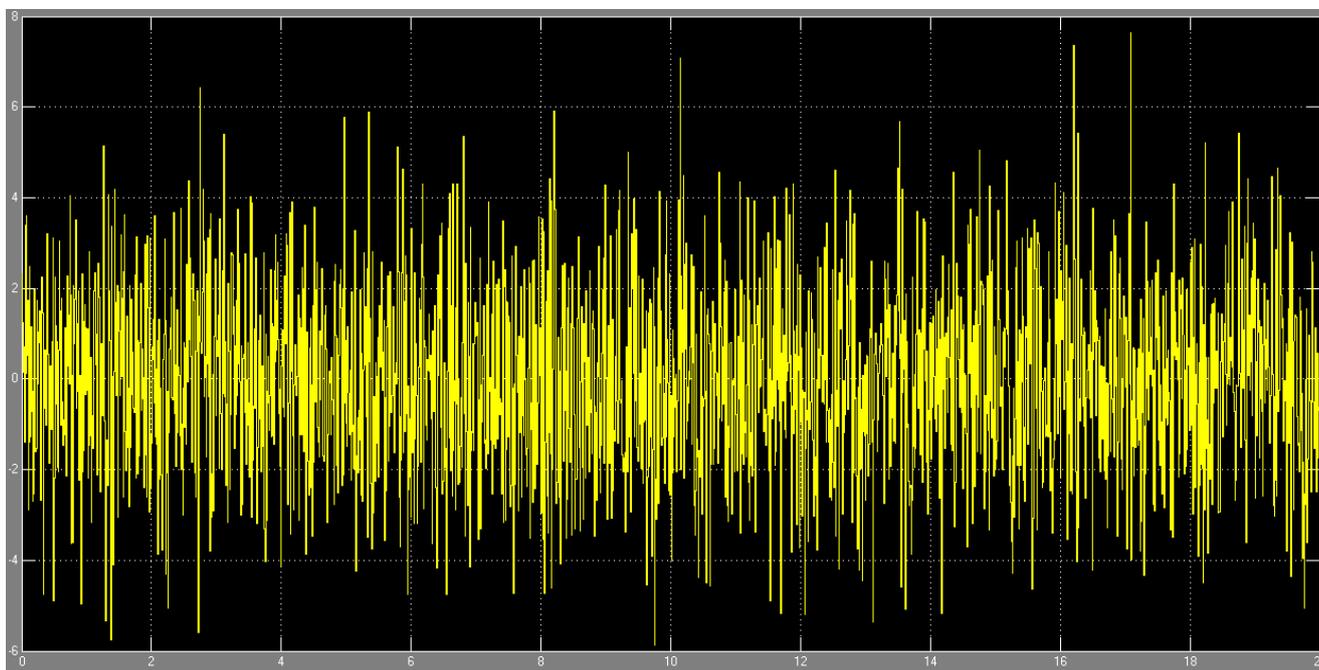


Figura 13.13 variazione del vento lungo l'asse N in m/s

Come si apprende dal diagramma tale disturbi comprende variazioni anche piuttosto consistenti, fino a quasi 8 m/s (circa 30 km/h).

Se quindi il modello risulta stabile possiamo considerarlo affidabile e passare all'analisi successiva.

$\dot{N}, \dot{E}, \dot{D}$

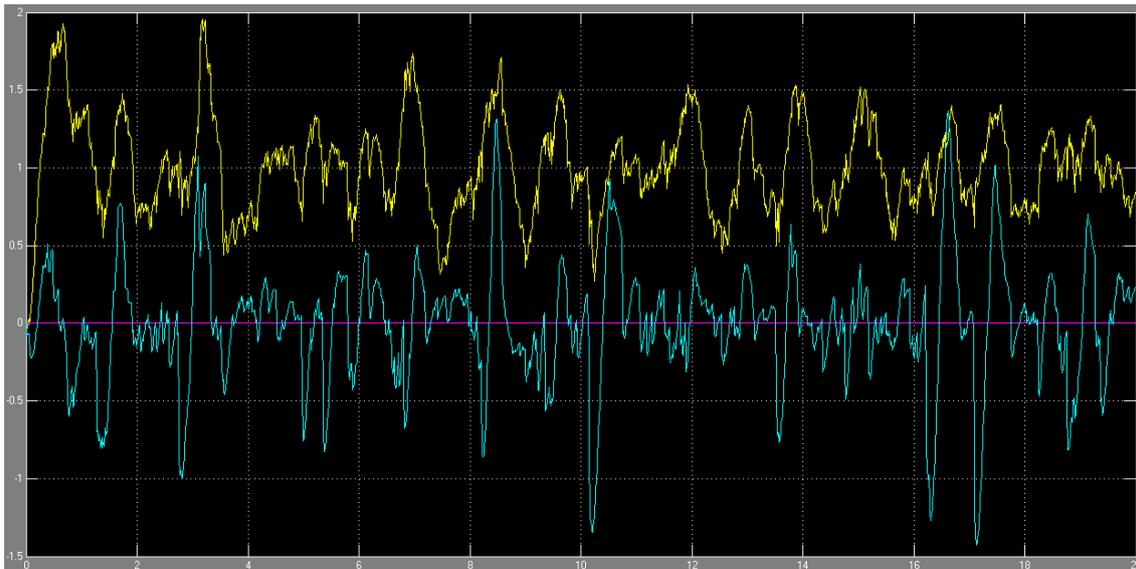


Figura 13.14 disturbo random e moto costante lungo N in $\dot{N}, \dot{E}, \dot{D}$

N, E, D

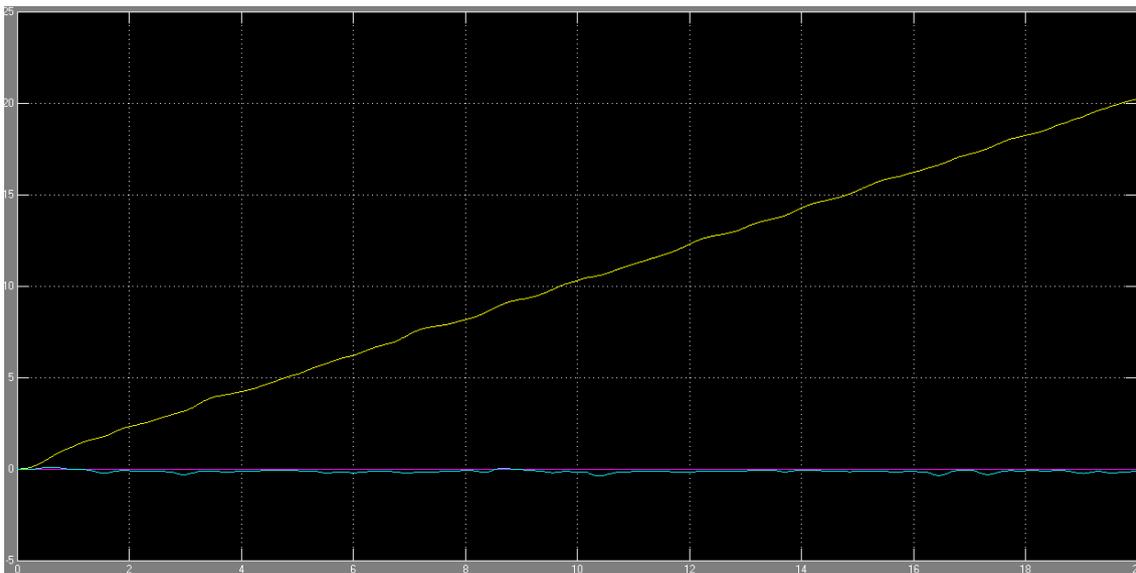


Figura 13.15 disturbo random e moto costante lungo N in N, E, D

Considerando che la variazioni avvengono ogni centesimo di secondo con un valore di discostamento fra ogni valore di 4 m/s possiamo ritenere accettabile il risultato in quanto, nelle posizioni assolute, l'errore è contenuto nella tolleranza desiderata.

Il modello dunque è valido, si deve solo accertare che lo sia anche una volta introdotta la dinamica dei motori e la frequenza di analisi dei diversi dati di input.

14. SVILUPPO DELLE FORZE AERODINAMICHE

Fino ad ora abbiamo trattato velocità di crociera molto modeste tali da permettere di trascurare il fattore aerodinamico.

Per valori alti di moto l'attrito aerodinamico produce effetti sul modello che lo porto ad instabilità.

Un corpo solido in movimento sospeso in un fluido non stazionario presenta due fattori fluidodinamici:

- Uno dovuto al movimento del corpo;
- Uno dovuto al movimento del fluido.

Nel caso corrente, avremo che il primo rientra nelle forze aerodinamiche conosciute, mentre le seconde nel disturbo del vento, quindi ignote.

È possibile quindi scomporre le forze aerodinamiche in queste due entità per produrre un sistema completo e più robusto.

Partiamo dunque riprendendo le equazioni del moto linearizzate:

$$\left\{ \begin{array}{l} \dot{u} = -Q_0 \cdot w - q \cdot W_0 + R_0 \cdot v + r \cdot V_0 - g \cdot \vartheta \cdot \cos\theta_0 + \frac{df_{wx}}{m} \\ \dot{v} = P_0 \cdot w + p \cdot W_0 - R_0 \cdot u - r \cdot U_0 + g \cdot \phi \cdot \cos\Phi_0 \cdot \cos\theta_0 - g \cdot \vartheta \cdot \sin\Phi_0 \cdot \sin\theta_0 + \frac{df_{wy}}{m} \\ \dot{w} = Q_0 \cdot u + q \cdot U_0 - P_0 \cdot v - p \cdot V_0 - g \cdot \phi \cdot \cos\theta_0 \cdot \sin\Phi_0 - g \cdot \vartheta \cdot \sin\theta_0 \cdot \cos\Phi_0 + \frac{-(\sum_i df_{tzi}) + df_{wz}}{m} \end{array} \right.$$

La procedura consiste nel tenere conto anche delle forze aerodinamiche dovute al movimento del drone nell'aria, si ottiene :

$$\left\{ \begin{array}{l} \frac{df_{wx}}{m} = \frac{df_{dx}}{m} + \frac{df_{ax}}{m} \\ \frac{df_{wy}}{m} = \frac{df_{dy}}{m} + \frac{df_{ay}}{m} \\ \frac{df_{wz}}{m} = \frac{df_{dz}}{m} + \frac{df_{az}}{m} \end{array} \right.$$

Dove il pedice “d” indica disturbo e il pedice “a” indica aerodinamico.

Le forze aerodinamiche sono equilibrate dalle corrispettive forze propulsive generate dalle eliche in rotazione, come si evince dall’esempio in figura 14.1

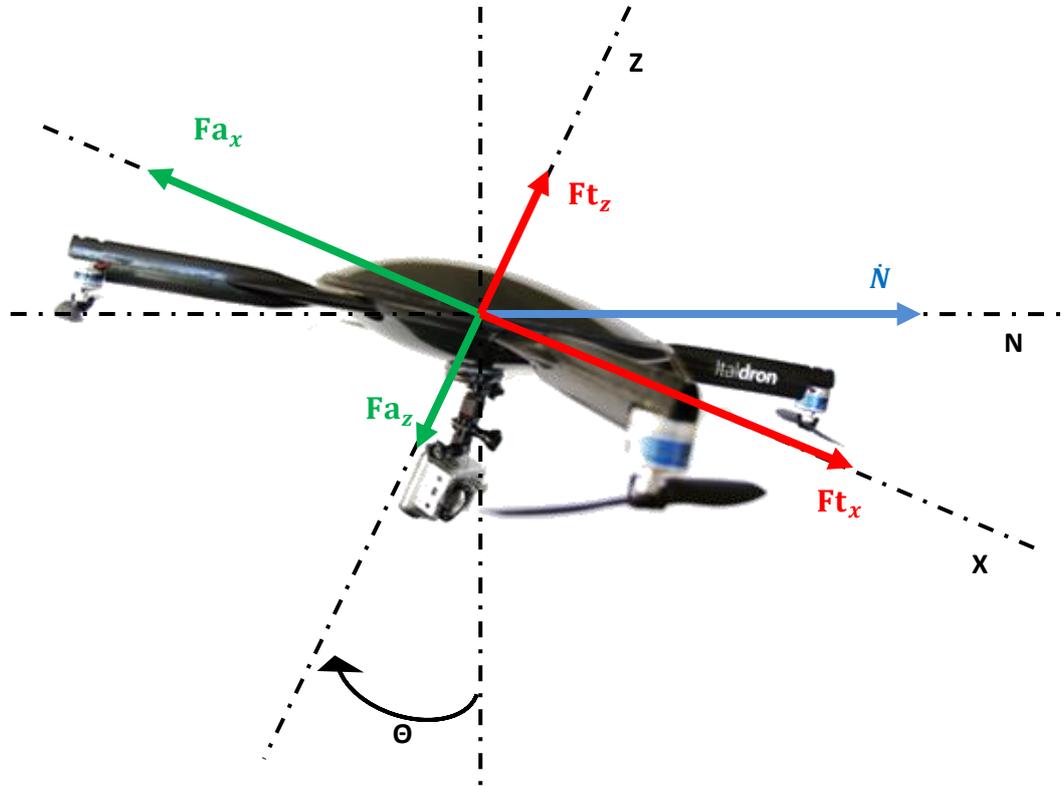


Figura 14.1 esempio di equilibrio fra forze di thrust e quelle aerodinamiche

Quindi in un moto lungo N (indicato dalla freccia blu) ho che il moto si scompone nelle due direzioni X e Z generando le corrispettive forze di attrito aerodinamico (freccie verdi) compensate da quelle di spinta (freccie rosse).

Vediamo come sono calcolate le forze di attrito aerodinamico:

$$\left\{ \begin{array}{l} F_{ax} = \frac{1}{2} \cdot \rho \cdot C_D \cdot S \cdot U_0^2 \\ F_{ay} = \frac{1}{2} \cdot \rho \cdot C_D \cdot S \cdot V_0^2 \\ F_{az} = \frac{1}{2} \cdot \rho \cdot C_D \cdot S \cdot W_0^2 \end{array} \right.$$

E le corrispettive forze di propulsione:

$$\left\{ \begin{array}{l} F_{txi} = K_{sx} \cdot \omega_{xi}^2 \\ F_{tyi} = K_{sy} \cdot \omega_{yi}^2 \\ F_{tzi} = K_{sz} \cdot \omega_{zi}^2 \end{array} \right.$$

Ovvero ogni azione ha la sua componente di velocità angolare e fattore di spinta e il pedice “i” indica il numero del rotore di riferimento perciò i=1,2,3,4.

Derivando per la linearizzazione si ottiene:

$$\left\{ \begin{array}{l} df_{txi} = 2 \cdot K_{sx} \cdot \omega_{0x} \cdot d\omega_{0xi} \\ df_{tyi} = 2 \cdot K_{sy} \cdot \omega_{0y} \cdot d\omega_{0yi} \\ df_{tzi} = 2 \cdot K_{sz} \cdot \omega_{0z} \cdot d\omega_{0zi} \end{array} \right.$$

E quindi reinserendo i valori nelle equazioni linearizzate del moto si ottiene:

$$\left\{ \begin{array}{l} \dot{u} = -Q_0 \cdot w - q \cdot W_0 + R_0 \cdot v + r \cdot V_0 - g \cdot \vartheta \cdot \cos\theta_0 + \frac{-\sum_i df_{ax} + df_{wx}}{m} \\ \dot{v} = P_0 \cdot w + p \cdot W_0 - R_0 \cdot u - r \cdot U_0 + g \cdot \phi \cdot \cos\Phi_0 \cdot \cos\theta_0 - g \cdot \vartheta \cdot \sin\Phi_0 \cdot \sin\theta_0 + \frac{-\sum_i df_{ay} + df_{wy}}{m} \\ \dot{w} = Q_0 \cdot u + q \cdot U_0 - P_0 \cdot v - p \cdot V_0 - g \cdot \phi \cdot \cos\theta_0 \cdot \sin\Phi_0 - g \cdot \vartheta \cdot \sin\theta_0 \cdot \cos\Phi_0 + \frac{-\sum_i df_{tzi} - \sum_i df_{az} + df_{wz}}{m} \end{array} \right.$$

Quindi le forze aerodinamiche producono una spinta ulteriore che deve essere considerata dal sistema tramite le matrici per il calcolo della funzione di Riccati.

Essendo una forza non può rientrare nella matrice **A** ma nella matrice che prende in considerazione le velocità di rotazione dei rotori, ovvero la matrice **G** che quindi inserendo questi nuovi valori diviene:

$$G = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 \cdot K_S \cdot \omega_0 \cdot D \cdot \sin\alpha & -2 \cdot K_S \cdot \omega_0 \cdot D \cdot \cos\alpha & -2 \cdot K_S \cdot \omega_0 \cdot D \cdot \sin\alpha & 2 \cdot K_S \cdot \omega_0 \cdot D \cdot \cos\alpha \\ 2 \cdot K_S \cdot \omega_0 \cdot D \cdot \cos\alpha & 2 \cdot K_S \cdot \omega_0 \cdot D \cdot \sin\alpha & -2 \cdot K_S \cdot \omega_0 \cdot D \cdot \cos\alpha & -2 \cdot K_S \cdot \omega_0 \cdot D \cdot \sin\alpha \\ -2 \cdot K_D \cdot \omega_0 & 2 \cdot K_D \cdot \omega_0 & -2 \cdot K_D \cdot \omega_0 & 2 \cdot K_D \cdot \omega_0 \\ -2 \cdot K_{Sx} \cdot \omega_{0x} & -2 \cdot K_{Sx} \cdot \omega_{0x} & -2 \cdot K_{Sx} \cdot \omega_{0x} & -2 \cdot K_{Sx} \cdot \omega_{0x} \\ -2 \cdot K_{Sy} \cdot \omega_{0y} & -2 \cdot K_{Sy} \cdot \omega_{0y} & -2 \cdot K_{Sy} \cdot \omega_{0y} & -2 \cdot K_{Sy} \cdot \omega_{0y} \\ -2 \cdot (K_S \cdot \omega_0 + K_{Sz} \cdot \omega_{0z}) & -2 \cdot (K_S \cdot \omega_0 + K_{Sz} \cdot \omega_{0z}) & -2 \cdot (K_S \cdot \omega_0 + K_{Sz} \cdot \omega_{0z}) & -2 \cdot (K_S \cdot \omega_0 + K_{Sz} \cdot \omega_{0z}) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Non rimane che implementare il nuovo modello, su cui viene aggiunto una parte fondamentale mostrata qui di seguito in linguaggio Matlab.

```

a=0.5*Cds*ro;
Fx=a*u^2/4;
Fy=a*v^2/4;
Fz=a*w^2/4;

if Fx>Ti(1,19)
    Fx=Ti(1,19);
else if Fx<Ti(1,1)
    Fx=0;
else
    end
end
} Saturazione della componente X

if Fy>Ti(1,19)
    Fy=Ti(1,19);
else if Fy<Ti(1,1)
    Fy=0;
else
    end
end
} Saturazione della componente Y

if Fz>Ti(1,19)
    Fz=Ti(1,19);
else if Fz<Ti(1,1)
    Fz=0;
else
    end
end
} Saturazione della componente Z

if Fx==0
    om0x=0;
    Ksx=0;
else
    nox=interp1(Ti,N,Fx);
    om0x=nox*pi/30;
    Ksx=Fx/(om0x^2);
End
} Calcolo dei coefficienti e velocità angolari
nella componente X

```

```

if Fy==0
    om0y=0;
    Ksy=0;
else
    noy=interp1 (Ti, N, Fy) ;
    om0y=noy*pi/30;
    Ksy=Fy/(om0y^2);
End

if Fz==0
    om0z=0;
    Ksz=0;
else
    noz=interp1 (Ti, N, Fz) ;
    om0z=noz*pi/30;
    Ksz=Fz/(om0z^2);
end

```

Calcolo dei coefficienti e velocità angolari
nella componente Y

Calcolo dei coefficienti e velocità angolari
nella componente Z

Per evitare che la risoluzione del calcolo con dati in input troppo piccoli, o troppo grandi generino valori senza senso (nan) si sono aggiunti questi cicli IF per aggirare tale problema, analizziamo ora nel particolare.

Con le parentesi graffe blu si effettua una saturazione dei valori in input.

Infatti se i valori sono troppo piccoli rispetto a quelli generabili dalle eliche, significa che sono trascurabili ovvero l'attrito aerodinamico non è un'entità consistente da influire sul modello e quindi considerate nulle.

Al contrario se tali dati sono più grandi del limite ammissibile viene saturato al valore massimo generabile dalle pale.

Con le graffe verdi invece si crea un ciclo IF che restituisce valori nulli per i casi di velocità piccole e quindi attrito aerodinamico trascurabile, mentre calcola il coefficiente di spinta e la relativa velocità angolare per i restanti casi.

In questo modo si fa agire il sistema solo nel campo delle velocità dove le forze di attrito aerodinamico iniziano a produrre valori non più trascurabili.

14.1 CREAZIONE DEL MODELLO IN SIMULINK

Per implementare questo modello è necessario dunque scomporre le forze aerodinamiche da quelle di disturbo del vento, la separazione viene evidenziata in figura 14.2.

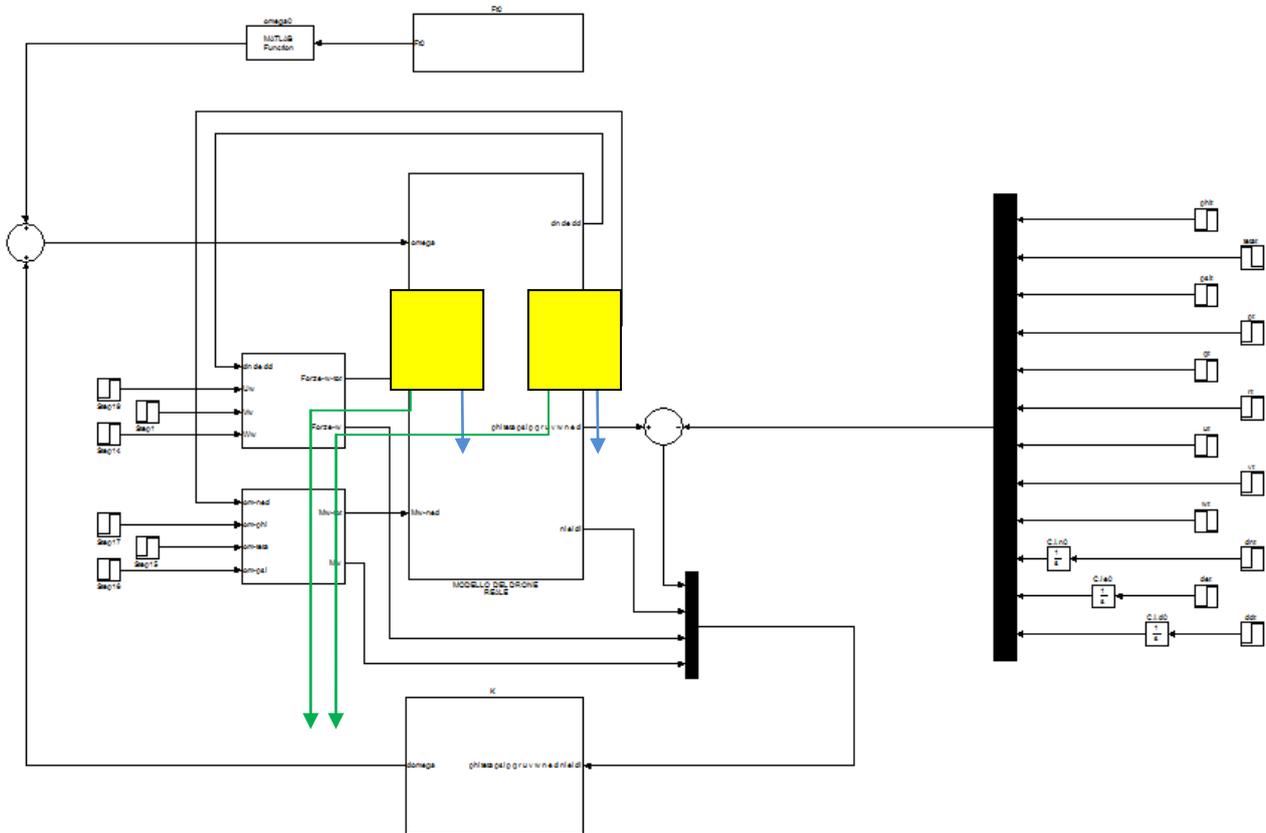


Figura 14.2 scomposizione delle forze aerodinamiche da quelle di disturbo

E come si mostra dalla miniatura con le frecce blu sono indicate le forze aerodinamiche totali che entrano nel modello, mentre in verde si hanno solo quelle di disturbo prelevate per il calcolo della funzione di Riccati nelle matrici estese.

Di seguito si mostrano come sono costituiti ora i blocchi per il calcolo delle forze e dei momenti dovuti al vento (evidenziati in giallo).

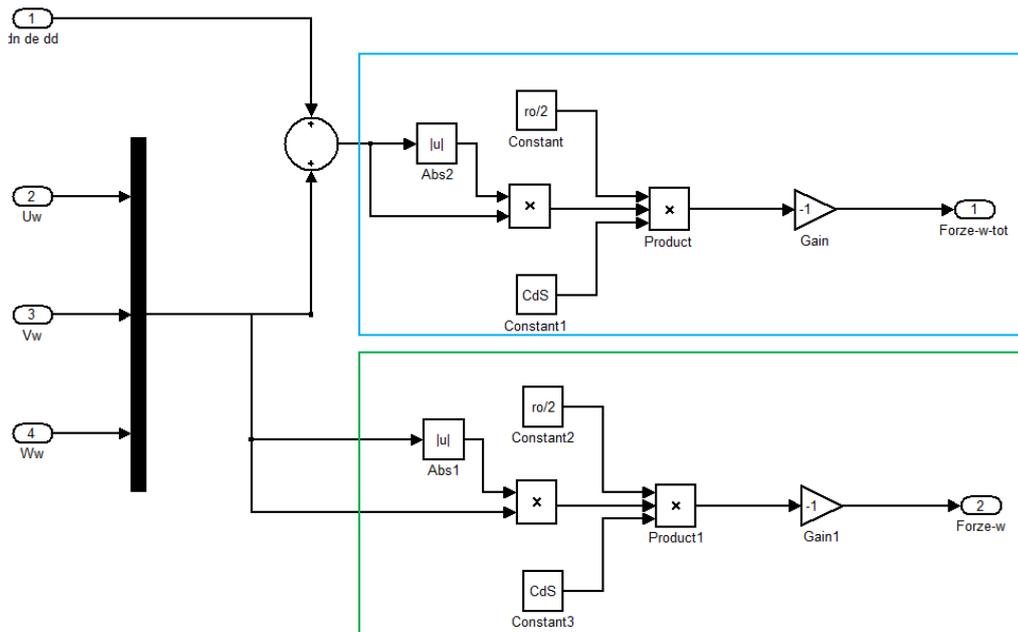


Figura 14.3 blocco per il calcolo delle forze aerodinamiche totali e di disturbo

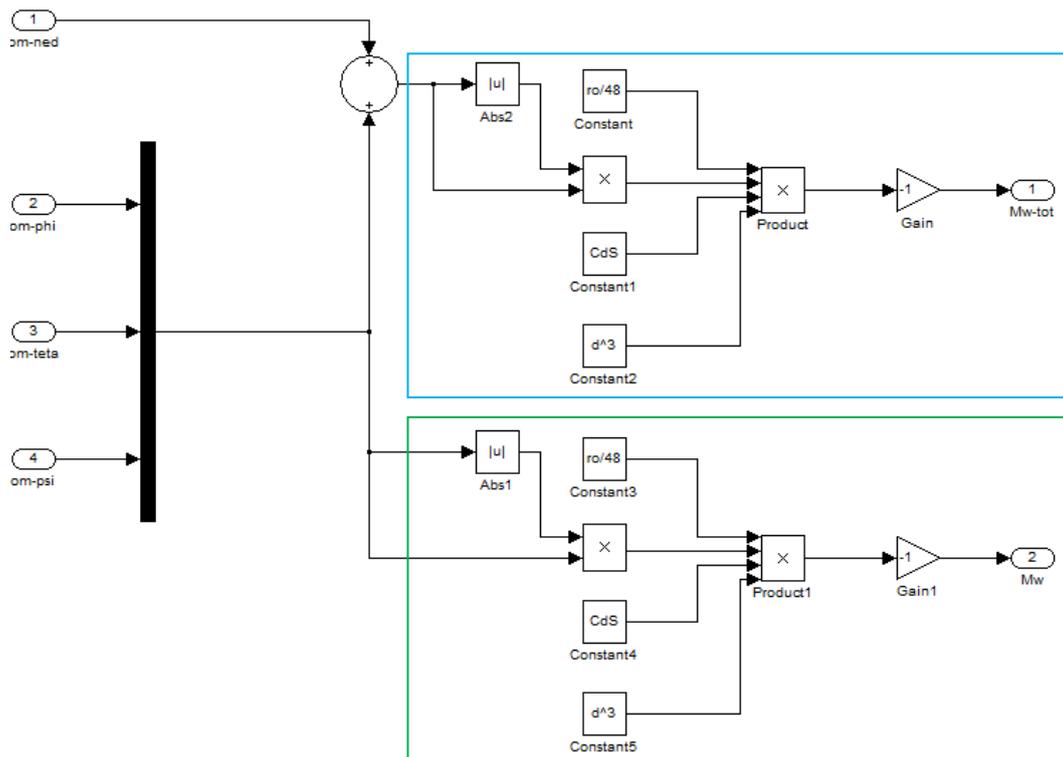


Figura 14.4 blocco per il calcolo delle coppie aerodinamiche totali e di disturbo

E come prima in verde si evidenzia il sistema per il calcolo delle forze e coppie di disturbo e in blu quelle aerodinamiche totali.

Non resta che verificare il modello ai valori limite con velocità ben oltre i casi nominali di utilizzo (1 ÷ 5 m/s) e verificare che si mantenga stabile anche in queste condizioni massime di utilizzo.

14.2 TEST E VERIFICA DEL MODELLO AERODINAMICO

Spingiamo il modello al limite, in condizioni di funzionamento estreme, si impone quindi una velocità di crociera di 15 m/s lungo l'asse N ed un vento di 10 m/s lungo l'asse E costante.

$\dot{N}, \dot{E}, \dot{D}$

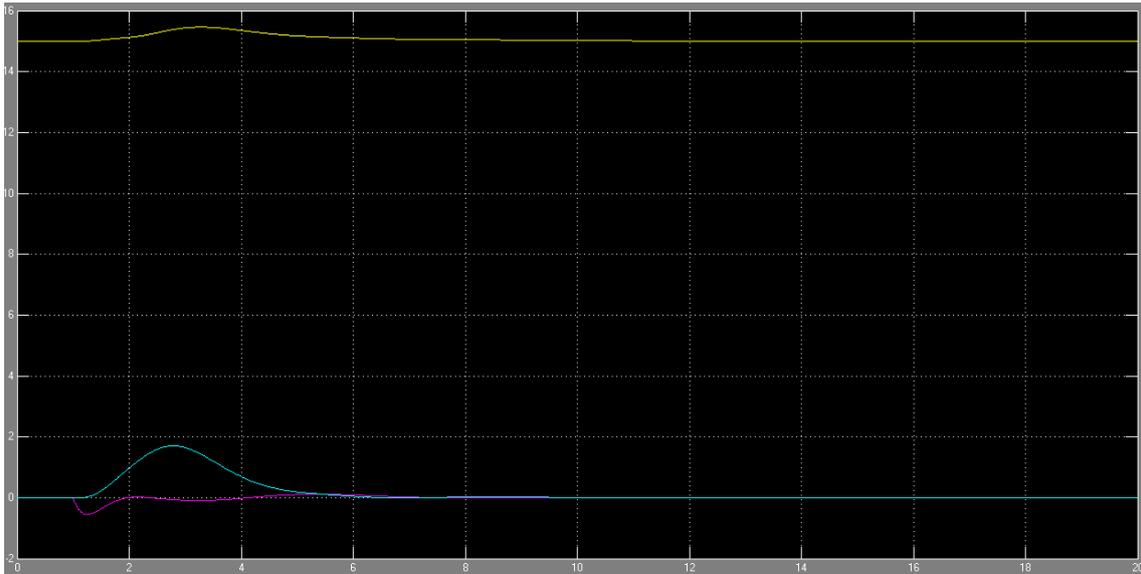


Figura 14.3 moto lungo N e vento costante lungo E in $\dot{N}, \dot{E}, \dot{D}$

N, E, D

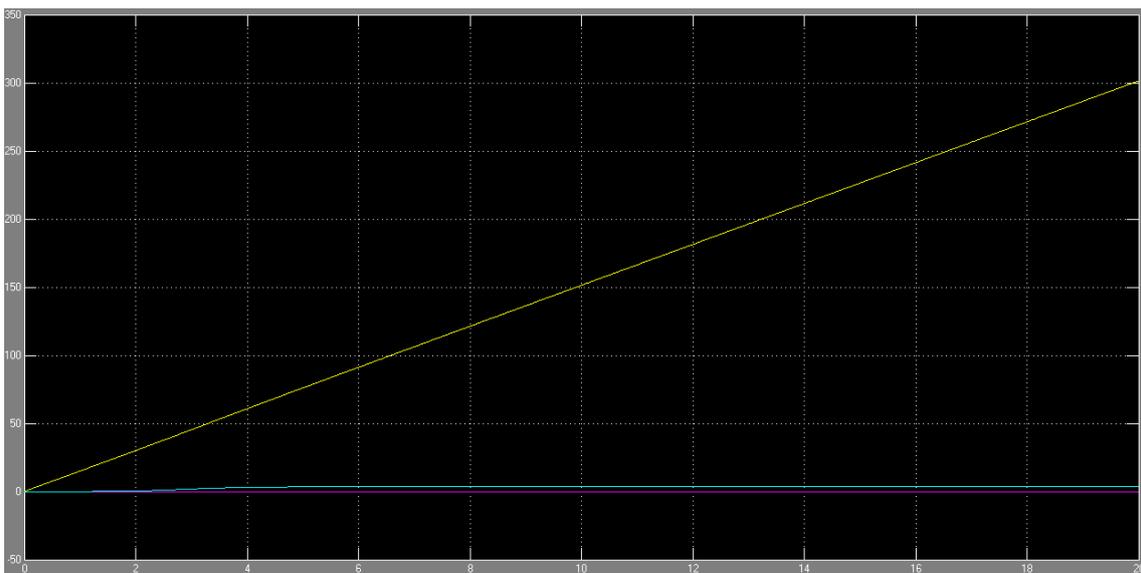


Figura 14.3 moto lungo N e vento costante lungo E in N, E, D

Complicando ancora le condizioni imponiamo una velocità di crociera di 20 m/s lungo N e un vento costante di 5 m/s lungo E.

$\dot{N}, \dot{E}, \dot{D}$

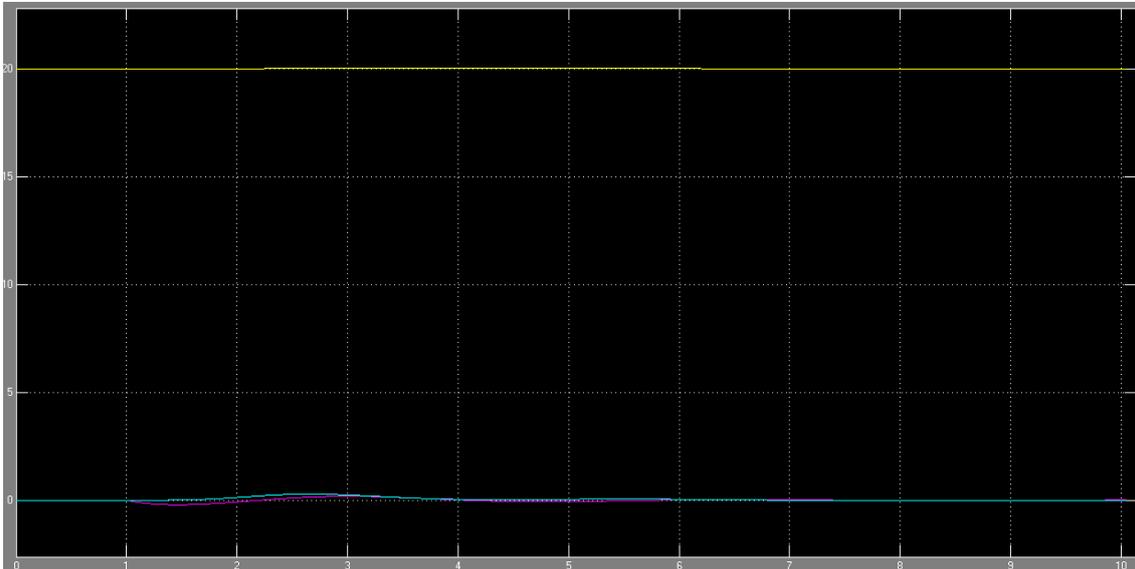


Figura 14.4 moto lungo N e vento costante lungo E in $\dot{N}, \dot{E}, \dot{D}$

N, E, D

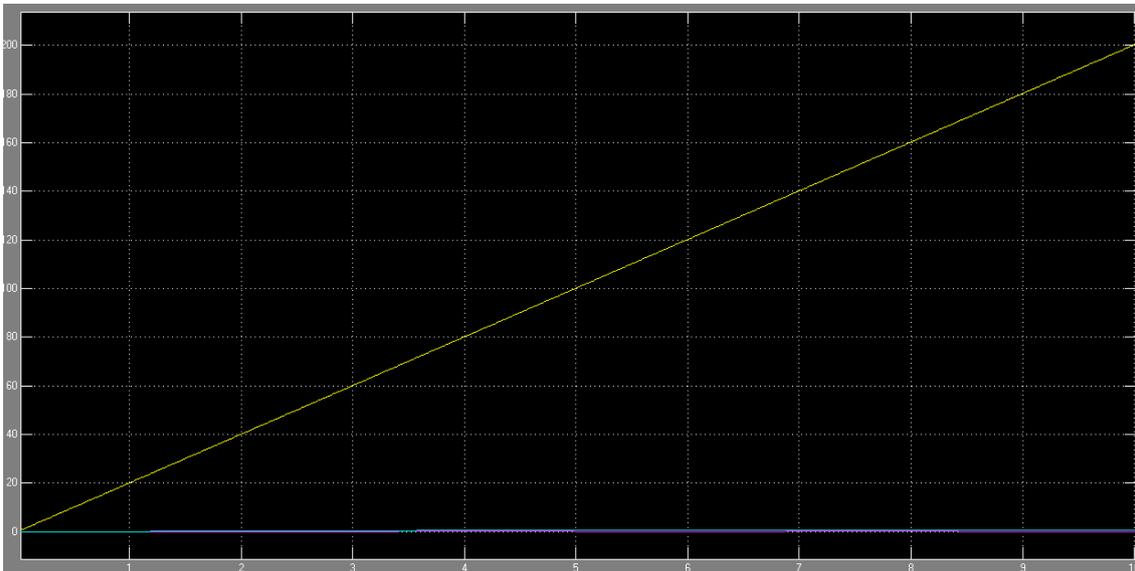


Figura 14.5 moto lungo N e vento costante lungo E in N, E, D

Come si evince dai grafici il modello è valido e funzionante, quindi in grado di comportarsi correttamente anche ad alte velocità di crociera.

È anche vero che ci sono errori di quota anche di 2 m nel primo test, ma bisogna considerare che ci sono condizioni di funzionamento tali da non permettere ai rotori talvolta di offrire velocità angolari superiori a quelle di esercizio limite e quindi si creano errori di posizione dovuta alla saturazione del sistema.

Nel complesso il sistema si può ad ogni modo ritenere stabile, pronto e robusto.

15. DISCRETIZZAZIONE DEL MODELLO REALE

Confrontiamo un sistema a tempo continuo da uno a tempo discreto:

Tempo continuo (trasformata di Laplace/analisi di frequenza)

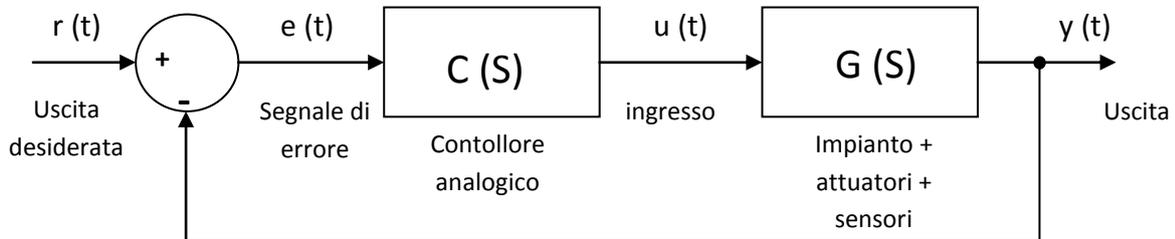


Figura 14.1 modello continuo

t = variabile continua (tempo).

Il processo da controllare è per lo più delle volte a tempo continuo:

- Descritto da un modello alle equazioni differenziali (spesso lineari);
- I segnali degli attuatori e le grandezze in uscita variano con continuità nel tempo.

Tempo discreto (trasformata zeta)

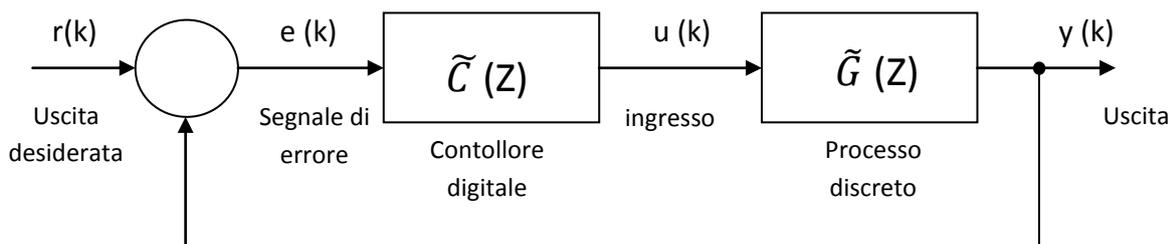


Figura 14.2 modello discreto

k = variabile intera (numero di intervalli di campionamento).

Il controllore che viene implementato è di tempo discreto:

- Più economico da implementare e riprogrammare;
- Più versatile.

Nella pratica quindi si crea un sistema continuo in retroazione con controllore discreto come mostrato in figura 14.3.

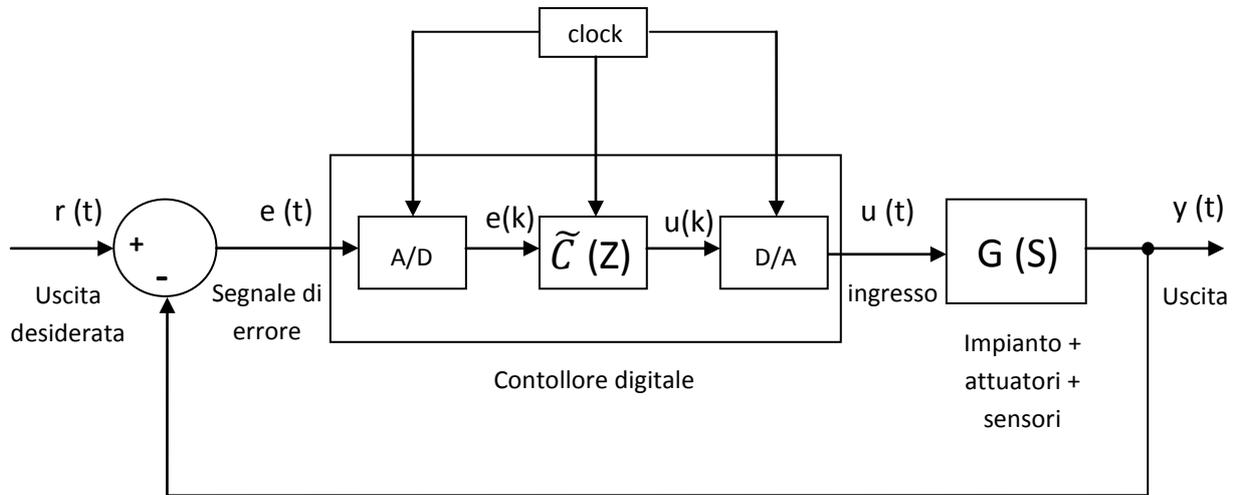


Figura 14.3 modello in retroazione continuo con regolatore discreto

E dunque avremo un sistema che lavora in continuo mentre il regolatore preleva tale segnale, lo discretizza, elabora il risultato e lo reimmette nel sistema ritrasformato in continuo.

I blocchi che campionano il segnale li si possono interpretare fisicamente come interruttori che si aprono, per il tempo stabilito nel clock, trattenendo dunque il valore per tale periodo costante (Figura 14.4).

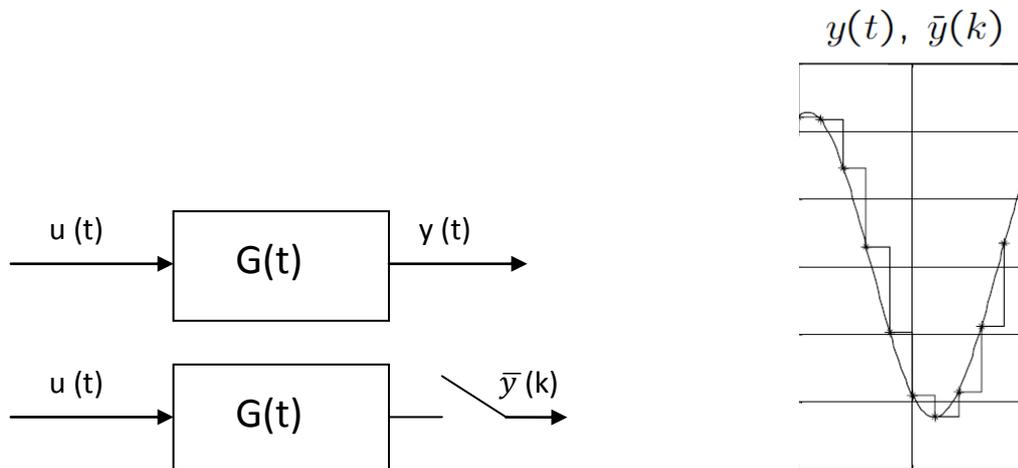


Figura 14.4 discretizzazione di un segnale continuo tramite interruttore

15.1 DISPOSITIVI PER L'INTERFACCIA DIGITALE E ANALOGICA

I segnali vengono convertiti da i dispositivi di interfaccia, e sono di due tipi:

- A/D, ossia convertitore analogico/digitale.

Effettua una campionamento di periodo T del segnale in ingresso $x(t)$ restituendo poi all'uscita una sequenza di valori $x(kT)$ quantizzati con k numero intero reale.



- D/A, ossia convertitore digitale/analogico.

È il dispositivo che prende in ingresso un segnale campionato e discretizzato e lo ricostruisce nel tempo continuo e quindi analogico realizzando una funzione contraria a quella descritta dal sistema precedente.



Il blocco indicato con "hold" non è altro che il ricostruttore di ordine zero, che è quello più utilizzato nei controlli.

Una volta quindi costruito l'intero sistema, questo non funzionerà più in trasformata S , ma secondo la sua analogia in tempo discreto, ovvero la trasformata Z .

È da specificare che il segnale uscente dal blocco hold è ancora a gradini, ma è in forma analogica, quindi nn più discretizzato, come si evince dalla figura14.5.

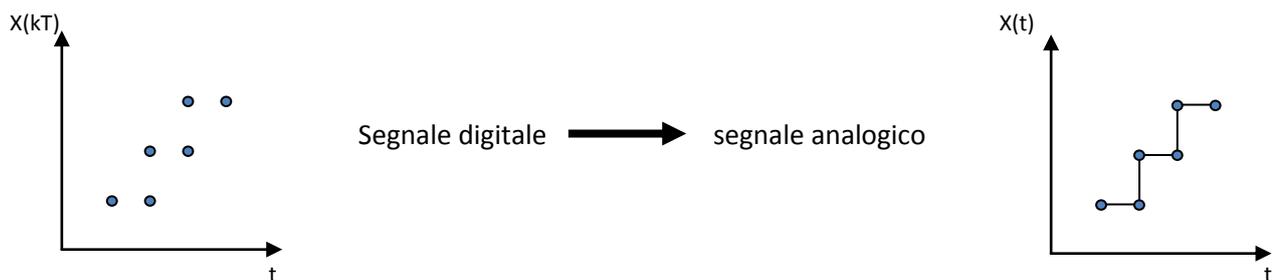


Figura 14.5 ricostruzione in analogico di un segnale digitale a gradini

15.2 VANTAGGI E SVANTAGGI DI UN CONTROLLO IN DIGITALE

Tra i vantaggi si elencano:

- Maggiore capacità e precisione di elaborazione: in quanto consente l'utilizzo di algoritmi più sofisticati;
- Maggiore flessibilità: è sufficiente modificare il programma mantenendo la stessa piattaforma hardware;
- Maggiore affidabilità e ripetibilità: l'elaborazione di un segnale digitale è sempre identica, mentre quella analogica è soggetta a variazioni ambientali che affligge i componenti elettrici;
- Maggiore sensibilità e trasmissibilità dei segnali: l'elaborazione digitali non presenta soglie o saturazioni e la trasmissione a distanza è priva di errori.

Tra gli svantaggi troviamo:

- Progettazione più difficile ed articolata: si richiede una competenza sia sui sistemi dinamici ad evoluzione discreta che sugli elaborati digitali;
- Stabilità più precaria: la discontinuità dei segnali implica ritardi negli interventi del regolatore e la scelta del periodo di campionamento implica un compromesso fra la qualità della regolazione e la quantità di impegno computazionale dell'elaboratore;
- Possibilità di arresti dovuti ai disturbi: se la programmazione non è accurata da prevedere tutte le situazioni anomale generate dai disturbi il regolatore può interrompere improvvisamente la regolazione;
- Necessità di utilizzare energia elettrica: i sistemi digitali sono sempre elettronici.

Nel nostro caso in particolare vedremo che il punto critico sarà dovuto alla stabilità del sistema che per valori troppo alti del periodo di campionamento T porterà la risoluzione a valori divergenti dalla soluzione esatta.

15.3 IMPLEMENTAZIONE IN MATLAB

Non resta che sviluppare i postulati fino ad ora esposti, per la discretizzazione del modello che abbiamo sviluppato e completato nel caso analogico.

Seguendo allora le direttive elencate nella teoria sopra riportata si ottiene lo schema di figura 15.1.

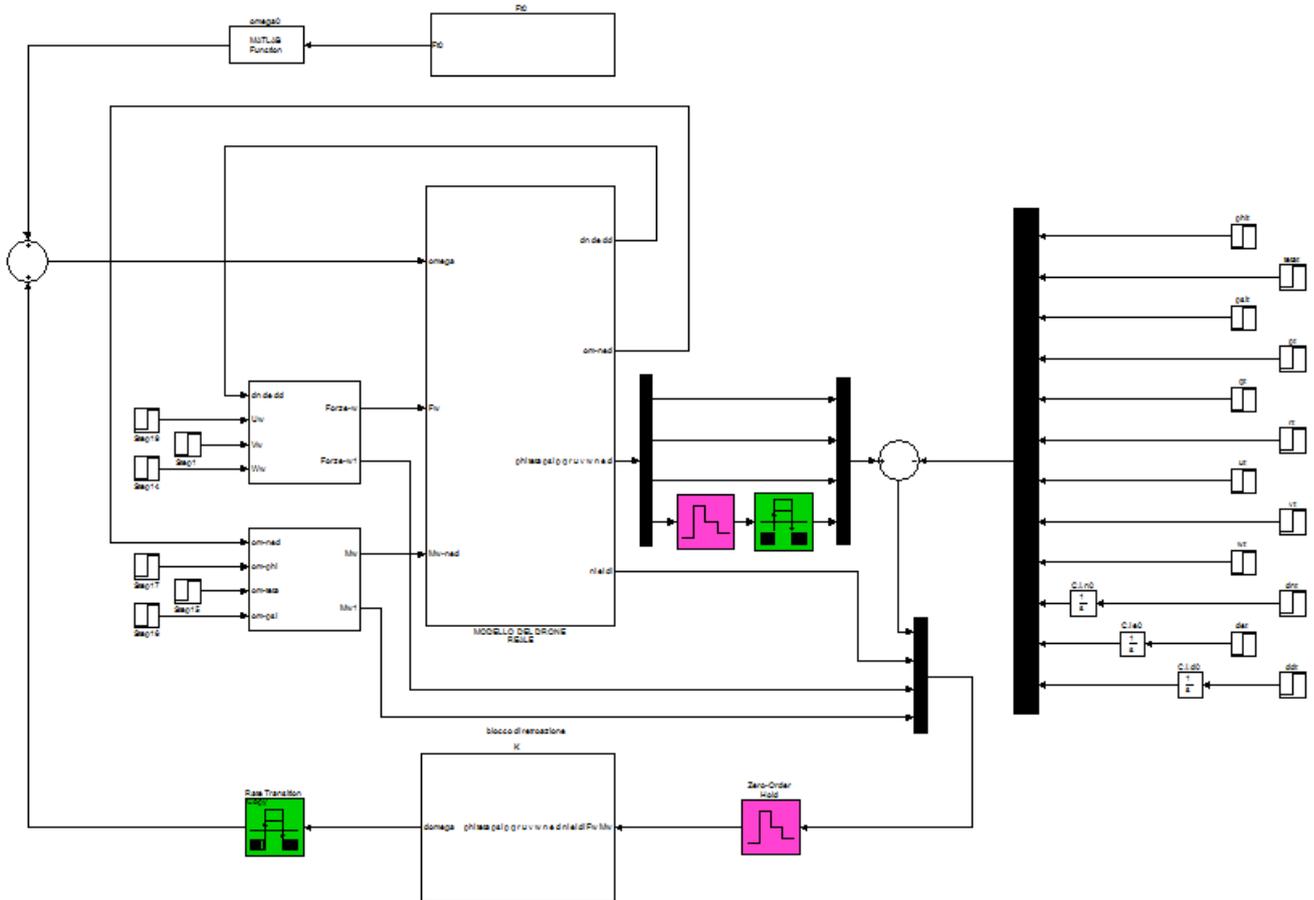


Figura 15.1 modello discretizzato

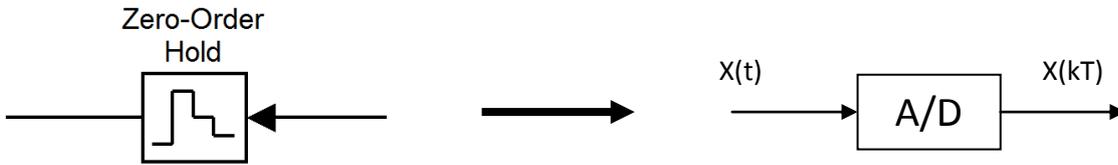
E come si è evidenziato in figura, con i colori magenta e verde, rispetto al modello analogico, si sono inseriti nuove tipologie di blocchi, vediamo le caratteristiche di questi nuovi elementi.

Prima di tutto è facile cogliere l’analogia con lo schema generale riportato all’inizio del capito 15.

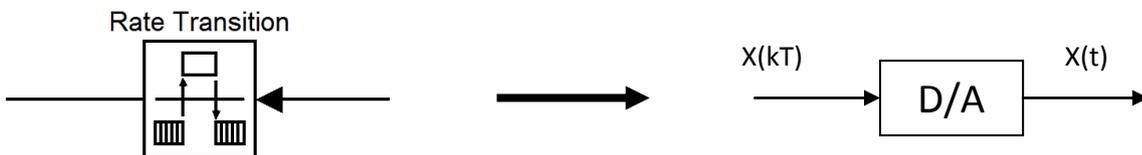
Infatti abbiamo che il regolatore è frapposto fra due nuovi elementi che altro non sono che i due sistemi di interfaccia analogica e digitale.

Per quanto riguarda i blocchi a monte del modello reale del drone si spiegherà anch’esso di seguito il motivo del suo posizionamento particolare.

Col colore magenta viene risaltato il blocco zero-order hold, che ha la funzione di prendere in ingresso il segnale analogico e di trasformarlo in digitale.



Nel rettangolo verde invece si denota il blocco rate transitino, ovvero un sistema in grado di prendere il segnale in ingresso e trasformarlo nella frequenza desiderata fino al valore che nel sistema è indicato come analogico (0.01 s).



Nel nostro caso specifico avremo le seguenti frequenze di elaborazione dei dati della piattaforma IMU e del GPS :

- Frequenza GPS = 5 Hz;
- Frequenza accelerometri = 400 Hz;
- Frequenza giroscopi = 50 Hz.

È facile capire dunque che la frequenza che penalizza il sistema è quella del GPS che interviene solo sulla posizione assoluta (N,E,D).

Trasformiamo ora le varie frequenze nei relativi tempi di campionamento, così poi da paragonarle al tempo di elaborazione del compilatore :

$$T_{GPS} = \frac{1}{f_{GPS}} = \frac{1}{5} = 0.2 \text{ s}$$

$$T_{acc} = \frac{1}{f_{acc}} = \frac{1}{400} = 0.0025 \text{ s}$$

$$T_{Giro} = \frac{1}{f_{Giro}} = \frac{1}{50} = 0.02 \text{ s}$$

E come accennato quella che ha un periodo più lungo di discretizzazione è quella del GPS.

Se però ponessimo tutte le variabili in input al regolatore a tale frequenza avremmo un sistema troppo lento e compromesso e quindi facilmente soggetto ad instabilità.

Ci si attiene allora al caso reale dove il regolatore assume frequenze diverse da ogni sensore, tranne per gli accelerometri che lavorano ad un tempo di campionamento più fine di quello di compilazione del computer.

Ecco spiegata la formazione dello schema in figura 15.1, ovvero i blocchi a monte del modello reale del drone prelevano i segnali di N,E,D, vengono discretizzati a 0.2 s e poi re introdotti nel sistema a 0.01 s.

A questo punto una volta costituito l'intero vettore, da moltiplicare per la matrice K in seguito, viene discretizzato tutto, per una stima più pessimistica dell'intero sistema, al tempo di 0.02s.

Si procede effettuando la moltiplicazione con K nel regolatore e si ritrasforma tutto a 0.01 s.

Proviamo dunque ad implementare questo modello con disturbi attivati e di valutarne la stabilità asintotica in un periodo di simulazione di 20 s.

15.4 VERIFICA E TEST DEL MODELLO DISCRETIZZATO

Per verificare tale modello in tutti i suoi campi di funzionamento sarà sufficiente metterlo su banco per i due casi canonici realizzati in questa relazione, testiamo dunque il modello prima in caso di hovering con un vento costante lungo E di 15 m/s.

$\dot{N}, \dot{E}, \dot{D}$

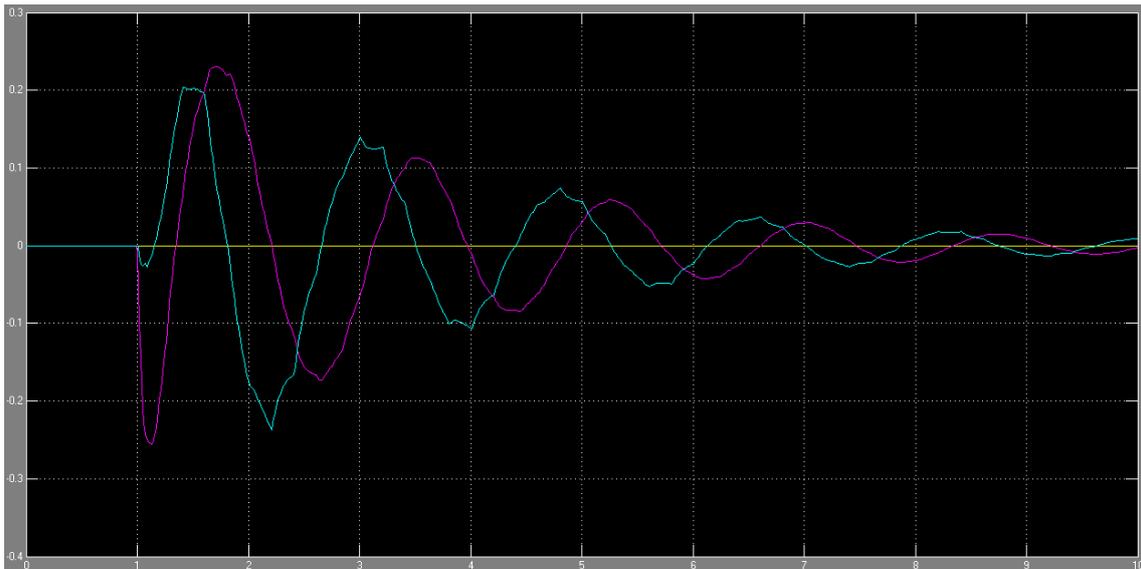


Figura 15.2 hovering e vento costante lungo E in $\dot{N}, \dot{E}, \dot{D}$

N, E, D

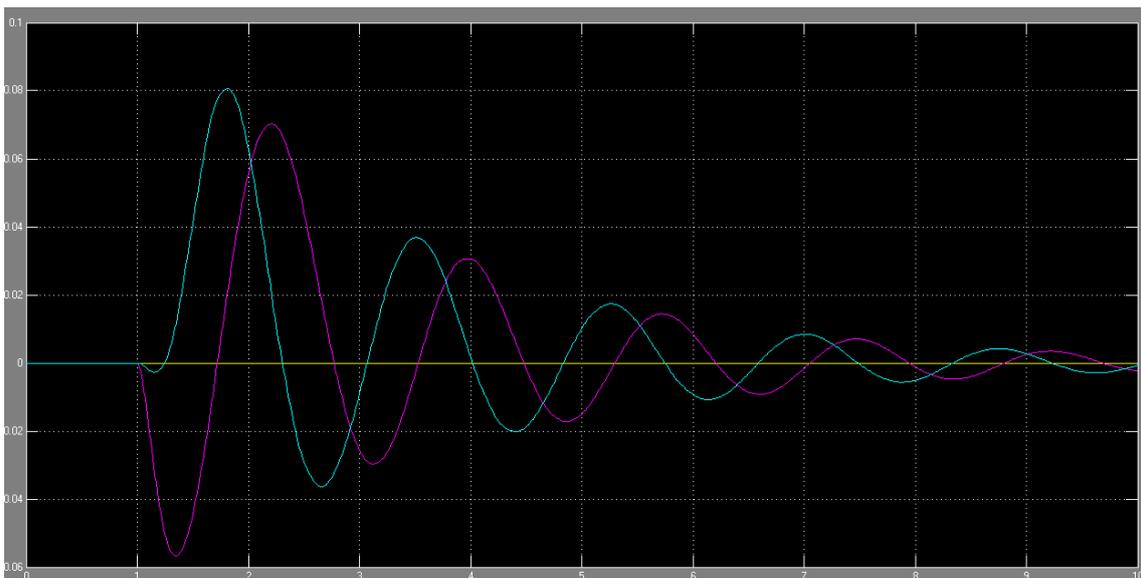


Figura 15.3 hovering e vento costante lungo E in N, E, D

E poi il caso di moto costante di 10 m/s lungo N ed un vento costante di 10 m/s lungo E.

$\dot{N}, \dot{E}, \dot{D}$

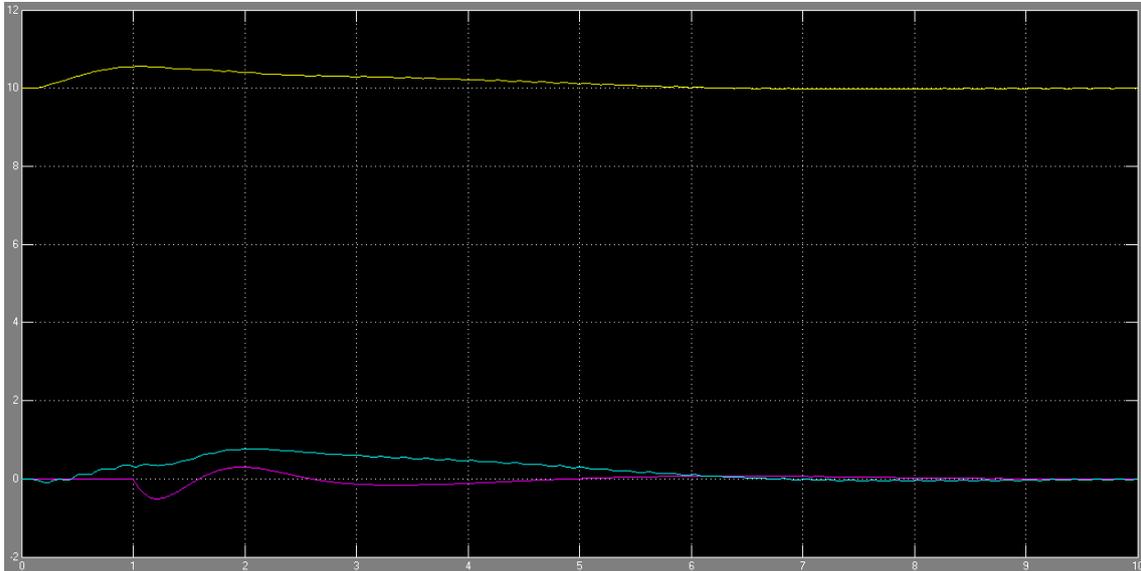


Figura 15.4 moto costante lungo N e vento costante lungo E in $\dot{N}, \dot{E}, \dot{D}$

N, E, D

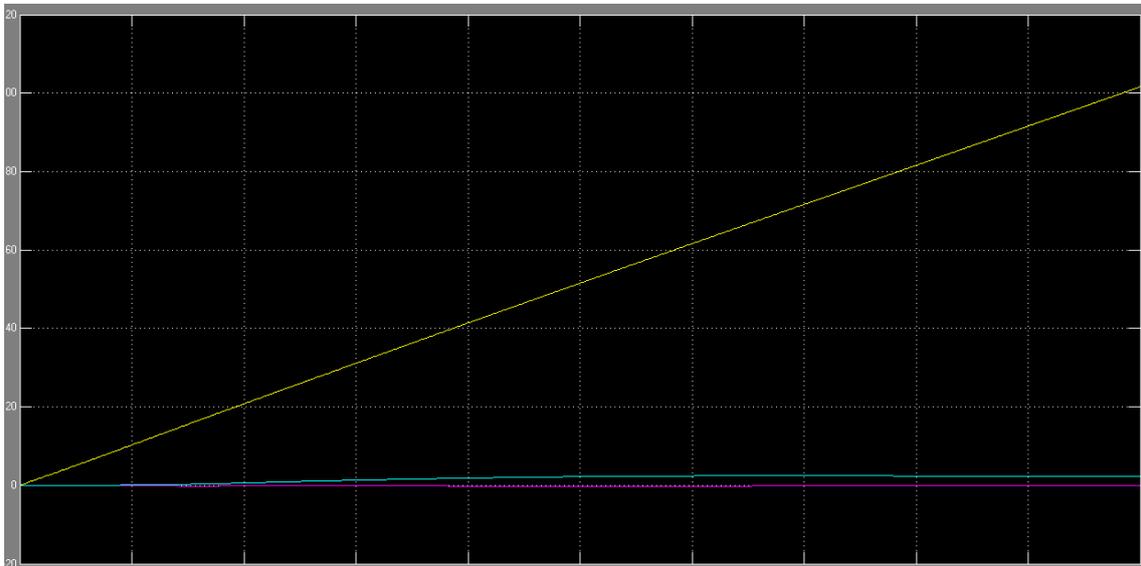


Figura 15.5 moto costante lungo N e vento costante lungo E in N, E, D

E come è facile verificare dai plot riportati il sistema è stabile e robusto anche dopo essere stato affetto da ritardi nell'assunzione di segnali in input.

Se un modello è stabile post discretizzazione allora implica che la modellazione è stata fatta correttamente quindi ciò valida in fine la bontà e la qualità del modello ottenuto.

16. MODELLO INERZIALE

Ultima fase di studio concerne il prendere in considerazione anche le inerzie di tutti i rotori che affliggono la risposta del sistema.

Durante lo studio, stilato fino a questo capitolo, abbiamo assunto una reazione istantanea del sistema, ovvero ad ogni incremento $\Delta\omega$ di velocità rotazionale dei propulsori si aveva una risposta immediata nella reazione del modello.

Il fatto invece che i rotori presentino una massa implicano che siano presenti accelerazioni inerziali che compromettano la risposta e la prontezza del sistema.

Quello che sarà aggiunto è un piccolo sistema che prende in ingresso le variazioni di velocità angolari di ogni singolo rotore e ne restituisca un segnale con i relativi ritardi.

In figura 16.1 viene mostrato come il segnale risente delle inerzie dei corpi meccanici assemblati.

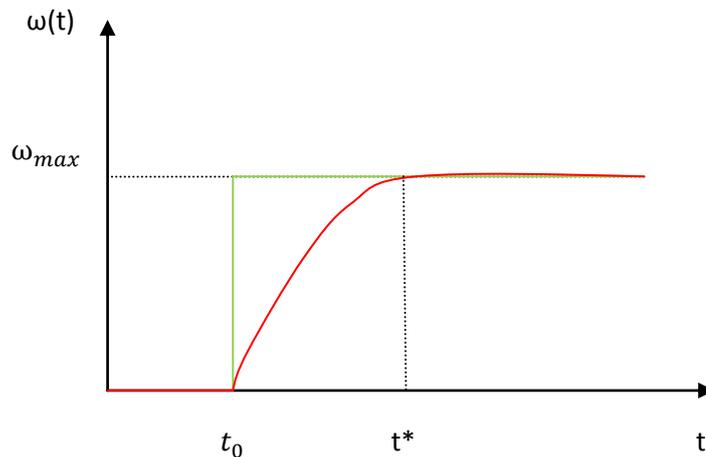


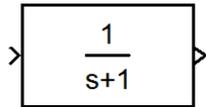
Figura 16.1 risposta di un sistema afflitto da inerzie

Il grafico esplicita bene il concetto fino ad ora esposto, difatti col colore verde si evidenzia la risposta di un segnale ideale, privo di masse, che quindi raggiunge la velocità ω_{max} istantaneamente.

Mentre la presenza di inerzia tende a procurare un rallentamento nella fase di variazione istantanea e quindi un'accelerazione che non ha più valore infinito ma ben definito, dalle caratteristiche fisiche dell'oggetto in esame, è necessario dunque al sistema impiegare un tempo $t^* - t_0$ per raggiungere la velocità nominale di rotazione (curva rossa).

Per implementare matematicamente questo ritardo si fa utilizzo di specifiche funzioni messe a disposizione dal software, vediamo ora in cosa consistono.

Transfer FCN:



La funzione in S sopra riportata non è altro che un sistema del primo tipo, ovvero:

$$G(S) = \frac{b_1 \cdot S + b_0}{a_1 \cdot S + a_0}$$

Ponendo $b_1 = 0$ e svolgendo questi semplici passaggi:

$$G(S) = \frac{b_0}{a_1 \cdot (S + \frac{a_0}{a_1})} = \frac{\tilde{k}}{S - P} = \frac{-\tilde{k}/P}{\frac{S}{P} + 1}$$

E ponendo $\tau = -1/P =$ costante di tempo:

$$G(S) = \frac{K}{\tau \cdot S + 1}$$

Imponendo $K = \tau = 1$ troviamo la forma della nostra funzione iniziale.

Se ora prendiamo come input un gradino (che equivale proprio al nostro caso), la cui trasformata di Laplace vale $1/S$, come uscita otterremo:

$$Y(S) = G(S) \cdot U(S) = \frac{K}{\tau \cdot S + 1} \cdot \frac{1}{S} = \frac{K}{S} - \frac{K}{S + 1/\tau}$$

E anti trasformando:

$$y(t) = K \cdot (1 - e^{-\frac{t}{\tau}})$$

Che descrive proprio la curva rossa di figura 16.1.

16.1 ESTENSIONE DEL MODELLO MATEMATICO

Valutiamo cosa comporta l'inserzione di un fattore inerziale nel modello reale ottenuto.

L'inserimento dei blocchi, con la funzione di trasferimento vista prima, implica l'iniezione della costante τ nel sistema circolatorio dei dati del sistema.

È necessario dunque considerare tale valore anche all'interno della matrice \mathbf{A} che verrà poi digerita dalla funzione di Riccati.

Il procedimento è del tutto analogo a quello delle matrici estese, vediamo ora la trattazione per la risoluzione di questo caso particolare.

Prima di tutto riscriviamo l'equazione generale delle variabili di stato:

$$\dot{x} = \mathbf{A}x + \mathbf{G}d\omega + \mathbf{D}u_d$$

Ricordando $u_d = [df_{wx} df_{wy} df_{wz} dm_{wx} dm_{wy} dm_{wz}]^T$

Quello che noi faremo è intervenire sulle variazioni di velocità angolare dei rotori, quindi si esprime di seguito la derivata di tale entità:

$$d\dot{\omega}_i = \frac{1}{\tau} \cdot (\omega_{ci} - d\omega_i)$$

Con $i=1,2,3,4$ e ω_c è la velocità angolare controllata.

Sotto questa nuova ipotesi è lecito introdurre un nuovo vettore delle variabili \bar{x}_e detto appunto esteso :

$$\bar{x}_e = \begin{pmatrix} x \\ d\omega \end{pmatrix}$$

Posto $d\omega = (d\omega_1, d\omega_2, d\omega_3, d\omega_4)'$

Si può procedere ora con un nuovo sistema matriciale così formulato :

$$\dot{\bar{x}}_e = \mathbf{A}_{11}\bar{x}_e + \mathbf{G}_1d\omega_c + \mathbf{D}u_d$$

Ed analogamente $d\omega_c = (d\omega_{c1}, d\omega_{c2}, d\omega_{c3}, d\omega_{c4})'$

Dove :

$$\mathbf{A}_{11} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \boldsymbol{\tau} \end{pmatrix} = \text{matrice estesa in } \omega \text{ di } \mathbf{A}$$

$$\mathbf{G}_1 = \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\tau} \end{pmatrix} = \text{matrice estesa in } \omega \text{ di } \mathbf{G}$$

Avendo definito la matrice:

$$\boldsymbol{\tau} = \frac{1}{\tau} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

E quindi si procede sostituendo \mathbf{A}_{11} ad \mathbf{A} e \mathbf{B}_1 ad \mathbf{B} nel procedimento delle matrici estese

Si riporta di seguito il nuovo script di Matlab con le modifiche appena trascritte ma solo per il caso di hovering, per il moto continuo la trattazione è del tutto analoga:

```
A=[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 -9.81 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    9.81 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0];
```

```
G=[0 0 0 0;
    0 0 0 0;
    0 0 0 0;
    2*Ks*om0*D*sin(al)/Ix -2*Ks*om0*D*cos(al)/Ix -2*Ks*om0*D*sin(al)/Ix 2*Ks*om0*D*cos(al)/Ix;
    2*Ks*om0*D*cos(al)/Iy 2*Ks*om0*D*sin(al)/Iy -2*Ks*om0*D*cos(al)/Iy -2*Ks*om0*D*sin(al)/Iy;
    -2*Kd*om0/Iz 2*Kd*om0/Iz -2*Kd*om0/Iz 2*Kd*om0/Iz;
    0 0 0 0;
    0 0 0 0;
    -2*Ks*om0/M -2*Ks*om0/M -2*Ks*om0/M -2*Ks*om0/M;
    0 0 0 0;
    0 0 0 0;
    0 0 0 0;
    0 0 0 0;
    0 0 0 0;
    0 0 0 0;
    0 0 0 0];
```

```

r=[1000 0 0 0;
   0 1000 0 0;
   0 0 1000 0;
   0 0 0 1000];

q=[100 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 100 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 100 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 100 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 100 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 100 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 100 0 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 100 0 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0.001 0 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0.001 0 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0.001 0 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0.001 0 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0.001 0 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0.001 0 0 0 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 100000000 0 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 100000000 0 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 100000000 0;
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 100000000];

```

```
A11=[A,G;zeros(4,15),1/tao*eye(4,4)];
```

```
G1=[zeros(15,4);1/tao*eye(4,4)];
```

```
C1=eye(19);
```

```
R=inv(r);
```

```

A12=[0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 1/Ix 0 0;
      0 0 0 0 1/Iy 0;
      0 0 0 0 0 1/Iz;
      1/M 0 0 0 0 0;
      0 1/M 0 0 0 0;
      0 0 1/M 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0];

```

```
A22=zeros(6,6);
```

```
[~,S11] = LQR(A11,G1,C1'/q*C1,R);
```

```
S12=lyap(A11'-S11*G1/R*G1',A22,S11*A12);
```

```
K=r*[G1'*S11,G1'*S12];
```

16.2 COMPILAZIONI MATRICI R E Q

La decisione dei coefficienti è di facile intuizione ma comunque si definiscono sempre in maniera empirica.

Infatti i nuovi dati inseriti sono relativi alle velocità angolari, la cui regolazione non è assolutamente di interesse per una buona retroazione del sistema.

Si assumono allora i numeri più grandi possibili in modo che il risultato poi sia scartato dalla funzione clean posta a monte dei calcoli di Riccati.

Nel capitolo precedente si può già constatare i valori assunti per le nuove variabili introdotte.

Per il caso di moto uniforme la considerazione è identica, quindi valori sempre alti per i coefficienti relativi, unica differenza rispetto al caso classico definito dal capitolo 14, risiede nella matrice **A** nella quale sono stati introdotti anche gli integrali delle posizioni.

Tali valori, come è già stato trattato, danneggiano la precisione della risposta a regime, ma sono obbligatori per il corretto calcolo delle funzioni di Riccati.

Unico stratagemma adottabile è penalizzare anch'essi nella matrice **Q** con coefficienti molto alti, si ottengono dunque le seguenti matrici **A** e **Q**.

```
A=[0 0 0 1 sin(phi)*tan(teta) cos(phi)*tan(teta) 0 0 0 0 0 0 0 0 0 0;
  0 0 0 0 cos(phi) -sin(phi) 0 0 0 0 0 0 0 0 0 0;
  0 0 0 0 sin(phi)/cos(teta) cos(phi)/cos(teta) 0 0 0 0 0 0 0 0 0 0;
  0 0 0 0 ix*r ix*q 0 0 0 0 0 0 0 0 0 0;
  0 0 0 iy*r 0 iy*p 0 0 0 0 0 0 0 0 0 0;
  0 0 0 iz*q iz*p 0 0 0 0 0 0 0 0 0 0;
  0 -9.81*cos(teta) 0 0 -w v 0 r -q 0 0 0 0 0 0;
  9.81*cos(teta)*cos(phi) -9.81*sin(teta)*sin(phi) 0 w 0 -u -r 0 p 0 0 0 0 0 0;
  -9.81*cos(teta)*sin(phi) -9.81*sin(teta)*cos(phi) 0 -v u 0 q -p 0 0 0 0 0 0;
  xphi xt xpsi 0 0 0 xu xv xw 0 0 0 0 0 0;
  yphi yt ypsi 0 0 0 yu yv yw 0 0 0 0 0 0;
  zphi zt 0 0 0 0 zu zv zw 0 0 0 0 0 0;
  0 0 0 0 0 0 0 0 0 1 0 0 0 0 0;
  0 0 0 0 0 0 0 0 0 0 1 0 0 0 0;
  0 0 0 0 0 0 0 0 0 0 0 1 0 0 0];
```

} Integrali alle posizioni

```
q=[100 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
  0 100 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
  0 0 100 0 0 0 0 0 0 0 0 0 0 0 0 0;
  0 0 0 100 0 0 0 0 0 0 0 0 0 0 0 0;
  0 0 0 0 100 0 0 0 0 0 0 0 0 0 0 0;
  0 0 0 0 0 100 0 0 0 0 0 0 0 0 0 0;
  0 0 0 0 0 0 100 0 0 0 0 0 0 0 0 0;
  0 0 0 0 0 0 0 100 0 0 0 0 0 0 0 0;
  0 0 0 0 0 0 0 0 0.001 0 0 0 0 0 0 0;
  0 0 0 0 0 0 0 0 0 0.001 0 0 0 0 0 0;
  0 0 0 0 0 0 0 0 0 0 0.001 0 0 0 0 0;
  0 0 0 0 0 0 0 0 0 0 0 100000000 0 0 0 0;
  0 0 0 0 0 0 0 0 0 0 0 0 100000000 0 0 0;
  0 0 0 0 0 0 0 0 0 0 0 0 0 100000000 0 0;
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 100000000 0;
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 100000000];
```

} Integrali alle posizioni
} Velocità angolare dei rotori

16.3 IMPLEMENTAZIONE DEL MODELLO INERZIALE

Applicando un blocco funzione di trasferimento per ogni rotore, in quanto sono tutti indipendenti l'uno dall'altro, si ottiene lo schema di figura 16.2.

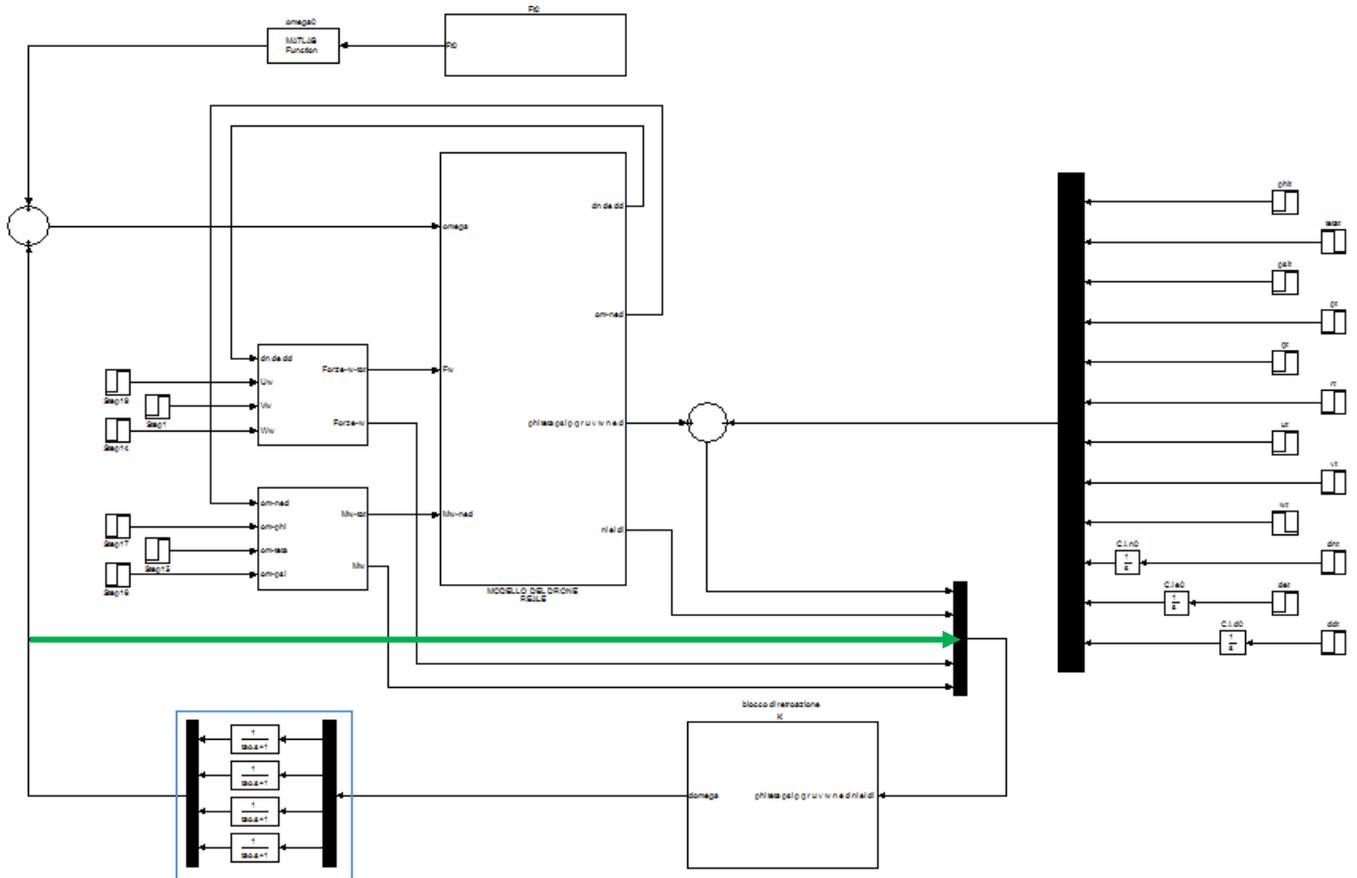


Figura 16.2 modello inerziale

L'implementazione delle masse dei rotori nel modello viene dunque effettuata all'uscita del regolatore indicata dal riquadro blu.

Con il vettore verde invece si evidenzia l'estrapolazione dei valori di velocità controllata assimilati poi per il calcolo delle nuove matrici estese.

Non resta che definire la costante di tempo τ .

È necessario assumere un valore approssimativamente il più reale possibile, e dipende dalla tipologia di rotori che monta la macchina.

Si assume dunque un valore pessimistico di sicurezza: $\tau = 0.1$ s.

La costante di tempo esprime la dilazione temporale per passare dalla condizione iniziale ad una che è al 95% di quella asintotica, ovvero il tempo che intercorre tra t_0 e t^* .

Prima di verificare il modello vediamo come reagisce il sistema dei rotori con l'aggiunta delle masse.

In figura 16.3 si visualizza l'andamento del sistema implementato a monte del nuovo blocco impiantato dopo il sommatore.

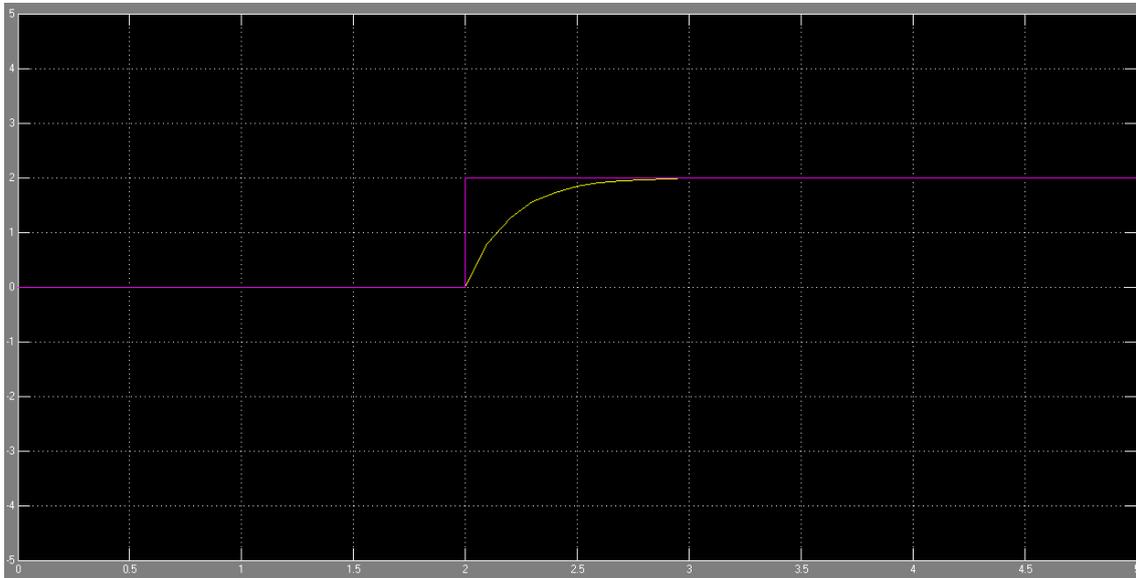


Figura 16.3 plot del sistema affetto da masse inerziali per un generico gradino in ingresso

In viola si evidenzia il segnale a gradino in input, mentre col colore giallo si risalta l'andamento dell'output affetto dal ritardo dovuto alle masse inerziali.

Non resta che mettere a banco il sistema e studiarne la stabilità globale dopo l'applicazione dei disturbi.

16.4 TEST E VERIFICA DEL MODELLO INERZIALE

Per valutare la bontà del software lo si prova nelle due condizioni canoniche di funzionamento soggetto sempre a disturbi. Quindi si fa una prova in hovering con un vento di 5 m/s lungo l'asse E applicato ad 1 s dall'inizio del test.

$\dot{N}, \dot{E}, \dot{D}$

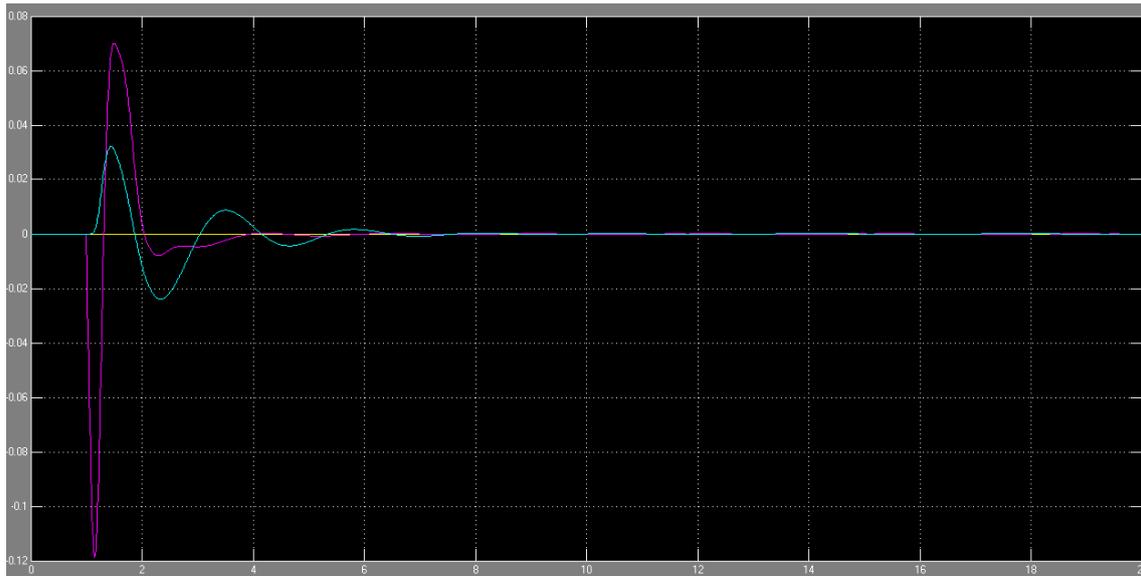


Figura 16.4 hovering e vento costante lungo E in $\dot{N}, \dot{E}, \dot{D}$

N, E, D

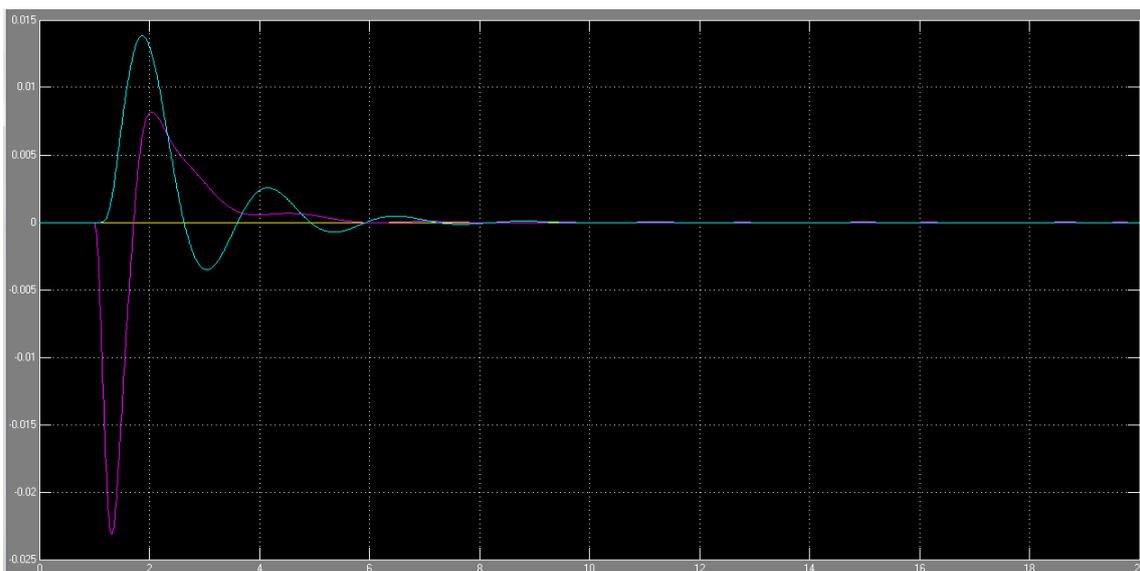


Figura 16.5 hovering e vento costante lungo E in N, E, D

Ora mantenendo lo stesso disturbo proviamo una condizione di volo livellato lungo l'asse N di 10 m/s costante.

$\dot{N}, \dot{E}, \dot{D}$

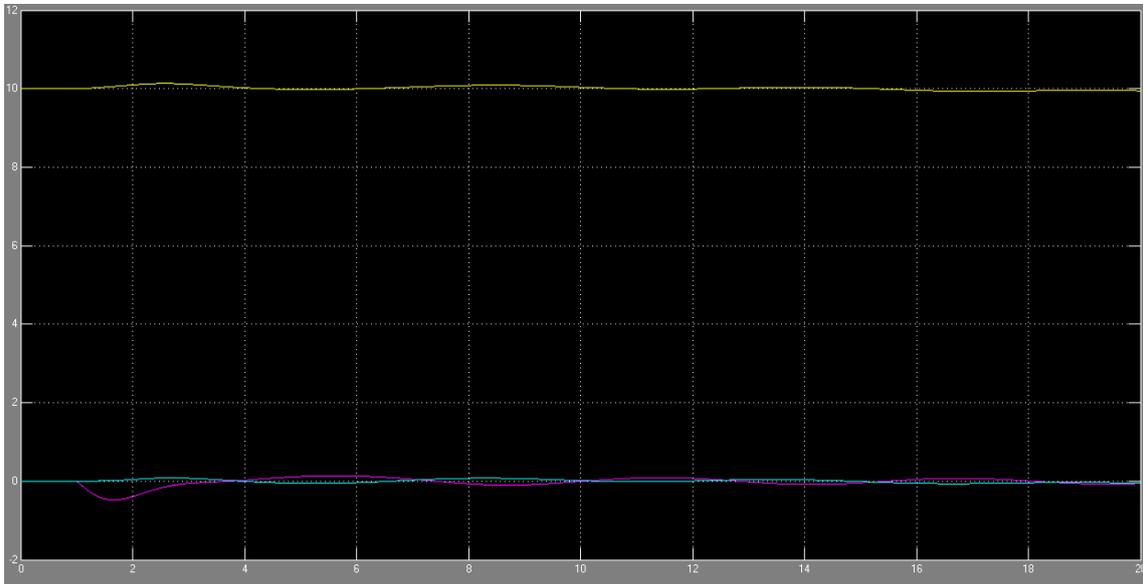


Figura 16.6 moto costante lungo N e vento costante lungo E in $\dot{N}, \dot{E}, \dot{D}$

N, E, D

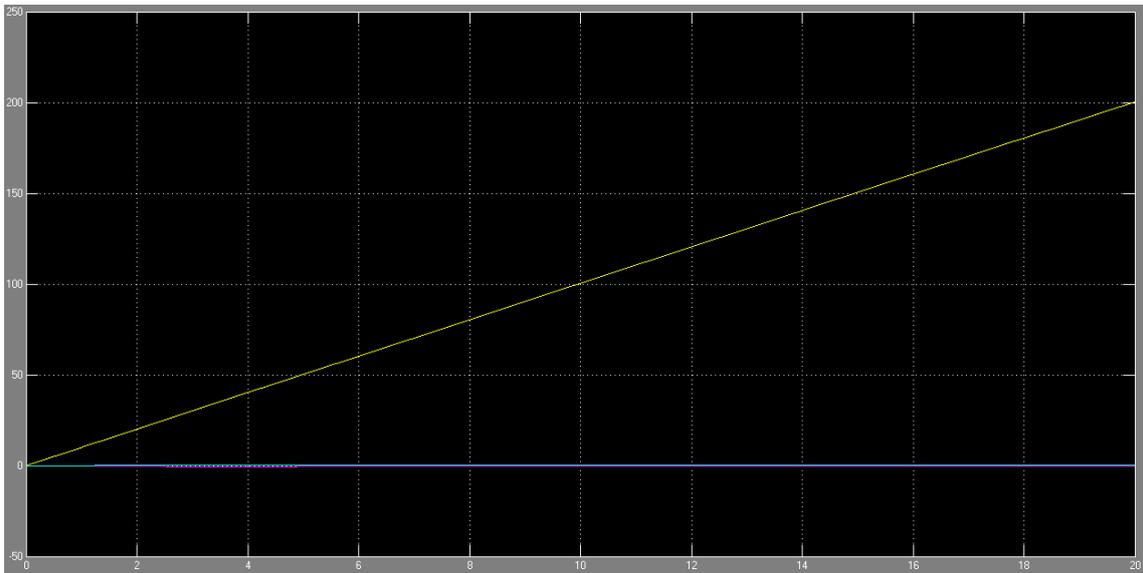


Figura 16.7 moto costante lungo N e vento costante lungo E in N, E, D

17. CONSIDERAZIONI CONCLUSIONI E SVILUPPI FUTURI

L'obiettivo di costruire una legge di controllo, per il pilotaggio assistito di un quadricottero, capace di compensare le forze di disturbo del vento, è stato ampiamente trattato in questa relazione di tesi.

Il modello finale, messo a banco, restituisce una risposta stabile e pronta, tale da asserire la sua totale affidabilità ed efficienza in tutti i range di utilizzo.

Dalle leggi canoniche del moto si è arrivati a costruire un sistema in grado di reagire attivamente alle condizioni ambientali sconosciute e di produrre un segnale ai propulsori in grado di compensare gli sbalzi di vento di qualsiasi natura.

L'implementazione di modelli matematici, atti a descrivere modalità di volo specifiche, ha reso raffinato il modello a tal punto da poterlo spingere alle condizioni limite di velocità e di poterne testare la sua bontà anche in condizioni reali di utilizzo tramite l'assunzione di segnali dai sensori discretizzati .

Si è partiti da un regolatore, costruito da un modello linearizzato, capace di operare solo in hovering e per disturbi a forma di impulso molto piccoli.

Di seguito, tramite lo sviluppo del modello reale, si è ottenuto un feedback capace di far muovere in moto costante e in tutte le direzioni il velivolo, ma solo grazie all'implementazione del modello a matrici estese.

Il sistema per la prima volta ha potuto reagire a disturbi costanti e continui restituendo un errore sulla posizione nullo a regime.

Congiunti i due tipi di regolatori si è ottenuto un sistema generalizzato capace di operare in tutte le condizioni possibili di volo, ma con velocità di crociera e dei venti di disturbo limitati entro i 2 m/s.

Il passaggio finale ha riguardato lo studio delle forze aerodinamiche; prendendo in considerazione l'attrito del vento dovuto al moto relativo tra drone e aria, si è stilato un modello matematico che una volta implementato ha consentito di ampliare il range di funzionamento a velocità dieci volte superiori quelle considerate fino ad ora fino a raggiungere le condizioni limite di 70 km/h .

I risultati, ottenuti dai test mettendo l'intero software al banco prova, sono stati, passo dopo passo, discussi e poi ritenuti validi solo se confrontabili con le prove sperimentali reali.

L'affidabilità e l'efficienza del regolatore sono tali da poterlo ritenere conforme per un suo utilizzo per una prova di volo su un quadricottero di tipo standard.

Un possibile sviluppo futuro concerne proprio la sua implementazione in un hardware collegato ad una piattaforma IMU e con un dispositivo GPS.

Il regolatore, il sistema di tracking e il blocco per il calcolo della velocità angolare iniziale dei quattro rotori verranno estrapolati dal modello finale e impiantati nella piattaforma di calcolo del drone per realizzare un sistema a catena chiusa in grado di fornire il feedback necessario a compensare i disturbi del caso.

L'utilizzo sempre più ampio di macchine costituite da più di quattro eliche può ritenere valido anche uno sviluppo successivo del software in grado di adattarsi a qualsiasi numero di rotori ed in qualsiasi formazione essi vengano montati, se simmetrici o speculari e controrotanti.

Un possibile passo avanti, quindi potrebbe riguardare una generalizzazione delle equazioni del moto in modo che la legge di controllo così ottenuta si adatti a tutte le tipologie di macchine e non il viceversa.

Per quando concerne tale relazione un piccolo passo in avanti è possibile tramite l'implementazione del modello inerziale con quello discretizzato, ovvero verificare se un modello doppiamente penalizzato risponde ancora fedelmente alle dinamiche ipotizzate e quali sono perciò i suoi limiti.

Con le leggi in arrivo del 30 aprile 2014, saranno varate normative per la certificazione delle macchine in circolazione nel commercio.

La possibilità quindi di disporre di droni che supportano software in grado di limitare l'effetto di disturbi dannosi per le persone presenti nelle zone di volo, oltre che per le riprese stesse, si propone già come un vantaggio futuro per un eventuale certificazione della macchina.

La scelta di sviluppare una legge di controllo ottimale per la compensazione del vento si è dimostrata vincente, rispetto all'ipotesi di montare sul velivolo dei sensori di collisione, per i seguenti motivi principali:

- Aumento del tempo di volo dovuto al risparmio di peso dei dispositivi di rilevamento;
- Risparmio economico per l'acquisto di tali elementi;
- Ottenere riprese fluide senza oscillazioni dannose;
- Governabilità più facile e sicura della macchina;
- Controllo e possibilità di ripresa anche in zone ventose o all'aperto.

Ovviamente i sensori forniscono un segnale di allerta in presenza di oggetti limitrofi alle aree di volo, ma il poter disporre di un sistema in grado di sommarsi ai comandi manuali di manovra e di isolare il disturbo del vento, a tal punto da limitare gli sbalzi di posizione, limita a sua volta la possibilità che si verifichi uno schianto contro pareti, oggetti e persone.

Tale conclusione pone la scelta di una legge di controllo, migliore rispetto a quella di disporre la macchina con sensori e quindi termina lo studio trattato in questa relazione positivamente.

18. BIBLIOGRAFIA

- Sistemi di controllo digitale, C. Bonivento, C. Melchiorri, R.Zanasi, progetto leonardo;
- Controlli automatici, G. Marro,Zanichelli;
- Elementi di meccanica classica, F. Mauro, Zanichelli;
- Sistemi lineari, S. Rinaldi, C. Piccardi, Cittàstudi;
- Progetto di sistemi di controllo, M. Tibaldi, Pitagora.

19. SITOGRAFIA

Figura 1.1 http://it.wikipedia.org/wiki/File:Szkic_%C5%9Bmig%C5%82owca.jpg

Figura 2.1 http://it.wikipedia.org/wiki/Aeromobile_a_pilotaggio_remoto

Figura 2.2 http://en.wikipedia.org/wiki/File:Parrot_AR.Drone_2.0_take-off,_Nevada.jpg

Figura 2.2 <http://www.italdron.com/nyx.php>

Figura 3.2 http://www.hobbyking.com/hobbyking/store/uh_viewItem.asp?idProduct=39708

Figura 3.3 http://www.modeltoy.it/product_info.php?products_id=7087

Figura 3.4 <http://www.aliexpress.com/store/product/Crazyradio-2-4Ghz-nRF24LU1-USB-radio>

Figura 3.5 <http://makezine.com/2009/01/06/ardupilot-arduino-compatible-uav-co/>

Figura 3.6 <http://www.geek.org/ar-drone-hacking-les-4-hacks-les-plus-impressionnants-338.html>

Figura 3.7 <http://scorpio.it/scheda-prodotto.dhtml?cod=7893&cat=v>

Figura 3.8 <http://www.singahobby.com/?q=node/25289>

Figura 4.1 http://en.wikipedia.org/wiki/File:Parrot_AR.Drone_2.0_take-off,_Nevada.jpg

Figura 5.1 <http://www.italdron.com/scrabble4.php>

Figura 5.2 <http://en.wikipedia.org/wiki/File:Rollpitchyawplain.png>

Figura 5.3 <http://people.ciram.unibo.it/~muracchi/Materiale%20didattico/eulero.pdf>

Figura 7.1 <http://www.ipad.it/ar-drone-2-0-parrot-svela-il-nuovo-quadricottero-al-ces-2012/>

Figura 7.3 <http://it.wikipedia.org/wiki/Portanza>

Figura 11.6 <https://twitter.com/ITALDRON>

