



Università degli Studi di Cagliari

Facoltà di Ingegneria e Architettura

**Corso di Laurea Magistrale in Ingegneria
Elettronica**

—

Corso di Controllo Digitale

**Modellazione e Controllo di un
Quadricottero in ambiente MATLAB
Simulink**

Prof. Alessandro Pisano

Augusto Mura

Numero di Matricola 70/83/65228

Roberto Ruda

Numero di Matricola 70/83/65240

Introduzione

I **quadricotteri** e altri tipi di **droni** sono **velivoli** senza pilota molto diffusi al giorno d'oggi. Anche le versioni commerciali più economiche contengono sofisticati **sistemi di controllo**, che consentono loro di essere **stabili** e di volare in maniera autonoma, con un intervento umano molto ridotto. In questo lavoro è stato progettato un sistema di controllo che consente di far seguire al drone una **traiettoria arbitraria predefinita**. Nello specifico, è stato scelto di prendere come riferimento un quadricottero, vista la loro sempre crescente diffusione e popolarità.



Figure 1 - Un quadricottero in volo.

1 - Cosa è un drone

Gli ultimi anni hanno visto una rapida espansione dell'utilizzo dei droni in svariati ambiti. Intrattenimento, sport, agricoltura, industria e sorveglianza sono solo alcuni dei settori coinvolti. Vista l'eterogeneità delle possibili applicazioni, esistono molteplici versioni di droni altrettanto diversi tra loro. Questo porta alla domanda: cosa è che rende un drone tale? La presenza di eliche, il numero di rotori, la dimensione del velivolo? **Non esiste** una **definizione univoca** di drone, ma bisogna inquadrare le **diverse tipologie**. Generalizzando, però possiamo, dire che si tratta di un **apparecchio** che vola tramite un sistema di **ali o eliche**, **senza figure umane a bordo e pilotato da remoto**.

Una descrizione più rigorosa può essere la seguente:

“Un aeromobile a pilotaggio remoto o APR, comunemente noto come drone, è un velivolo caratterizzato dall'assenza del pilota umano a bordo. Il suo volo è controllato dal computer a bordo del velivolo, sotto il controllo remoto di un navigatore o pilota, sul terreno o in un altro veicolo.”

A proposito di droni, l'**ENAC**, l'**Ente Nazionale per l'Aviazione Civile** in Italia, parla di **“Mezzi Aerei a Pilotaggio Remoto”**, che divide poi in due sottocategorie:

- **Sistemi Aeromobili a Pilotaggio Remoto (SAPR)**
mezzi impiegati o destinati all'impiego in **operazioni specializzate** (lavoro aereo).
- **Aeromodelli**
mezzi impiegati esclusivamente per **scopi ricreazionali e sportivi** e che non sono considerati aeromobili ai fini del loro assoggettamento al Codice della Navigazione.

Come è facile immaginare, un'oggetto volante senza pilota a bordo, con possibilità di trasportare carichi, fu azionato per la prima volta in **guerra**. Il primo drone ad uso militare fece la sua comparsa nel 1849, e venne utilizzato per bombardare Venezia.

Da questo primo prototipo, somigliante più a un pallone (con attaccate delle bombe) che a quello che oggi chiamiamo drone, la tecnologia è poi esplosa. Dall'utilizzo prettamente militare, questi mezzi si sono molto evoluti, ma è degli anni 2000 il boom che li ha fatti uscire dall'ambito militare. Ad oggi i droni vengono comunemente utilizzati nel **settore industriale, agricolo, ambientale, cinematografico, intrattenimento**.

Esistono diversi tipi di droni, raggruppati principalmente in base al loro **utilizzo** o al **numero di motori**.

Classificazione in base all'utilizzo

Per quanto riguarda l'utilizzo, possiamo fare una macro-distinzione tra droni ad **uso militare** e droni ad **uso civile**. Come spesso accade per le tecnologie avanzate, come detto prima, anche i droni sono nati per scopi militari e questi rappresentano, ancora oggi, una larga fetta del mercato. Gli ultimi anni, però, hanno visto l'ascesa potente ed inarrestabile dei droni ad uso civile, protagonisti di un successo senza pari determinato dalla sempre più ampia sfera di applicazione.

Per quanto riguarda gli usi civili, è poi possibile distinguere tra l'utilizzo professionale e quello amatoriale dei droni.

Ecco i principali usi di un **drone non professionale**:

1. **Video**, per riprese aeree anche ad elevata risoluzione in contesti amatoriali.
2. **Sport**, i droni più moderni possono essere impostati per seguire gli atleti durante un tragitto in moto o in bici, ad esempio.
3. **Gare**: i droni da gara vengono modificati dagli appassionati per essere sempre più veloci, vince chi conclude il percorso nel minor tempo.

I droni non destinati all'uso lavorativo oggi sono provvisti di sensori, integrazioni con app e tecnologie all'avanguardia. Questo semplifica l'interazione dell'utente con il sistema di volo, consentendo anche a persone prive di background in ambito aviazione di pilotare un drone.

I droni impiegati in **ambiti professionali** trovano largo impiego nei più disparati settori. In **agricoltura**, per il monitoraggio di campi e colture, le **forze dell'ordine** li impiegano per monitorare città e luoghi di interesse. In ambito militare, come detto prima, rivestono un interesse particolare per il fatto di non mettere a rischio la vita del pilota. Non da ultimo, il settore dell'intrattenimento, dello spettacolo e della cultura. È riportato qui un esempio particolarmente appropriato agli autori: la ripresa aerea (eseguita ovviamente mediante un drone) dell'esecuzione di "Learn To Fly" dei Foo Fighters, da parte dei Rockin'1000. Il 26 luglio 2015 al Parco Ippodromo di Cesena, 1000 musicisti italiani, fan dei Foo Fighters, hanno eseguito "Learn To Fly" per invitare i Foo Fighters a fare una tappa anche nella cittadina romagnola. I musicisti (350 chitarristi, 250 cantanti, 250 batteristi, 150 bassisti) hanno registrato una performance da record del brano contenuto nell'album "There Is Nothing Left to Lose", del 1999. Quale canzone risulta più appropriata di questa, visto il tema della relazione? Anche qui si tratta infatti di imparare a (far) volare (un drone)!



Figure 2 - I Rockin'1000 suonano "Learn to Fly" dei Foo Fighters (link: <https://www.youtube.com/watch?v=JozAmXo2bDE>). Ripresa aerea eseguita mediante drone. Quale canzone più appropriata per modellare un velivolo che deve imparare a volare?

Classificazione in base al numero di motori

Un altro possibile tipo di distinzioni è quella basata sul **numero di motori**. In generale, a parità di motore, maggiore è il numero di motori, maggiore sarà la spinta massima erogabile. Questa permette di portare in volo un peso maggiore, garantendo anche maggiore stabilità in aria, specie in presenza di vento. Per contro un drone con più motori dovrà essere dotato di un telaio più resistente, di un'elettronica più complessa e di conseguenza avrà un prezzo più alto. Non a caso, il quadricottero si è affermato come drone per uso ricreativo, mentre esacotteri e ottocotteri sono droni più utilizzati da professionisti.



Figure 3 - Diversi modelli di drone, ciascuno identificato da un caratteristico numero di rotori.

2 - Analisi del sistema da controllare

Come accennato nel precedente paragrafo, il **sistema fisico** costituito dal drone è piuttosto **complesso**, in quanto è possibile eseguire una vasta gamma di movimenti, regolando in maniera indipendente la **velocità di rotazione** di ciascun **motore**, e quindi la **spinta erogata**. Al fine di rendere questi velivoli accessibili al pubblico in termini di usabilità sono stati implementati opportuni sistemi di controllo che semplificano le manovre del velivolo stesso. Anche questa tendenza ha avuto origine in ambito di aviazione militare, con i sistemi di controllo **fly by wire**. Questi consentono di mascherare la reale azione da compiere sul velivolo (piegare timone, sollevare flapper, incrementare giri del motore...) sollevando il pilota dal controllo meccanico diretto delle componenti, consentono quindi di concentrarsi esclusivamente sull'esperienza di volo. È inoltre possibile definire alcuni automatismi per consentire una pronta risposta a delle condizioni esterne (ad esempio, inclinarsi contro una raffica di vento al fine di mantenere la posizione, senza richiedere l'intervento del pilota).

L'**obiettivo** è quindi **sviluppare** un possibile **sistema di controllo** che implementi anche solo in minima parte le funzioni descritte sopra. Chiaramente, tale algoritmo non può prescindere dal sistema fisico da controllare. Per questo lavoro si concentra sul modello a quattro rotori, in quanto è il più diffuso in commercio.

Prima di poter implementare un adeguato sistema di controllo, è necessario analizzare il sistema da controllare, individuare le variabili e i segnali di interesse e capire come è possibile agire su di essi.

Se si fosse coinvolti nel processo di progettazione e costruzione del drone, il compito sarebbe quello di guidare e influenzare il design del velivolo. Questo, nell'ottica di poter soddisfare i requisiti di controllo. Tuttavia, non è questo il caso in studio: il sistema oggetto di analisi è già stato realizzato, e si vuole sviluppare un opportuno sistema di controllo per questo sistema già costruito.

2.1 – I sensori

Può essere utile partire dai **dati** che è possibile misurare, e che vengono usati come **segnale di riferimento** nel **sistema di controllo**. Questi sono acquisiti per mezzo di diversi **sensori**: opportuni **dispositivi** che consentono di **effettuare misure** su **grandezze** relative all'**ambiente esterno** e allo **stato del drone**, restituendo un'informazione a partire dalla quale poter effettuare delle operazioni.

Quasi tutti i modelli di drone presentano:

1. un **sensore di distanza**;
2. una **videocamera**;
3. un **accelerometro**.

2.1.1 - Sensore di distanza

Il **sensore di distanza** viene utilizzato per **misurare la distanza dal suolo**. Sono possibili diverse soluzioni (sonar a **ultrasuoni**, **lidar** infrarosso...), ciascuna con vantaggi e svantaggi sia in termini di performance che di costi. Attraverso questo sensore il drone può determinare la propria quota rispetto al livello del suolo. A partire da questa informazione è possibile regolare opportunamente la spinta che il drone esercita verso il basso in modo che possa tenere la posizione fissata (*hover*), o segua la traiettoria predefinita.



Figure 4 - A sinistra: esempio di sensore di distanza per droni. Destra: sensore LiDAR per droni.

2.1.2 - Videocamera

Anche per il secondo sensore, la **camera**, sono possibili diverse configurazioni, che differiscono tra loro per **risoluzione**, **frequenza dei fotogrammi (frame rate)**, misurato in **fps**, frame per second), ecc. Usando delle opportune tecniche di elaborazione immagini, chiamate in generale **optical flow**, è possibile capire come gli oggetti ripresi si “muovono” tra un fotogramma e quello successivo, relativamente al drone. Da questo **movimento** il drone può determinare (o almeno **stimare**) il suo **spostamento** nello **spazio** e la sua **velocità**.



Figure 5 - Videocamera per uso drone.

2.1.3 - Accelerometro e giroscopio

Il terzo sensore è un **accelerometro**. Più propriamente, si tratta di una **IMU (inertial measurement unit)**, in quanto esegue la **misura dell’accelerazione** non su un solo asse, ma bensì su **tre assi ortogonali**. Contiene inoltre un **giroscopio a tre assi**. In questo modo è possibile caratterizzare in maniera esaustiva il **vettore accelerazione** (lineare) nello **spazio**, così come le **accelerazioni angolari**. Da questi dati si possono quindi determinare la **posizione** del drone e il suo **assetto di volo** (in inglese **flight attitude**), cioè l’**orientazione** del velivolo nello spazio.



Figure 6 - Rappresentazione accelerometro e giroscopio. Il primo misura accelerazioni rettilinee, il secondo quelle angolari (quelle che tenderebbero a far ruotare il drone intorno al suo centro di massa).

2.1.4 - GPS

A seconda dell'ambiente di utilizzo, potrebbe essere utile un **transponder GPS**. Tuttavia, questo da solo non fornisce un'accuratezza sufficiente (uno scarto di qualche metro basta infatti a fare la differenza tra un volo tranquillo e il prendere in pieno un palo, o peggio altri esseri umani). Da cui la necessità di utilizzare altri sensori, come visto per l'utilizzo della camera e di tecniche di optical flow.



Figure 7 - Esempio di modulo GPS per droni.

Questi sono i sensori necessari per automatizzare la guida del drone. Senza di questi, il pilota dovrebbe pilotare a vista, cioè basandosi sulle sole informazioni visive raccolte dai suoi occhi. In queste condizioni, la guida risulta ostica. Tuttavia, l'obiettivo di questo lavoro è quello di studiare e implementare un sistema di controllo automatico, per cui assumiamo che il velivolo in questione sia dotato di queste interfacce verso il mondo esterno.

Ci possono essere poi molti altri sensori: telecamere termiche, sensori di pressione, rilevatori di specie chimiche, e così via. Questi però sono specifici della particolare applicazione per la quale il drone viene impiegato, e non sono rilevanti ai fini della trattazione in oggetto.

2.2 – Gli attuatori

Dopo aver discusso i sensori, parliamo della loro controparte, gli **attuatori**. Questi **dispositivi** svolgono il compito complementare rispetto ai sensori. I sensori misurano una grandezza esterna e

restituiscono tale informazione. Gli attuatori ricevono un'informazione dal sistema di elaborazione e la interpretano come grandezza fisica, effettuando un'azione sull'ambiente fisico. Nel drone gli attuatori sono i **motori**. Ciascuno di questi è dotato di un proprio **rotore (propeller)**.



Figure 8 - A destra motori DC brushless per droni. A sinistra, le eliche.

2.2.1 – Motori brushless

A seconda del **tipo di drone** e della sua **dimensione**, sono possibili soluzioni diverse. Si va da **piccoli motori elettrici**, fino a potenti **motori jet**, impiegati nei droni ad ala fissa per applicazioni militari. Per i dispositivi commerciali a rotore oggetto di questo progetto, la scelta di default ricade sui **motori elettrici DC brushless**. Questi richiedono infatti una minore manutenzione rispetto a quelli brushed, che hanno bisogno di continue sostituzioni dei carboncini al fine di mantenere alta l'efficienza del motore stesso. Per contro questi ultimi sono meno complessi da controllare. Tuttavia, con l'arrivo sul mercato di sistemi di controllo elettronici, i vantaggi dei primi li hanno resi la scelta più diffusa.

Questo può essere lo spunto per discutere del fatto che, vista la complessità del funzionamento del motore, al fine di garantire un controllo fine e puntuale dello stesso, questi siano dotati di un proprio sistema di controllo in retroazione, come abbiamo avuto modo di affrontare durante il corso di Controllo Digitale. Potrebbe valere la pena di estendere lo studio del sistema fisico del drone, osservando più nello specifico il comportamento dei singoli motori. Tuttavia, si rischierebbe di uscire fuori tema e appesantire inutilmente la trattazione. Pertanto, questa possibilità viene lasciata come possibile sviluppo futuro.

2.2.2 – Disposizione spaziale dei rotori

I **rotori**, in un **quadricottero**, sono in genere disposti ai **vertici** di un **quadrato**. A seconda della **direzione di avanzamento** del drone rispetto alla **disposizione** dei **rotori**, sono possibili **due** principali **configurazioni**, quella a “+” e quella a “X”, mostrate di seguito.

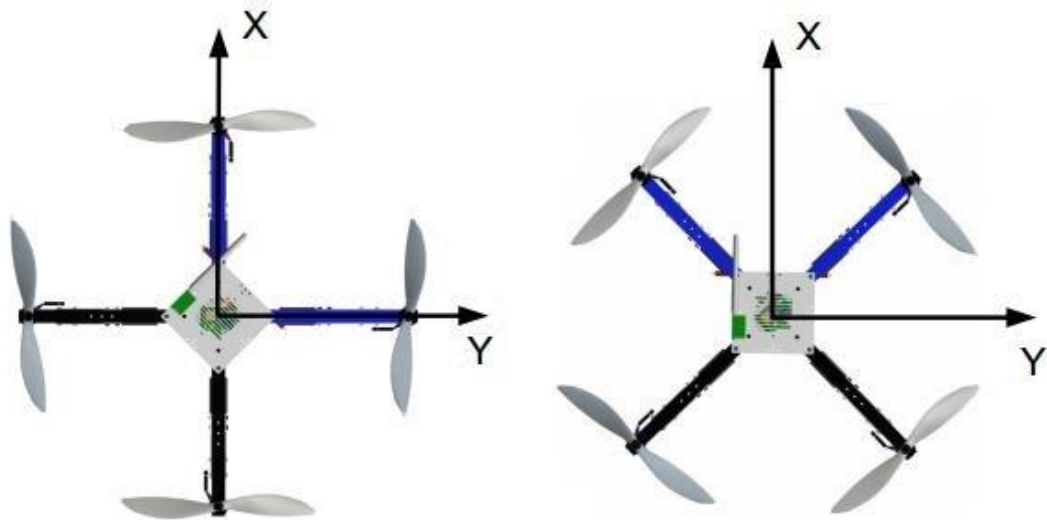


Figure 9 - Sinistra: configurazione a + di un quadricottero. Destra: configurazione a X.

Nella nostra trattazione ci concentreremo su un modello di drone con **configurazione a X**, cioè con i rotori posti ai vertici di un quadrato, del quale due lati sono paralleli al senso di avanzamento del drone, i restanti due ortogonali.

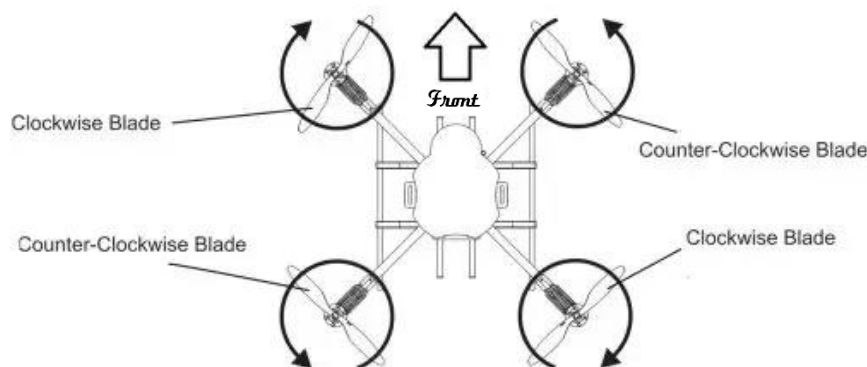


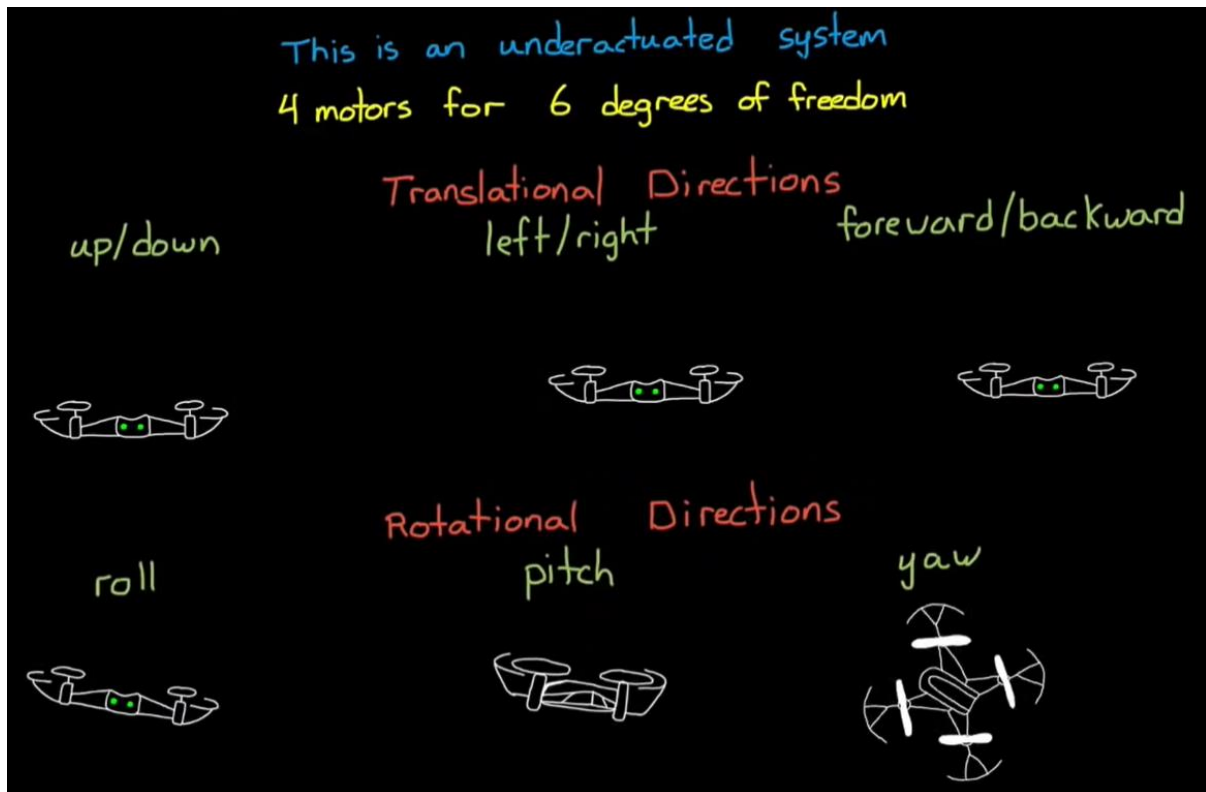
Figure 10 - Sensi di rotazione dei diversi rotori in un quadricottero.

I **rotori** del drone **ruotano** in modo tale che, in **condizioni stazionarie**, il **momento torcente totale** esercitato dalle **eliche** sul **corpo** del **drone** sia **nullo**. In questo modo si evita che il drone ruoti su sé stesso quando è invece preferibile che stia fermo. Vorremo infatti che a questa situazione corrisponda solo una spinta netta verso l'alto, senza altri contributi di movimenti in altre direzioni, né rotazioni.

Per ottenere questo effetto, si impone che i **motori** posti sulla **stessa diagonale** ruotino nello **stesso senso**, e che questo sia **opposto** a quello dell'**altra coppia**. Questo garantisce che, almeno in prima approssimazione, **spinta**, **rollio**, **beccheggio** e **imbardata** possano essere **controllati indipendentemente** l'uno dall'altro. In realtà, anche se in maniera lieve, tutti i movimenti sono legati tra loro, ma per i fini di questa trattazione è possibile trascurare questi dettagli.

3 - Descrizione del sistema dal punto di vista fisico

Il sistema è costituito da una macchina con quattro rotori. Questi sono disposti ai vertici di un quadrato. Per ciascuno di questi è possibile controllare la velocità di rotazione. Gestendo opportunamente questi quattro parametri è possibile far compiere al drone i movimenti riportati sotto, sia separatamente, sia una combinazione di questi.



- **Tre movimenti di traslazione** (uno lungo ciascuno degli assi 3d)
 1. **Traslazione** lungo **x** (avanti e indietro)
 2. **Traslazione** lungo **y** (destra e sinistra)
 3. **Traslazione** lungo **z** (sopra e sotto)
- **Tre movimenti di rotazione:**
 1. **rollio** (roll)
 2. **beccheggio** (pitch)
 3. **imbardata** (yaw)

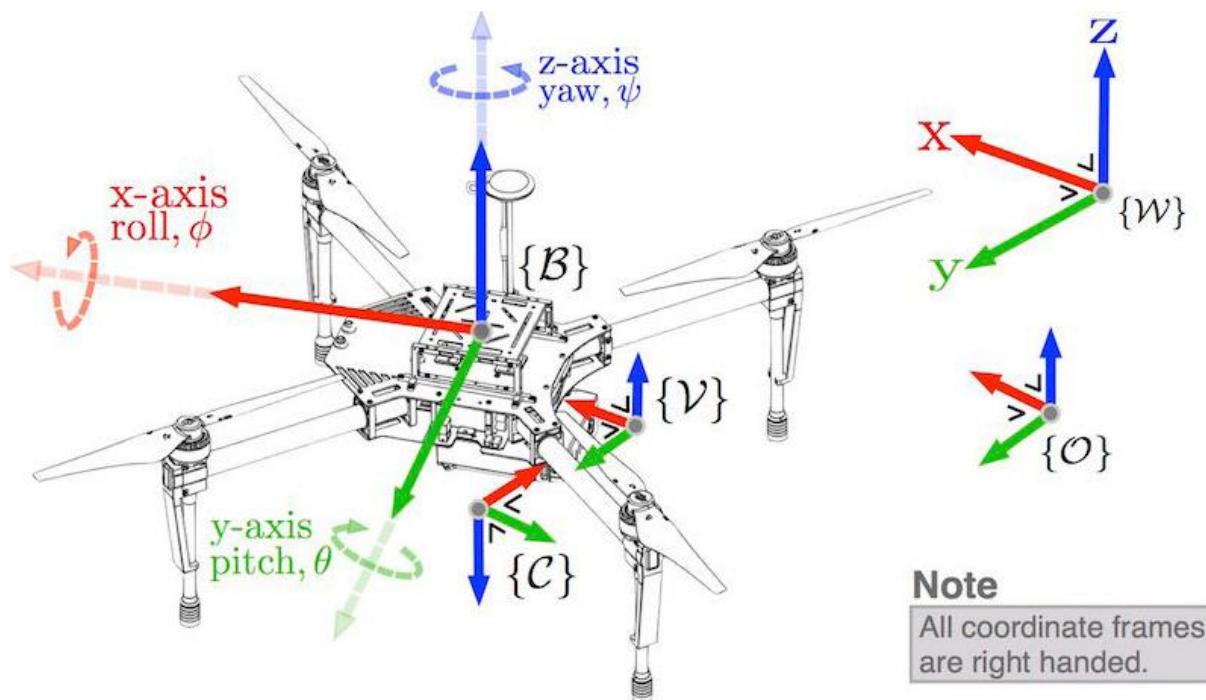


Figure 11 - Rappresentazione dei tre movimenti di traslazione e i tre di rotazione. È importante notare che il davanti del drone è rivolto nel verso positivo dell'asse x, il semiasse positivo y si trova a sinistra, e il verso positivo della quota z è verso l'alto.

Oltre alle considerazioni riportate nella nota precedente, va detto che tale sistema di riferimento è solidale col drone. È poi possibile definire un sistema di riferimento assoluto, solidale con il terreno, rispetto al quale definire i movimenti del drone stesso. Si può passare da un sistema di coordinate all'altro mediante un'opportuna matrice di trasformazione.

La posizione del drone è definita per mezzo di coordinate cartesiane in uno spazio tridimensionale. L'assetto di volo è identificato attraverso un'altra terna di valori, che prendono il nome di **angoli di Eulero** (indicati nella precedente figura con le lettere greche θ , ϕ e ψ).

In primo luogo, viene identificato e proposto un modello matematico del dispositivo in studio. Successivamente viene portata questa rappresentazione in ambiente di simulazione. Si è scelto di utilizzare il tool di sviluppo MATLAB, con l'aggiunta di Simulink.

Inoltre, vale la pena riportare che la terna di movimenti rotazionali, identifica altrettante posizioni angolari. Tale terna prende il nome di angoli di Eulero. Questi, infatti, descrivono l'orientamento di un corpo rigido nello spazio. Nello specifico, descrivono la posizione di un sistema di riferimento XYZ solidale con un corpo rigido attraverso una serie di rotazione, a partire da un sistema di riferimento fisso xyz.

Infine, dal punto di vista fisico, è importante sottolineare che, ai fini di semplificare la trattazione in esame si è optato per **trascurare l'attrito viscoso** dato dall'aria in cui il velivolo è immerso.

3.1 - Parametri fisici del drone

In questo progetto, il velivolo è stato modellizzato come un **corpo rigido**. In particolare, possiamo considerarlo come un **parallelepipedo**, a **sezione quadrata**, più largo che alto. Una sorta di mattonella, capace di traslare nello spazio e ruotare attorno ai suoi tre assi. Oltre che da una certa massa, il corpo è caratterizzato da tre valori di **momento d'inerzia**, uno per ciascuno degli **assi di rotazione**. Questi dipendono, oltre che dalla massa del drone, da come questa è distribuita intorno a ciascun asse. In

generale, questa risulterà non uniforme. Al centro del drone si trova in genere la batteria, che è il componente più pesante dell'intero sistema. I bracci sono praticamente vuoti, mentre invece i motori sulle estremità sono più pesanti, a causa degli avvolgimenti elettrici e degli eventuali magneti. Questa struttura risulta complessa da trattare in maniera analitica. La sua analisi complicherebbe la trattazione, senza però apportare informazioni utili ai fini del progetto. Tuttavia, piuttosto che calcolare il valore dei singoli momenti d'inerzia, è possibile recuperare dei valori di riferimento da alcuni studi relativi ad altri progetti. Facendo quindi riferimento ai dati reperiti in rete, sono stati assunti i seguenti valori dei parametri sopracitati: massa e momento d'inerzia.

| Grandezza | Valore | Unità di Misura |
|----------------------------------|--------|-------------------|
| Massa | 0,2 | kg |
| Momento d'inerzia (lungo asse x) | 0,1 | kg*m ² |
| Momento d'inerzia (lungo asse y) | 0,1 | kg*m ² |
| Momento d'inerzia (lungo asse z) | 0,12 | kg*m ² |

3.2 – Il sistema di coordinate

Nel fenomeno in studio è possibile identificare **due** diversi **sistemi di riferimento**: uno **solidale col drone (body, B)** e uno **assoluto esterno (earth, E)**. È utile notare che nel primo, essendo questo solidale col velivolo, questo occupa una posizione fissa, convenzionalmente scelta nell'origine. L'utilità di questo sistema di riferimento è l'identificazione dell'orientazione del drone nello spazio. Questa può essere descritta mediante gli angoli di Eulero. Un altro strumento per indicare l'orientazione del drone è la **DCM (Direct Cosine Matrix)**. Ancora, è possibile avvalersi dei quaternioni per descrivere i cambiamenti dell'orientazione dei corpi nello spazio.

Tuttavia, nel caso bidimensionale, vista la semplicità del sistema, non è necessario scomodare tali notazioni, essendoci un solo angolo da considerare.

4 - Descrizione del sistema complessivo

Dopo aver trattato sensori e attuatori, si passa ora a descrivere il sistema complessivo. Il sistema da modellizzare è il drone. Sono presenti quattro attuatori che imprimono forze e coppie sul sistema. Si ha inoltre accesso a un insieme di misure dirette e indirette dello stato del drone (velocità e **accelerazioni**, sia **spaziali** che **angolari**, acquisite mediante i **sensori** presenti a bordo). A partire da queste è possibile conoscere lo stato del sistema e l'evoluzione del sistema nel tempo.

5 - Esecuzione dei movimenti dei sei gradi di libertà

Complessivamente, abbiamo quindi **quattro** diversi **attuatori** che agiscono sul sistema (i **rotori**), e **sei gradi di libertà (tre traslazionali e tre rotazionali)**. In altre parole, sono presenti solo quattro variabili di controllo (la velocità, e quindi la spinta, di ciascun rotore), ma sei variabili di stato (la posizione x, y, z e l'orientazione θ, ϕ, ψ nello spazio). Non avendo un attuatore specifico per ciascuno di questi movimenti, questi non sono indipendenti tra loro. Il **sistema** risulta quindi **sotto-attuato**. Un meccanismo viene detto sotto-attuato se il numero di attuatori è minore del numero di gradi di libertà. Questo comporta che non si possano controllare in maniera indipendente tra loro ciascuno dei possibili movimenti del drone. Ad esempio, partendo da una condizione stazionaria (drone in sorvolo statico, hovering), se si volesse far traslare in avanti il drone, non si potrebbe ottenere questo effetto da solo, in maniera indipendente dagli altri movimenti elementari. Sarà necessario prima sbilanciare il drone in avanti, accelerando i rotori posteriori e/o rallentando quelli anteriori. In questo modo il

mezzo si inclina in avanti. Solo a questo punto la spinta totale che agisce sul dispositivo sarà non più solo verticale, ma avrà anche una componente orizzontale, responsabile della traslazione in avanti. Per ottenere la traslazione in avanti desiderata, è quindi necessario eseguire anche una rotazione sul piano mediano, movimento che prende il nome di beccheggio. In generale, questo è l'approccio che attueremo per ovviare al problema della sotto-attuazione: **combinare rotazioni** dell'intero sistema e **spinta** dei singoli motori per **ottenere lo spostamento desiderato**. Si discute quindi come generare sei diversi movimenti (ricordiamo, tre traslazioni e tre rotazioni) con i soli quattro motori disponibili.

Un **motore** produce spinta facendo girare l'**elica**, la quale spinge l'**aria** verso il **basso**, imprimendo una **forza** di pari intensità ma con verso opposto sul **drone**. Questo in accordo con il **terzo principio della dinamica**. Se questa forza fosse applicata sulla verticale passante per il baricentro del corpo, avremmo una traslazione verticale pura. Applicando la forza fuori asse avremmo invece una combinazione di traslazione verticale e rotazione intorno al baricentro del sistema fisico. Ma, ancora, se applicassimo una stessa forza su ciascuno dei quattro rotori, avremmo di nuovo una traslazione pura. Infatti, data la simmetria del sistema, il contributo rotazionale dato da ciascuno dei quattro rotori si semplificherebbe con quello del rotore in posizione diametralmente opposta. A questo punto è legittimo chiedersi se non sia possibile realizzare un dispositivo più semplice, con due soli rotori, possa svolgere le medesime operazioni della versione a quattro rotori, con minore impiego di componenti. La risposta è no: supponendo i due rotori posti perpendicolarmente alla direzione di avanzamento desiderata, questi consentirebbero di controllare il rollio del velivolo, ma non il beccheggio. Analogamente per le altre disposizioni spaziali, ci si accorge presto che questa configurazione lascia dei movimenti non controllabili. Con tre rotori, diventa più problematico effettuare traslazioni pure, senza rotazioni, vista la difficoltà di bilanciare opportunamente spinte e rotazioni del velivolo attorno al suo asse.

Aggiungendo un secondo asse, otteniamo la configurazione a quattro rotori dalla quale siamo partiti. A partire da questa, possiamo effettuare un sorvolo statico (*hover*) ripartendo equamente una spinta pari alla forza di gravità fra i quattro rotori, i quali devono essere controrotanti a due a due. Quest'ultima condizione garantisce che il momento torcente totale sul drone sia nullo, e che questo non giri su stesso quando i motori sono nella condizione appena descritta. Tuttavia, tutti i droni in commercio presentano un ulteriore vincolo: i motori che ruotano nello stesso senso sono opposti. Perché questa scelta? Per annullare il momento torcente in condizioni stazionarie basterebbe che i motori con stesso senso di rotazione stiano su uno stesso lato del drone. La ragione dietro questa scelta ha a che vedere con il come l'imbardata interagisce con rollio e beccheggio. Abbiamo visto come la scelta di motori controrotanti a due a due garantisca che in condizioni di simmetria della forza applicata, il momento torcente sul piano dell'imbardata sia nullo. Ha senso quindi pensare che sbilanciando questo equilibrio in favore di una coppia o dell'altra consenta di far ruotare il drone sul piano X-Y in un certo senso piuttosto che nell'altro. Rallentando una coppia e accelerando l'altra in maniera opportuna, è possibile tenere costante la spinta verso il basso, garantendo che la posizione lungo l'asse z resti fissa. Se però le coppie di motori si trovano su uno stesso lato, allora rallentarne/accelerarne una causerebbe uno squilibrio delle forze agenti sul baricentro, e il velivolo necessariamente si inclinerà rispetto all'asse x o all'asse y, a seconda del lato su cui si trova la coppia di motori. Questo mostra come anche la configurazione con tre rotori necessiti di ulteriori considerazioni per compensare spinta e rotazioni che agiscono sul sistema complessivo.

Questa configurazione appena descritta consente di rendere indipendenti quattro dei sei movimenti che il drone può compiere. Analogamente a quanto visto per l'imbardata, possiamo compiere i restanti due movimenti. Per rollare, si può diminuire la spinta dei motori di uno dei due lati e aumentare quella dell'altro lato. Per il beccheggio, stessa cosa, ma considerando le coppie di motori

anteriore e posteriore. Nessuno di questi movimenti influenza l'imbardata, in quanto il momento torcente totale esercitato da ciascun drone è perfettamente bilanciato da uno dei motori dell'altra coppia. Per modificare la spinta è necessario modificare la velocità di tutti quanti i motori simultaneamente.

In questo modo, si ha diretto controllo su questi quattro movimenti: traslazione verticale, rollio, beccheggio e imbardata. Le traslazioni in orizzontale (e tutte le combinazioni dei movimenti base) saranno ottenute combinando opportunamente questi quattro controlli.

Riassumendo:

1. È possibile modificare la spinta regolando tutti e quattro i motori alla medesima velocità.
2. Il beccheggio è controllato aumentando o diminuendo la spinta esercitata dalla coppia di motori frontali, e aggiustando la coppia posteriore in maniera contraria.
3. Per il rollio stessa cosa, ma usando le coppie di motori di destra e sinistra.
4. Per l'imbardata, incrementiamo la spinta esercitata dalla coppia di motori opposti che ruotano nello stesso senso in cui vogliamo che il drone giri, riducendo la spinta dell'altra coppia, il tutto in modo tale che la spinta totale risulti costante.
5. Le traslazioni in orizzontale non possono essere ottenute come movimenti puri, ma saranno una combinazione di rollio e/o beccheggio con un'opportuna regolazione della spinta.

Almeno intuitivamente, questo è l'algoritmo di motor mixing (*MMA, motor mixing algorithm*), che si occupa di convertire comandi di controllo espressi in termini di movimenti basi, in termini di numero di giri al minuto di ciascuna delle eliche. È doveroso specificare che le precedenti considerazioni reggono nel caso particolare in cui non intervengano altri disturbi dall'esterno e che il drone sia immerso in un fluido (l'aria) con proprietà spazialmente uniformi.

Si hanno quindi tutti gli ingredienti necessari per conoscere lo stato del drone e agire su di esso. Manca ancora però il *come* agire sul sistema. In altre parole, è ancora da realizzare un opportuno sistema di controllo in retroazione.

5.1 - Semplificazione del sistema in studio

Ci si è accorti che, fatte le opportune premesse teoriche, implementare un modello matematico del quadricottero, che dotato di sei gradi di libertà, è un'operazione piuttosto complicata. La portata di tale compito esula dagli obiettivi del corso di Controllo Digitale. Si è pertanto deciso di concentrarci su un problema più semplice.

Si è partiti con lo studiare un fenomeno semplificato. Più nello specifico, si è considerato il caso di un drone dotato di **due soli rotori**, capace di spostarsi su un **piano verticale**. Tale oggetto è un'astrazione e vive, è definito, su un piano. Si tratta in sostanza di una barra con due rotori posti sulle estremità. In questo contesto, i possibili movimenti sono traslazioni orizzontali e verticali (e le loro combinazioni), e la rotazione attorno al centro di massa, perpendicolarmente al piano identificato. Si hanno quindi ora in totale tre gradi di libertà (due traslazionali e uno rotazionale). Le considerazioni sono analoghe a quelle svolte per il modello tridimensionale, dotato di sei gradi di libertà. Anche nel caso semplificato, infatti, il sistema risulta sotto-attuato (due soli attuatori per tre movimenti base complessivi). Anche qui la soluzione è la medesima: si possono effettuare traslazioni verticali e rotazioni in maniera indipendente, ma le traslazioni orizzontali risulteranno da un'opportuna combinazione dei due precedenti movimenti. Prima di poter traslare lateralmente, sarà infatti necessario inclinare il drone verso direzione nella quale si intende proseguire.

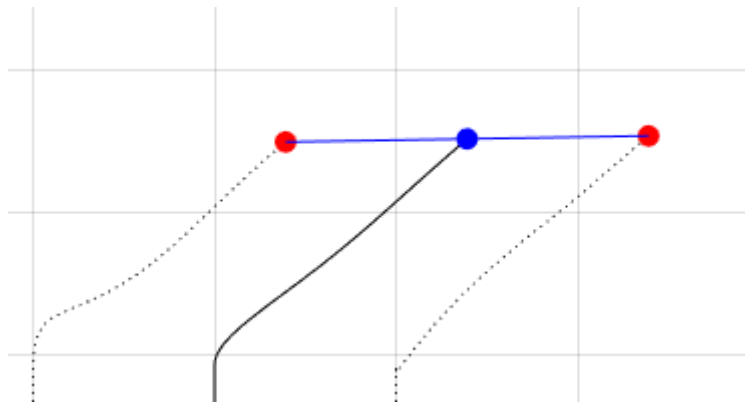


Figure 12 - Possibile rappresentazione del sistema semplificato. Tale drone risiede e sarebbe in grado di spostarsi in uno spazio 2D.

Questa semplificazione introduce delle limitazioni anche nei movimenti del drone. Questo, infatti, non potrà più muoversi nell'intero spazio tridimensionale, ma sarà vincolato al solo piano sagittale (se immaginiamo di osservare il drone originale di profilo. Se invece lo guardiamo frontalmente o d retro, il piano in cui il drone si può muovere sarà quello frontale, in maniera analoga al caso precedente).

Di seguito è riportata la modellizzazione matematica del sistema in studio.

6 – Formulazione matematica

6.1 - Identificazioni delle grandezze di interesse

È di fondamentale importanza identificare i **segnali di interesse** per il nostro modello.

L'obiettivo è quello di creare un algoritmo che regoli opportunamente spinta complessiva impressa al drone, al fine di garantire la stabilità del mezzo, e il poterne controllare posizione e traiettoria. Questa sarà poi l'ingresso dell'algoritmo di motor mixing, che tradurrà questa sollecitazione in termini di velocità di rotazione di ciascuno dei rotori. Siamo quindi interessati alla traiettoria percorsa dal drone dal punto di partenza al setpoint imposto, nonché alla sua inclinazione (assetto di volo). La prima informazione è descritta in maniera esaustiva dalla coppia di valori (x, y) che identificano le coordinate spaziali del baricentro del drone. Nell'ipotesi di partenza questo è anche il punto medio della barra che modella il drone. La seconda informazione, l'inclinazione è l'angolo compreso tra la verticale e l'asse del drone-barra.

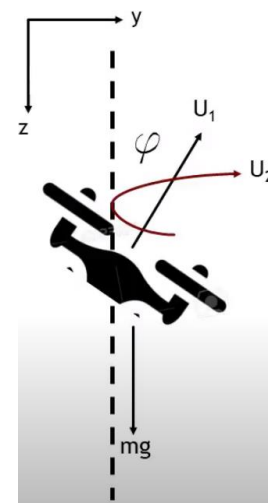


Figure 13 - Rappresentazione schematica del drone all'interno dello spazio cartesiano in cui risiede.

6.2 - Modellizzazione

Per ragioni di leggibilità del modello, piuttosto che lavorare in termini di spinte dei singoli rotori, si è preferito lavorare in termini di **spinta totale** e **momento torcente totale**, in accordo con l'**approccio più diffuso in letteratura**. Questa descrizione ha l'ulteriore vantaggio di essere più esplicativa dello stato del drone. È poi possibile passare dalla formulazione utilizzata ai comandi per i singoli motori mediante un opportuno algoritmo di Motor Mixing. I parametri sui quali la simulazione può agire per

controllare posizione e assetto di volo del drone sono proprio la **spinta totale** (*thrust*, **T**), esercitata complessivamente dai motori sul velivolo, e il **momento torcente** agente sul velivolo stesso, causato dall'eventuale squilibrio tra la spinta dei diversi rotori. Questa seconda grandezza è indicata con il simbolo τ . In generale si tratta di un vettore, solo che, in questo caso semplificato, la direzione è fissata, ed è quella ortogonale al piano sul quale questo drone 2D può muoversi. Il verso, indicato dal segno, distingue invece una rotazione che avviene in senso positivo o in senso negativo

La domanda che sorge spontanea è: come passare da queste due grandezze, alla terna di posizione (sia lungo l'asse orizzontale che verticale, quindi sull'intero piano) e l'angolo indicante l'assetto di volo del drone?

Si è partiti dalle basi, cioè dalle equazioni del moto viste a Fisica I. Forza massa e accelerazione sono legate tra loro dalla seguente relazione:

$$\mathbf{F} = m \cdot \mathbf{a}$$

(Secondo Principio della Dinamica)

E che, in maniera analoga, possiamo trovare una relazione formalmente simile per quanto riguarda i moti rotazionali:

$$\tau = m \cdot \alpha \cdot \hat{r}$$

dove τ è il momento torcente, risultante dall'applicazione della coppia di forze. Con \hat{r} si identifica il versore nella cui direzione giace il vettore momento torcente. Data la simmetria geometrica del problema, quest'ultimo è nullo quando le spinte generate dai rotori sono uguali in modulo (dato che per direzione e verso sempre identici). Inoltre, avendo il sistema un solo grado di libertà rotazionale, il vettore τ sarà necessariamente ortogonale al piano su cui il drone vive.

Utilizzando le relazioni inverse, possiamo ricavare accelerazione spaziale e angolare, noti forza e momento torcente applicati (risultanti dall'azione congiunta dei due rotori) e i parametri propri del drone (la sua massa e il suo momento di inerzia).

A partire poi dalle leggi della cinematica, sappiamo che l'accelerazione è definita come derivata della velocità e che, a sua volta, la velocità è la derivata della posizione. Queste relazioni sono valide sia per le grandezze spaziali, sia per le controparti angolari. Note quindi accelerazione spaziale e angolare, possiamo procedere a effettuare due integrazioni nel tempo su ciascuna di queste, in modo da ottenere la posizione del drone nello spazio, e la sua orientazione.

| Grandezze spaziali | Grandezze rotazionali |
|--------------------------------|------------------------------------------|
| $v = \int_{t_i}^{t_f} a \, dt$ | $\omega = \int_{t_i}^{t_f} \alpha \, dt$ |
| $s = \int_{t_i}^{t_f} v \, dt$ | $\theta = \int_{t_i}^{t_f} \omega \, dt$ |

Figure 14 - Relazioni che legano l'uscita del sistema (posizione) agli ingressi (forze, o, in questo caso, accelerazioni). I pedici "i" ed "f" indicano rispettivamente iniziale e finale.

Le due grandezze posizione spaziale **s** e posizione angolare θ non sono indipendenti l'una dall'altra. In particolare, la posizione del drone è individuata da una coppia di coordinate, **x** e **y**. Le relazioni illustrate sopra però si riferiscono solo a termini scalari. Possiamo però considerare distintamente i due contributi, scomponendo il vettore posizione nelle sue componenti parallele agli assi cartesiani.

Per ciascuna delle due componenti valgono le relazioni descritte per le grandezze rettilinee. Come operare questa scomposizione? A partire dall'angolo θ , il quale indica l'inclinazione del drone. Si è visto infatti come questo possa essere ricavato, noto il momento torcente esercitato sul drone. Scomponendo quindi la spinta T lungo gli assi x e y , dividendo poi entrambe per la massa m , e operando due integrazioni nel tempo, otteniamo infine lo spostamento del drone rispetto alla posizione iniziale. Quello che si è descritto è il passaggio da coordinate sferiche a quelle cartesiane nel caso del piano 2D. Di seguito sono riportate le equazioni che descrivono tale modello (non lineare).

$$\ddot{z} = g - \frac{|F|}{m} \cdot \cos(\phi)$$

$$\ddot{y} = \frac{|F|}{m} \cdot \sin(\phi)$$

$$\ddot{\theta} = \frac{|\tau|}{I_{xx}}$$

Vale la pena notare che nella prima equazione l'accelerazione di gravità ha segno positivo. Questo va contro la notazione usuale che si è abituati a vedere nei corsi di fisica. Non si tratta tuttavia di un errore, ma di una diversa convenzione. In particolare, in ambito di aviazione, si assume l'asse delle quote, z , con verso positivo rivolto verso il basso. Per coerenza con la letteratura presente sul tema, i è deciso di adottare questa stessa notazione. Per lo stesso motivo, la spinta prodotta dai rotori, rivolta verso l'alto risulta avere segno negativo.

Un'altra considerazione importante è che, come per tutti i problemi descritti da equazioni differenziali, è necessario definire le $n-1$ condizioni iniziali al fine di determinare una soluzione univoca dell'equazione (problema di Cauchy). Nel nostro caso, queste condizioni iniziali sono posizione e velocità spaziali iniziali e l'inclinazione iniziale del drone. Per semplicità di trattazione sono sempre state assunte come nulle. In altre parole, si è assunto che il drone parte sempre da fermo, in orizzontale, dall'origine del sistema di riferimento.

Le equazioni appena descritte modellizzano il comportamento del drone. Tale modello è stato poi implementato su **Simulink**. Lo schema è mostrato in figura:

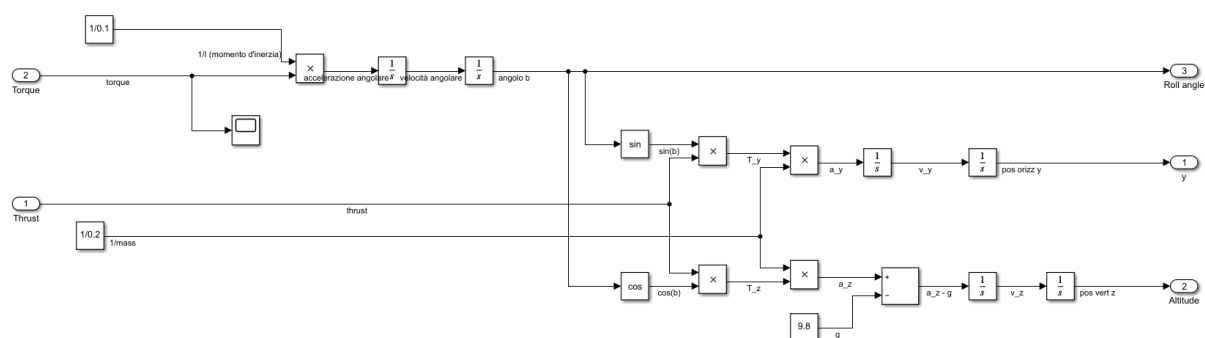


Figure 15 - Modello non lineare tempo continuo del drone oggetto di studio, nel caso semplificato bidimensionale.

6.3 - MATLAB e Simulink

MATLAB è una **suite di sviluppo software** proprietaria sviluppata dalla **MathWorks**. È un software professionale, completo e fornisce molti **toolbox** specifici per diversi campi specifici delle **scienze** e dell'**ingegneria**. È inoltre molto usato sia in ambito **accademico** che **industriale**. Nello specifico presenta potenti toolbox per simulazione e implementazione di **sistemi di controllo**, sia in dominio **continuo** che **digitale**. Ciò ne ha motivato l'adozione per questo progetto.



6.4 - Linearizzazione del modello

Il **modello** di un **drone reale** è, come molti sistemi fisici reali, **non lineare**. Ma spesso è possibile individuare un **punto di lavoro**, e operare una **linearizzazione** in un intorno di esso. Dal punto di vista strettamente matematico, questo procedimento coincide con lo **sviluppare** le **equazioni** del modello in **serie di Taylor**, e **troncarle** al **primo ordine**.

Come **punto di lavoro** è stato scelto quello in cui il **drone** è in **posizione stazionaria**, poiché è anche la condizione di maggiore **stabilità**, intorno alla quale il drone oscilla per compiere i movimenti richiesti. Nello specifico, nel caso del sistema 2D, si è effettuata un'**approssimazione** per **piccole oscillazioni**. Per valori di angoli prossimi allo 0, il seno dell'angolo si confonde col suo argomento, mentre il coseno è vicino all'unità.

$$\ddot{z} = g - \frac{|F|}{m} \cdot \cos(\phi) \rightarrow g - \frac{|F|}{m}$$

$$\ddot{y} = \frac{|F|}{m} \cdot \sin(\phi) \rightarrow \frac{|F|}{m} \cdot \phi$$

$$\ddot{\theta} = \frac{|\tau|}{I_{xx}}$$

Anche in questo caso si è adottata la medesima convenzione già descritta per il caso non lineare. Si può inoltre notare come la terza equazione, quella sul calcolo dell'angolo, resti invariata rispetto al set di equazioni non lineari, in quanto già lineare. Analogamente, si è poi implementato tale modello su Simulink.

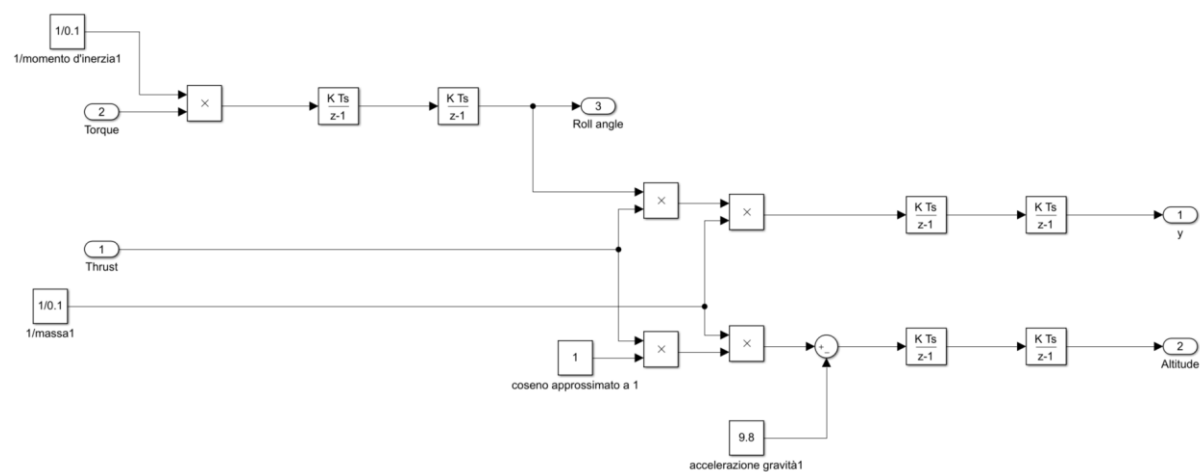


Figure 16 - Modello Simulink del plant del drone linearizzato.

Al fine di rispettare le condizioni che consentono di introdurre tale semplificazione, il sistema deve necessariamente **limitare** le proprie **oscillazioni** per restare fedele all'approssimazione fatta. Si è pertanto imposta una saturazione del valore di comando dell'inclinazione da raggiungere. Limitare l'inclinazione del drone ha un prezzo: **ridurne** la **reattività** e in generale renderlo più **lento** nel

soddisfare il raggiungimento della **componente orizzontale** del **set-point**. Infatti, come spiegato prima, per spostarsi di lato il drone ha bisogno di inclinarsi nella direzione desiderata. Maggiore l'inclinazione, maggiore sarà la forza applicata lungo quella direzione e verso e più velocemente il drone percorrerà la distanza fino al setpoint. Non è detto comunque che il sistema risultante sia necessariamente "troppo" lento, come infatti si è poi verificato sperimentalmente in fase di simulazione. Maggiori dettagli sono forniti al relativo paragrafo.

6.5 - Perché simulare un sistema?

Per quanto si sia visto solo l'inizio della parte sperimentale, può essere già sorta nel lettore un dubbio relativo alla necessità o anche solo all'utilità dello svolgere una **simulazione** preventiva del sistema. Non solo questa, infatti, richiede una certa destrezza nell'utilizzo di **tool professionali** (che per di più spesso non sono disponibili gratuitamente), ma anche una precedente fase di analisi del sistema. Queste operazioni non sono affatto banali e comportano un ingente investimento di tempo e risorse, che potrebbero essere apparentemente risparmiati saltando queste fasi di studio e simulazione.

In realtà, in generale, a saltare le fasi di modellizzazione e simulazione, si otterrebbe il risultato opposto. Senza queste prove, si potrebbero effettuare i test solo sui sistemi fisici una volta realizzati. Grazie alle simulazioni è invece possibile **scoprire criticità, difetti e mancanze** di un **sistema**, ancora prima di aver iniziato a realizzarlo nella pratica. In generale poi il costo dei software utilizzati è nettamente inferiore a quello di eventuali test diretti sul sistema fisico. Vi sono poi casi e sistemi sui quali non è possibile permettersi il lusso di svolgere dei test senza garanzie di successo, non tanto in termini economici, ma di vite umane (velivoli, impianti nucleari, automotive...). Da cui l'esigenza di risolvere quante più criticità possibile ancora prima di avere un prototipo fisico tra le mani.

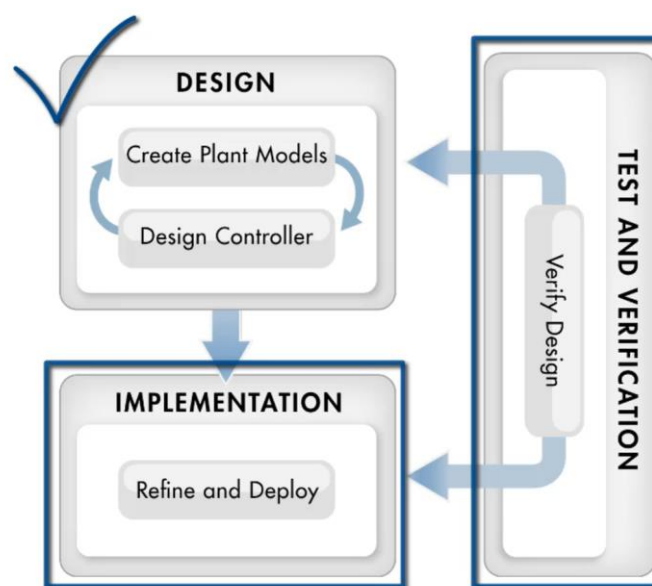


Figure 17 - Ciclo di produzione di un dispositivo.

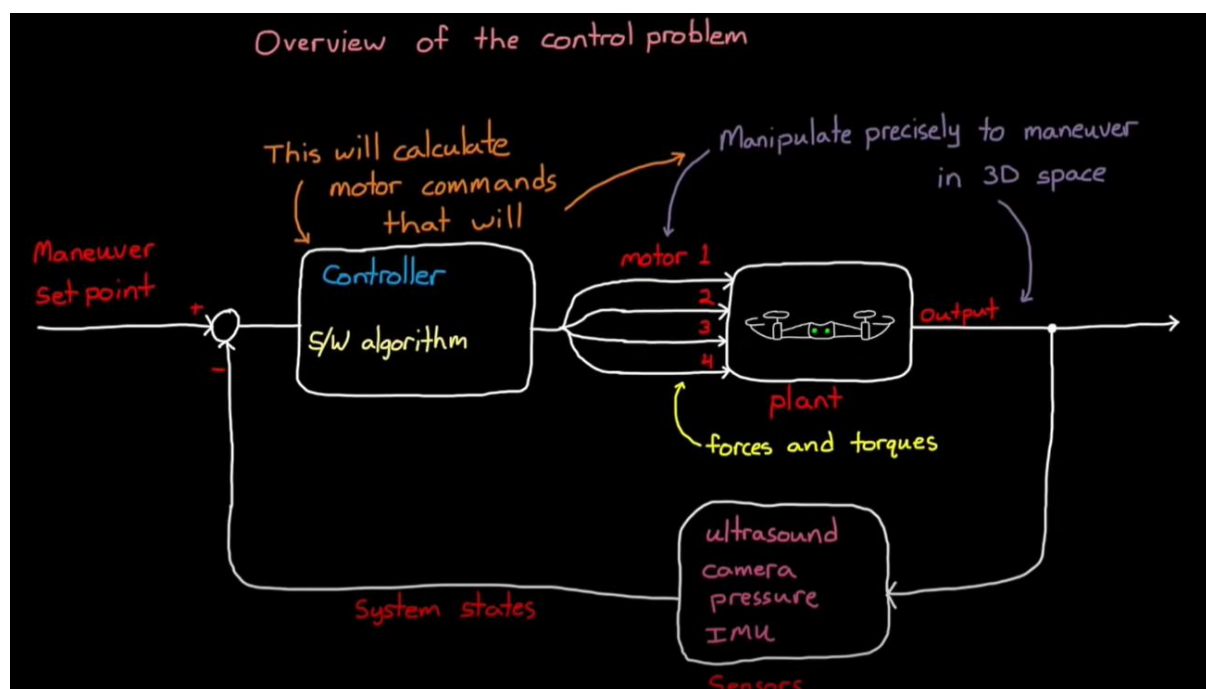
7 - Implementazione del sistema di controllo digitale

Prima di entrare nel merito dell'implementazione del sistema di controllo, è necessario specificare che vista l'eterogeneità dei progetti che era possibile svolgere come parte del corso, e visto il caso in studio ancora generico, non specifico di un determinato modello di drone, non si avevano specifiche numeriche né qualitative imposte dall'esterno da rispettare. Pertanto, si sono scelti dei requisiti piuttosto arbitrari circa le specifiche del sistema di controllo, relative in particolare a prontezza del

sistema e sovraelongazione. Va detto infatti che, normalmente, chi progetta un sistema di controllo per una determinata applicazione, ha un set di specifiche, che si vorrebbero soddisfare. Anche se magari quelle di partenza non sono espressamente relative alla banda passante del sistema, o alla sovraelongazione percentuale massima tollerabile, conoscendo il problema in studio è possibile passare dai requisiti e limiti del sistema alle specifiche del sistema di controllo. Le specifiche, la disponibilità economica e la complessità del sistema consentono inoltre di individuare non solo le specifiche che si vogliono soddisfare, ma anche l'architettura del sistema di controllo che meglio risponde a questi requisiti. Per cui si è pensato di chiedere che la risposta del drone si scosti di solo qualche punto percentuale dal setpoint, nel caso di segnali variabili nel tempo. Mentre per quanto riguarda setpoint costanti, si desidera che questo venga raggiunto senza errori statici. Per quanto concerne l'architettura del sistema di controllo, si è invece pensato di rivolgersi alla florida letteratura presente nel settore, e reperita su internet.

Dalla letteratura disponibile in rete si evince come, in generale, come, per i droni, si impieghino tecniche di controllo avanzate, vista la non linearità del sistema in studio. Sono normalmente impiegati sistemi di controllo **LQR** (*Linear Quadratic Regulator*) e metodi **MPC** (*Model Predictive Control*). Tuttavia, queste tecniche esulano dal corso di Controllo Digitale e ancor più dai precedenti corsi di Sistemi di Controllo. Pertanto, si è cercato di ricondurre il sistema di controllo a una tipologia che fosse già stata trattata durante il corso di studio. Si è deciso di scegliere i **regolatori PID** per la loro versatilità e relativa semplicità, rispetto alle tecniche non lineari nominate sopra. Un ulteriore punto a favore di questa scelta è la sua diffusione piuttosto contenuta nei sistemi di controllo dei droni, almeno stando alla letteratura reperita. Può infatti essere interessante valutare la qualità di questa particolare architettura.

Di seguito è inoltre mostrata l'architettura inizialmente adottata per implementare il modulo di controllo del drone.



Tuttavia, più che agire direttamente sugli angoli dell'assetto di volo del drone, a un pilota umano interessa la posizione del drone. Possiamo quindi derivare la seguente architettura, nella quale gli angoli di rollio e beccheggio, dai quali dipendono posizione orizzontale lungo i due assi x e y , siano controllati da un apposito sottosistema di controllo.

Come descritto nei precedenti paragrafi, i PID hanno un comportamento noto quando si utilizzano su sistemi lineari. Quello in studio, almeno nel caso più generale, non lo è. È però possibile identificare una regione del suo comportamento nel quale questo risulta approssimabile come lineare. Si è pertanto proceduto a limitare la validità del caso in studio e a linearizzare il sistema. L'operazione è già stata descritta nei precedenti paragrafi, per cui non si ritiene opportuno appesantire ulteriormente la lettura riportandola qui sotto. Una volta linearizzato il modello da controllare è possibile implementare un controllore PID. Tuttavia, oltre che alla sola linearizzazione, siamo interessati all'implementazione di un sistema digitale, per i vantaggi che discuteremo nel prossimo paragrafo.

7.1 - Motivazione per la realizzazione di un sistema di controllo digitale piuttosto che non un tempo continuo

Una domanda solo apparentemente banale potrebbe essere “Perché implementare un controllore discreto per un sistema continuo? Non converrebbe tenere il sistema in tempo continuo, eliminando così il bisogno di convertitore AD e DA?” No, non sempre questa scelta risulta vantaggiosa. Anzi, attualmente la maggior parte dei controlli in ambito industriale vengono svolti mediante **metodi digitali**. Vale dunque la pena spendere qualche parola sul perché di tale scelta, e su quali vantaggi essa comporti. Questi sono molteplici, e hanno a che fare con **costi** e **velocità di sviluppo**. Vediamo brevemente alcuni di questi vantaggi. Un sistema di controllo analogico va costruito ad hoc di volta in volta. Con un sistema discreto è invece possibile utilizzare un processore standard, con degli attuatori standard, caricando un programma più o meno standard, e tarando opportunamente i parametri del controllore. Come è facile intuire, questo approccio concorre a migliorare sia tempo di sviluppo e investimento economico. Un sistema digitale è inoltre più facile da replicare, in quanto risente meno di differenze nella realizzazione dei componenti. L'utilizzo di componenti off-the-shelf consente inoltre una maggiore scalabilità dei sistemi. Risulta inoltre più facile da aggiornare e mantenere. Infine, l'eventuale perdita di informazioni dovuta al campionamento e alla quantizzazione è largamente compensata dalla robustezza al rumore elettronico.

Tuttavia, prima di poter far girare un sistema di controllo su processore, occorre avere un modello discreto di tale sistema di controllo. Serve quindi progettare un sistema discreto da capo, oppure discretizzare un modello tempo-continuo, al fine di implementare il controllo mediante algoritmi digitali.

7.2 - Tecniche di discretizzazione

Nel paragrafo precedente si è discusso dei vantaggi delle tecniche di controllo digitale. Per contro, però, i sistemi di controllo analogico presentano un valido ed importante punto di forza: una forte base teorica cresciuta nel corso di più di un secolo e uno studio che ha portato alla scoperta di molteplici teoremi, che le tecniche discrete non possono vantare (almeno non in maniera diretta).

Tuttavia, entro certi limiti, è possibile usufruire dei vantaggi di entrambi gli approcci. Si può infatti progettare prima nel dominio continuo, contando su una robusta teoria dei controlli tempo-continuo. Si possono quindi costruire, calibrare e raffinare i regolatori al fine di soddisfare le specifiche richieste. A questo punto si può poi **discretizzare** il sistema di controllo così progettato, e verificare se il risultato così ottenuto è conforme alle specifiche. In caso contrario, si può agire sul metodo di discretizzazione e sui relativi parametri (tempo di campionamento i primis). Oppure, rivedendo il controllore tempo-continuo e operando una nuova discretizzazione, in maniera iterativa, fino a che non trovo un risultato soddisfacente. O ancora, dopo una prima discretizzazione si prosegue poi col raffinare il controllo direttamente nel dominio discreto. La **sintesi per discretizzazione** consiste quindi nell'effettuare il progetto del controllore (o del filtro) secondo i metodi propri dei sistemi a tempo continuo, e nel cercare poi un “corrispettivo digitale” in grado di approssimare al meglio possibile il comportamento

del controllore/filtro analogico. Vi sono diversi approcci attraverso i quali convertire una funzione di trasferimento analogica in una funzione di trasferimento digitale che ne “emuli” il comportamento:

1. Approssimazione tramite integrazione numerica

Sono i metodi di discretizzazione maggiormente utilizzati nella pratica (in particolare uno di questi, il **Metodo di Tustin**). All'atto pratico, tali metodi (ne vengono presentati tre) si traducono in una **sostituzione di variabile $s:=s(z)$** da applicare alla funzione di trasferimento $C(s)$ del controllore onde ricavare una corrispondente versione a tempo discreto $C(z)$. I metodi che presentiamo sono i seguenti:

1. **Metodo delle differenze in avanti**
2. **Metodo delle differenze all'indietro**
3. **Metodo di Tustin** (o della trasformazione bilineare, o ancora “metodo dei trapezi”)

| Metodo | Sostituzione di variabile |
|--------------------------------------|-----------------------------------------|
| Metodo delle differenze in avanti | $s = \frac{1}{T_c} (z - 1)$ |
| Metodo delle differenze all'indietro | $s = \frac{1}{T_c} \frac{z - 1}{z}$ |
| Metodo di Tustin | $s = \frac{2}{T_c} \frac{z - 1}{z + 1}$ |

2. Approssimazione tramite invarianza della risposta

Un differente approccio alla discretizzazione mira a ricercare un regolatore discreto $C(z)$ che conservi le caratteristiche della risposta del regolatore continuo $C(s)$ a determinati segnali di ingresso. In particolare, si fa riferimento alla risposta al gradino, $\delta_{-1}(t)$. Tale metodo di discretizzazione conduce esattamente alla stessa relazione che modella un processo a tempo continuo preceduto da un ricostruttore ZOH.

$$C(z) = \frac{z-1}{z} Z \left\{ \left[\mathcal{L}^{-1} \left(\frac{C(s)}{s} \right) \right]_{t=kT_c} \right\}$$

Il metodo dell'invarianza della risposta al gradino preserva la stabilità del controllore discretizzato. Pertanto, poli stabili di $C(s)$ vengono mappati in poli stabili di $C(z)$. La risposta in frequenza di $C(z)$ risulta tuttavia alquanto distorta rispetto a quella di $C(s)$, e, per effetto del termine $1/s$, si ha una certa attenuazione alle alte frequenze.

Alla luce delle considerazioni appena viste, si è pertanto **scelto** di adottare il **metodo di Tustin**, il quale è solitamente, anche il più diffuso.

All'atto pratico, le operazioni di campionamento e ricostruzione dei segnali sono svolte in automatico dal motore di simulazione di Simulink (il solver del sistema). Il modello è stato infatti implementato utilizzando MATLAB e Simulink, per i vantaggi discussi qualche paragrafo sopra.

7.3 - Modello di controllo realizzato

Per questioni di complessità si è scelto di fare riferimento all'architettura standard di regolatore a retroazione unitaria. Di seguito è riportata l'architettura di riferimento.

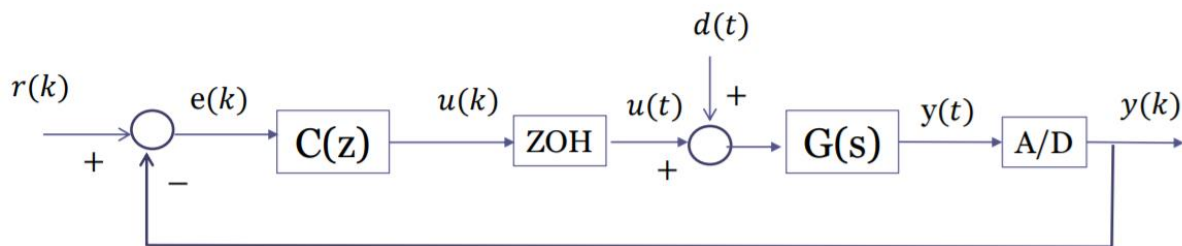


Figure 18 - Schema generale sistema di controllo digitale applicato a un processo tempo-continuo.

7.3.1 - Scelta del periodo di campionamento

Un aspetto delicato della simulazione tempo-discreto e implementazione di un sistema di controllo è la **scelta del periodo di campionamento**. Da questo dipende infatti la risoluzione temporale con la quale il simulatore andrà a ricalcolare i parametri del modello, o ad apportare correzioni allo stato del sistema fisico.

Al fine di garantire un sistema di controllo sufficientemente accurato rispetto ai risultati reali, è utile scegliere un tempo di campionamento che sia comparabile con quello con il quale il drone reale andrà a ricalcolare il proprio assetto di volo.

I moderni velivoli effettuano questi calcoli e le successive correzioni anche centinaia di volte al secondo. Per alcuni velivoli militari sono intrinsecamente instabili, sia a causa di alcune scelte progettuali relative all'implementazione di caratteristiche che rendano l'aereo stealth, sia per aumentare la banda passante e rendere il sistema più reattivo ai comandi del pilota, specie in situazioni critiche di combattimento. Tutto ciò fa sì che anche compiere le normali operazioni di decollo e di volo a velocità di crociera risulti impossibile senza un adeguato sistema di controllo. In questi casi si arriva ad effettuare diverse migliaia di ri-configurazioni dell'assetto di volo al secondo.

Il sistema di controllo apporta in automatico delle correzioni all'assetto di volo, senza che il pilota debba necessariamente conoscere l'entità di tali correzioni, garantendo il risultato inteso dal pilota stesso. Questa tecnica prende il nome di "fly by wire", nella quale i comandi vengono inviati dalla cabina di pilotaggio agli attuatori mediante impulsi elettronici trasmessi via cavo. Come è facile immaginare, anche le origini di questa innovazione affondano le loro radici nel settore militare. Si distingue dalle più vecchie tecniche di controllo, nelle quali era il pilota stesso a imprimere un'azione sulle superfici di controllo, mediante cavi, pulegge e in generale altri componenti puramente meccanici.

7.3.2 – Processo di realizzazione

Siamo partiti col modellizzare il sistema tempo continuo, non lineare, per poi introdurre delle semplificazioni, atte a linearizzare il modello intorno a un punto di lavoro. Successivamente, ci si è concentrati sul modello tempo linearizzato tempo-discreto. I due modelli continui sono stati presi come riferimento, sia in termini di struttura che per confrontare la risposta del sistema linearizzato.

A livello di progettazione si è inoltre sfruttato un approccio modulare che consentisse di riutilizzare agevolmente porzione di schema per le diverse versioni del modello. Come valore aggiunto, il sistema risulta più leggibile e ordinato.

7.3.3 – Architettura del sistema di controllo realizzato

Il sistema di controllo consiste complessivamente di tre anelli di retroazione, uno per ciascuna delle variabili di stato. In particolare, possiamo individuare il primo, impiegato per la retroazione del valore dell'altitudine del drone, che consente di regolare la spinta che i rotori complessivamente imprimono al drone, e che tenderebbe a far procedere il drone in direzione perpendicolare al piano in cui giace.

Come abbiamo potuto osservare, la spinta totale del drone non è tutta diretta lungo l'asse z , ma viene usata in parte anche per consentire le traslazioni orizzontali, regolando l'inclinazione del drone. Pertanto, per poter raggiungere un certo set-point orizzontale, occorre passare per l'inclinazione del drone. Si ha quindi bisogno di un anello di controllo per la componente orizzontale dello spostamento, e di uno per controllare l'angolo di inclinazione del drone che consente di ottenere lo spostamento desiderato. Questa redistribuzione dei contributi della forza totale sui due assi fa sì che lungo Z ci sia una diminuzione della spinta, che tenderebbe a far abbassare il drone. Fortunatamente, il primo PID interviene a compensare questo calo, incrementando il modulo della forza totale. Questo potrebbe comportare una sovraelongazione anche lungo la componente Y . Il secondo loop potrebbe quindi richiedere una diminuzione dell'inclinazione, con conseguente riduzione della spinta totale richiesta. Tarando opportunamente i diversi loop di controllo in maniera coordinata, è possibile far convergere queste correzioni in un tempo finito, garantendo un sistema di controllo stabile, reattivo e con oscillazioni contenute.

È inoltre doveroso specificare che si sta assumendo che i motori siano in grado di rispondere prontamente e in maniera fedele al comando impartito dai controllori.

7.3 – Giustificazione utilizzo tecniche di controllo lineare su plant non lineare [9] [10]

In letteratura si trovano molteplici implementazioni di sistemi di controllo per diversi modelli di droni. Tutti impiegano architetture e tecniche più esotiche rispetto a quelle viste finora nel corso dei nostri studi. Si è pertanto pensato che potrebbe essere interessante studiare se un più semplice PID possa essere sufficiente ad assolvere a questo compito.

In generale, è possibile utilizzare un controllore lineare per un sistema non-lineare. In particolare, se il sistema di controllo implementa una risposta integrale, può essere una valida scelta per lo stabilizzare un sistema non-lineare, specialmente tunando opportunamente i parametri di controllo. La principale limitazione sta nel fatto che un regolatore PID ha una risposta lineare e dei coefficienti costanti. Per un sistema lineare questo comportamento è adeguato. Per un sistema che ha diversi comportamenti in diverse regioni di funzionamento (un sistema non-lineare), questo controllo potrebbe essere sub-ottimale. Questa non-ottimalità può variare da trascurabile a così grave da rendere il sistema instabile.

Un approccio intermedio tra PID e tecniche più avanzate è quella di usare un PID a guadagno regolabile, capace di variare i propri coefficienti in funzione della regione di funzionamento del plant. Sarebbe come avere una disposizione una famiglia di PID costanti, dei quali viene scelto di volta in volta quello più adatto per la condizione attuale del plant.

In sostanza, si tratta in generale di valutare se il sistema di controllo-PID risulta adeguato caso per caso.

Va infine detto che tutti i sistemi reali presentano un certo grado di non-linearità. Pertanto, anche quelli su cui i PID vengono regolarmente sfruttati con successo, sono non-lineari. Tuttavia, questo non sempre causa problemi insormontabili. Anzi, i PID vengono ampiamente utilizzati in moltissimi processi industriali. Si è proceduto quindi a verificare anche se questo ambito potesse beneficiare di tale sistema di controllo e con quali riscontri.

7.3 - Calibrazione dei coefficienti del controllore PID

Di seguito vengono sinteticamente descritti i principali effetti di ciascuna delle tre azioni di controllo svolte dal PID. Per una questione di brevità non ci si dilungherà su questi aspetti, né verrà fornita una trattazione rigorosa che potrebbe appesantire l'elaborato.

1. P - contributo proporzionale

Il suo contributo è proporzionale all'errore, cioè alla differenza tra posizione attuale e setpoint (o target). Un contributo proporzionale P maggiore vuol dire sistemi più veloci, ma eventualmente meno stabile.

2. I - contributo integrativo

È il più lento dei tre contributi per quanto riguarda la convergenza a regime. Per contro, è quello che garantisce la sovralongazione minore (non ne introduce affatto). Riduce effetti del rumore in ingresso sull'uscita. Il contributo integrale I garantisce il raggiungimento del setpoint a regime (in assenza di disturbi particolari). Garantisce inoltre la reiezione totale dei disturbi costanti.

3. D - contributo derivativo

Garantisce la convergenza a regime più veloce e reattivo. Tuttavia, è quello che può causare maggiore sovralongazione e oscillazioni (*ripple*). Può anche portare a problemi di instabilità. Risente molto di rumore e disturbi variabili nel tempo (in un certo senso amplifica l'effetto differenze tra campioni successivi, quindi esalta la variabilità del segnale). Tuttavia, nel caso in studio si sta ipotizzando un sistema ideale, nel quale il rumore è nullo.

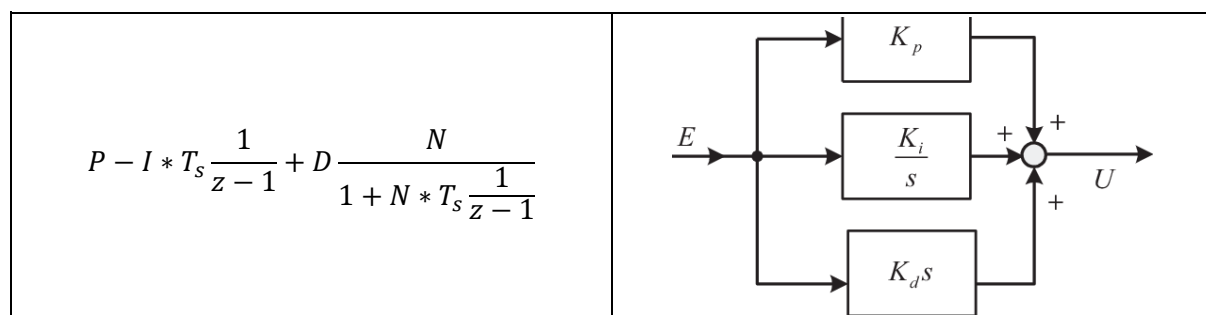


Figure 19 - Destra: funzione di trasferimento del generico PID. Sinistra: schema a blocchi della funzione di trasferimento di un PID.

Nel caso specifico è stata utilizzato un approccio ibrido per la calibrazione a partire dai coefficienti determinati tramite la funzione di autotuning. Mediante il **luogo delle radici** è stato studiato il variare della posizione dei poli a ciclo chiuso. Mediante diverse iterazioni con valori di I e D diversi si è determinata una configurazione del sistema di controllo soddisfacente.

Può essere utile inoltre spendere qualche parola sull'effetto dei disturbi e su come contenerlo, Condizione sufficiente per la reiezione a regime di un disturbo è che la funzione di trasferimento G del controllore contenga al denominatore il polinomio generatore del disturbo. Più in generale, è sufficiente che il sistema a ciclo aperto (la cascata di controllore e plant) contenga al denominatore il polinomio generatore del disturbo. Questo enunciato prende il nome di **principio del modello interno**. Tornando al modello del drone descritto, la forza di gravità agisce come un gradino, la cui trasformata di Laplace è $1/s$. Ma questo modello è contenuto nel controllore (grazie all'azione integrante, ma anche nel modello del plant sono presenti due integratori). Per cui a regime si dovrebbe avere una reiezione totale del disturbo dalla forza di gravità sull'uscita. Questa caratteristica prende il nome di **precisione statica**, a regime. Vedremo in seguito come è infatti raggiunto il setpoint senza alcuno scarto, a regime. Per il momento però può essere opportuno descrivere più nel dettaglio alcuni aspetti relativi ai disturbi che possono agire su un drone.

7.4 - Disturbi

Ai fini della trattazione svolta, si è preferito non appesantire il lavoro con analisi approfondite sulla **presenza di disturbi**, che infatti nel modello sviluppato sono quasi del tutto assenti, eccetto per la forza di gravità. Vale comunque la pena approfondire il discorso.

Oltre alla spinta esercitata dai rotori, sul drone agisce anche la forza di attrazione gravitazionale. Questa può essere modellizzata come un disturbo sul sistema. In particolare, possiamo identificare alcune proprietà specifiche di questo disturbo. Primo, agisce sempre nella stessa direzione, aspetto non banale dato che stiamo comunque parlando di un sistema che presenta due gradi di libertà spaziale e uno rotazionale. Ad esempio, un altro disturbo piuttosto comune, il vento, può soffiare da direzioni e inclinazioni diverse. Il secondo aspetto interessante è che anche l'intensità di questo disturbo è costante, e noto. In particolare, sulla terra, vale $9,8 \text{ m/s}^2$. Anche questo aspetto non è da dare per scontato in quanto, sempre il vento, può soffiare con intensità variabile. Il caso estremo è quello delle raffiche, fenomeno nel quale la velocità del vento cambia in maniera frequente e brusca, rendendo complicato il volo.

Oltre alla sopracitata forza di gravità, in generale, mentre il drone è in volo, questo risente degli effetti del **vento**, come si diceva pocanzi. Questo che può essere più o meno costante, sia in termini di intensità che di direzione. È importante quindi che il velivolo sia in grado di rispondere ai cambiamenti del vento in maniera tempestiva (nessun gioco di parole inteso), mantenendo la posizione o traiettoria desiderata, correggendo opportunamente l'assetto di volo e la spinta fornita dai motori.

Un'altra non idealità presente in un sistema reale è il cosiddetto **effetto suolo** (in inglese *Ground Effect*). Con questo termine si indica la conseguenza della **vicinanza** del **suolo** rispetto al velivolo, e l'effetto che provoca. Nel caso di elicotteri e droni, si ha che a parità di potenza fornita e angolo di attacco delle pale del velivolo, si ha la **formazione** di un **cuscinio d'aria** che tende a **sostenere** il velivolo più di quanto non avverrebbe, a parità di spinta, lontano dal terreno. Tale effetto consente quindi di risparmiare potenza quando il drone è in prossimità del suolo, specie in fase di decollo e atterraggio. Presenta però anche un problema, in quanto una potenza relativamente bassa può essere sufficiente per far staccare il drone dal suolo, ma non per sostenerlo in volo in maniera duratura. Senza quindi un tempestivo incremento, c'è il rischio concreto che il **mezzo ricada al suolo** subito dopo il decollo, con conseguenti eventuali danni meccanici, in particolare alla parte inferiore della fusoliera. Per di più, tornando in prossimità del suolo, si ripresenta l'effetto suolo, e come conseguenza, se non opportunamente gestito, si può andare incontro a una serie di "rimbalzi" al suolo, con conseguente accumulo di urti e possibili danneggiamenti del mezzo. Un'alternativa naïve sarebbe quella di fornire da subito la potenza necessaria al drone per stare in volo, in modo che terminato l'effetto suolo, questo possa continuare a stare in volo. Ma anche questa opzione presenta degli svantaggi: tale potenza, sommata al *ground effect*, potrebbe comportare delle accelerazioni verticali eccessive in fase di decollo, con stress dei componenti meccanici e sollecitazioni eccessive per eventuali carichi o peggio ancora persone, se l'aeromobile è predisposta al trasporto di piloti e passeggeri.

Pertanto, è necessario un opportuno sistema di controllo, sia esso implementato mediante un bravo pilota in grado di tener conto di tale effetto, oppure di un algoritmo che in automatico apporti delle correzioni ai comandi impartiti da un operatore meno tecnico.

Può essere interessante che questo stesso effetto è particolarmente importante anche nel settore automobilistico, in particolare delle auto da corsa, dove è noto con lo stesso nome. Tuttavia, in questo settore, viene usato al contrario: le auto vengono progettate come ali rovesciate, in modo da, primo, evitare che la macchina si stacchi dal suolo, o perda aderenza. Secondo: aumentare attivamente la tenuta di strada del veicolo alle alte velocità, consentendo quindi maggiore aderenza (grip) al manto stradale, e quindi di poter, ad esempio, affrontare le curve a velocità superiori di quanto non sarebbe possibile altrimenti.

8 - Sistema complessivo

Nell'architettura del sistema complessivo possiamo identificare tre macro-blocchi. Partendo da sinistra verso destra troviamo il sistema di controllo, il sistema da controllare (*plant*) e infine il blocco relativo alla visualizzazione dei dati.

1. **Plant.** Rappresenta e **modellizza** la **fisica** del **drone**. Si tratta di un **modello tempo-continuo**, in quanto le leggi della fisica, almeno per quanto concerne la meccanica newtoniana, determinano un'evoluzione dello stato continua nel tempo, che non procede a balzi, come invece accade nella meccanica quantistica.
2. **Controllo.** Contiene l'insieme degli algoritmi che consentono di **governare** il **comportamento** del **drone** e di **imporgli** una **traiettoria desiderata** (entro certi limiti). Contrariamente al plant, è un sistema tempo-discreto, in quanto con questo si vuole modellizzare il comportamento di un controllore digitale. È quindi necessario inserire **sample&hold** e **AD converter** sono necessari per un sistema di controllo discreto che agisce su un plant analogico. Nel nostro modello non sono presenti in maniera esplicita, ma sono "nascosti" all'interno dei regolatori PID discreti. Questi infatti, in maniera implicita, effettuano il campionamento del segnale.
3. Blocco di **visualizzazione**. Contiene diversi (due) metodi di rappresentazione dei risultati della simulazione. Il primo e più semplice consiste nell'utilizzo di uno scope per rappresentare le tra variabili di interesse (posizione x, y e angolo di rollio), al variare del tempo, ciascuna confrontata col proprio set-point.

Il secondo modo di visualizzazione è più articolato ed è stato preso in prestito da un progetto di esempio contenuto nel **toolbox Aerospace Blockset**. Questo introduce tutta una serie di blocchi utili per la modellizzazione e il **controllo** di **aerei**, **velivoli** e **satelliti**. Il progetto al quale si è fatto riferimento è "**asbQuadcopter**". Questo modella in maniera dettagliata e formale un drone e il suo sistema di controllo, basandosi sul lavoro di due professori Sertac Karaman e Fabian Riether [8]. Questa rappresentazione fa uso di blocchi che implementano funzionalità di alto livello, oltre lo scopo di questa trattazione. Tuttavia, mette altresì a disposizione un'efficace rappresentazione grafica del drone. È infatti possibile aprire una finestra, nella quale viene rappresentato uno scorcio di una città, all'interno della quale è possibile far volare un drone, passando posizione e assetto di volo in ingresso al blocco. Copiando il relativo modulo nel nostro design, e interfacciando opportunamente i segnali è possibile far eseguire al drone la sequenza di movimenti e spostamenti graficato con il precedente metodo.

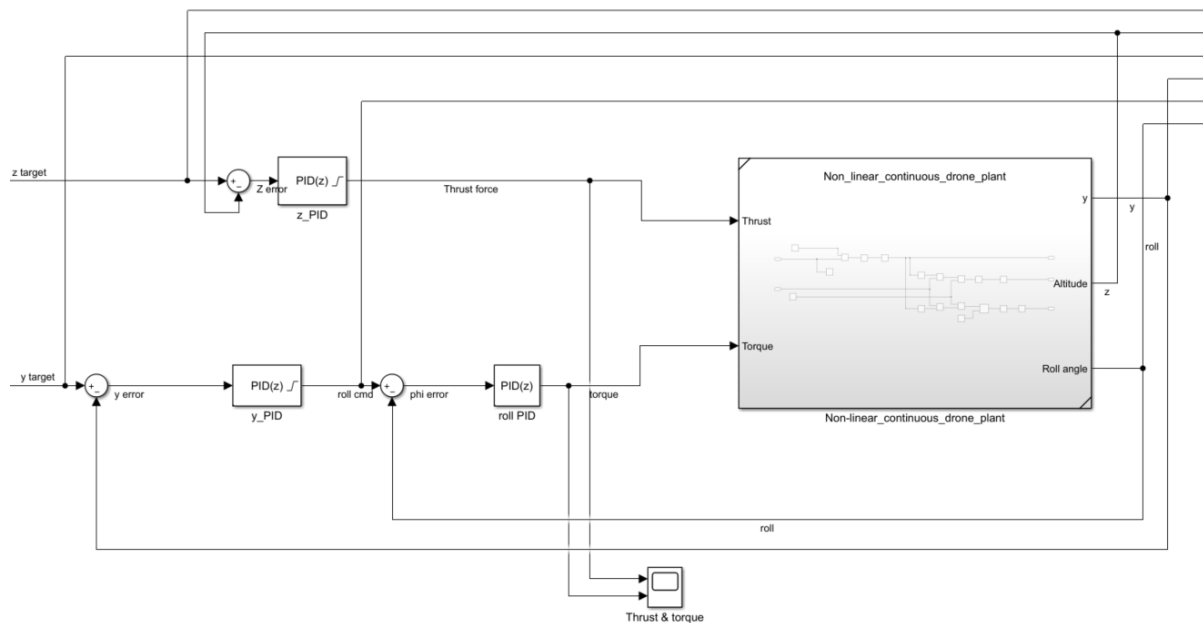


Figure 20 - Sistema complessivo. Il plant è racchiuso all'interno del blocco rettangolare sulla destra. Le uscite confluiscono nel blocco di visualizzazione (non presente in figura).

9 - Simulazione

Si è quindi passati a simulare il sistema complessivo così ottenuto.

9.1 - Sistema 2D

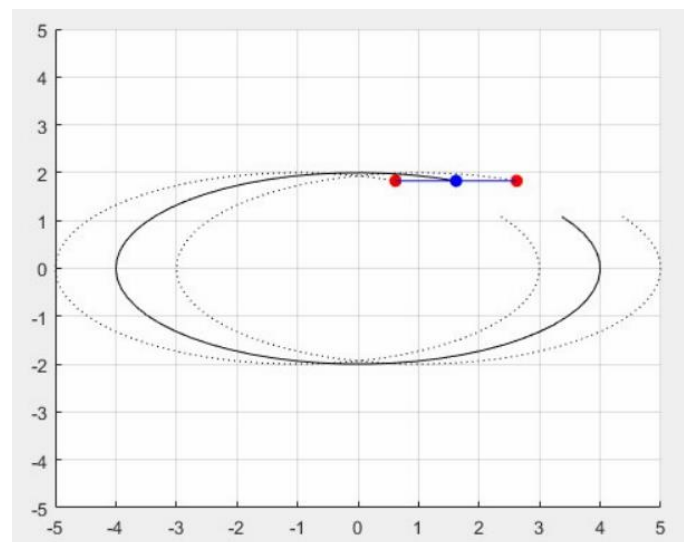


Figure 21 - Rappresentazione stilizzata del drone che si muove su un piano verticale.

Di seguito sono riportati i grafici rappresentanti l'andamento nel tempo delle tre grandezze di interesse (posizione lungo z , posizione lungo y e angolo di rollio) a diversi stimoli (gradino, in primo luogo, poi segnale sinusoidale). Per ciascuna di queste viene riportato il segnale di riferimento (o set-point) e il valore effettivo del segnale stesso. Se il sistema è stato progettato correttamente, lo scostamento tra questi due dovrebbe essere piccolo, trascurabile. Al più, si potrebbe manifestare qualche non idealità durante i transitori. Di seguito sono riportati i relativi grafici.

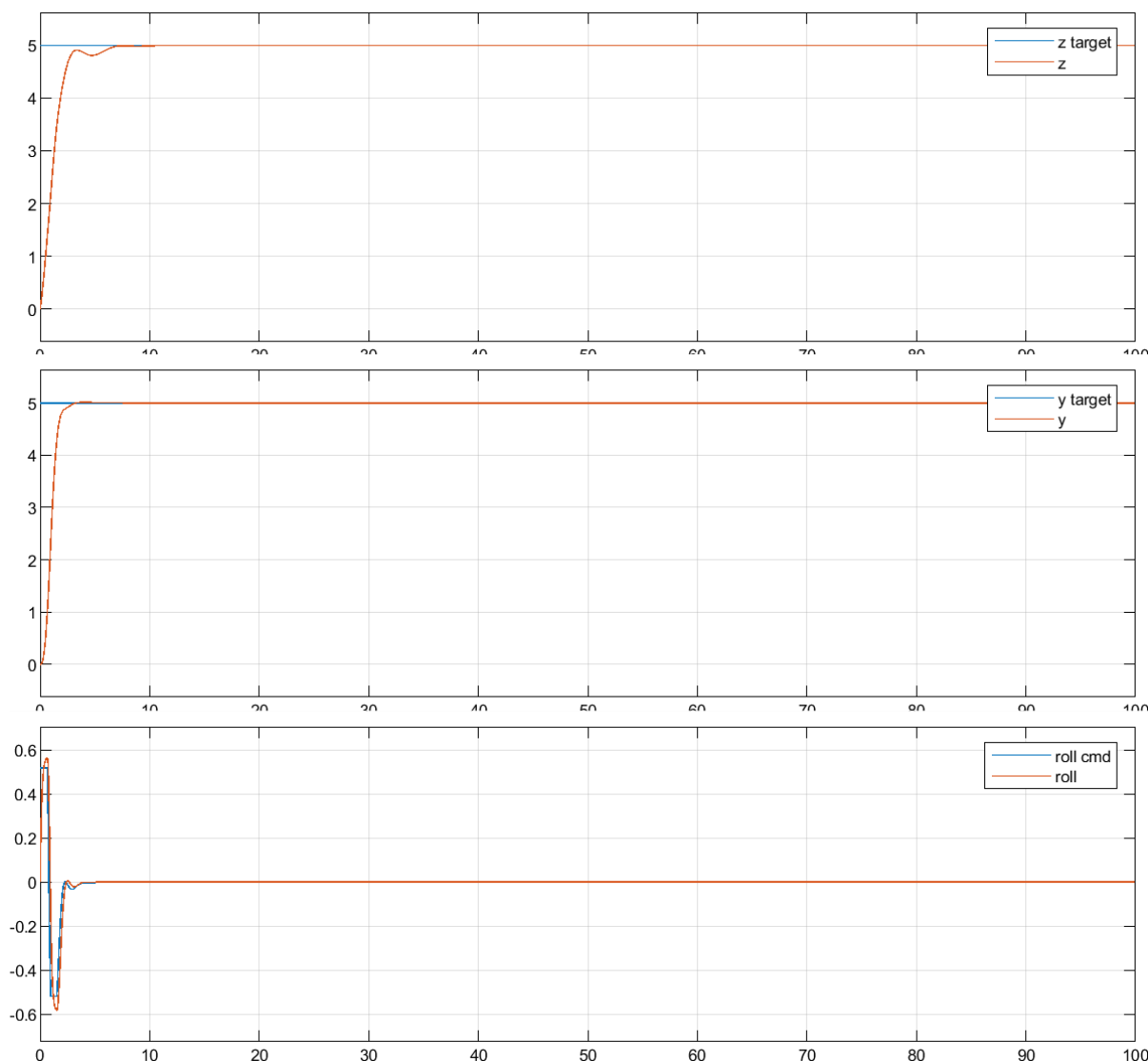


Figure 22 – Risposta al gradino. Evoluzione del sistema (in arancio) con set point costante (in blu). Dopo un primo transitorio piuttosto breve, la risposta converge al valore desiderato con errore a regime nullo.

È possibile osservare come il sistema evolva fino al raggiungimento della condizione desiderata, senza alcun errore statico a regime. Quest'ultimo risultato si deve alla presenza del blocco integrale all'interno del PID, il quale consente la reiezione totale dei disturbi costanti a regime, come conseguenza del principio del modello interno.

Il risultato ottenuto ci porta a dire che il sistema è stabile per tale ingresso, ma ancora non possiamo dire che il sistema sia stabile in senso assoluto.

In generale, il set-point è, come il nome lascia intendere, un punto statico, che si vuole raggiungere. Tuttavia, se si assume il sistema di controllo, e il plant controllato, sufficientemente veloci e robusti, si può pensare di far cambiare il set-point nel tempo, in modo che il sistema complessivo segua una traiettoria desiderata nello spazio degli stati. Nel nostro caso, questa è anche la traiettoria del drone nello spazio. È quindi stata definita una sequenza di prova che corrisponde a una forma simile al simbolo di infinito ∞ , che il drone percorre alcune volte consecutive. Il grafico mostra in giallo il set-point al variare del tempo, in blu lo stato (posizione e assetto di volo) del drone. Il segnale rappresentato in giallo potrebbe cambiare in maniera continua, arbitraria, poiché, come spiegato prima, i PID sono stati opportunamente configurati in modo da effettuare un campionamento del

segnale. L'unico requisito è che il segnale vari in maniera sufficientemente lenta affinché i PID possano far correggere lo stato del sistema prima del successivo cambiamento.

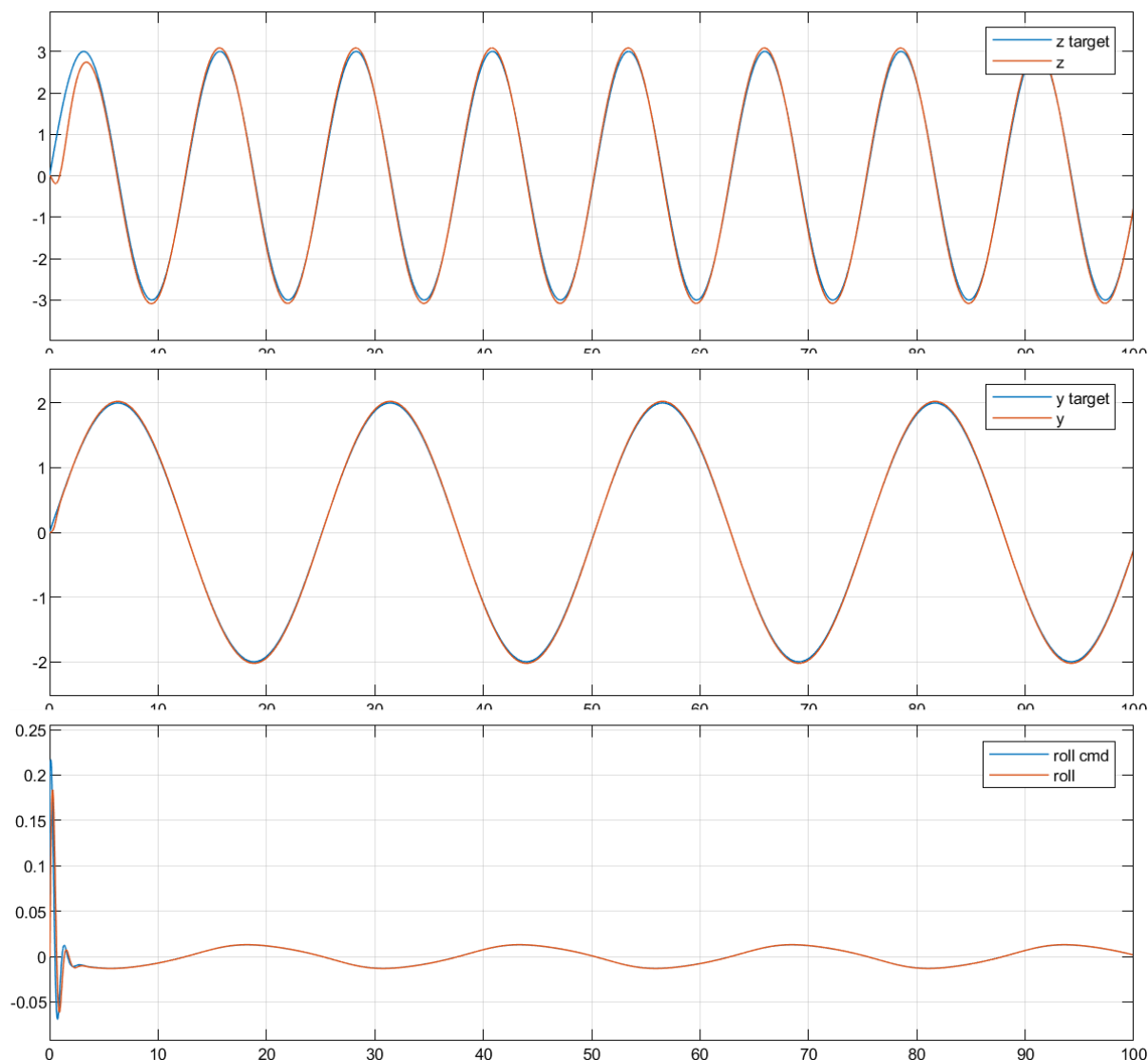


Figure 23 - Setpoint e risposta del drone al variare del tempo. Nell'istante iniziale il drone è sospeso in aria, e su di esso agisce la sola forza peso, imponiamo un setpoint sinusoidale nelle due componenti spaziali in blu). In arancio è riportata la posizione.

Anche questo risultato rafforza l'ipotesi precedente e cioè che il sistema sia stabile, ma ancora non abbiamo una conferma definitiva.

Un risultato comunque notevole è la conferma dell'ipotesi iniziale che consente di effettuare l'approssimazione di piccole oscillazioni. Questa, ricordiamo, ci ha consentito di linearizzare il sistema rispetto alle funzioni trigonometriche originariamente presenti. Il grafico sopra, infatti, mostra come l'angolo non ecceda mai i 0,2 radianti (circa 12°). Ma ancora, dopo un primissimo transitorio, l'inclinazione del drone rimane compresa entro 0,02 radianti (circa $1,2^\circ$). Si tratta quindi di angoli piuttosto piccoli che giustificano l'approssimazione introdotta.

Sono stati poi introdotti dei blocchi Simulink per caratterizzare il sistema dal punto di vista della risposta in frequenza. In particolare, scegliendo opportunamente i segnali da considerare come ingresso e come uscita, si è potuto tracciare il diagramma di Bode del sistema, sia a ciclo chiuso che a ciclo aperto. Ovviamente il diagramma di Bode è uno strumento utile quando si parla di sistemi SISO

(Single input Single Output), mentre nel nostro caso abbiamo un sistema MIMO (Multiple Input Multiple Output) anche quando consideriamo la versione 2D. Abbiamo infatti due diversi ingressi (forza e momento torcente). In uscita abbiamo le due coordinate spaziali y e z , e l'angolo di inclinazione del drone. È pertanto importante identificare sistema a ciclo chiuso si vuole studiare, in altre parole determinare quale anello di retroazione considerare, e scegliere quindi il segnale di input e quello di output di tale sistema. Si è scelto in particolare, come prova, di studiare l'anello relativo alla posizione z (uscita), che dipende in maniera stretta dalla spinta F impressa dal drone. Il blocco inserito nel progetto effettua una sweep in frequenza, riportando sul grafico la risposta del sistema nel range predefinito. In questo modo abbiamo potuto determinare anche i margini di stabilità del sistema.

Per rendere più diretta l'**individuazione** di **poli** e **zeri** del sistema complessivo è stato inserito un ulteriore blocco Simulink che assolve a tale funzione. Anche questa rappresentazione conferma però quanto precedentemente trovato dai precedenti risultati: il sistema a ciclo chiuso è stabile, non essendoci alcun polo a parte reale positiva, né molteplici poli nell'origine.

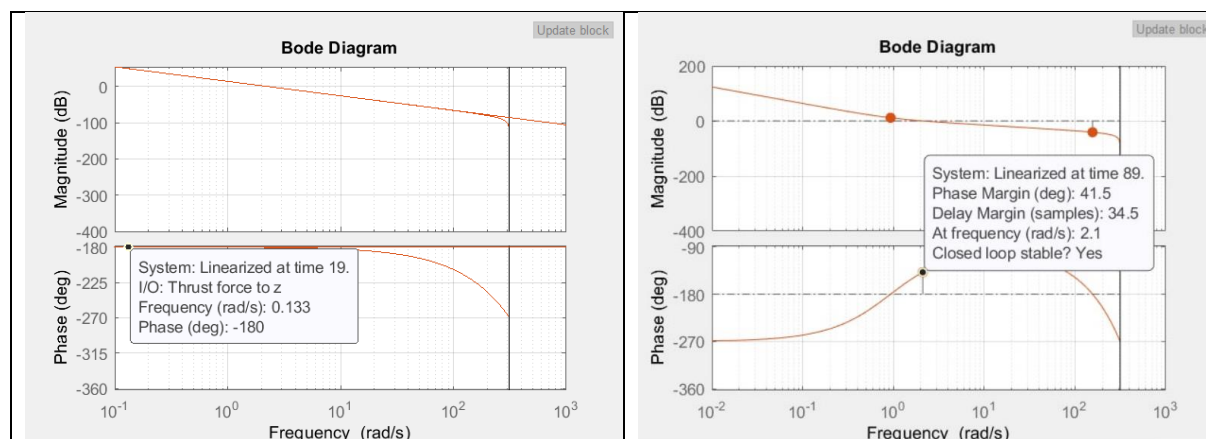


Figure 24 - Diagramma di Bode del sottosistema della posizione Z . A sinistra, sistema a ciclo aperto. A destra, sistema a ciclo chiuso.

Tuttavia, questo tipo di verifica della stabilità, per quanto automatizzato, presenta alcune pecche. Primo descrive un sistema linearizzato dal simulatore, operazione sulla quale non si ha un controllo fine. In secondo luogo, non mette in luce le tecniche studiate il corso di Controllo Digitale, in quanto studia il sistema a tempo continuo (che risulta dalla ricostruzione del segnale). Pertanto, per una verifica formale ed esaustiva, si rimanda al capitolo successivo. In questo si individueranno e tratteranno le espressioni analitiche dei diversi sottosistemi, e si dimostrerà la stabilità dei sistemi a ciclo chiuso.

Infine, oltre che mediante i grafici soprariportati, è possibile visualizzare l'evoluzione del drone mediante un'apposita finestra che mostra una rappresentazione 3D del drone, che si muove fedele ai parametri calcolati dal sistema a ciclo chiuso. Questo blocco di visualizzazione, infatti, riceve in ingresso solo la posizione e l'inclinazione del drone, mentre questi parametri fisici vengono elaborati dal sistema oggetto di studio, realizzato in questo lavoro. Di seguito è mostrato un frame della suddetta animazione.

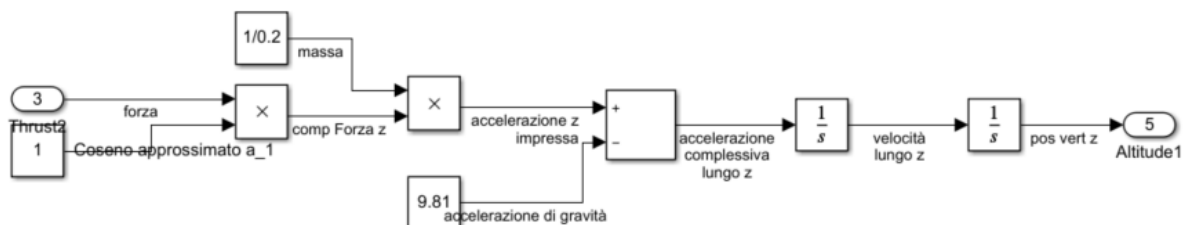


Figure 25 - Visualizzazione 3D dell'evoluzione del drone

9.2.1 – Risposta analitica del sistema

Di seguito sono riportate le **funzioni di trasferimento** (FdT) dei diversi sottosistemi ingresso-uscita che è possibile individuare all'interno del modello linearizzato. Si è partiti dal modellare il plant come modello in variabili di stato. Sono poi state individuate le rispettive FdT e poi sono state caricate su MATLAB, e da qui rese in forma discreta. Per ciascuna sono poi stati calcolati la funzione ad anello chiuso e i relativi margini di fase e posizione di zeri e poli, al fine di caratterizzare la stabilità del sistema. Per ciascuna è stata poi graficata la risposta in frequenza e quella al gradino, al fine di analizzare i parametri dinamici del sistema.

1. **Catena quota z a catena diretta** (drone senza sistema di controllo, relazione tra la forza applicata lungo l'asse z e la posizione risultante)



a. Funzione di trasferimento tempo-continuo

```
fdtPlantZ =
    5
    ---
    s^2
Continuous-time transfer function.
```

Come si può notare, la funzione di trasferimento del sistema in catena diretta presenta non uno ma due poli nell'origine. Ne risulta quindi un sistema instabile (per una trattazione più approfondita si rimanda alla sezione successiva). Contemporaneamente, questo ci garantisce la reiezione totale di disturbi costanti (e a rampa) a regime, in accordo col principio del modello interno. Infatti, il processo contiene di per sé un (due) polo nell'origine; quindi, il sistema di controllo sarà sempre almeno di tipo 1, e i sistemi di tipo 1 garantiscono l'azzeramento dell'errore a regime per un set point costante indipendentemente dal valore del guadagno del controllore (e del processo). Quindi la specifica sul raggiungimento del setpoint sarà automaticamente soddisfatta da qualunque scelta si adotti per il controllore (purché, ovviamente, quest'ultimo garantisca la stabilità a ciclo chiuso del sistema di

controllo). Un discorso simile si può inoltre fare per altri sottosistemi descritti di seguito, i quali presentano a loro volta dei poli nell'origine.

b. Funzione di trasferimento tempo-discreto (discretizzato con Tustin, tempo di campionamento = 0.01s)

```
fdtPlantZDisc =

    0.000125 z^2 + 0.00025 z + 0.000125
    -----
                z^2 - 2 z + 1

Sample time: 0.01 seconds
Discrete-time transfer function.
```

2. Catena posizione z a catena diretta, con ingresso accelerazione di gravità g



a. Funzione di trasferimento tempo-continuo

```
fdtPlantG =

    9.81
    ----
    s^2

Continuous-time transfer function.
```

b. Funzione di trasferimento tempo-discreto (discretizzato con Tustin, tempo di campionamento = 0.01s)

```
fdtPlantGDisc =

    0.0004905 z + 0.0004905
    -----
                z^2 - 2 z + 1

Sample time: 0.01 seconds
Discrete-time transfer function.
```

3. Catena posizione y a catena diretta (drone senza sistema di controllo)



a. Funzione di trasferimento tempo-continuo

```
fdtPlanty =

    5
    ---
    s^2

Continuous-time transfer function.
```

b. Funzione di trasferimento tempo-discreto (discretizzato con Tustin, tempo di campionamento = 0.01s)

```

fdtPlantyDisc =

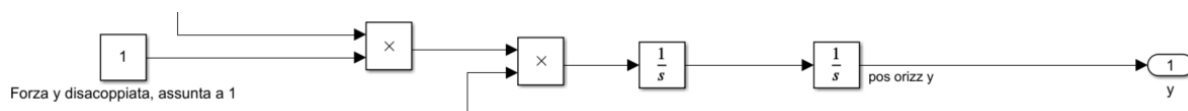
      0.000125 z^2 + 0.00025 z + 0.000125
      -----
              z^2 - 2 z + 1

Sample time: 0.01 seconds
Discrete-time transfer function.

```

Nelle funzioni di trasferimento relative alla posizione orizzontale y appena descritte, manca il contributo esplicito della forza applicata. Questo perché la forza applicata è, a meno di un fattore moltiplicativo pari all'inverso della massa, la derivata seconda della posizione in y . Moltiplicando questa per l'angolo θ , si avrebbe il prodotto tra due variabili di stato, il che renderebbe il sistema non lineare. Dal punto di vista strettamente pratico, ciò potrebbe non rappresentare un particolare problema, in quanto possiamo infatti vedere che il sistema di controllo riesce comunque a governare il drone, in quanto la non linearità risulta piuttosto contenuta (θ è piccolo e varia lentamente). Tuttavia, questo renderebbe più difficile la trattazione teorica del sistema. Sarebbe necessario, infatti, ricorrere alla teoria dei sistemi non-lineari, i quali però non sono stati oggetto di studio dei corsi frequentati finora. Si introduce quindi un'approssimazione piuttosto forte ma necessaria, e cioè che esista un contributo di spinta del drone che non influenzi la componente spaziale z , ma sia a disposizione della sola y . In sostanza, è come se si stessero disaccoppiando le due componenti del sistema.

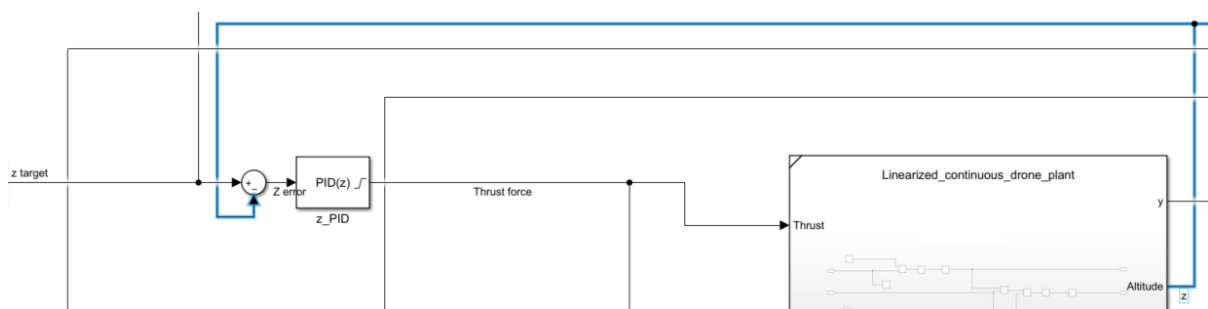
Pertanto, lo schema che rappresenta la funzione di trasferimento descritta e studiata è il seguente:



Si ribadisce però che lo stesso sistema di controllo, pur non essendo riusciti a dimostrarlo formalmente, riesce a ottenere risultati molto simili (praticamente uguali) anche sul modello di plant non lineare (contenente cioè sia funzioni goniometriche per la scomposizione della forza lungo i due assi, sia il prodotto tra le due variabili di stato).

4. Catena posizione z a ciclo chiuso

In questo schema e nei successivi relativi ai controlli in retroazione, viene evidenziato lo specifico anello al quale si fa riferimento.



a. Funzione di trasferimento tempo-continuo

$$\text{fdtClosedLoopz} = \frac{178.3 s^2 + 326.5 s + 150}{s^4 + 100 s^3 + 178.3 s^2 + 326.5 s + 150}$$

Continuous-time transfer function.

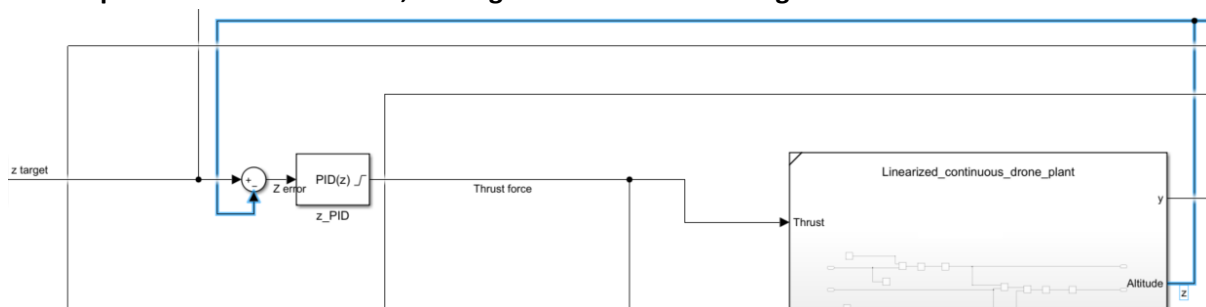
b. Funzione di trasferimento tempo-discreto (discretizzato con Tustin, tempo di campionamento = 0.01s)

$$\text{fdtClosedLoopzDisc} = \frac{0.002989 z^4 + 5.45e-05 z^3 - 0.005924 z^2 - 5.4e-05 z + 0.002935}{z^4 - 3.323 z^3 + 3.982 z^2 - 1.994 z + 0.3353}$$

Sample time: 0.01 seconds

Discrete-time transfer function.

5. Catena posizione z a ciclo chiuso, con ingresso accelerazione di gravità



a. Funzione di trasferimento tempo-continuo

$$\text{fdtClosedLoopG} = \frac{9.81 s^2 + 981 s}{s^4 + 100 s^3 + 349.7 s^2 + 640.6 s + 294.3}$$

Continuous-time transfer function.

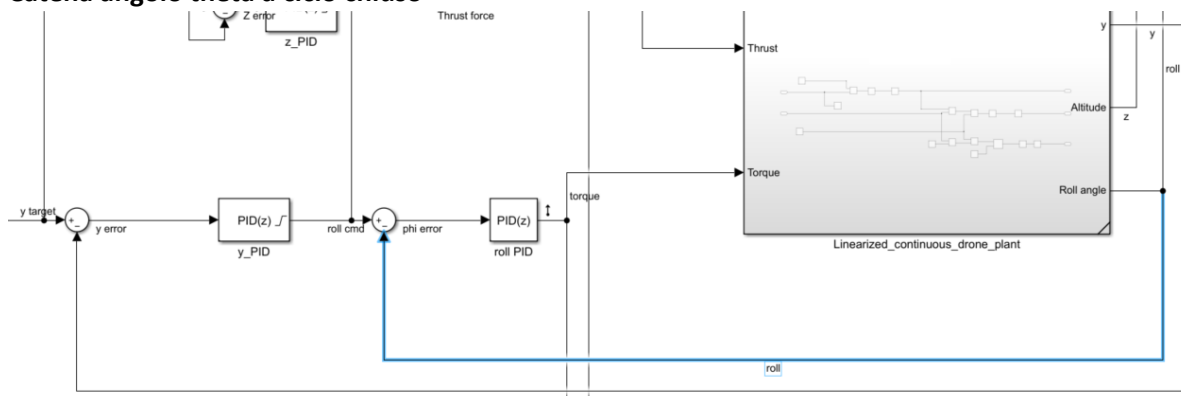
b. Funzione di trasferimento tempo-discreto (discretizzato con Tustin, tempo di campionamento = 0.01s)

$$\text{fdtClosedLoopGDisc} = \frac{0.0004893 z^3 - 0.0001805 z^2 - 0.0004886 z + 0.0001798}{z^4 - 3.346 z^3 + 4.059 z^2 - 2.082 z + 0.3679}$$

Sample time: 0.01 seconds

Discrete-time transfer function.

6. Catena angolo theta a ciclo chiuso



a. Funzione di trasferimento tempo-continuo

`fdtClosedLooppt =`

$$\frac{706 s + 600}{s^3 + 100 s^2 + 706 s + 600}$$

Continuous-time transfer function.

b. Funzione di trasferimento tempo-discreto (discretizzato con Tustin, tempo di campionamento = 0.01s)

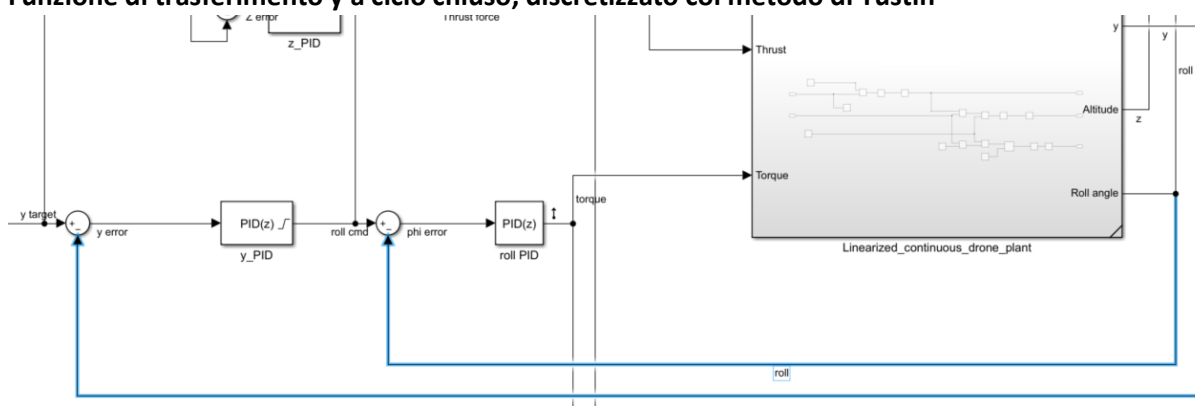
`fdtClosedLoopptDisc =`

$$\frac{0.01168 z^3 + 0.01178 z^2 - 0.01148 z - 0.01158}{z^3 - 2.294 z^2 + 1.636 z - 0.341}$$

Sample time: 0.01 seconds

Discrete-time transfer function.

7. Funzione di trasferimento y a ciclo chiuso, discretizzato col metodo di Tustin



Come si può notare, nel sistema di controllo della posizione verticale y sono presenti due anelli di retroazione annidati che identificano altrettanti loop di controllo. Questa struttura ha richiesto un'attenzione particolare in fase di calibrazione in quanto, affinché svolga correttamente la sua funzione, è preferibile che la dinamica del loop interno sia più veloce di quella del loop esterno. In questo modo il controllore interno vede l'evoluzione del sistema esterno quasi come costante e ha modo di rispondere in maniera efficace al comando iniettato.

a. Funzione di trasferimento tempo-continuo

```
fdtClosedLoopy =

          142965 s^2 + 298000 s + 150000
-----
s^6 + 200 s^5 + 10706 s^4 + 71200 s^3 + 202965 s^2 + 298000 s + 150000

Continuous-time transfer function.
```

a. Funzione di trasferimento tempo-discreto (discretizzato con Tustin, tempo di campionamento = 0.01s)

```
fdtClosedLoopyDisc =

3.966e-05 z^6 + 8.014e-05 z^5 - 3.719e-05 z^4 - 0.000157 z^3 - 4.128e-05 z^2 + 7.686e-05 z + 3.884e-05
-----
z^6 - 4.627 z^5 + 8.655 z^4 - 8.315 z^3 + 4.286 z^2 - 1.113 z + 0.1137

Sample time: 0.01 seconds
Discrete-time transfer function.
```

Per passare da una funzione di trasferimento tempo continuo alla sua controparte tempo-discreto è stata utilizzata la funzione MATLAB **c2d()** (continuous to digital).

```
% ----- discretization -----
fdtPlantZDisc = c2d(fdtPlantZ, Tc, method);
fdtPlantGDisc = c2d(fdtPlantG, Tc, method);
fdtPlantyDisc = c2d(fdtPlanty, Tc, method);

fdtClosedLoopzDisc = c2d(fdtClosedLoopz, Tc, method);
fdtClosedLoopGDisc = c2d(fdtClosedLoopG, Tc, method);
fdtClosedLoopTDisc = c2d(fdtClosedLoopT, Tc, method);
fdtClosedLoopyDisc = c2d(fdtClosedLoopy, Tc, method);
% ----- end discretization -----
```

Figure 26 - Porzione dello script MATLAB sviluppato incaricata di convertire i modelli tempo continuo descritti nei corrispondenti modelli tempo-discreto.

La funzione **c2d()** restituisce un modello discreto a partire da un modello tempo-continuo passato in ingresso. La funzione richiede inoltre il tempo di campionamento da utilizzare per la conversione. Un parametro facoltativo è invece il metodo di discretizzazione. Per questi test è stato utilizzato il metodo di Tustin, mentre il tempo di campionamento è **Tc = 0.01 secondi**. Questa scelta è frutto di un compromesso tra la necessità di garantire un campionamento sufficientemente veloce da garantire un controllo responsivo del drone e il carico di lavoro computazionale da fornire al microcontrollore del drone. Essendo il target di questo progetto un drone commerciale, le performance delle componenti elettroniche installate a bordo non saranno particolarmente elevate.

```
% parametri della discretizzazione

Tc = 0.01;
method = 'tustin';
```

Figure 27 - Porzione dello script in cui sono definiti i parametri relativi alla discretizzazione da effettuare sui modelli tempo-continuo.

L'ambiente di sviluppo MATLAB mette inoltre a disposizione una funzione nativa per la verifica della stabilità dei sistemi dinamici. La funzione in questione è **isstable(sys)**. Questa restituisce il valore logico

1 (vero) il sistema dinamico è stabile, o il valore logico 0 (falso) altrimenti. La funzione aspetta in ingresso un sistema dinamico, sia esso tempo continuo o tempo discreto. `Isstable` restituisce 1 (true) per la stabilità del sistema se:

- Per sistemi tempo-continui, tutti i poli giacciono nel semipiano complesso a parte reale negativa e quelli presenti nell'origine hanno molteplicità non superiore a 1.
- Per sistemi tempo-discreto, tutti i poli sono contenuti all'interno del cerchio unitario e quelli presenti sul cerchio unitario hanno molteplicità non superiore a 1.

Applicando la funzione ***isstable(sys)*** ai modelli (ci concentreremo maggiormente su quelli discreti, che rivestono maggiore interesse ai fini della trattazione in corso) otteniamo:

| Sub-systems tempo continuo | Sub-systems tempo-discreto |
|---------------------------------------------|-------------------------------------------------|
| <code>Is fdtPlantZ stable?</code> 0 | <code>Is fdtPlantZDisc stable?</code> 0 |
| <code>Is fdtPlantG stable?</code> 0 | <code>Is fdtPlantGDisc stable?</code> 0 |
| <code>Is fdtPlanty stable?</code> 0 | <code>Is fdtPlantyDisc stable?</code> 0 |
| <code>Is fdtClosedLoopz stable?</code> 1 | <code>Is fdtClosedLoopzDisc stable?</code> 1 |
| <code>Is fdtClosedLoopG stable?</code> 1 | <code>Is fdtClosedLoopGDisc stable?</code> 1 |
| <code>Is fdtClosedLoopt stable?</code> 1 | <code>Is fdtClosedLooptDisc stable?</code> 1 |
| <code>Is fdtClosedLoopy stable?</code> 1 | <code>Is fdtClosedLoopyDisc stable?</code> 1 |

Figure 28 - Stabilità dei sotto-modelli. A destra quelli tempo-continuo, a sinistra quelli tempo-discreto.

Come mostrato in figura, il solo plant risulta instabile. Apportando la correzione introdotta dal sistema di controllo progettato, la risposta del sistema risulta invece stabile. Questa osservazione è valida tanto per i sistemi tempo-contino di partenza quanto per quelli tempo-discreto ottenuti a partire dai primi. Questo approccio prende il nome di sintesi per discretizzazione.

Il precedente risultato è soddisfacente. Tuttavia, non conosciamo con esattezza la posizione dei poli; al più sappiamo che le condizioni di appartenenza a specifiche regioni del piano (semipiano a parte reale negativa per i tempo-continuo, cerchio unitario per i tempo-discreto) vengono soddisfatte. Anche in questo caso MATLAB si rivela utile. Il tool, infatti, presenta una pratica funzione per il calcolo dei poli, `pole(sys)`, che, dato in input un modello dinamico, sia tempo-continuo che tempo-discreto, ne restituisce i poli. Per completezza, vengono mostrati di seguito i poli dei diversi sub-systems.

| Porzione del sub-system (descritto mediante ingresso e uscita della FdT) | Poli modello tempo continuo | Poli modello tempo discreto | Modulo dei poli del sub-system tempo-discreto |
|---------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| Plant, posizione Z, con ingresso forza applicata dai rotori | $\text{polesZPlant} =$ $\begin{matrix} 0 \\ 0 \end{matrix}$ | $\text{polesZPlantD} =$ $\begin{matrix} 1 \\ 1 \end{matrix}$ | $\text{modPolesZPlantD} =$ $\begin{matrix} 1 \\ 1 \end{matrix}$ |
| Plant, posizione Z, con ingresso accelerazione di gravità | $\text{polesGPlant} =$ $\begin{matrix} 0 \\ 0 \end{matrix}$ | $\text{polesGPlantD} =$ $\begin{matrix} 1 \\ 1 \end{matrix}$ | $\text{modPolesGPlantD} =$ $\begin{matrix} 1 \\ 1 \end{matrix}$ |
| Plant, posizione Y | $\text{polesYPlant} =$ $\begin{matrix} 0 \\ 0 \end{matrix}$ | $\text{polesYPlantD} =$ $\begin{matrix} 1 \\ 1 \end{matrix}$ | $\text{modPolesYPlantD} =$ $\begin{matrix} 1 \\ 1 \end{matrix}$ |
| Sistema controllato, posizione Z, con ingresso forza applicata dai rotori | $\text{polesZClosedLoop} =$ $\begin{matrix} -98.2189 + 0.0000i \\ -0.5979 + 1.5004i \\ -0.5979 - 1.5004i \\ -0.5854 + 0.0000i \end{matrix}$ | $\text{polesZClosedLoopD} =$ $\begin{matrix} 0.9939 + 0.0149i \\ 0.9939 - 0.0149i \\ 0.9942 + 0.0000i \\ 0.3413 + 0.0000i \end{matrix}$ | $\text{modPolesZClosedLoopD} =$ $\begin{matrix} 0.9940 \\ 0.9940 \\ 0.9942 \\ 0.3413 \end{matrix}$ |
| Sistema controllato, posizione Z, con ingresso accelerazione di gravità | $\text{polesZClosedLoop} =$ $\begin{matrix} -98.2189 + 0.0000i \\ -0.5979 + 1.5004i \\ -0.5979 - 1.5004i \\ -0.5854 + 0.0000i \end{matrix}$ | $\text{polesGClosedLoopD} =$ $\begin{matrix} 0.9936 + 0.0000i \\ 0.9854 + 0.0159i \\ 0.9854 - 0.0159i \\ 0.3812 + 0.0000i \end{matrix}$ | $\text{modPolesGClosedLoopD} =$ $\begin{matrix} 0.9936 \\ 0.9855 \\ 0.9855 \\ 0.3812 \end{matrix}$ |
| Sistema controllato, angolo θ | $\text{polesTClosedLoop} =$ $\begin{matrix} -92.4322 \\ -6.5815 \\ -0.9863 \end{matrix}$ | $\text{polesTClosedLoopD} =$ $\begin{matrix} 0.9902 \\ 0.9363 \\ 0.3678 \end{matrix}$ | $\text{modPolesTClosedLoopD} =$ $\begin{matrix} 0.9902 \\ 0.9363 \\ 0.3678 \end{matrix}$ |
| Sistema controllato, posizione Y | $\text{polesYClosedLoop} =$ $\begin{matrix} -96.3906 + 1.4240i \\ -96.3906 - 1.4240i \\ -3.2151 + 0.0000i \\ -1.5437 + 1.7596i \\ -1.5437 - 1.7596i \\ -0.9162 + 0.0000i \end{matrix}$ | $\text{polesYClosedLoopD} =$ $\begin{matrix} 0.9909 + 0.0000i \\ 0.9845 + 0.0173i \\ 0.9845 - 0.0173i \\ 0.9684 + 0.0000i \\ 0.3495 + 0.0065i \\ 0.3495 - 0.0065i \end{matrix}$ | $\text{modPolesYClosedLoopD} =$ $\begin{matrix} 0.9909 \\ 0.9847 \\ 0.9847 \\ 0.9684 \\ 0.3496 \\ 0.3496 \end{matrix}$ |

Figure 29 - Poli dei diversi sub-systems, sia del modello tempo-continuo, sia del corrispondente modello tempo-discreto.

Come si può notare, per quanto riguarda il modello tempo continuo, tutti i poli di ciascun sub-system a ciclo chiuso sono a parte reale negativa. Questo in accordo con la prima verifica svolta, che affermava la stabilità di questi sistemi. In maniera analoga, per i sistemi closed-loop tempo-discreto è possibile vedere che tutti i poli ricadono all'interno del cerchio unitario (il loro modulo è inferiore a 1).

È inoltre possibile osservare come il metodo di discretizzazione va ad alterare il sistema tempo-discreto ottenuto a partire da quello tempo-continuo. In particolare, data la struttura parametrica del codice sviluppato, è sufficiente modificare la riga di definizione del metodo e rilanciare l'esecuzione. Per questioni di brevità non verranno riportati i risultati di questi test, ma si può facilmente verificare sperimentalmente che i risultati sono praticamente uguali nei tre casi studiati. In particolare, sono stati usati i metodi di Tustin, Zero Order Hold (ZOH) e First Order Hold (FOH), ottenendo risultati molto simili tra loro. Questa somiglianza può essere dovuta alla frequenza di campionamento utilizzata, che risulta sufficientemente alta rispetto alla dinamica del modello studiato.

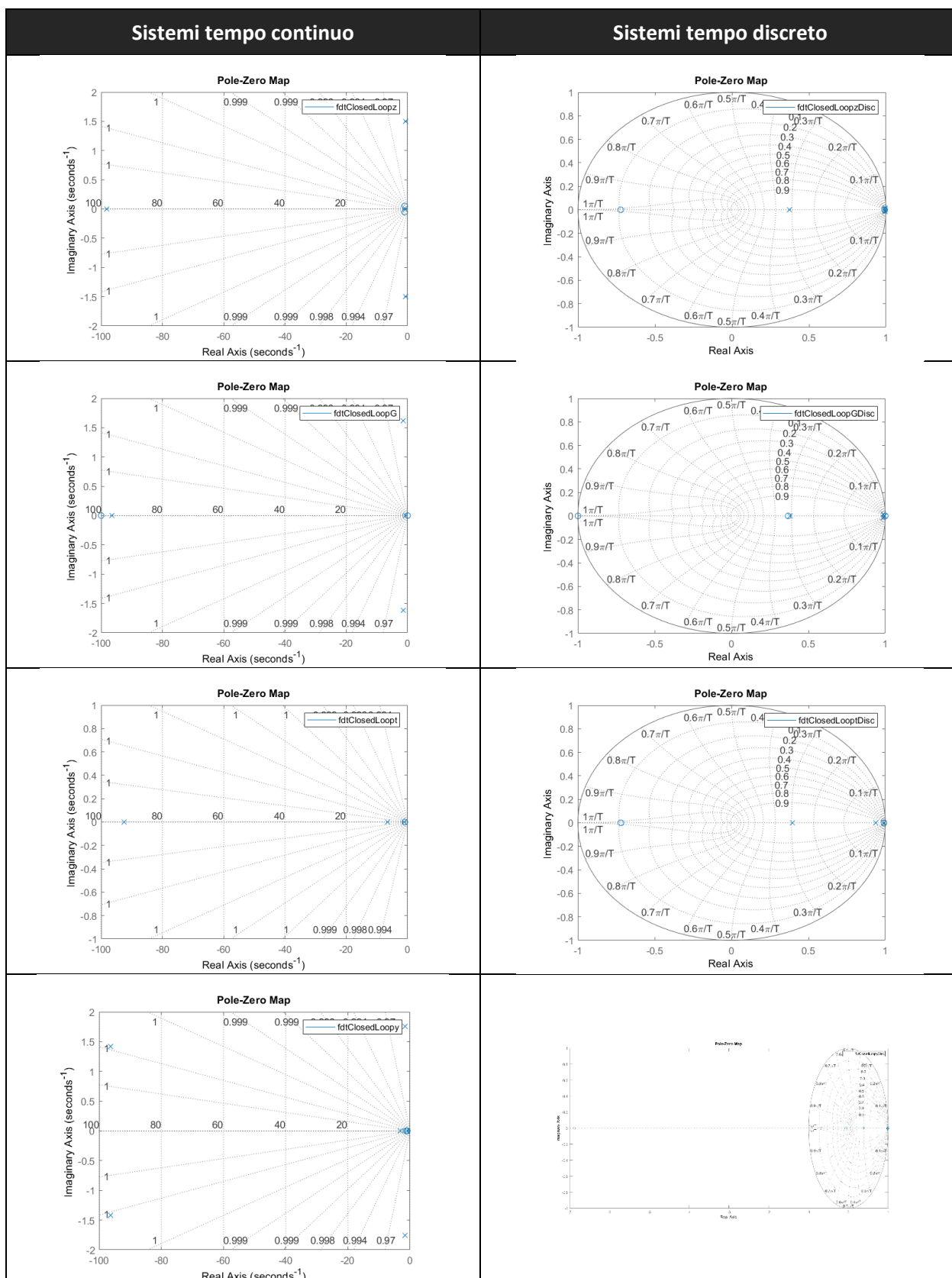


Figure 30 - Posizione di poli e zeri per i quattro sottosistemi a ciclo chiuso tempo-continuo (colonna a sinistra) e tempo-discreto (colonna a destra). Nell'ordine: posizione z dato il setpoint, posizione z dato il disturbo accelerazione di gravità, angolo ϑ , posizione y .

Un'altra considerazione utile riguarda i margini di fase dei sistemi controllati tempo continuo. Questi presentano tutti margini positivi (determinati mediante la funzione **allmargin(sys)** in via numerica, oppure grafica plottando il diagramma di Bode del relativo sistema, usando la funzione **bode(sys)**).

È infine possibile confrontare le risposte ottenute mediante simulazione Simulink con quelle ottenute a partire dall'integrazione numerica delle formule analitiche descritte su MATLAB. Di seguito il confronto.

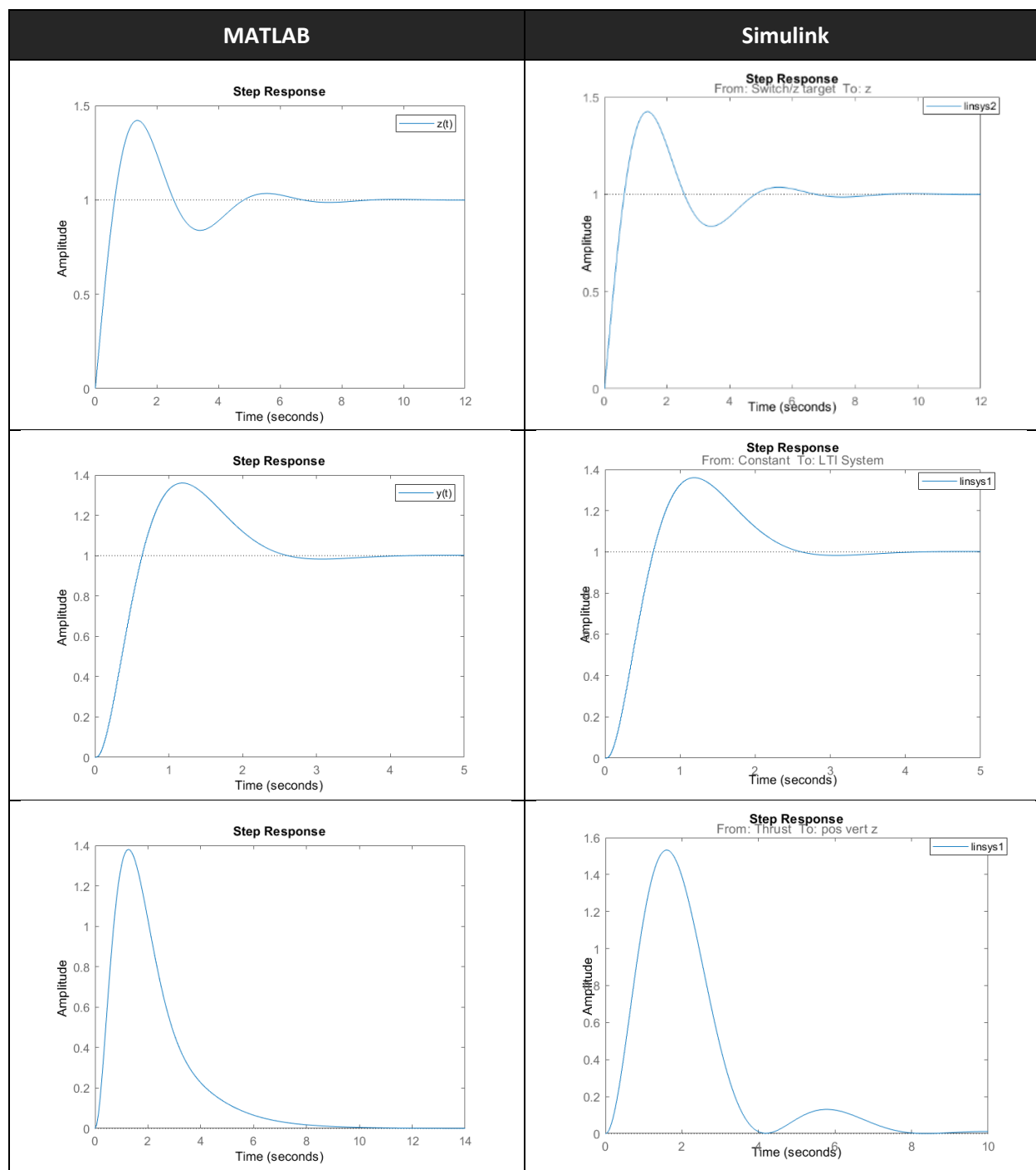


Figure 31 – Risposta al gradino dei subsystems della posizione z (prima riga), y (seconda riga) e z causata da g (terza riga). Nella prima colonna a sinistra è riportato il risultato ottenuto dalla simulazione su MATLAB, in quella a destra i grafici ottenuti mediante Simulink. Le risposte sono molto simili. Possibile causa della differenza potrebbe essere la discretizzazione operata da Simulink.

Come si può notare, le singole sollecitazioni lungo x e y presentano una sovraelongazione non trascurabile. Tuttavia, combinando i due movimenti insieme, le sovraelongazioni tendono a compensarsi (come mostrato nei grafici della risposta al setpoint $A = (5, 5)$ e nel caso del setpoint sinusoidale). Il risultato è quindi auspicabile in quanto, solitamente, un drone è impiegato per compiere movimenti più o meno inclinati, ma quasi mai solo lungo una direzione parallela agli assi di riferimento.

Può essere interessante confrontare il comportamento dei sistemi Z e Y , nel caso di modello tempo continuo e tempo-discreto. Vengono quindi riportate le risposte al gradino nei due casi:

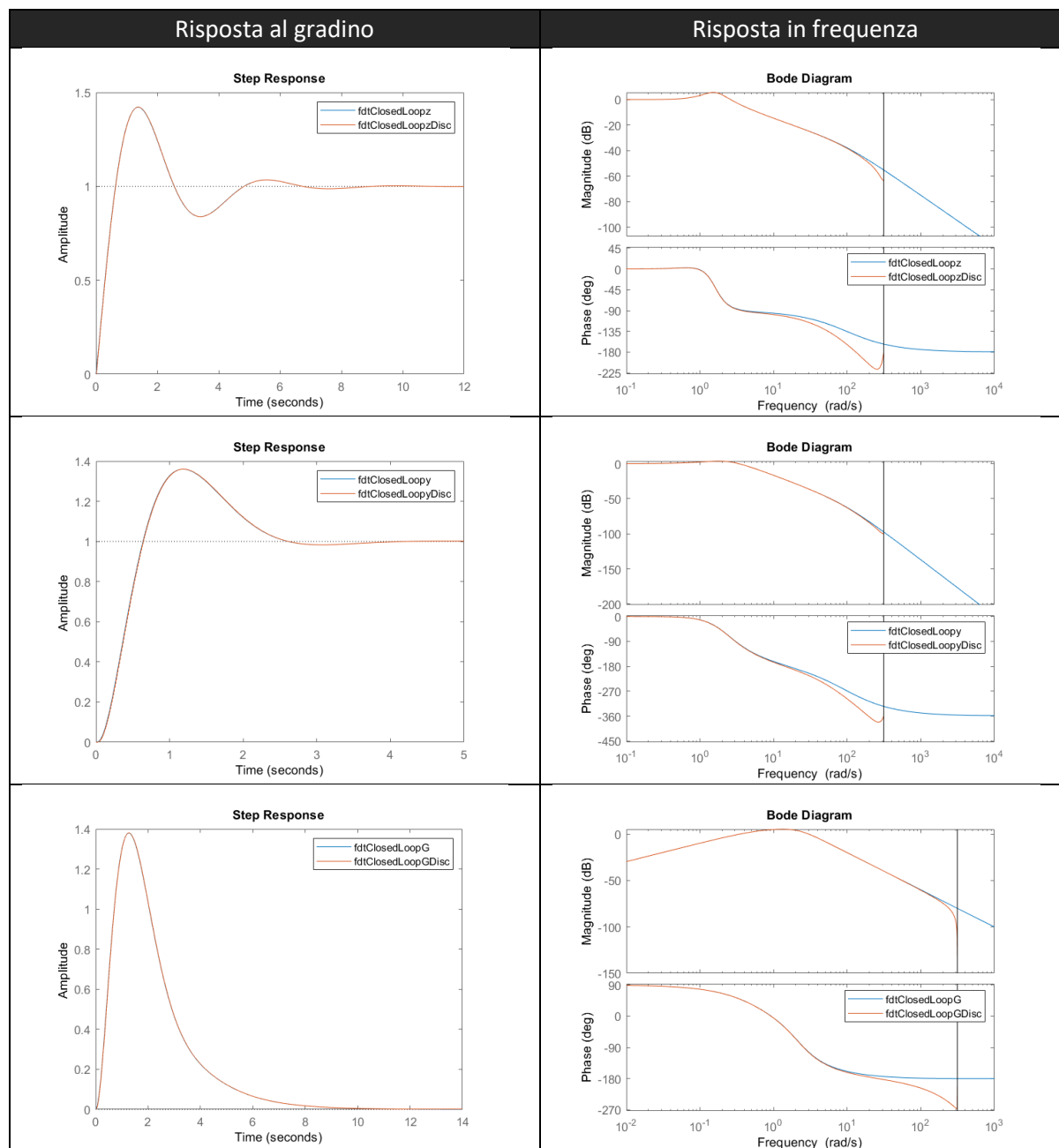


Figure 32 – Colonna a sinistra: risposta al gradino dei subsystems Z dato il setpoint (prima riga), Y dato il setpoint (seconda riga) e Z dato il disturbo g (terza riga). A destra: confronto della step response tra i medesimi sistemi tempo continuo (blu) e discreto (arancio). Per i sistemi tempo discreto è riportata, come linea verticale in grigio, la frequenza di Nyquist.

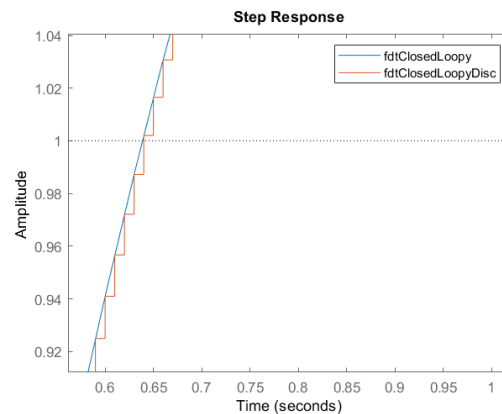


Figure 33 - Zoom sulle risposte impulsive. Nella precedente figura non è possibile apprezzare il dettaglio, ma la curva arancione è in realtà una spezzata, come atteso dal fatto che il sistema che la genera è tempo-discreto.

Come si può osservare, in entrambi i casi, lo scostamento delle risposte dei sistemi tempo-discreto da quelle dei corrispondenti sistemi tempo-continuo è trascurabile. Pertanto, si ritiene che il sistema di controllo progettato sia in grado di soddisfare le specifiche e di garantire una buona manovrabilità del velivolo. Per completezza, sono infine stati calcolati i margini di fase, calcolati mediante il comando **allmargin(sys)**, e verificati poi graficamente mediante il plot della risposta in frequenza appena mostrata. Il subsystem responsabile del posizionamento lungo l'asse verticale z presenta 99° di margine di fase. Quello che mette in relazione l'uscita z con l'ingresso dato dall'accelerazione di gravità 74° . Il sistema relativo al posizionamento lungo l'asse orizzontale y , invece, 82° .

Vediamo quindi un esempio di risposta complessiva che comprende gli ingressi della forza data dai rotori e dall'accelerazione di gravità. In questo esempio, si vuole raggiungere la quota di 10 m, partendo da terra, senza spostarsi di lato.

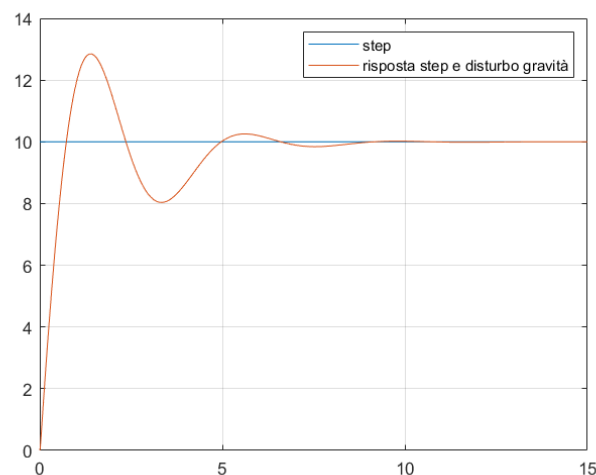
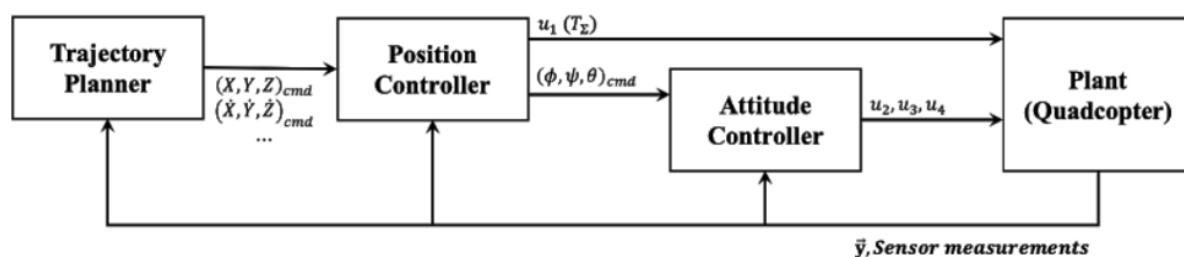


Figure 34 - Step response con set point statico, sulla verticale del punto di partenza, e forza di gravità applicata.

Si può notare che la sovralongazione è piuttosto pronunciata. Questa sovralongazione non era inizialmente presente nella risposta del modello Simulink, in quanto, su Simulink, è piuttosto agevole imporre un limite all'uscita di un PID. Per un drone di piccole dimensioni, 30 N è un valore onesto. Descrivendo la funzione di trasferimento su MATLAB, imporre questa saturazione è più complicato. Cionondimeno si può apprezzare come il sistema, oltre a garantire la stabilità, come dimostrato,

converga al valore di regime desiderato, e che quindi il disturbo costante dato dall'accelerazione di gravità venga completamente rigettato.

Oltre che effettuando un tuning ancora più fine, o utilizzando proprio una nuova architettura di sistema di controllo, è possibile contenere la sovraelongazione mediante un artificio. Si può infatti fare in modo, che pur avendo la medesima sovraelongazione al gradino, il sistema non debba mai rispondere a un gradino. In altre parole, è possibile fare sì che, ogni volta che al sistema venga richiesto di seguire un setpoint a gradino, questo venga prima “smussato”, al fine di ottenere una risposta più dolce. Questo accorgimento può essere attuato consapevolmente dal pilota, o in maniera più automatica, mediante un opportuno algoritmo. Questo compito, unitamente a quello più generale di calcolare la traiettoria migliore note le coordinate dei punti da visitare, viene svolto dal *Trajectory Planner*. Alla luce di questa considerazione viene mostrato uno schema di sistema contenente un Trajectory Planner, e in quale gerarchia questo è posto rispetto al resto del sistema.



Quelli sviluppati e analizzati nella presente trattazione sono i blocchi *Position Controller* e *Attitude Controller*. Questi si occupano di far seguire al drone la traiettoria desiderata. Questa può essere determinata dal pilota, o in caso di guida automatizzata, da un apposito programma. Questo è infatti il compito del trajectory planner (il blocco più a sinistra dello schema). Il trajectory planner, in generale, è costruito in modo da ricevere in ingresso una serie di punti, in generale lontani tra loro, che si vogliono visitare. Partendo da questi, calcola una traiettoria che sia ottimizzata (di solito in base al tempo di percorrenza e per ridurre la bruschezza delle manovre) da far seguire al drone. Questa traiettoria sarà il target che il sistema di controllo cercherà di seguire il più fedelmente possibile.

È infatti possibile, gestendo meglio il calcolo della traiettoria (smussando discontinuità esagerate e sostituendo i segnali di target con linee più morbide, tipo una rampa), raggiungere il target finale desiderato in tempi comparabili al caso precedente, ma riducendo in maniera consistente sovraelongazione e ampiezza delle oscillazioni. Di seguito si considera il medesimo caso dell'esempio precedente. Si vuole raggiungere la quota di 10 metri, partendo da terra. Se si ordina di raggiungere il setpoint in maniera repentina, il sistema va incontro a una forte sovraelongazione. Facendo variare il setpoint in maniera lineare per un certo intervallo di tempo, fino al raggiungimento del punto finale desiderato, il drone risponde con una sovraelongazione più contenuta e con oscillazioni minori, convergendo al valore desiderato e raggiungendolo con tempi molto simili. Questa proprietà prende il nome di **precisione dinamica**, in transitorio. Con calibrazioni più fini della traiettoria potrebbe essere possibile migliorare ulteriormente le performance del sistema, ma questo aspetto meriterebbe una trattazione separata. Per completezza è riportato un esempio relativo al medesimo gradino studiato al paragrafo precedente. La risposta a questo viene confrontata con quella a una rampa di pari altezza finale. Il grafico sotto riportato mostra i risultati ottenuti, in accordo con le considerazioni appena discusse.

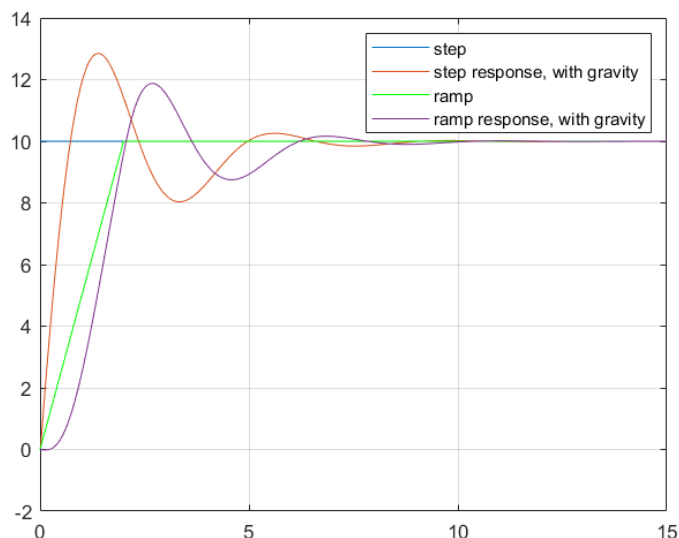


Figure 35 - Confronto tra risposta al gradino e risposta a una rampa saturata di pari altezza finale. Nel secondo caso si ha una sovralongazione e delle oscillazioni molto più contenute che non nel primo, pur garantendo la medesima precisione statica a regime e durata del transitorio comparabile.

10 – Considerazione su sistema 3D

Un'analisi analoga può essere ripetuta per il sistema dotato di sei gradi di libertà. Tuttavia, non è stato possibile effettuare tutte le verifiche del caso precedente, in quanto non è stata individuata una configurazione dei PID che rendesse il sistema complessivo stabile. Infatti, nonostante il sistema 3D sia un'estensione della versione 2D, il sistema di controllo non è risultato adeguato a garantire il raggiungimento del set point nelle direzioni orizzontali. Risulta invece efficace per il target in altitudine. Possibili sviluppi futuri potrebbero riguardare il perfezionamento del sistema di controllo di tale sistema. Per completezza, uno schema funzionale è riportato di seguito:

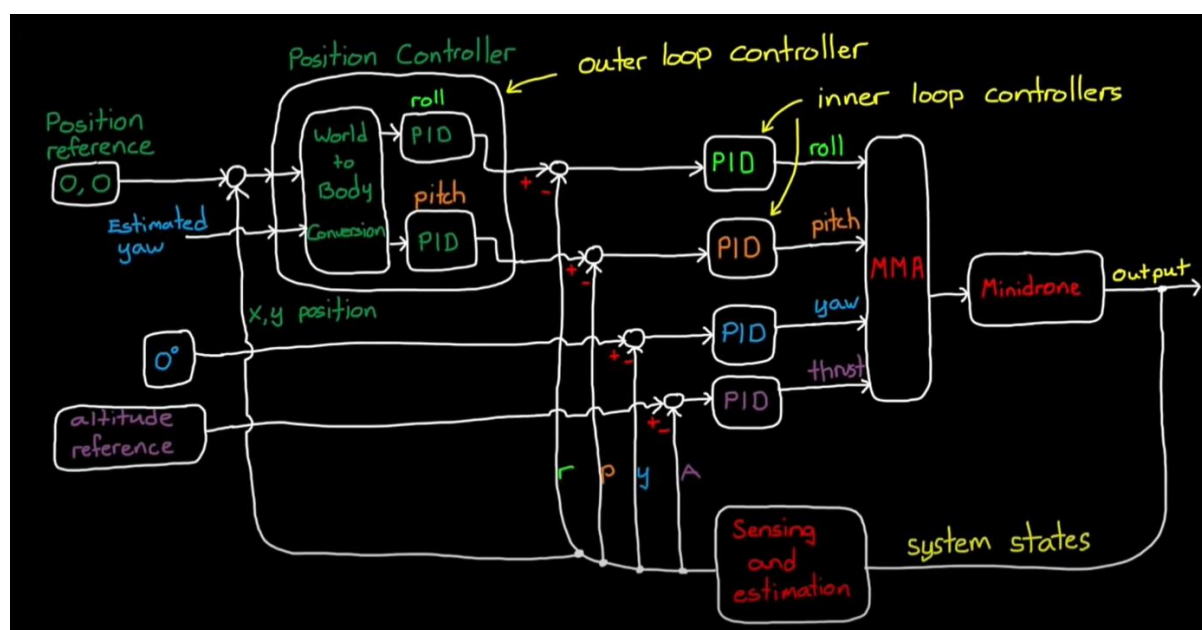


Figure 36 - Schema di un possibile sistema di controllo di un drone reale, basato su regolatori PID.

11 – Traduzione della spinta e del momento complessivo in spinta dei singoli motori

Prima di avviarsi alla conclusione di questo elaborato, viene riportata di seguito una possibile formulazione che consente di tradurre forze e momenti da applicare al drone in azioni concrete direttamente eseguibili dal drone. Questa parte esula dal controllo vero e proprio del drone, per cui non verrà trattata in dettaglio. Per completezza, vengono comunque riportate le formule che consentono di calcolare la spinta che ciascun motore deve esercitare per ottenere il comportamento desiderato da parte del drone, comportamento deciso dal nostro sistema di controllo. Noti quindi la spinta complessiva e le tre componenti del momento torcente (relative a rollio, beccheggio e imbardata), possiamo determinare la spinta che ciascun motore dovrà esercitare mediante il seguente insieme di equazioni:

$$\begin{aligned}
 \text{Motor}_{\text{front right}} &= \text{Thrust}_{\text{cmd}} + \text{Yaw}_{\text{cmd}} + \text{Pitch}_{\text{cmd}} + \text{Roll}_{\text{cmd}} \\
 \text{Motor}_{\text{front left}} &= \text{Thrust}_{\text{cmd}} - \text{Yaw}_{\text{cmd}} + \text{Pitch}_{\text{cmd}} - \text{Roll}_{\text{cmd}} \\
 \text{Motor}_{\text{back right}} &= \text{Thrust}_{\text{cmd}} - \text{Yaw}_{\text{cmd}} - \text{Pitch}_{\text{cmd}} + \text{Roll}_{\text{cmd}} \\
 \text{Motor}_{\text{back left}} &= \text{Thrust}_{\text{cmd}} + \text{Yaw}_{\text{cmd}} - \text{Pitch}_{\text{cmd}} - \text{Roll}_{\text{cmd}}
 \end{aligned}$$

Figure 37 - Una possibile implementazione dell'algoritmo di Motor Mixing.

Questo algoritmo prende il nome di Motor Mixing Algorithm. A partire dal datasheet del singolo motore (e dalle proprietà fisiche dell'aria) è possibile determinare la velocità di rotazione necessaria per garantire la spinta desiderata.

Il sistema complessivo risultante può essere descritto dal seguente schema funzionale:

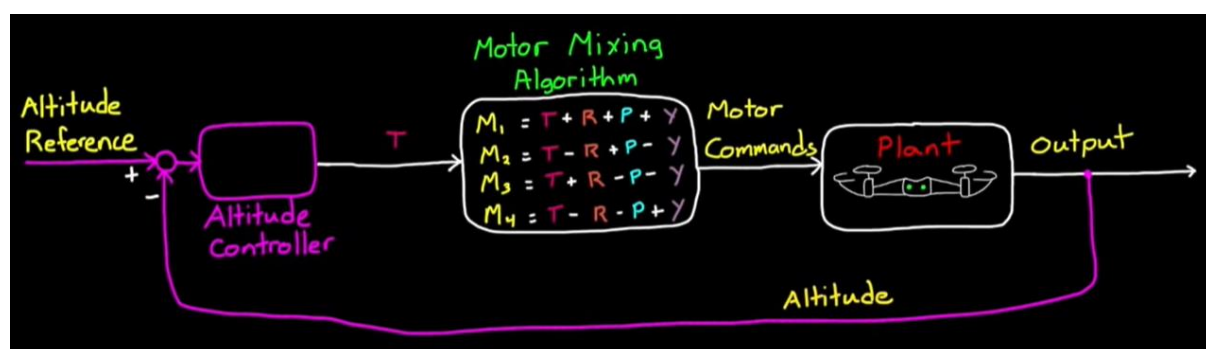


Figure 38 - Rappresentazione del sistema complessivo, comprensivo di algoritmo di Motor Mixing.

Conclusione

In questo progetto è stato analizzato un sistema fisico costituito da un drone. Nello specifico ci si è concentrati sui quadricotteri, il modello più diffuso in commercio, sia per applicazioni professionali sia amatoriali.

È stato presentato il modello teorico del sistema, con un focus particolare sulle grandezze di interesse (posizione e assetto di volo), su come è possibile agire su di esse (attuatori), partendo dalla differenza tra stato desiderato e stato reale, quest'ultimo ottenibile dalle misure raccolte dai sensori presenti a bordo.

Dopo aver riscontrato che il modello risulta non lineare, e quindi almeno in teoria non controllabile mediante le tecniche viste durante il corso, si è passati a linearizzare il modello. Nonostante la semplificazione introdotta, è stato riscontrato che anche questa seconda versione era piuttosto complessa da gestire, per cui si è passati a un caso di studio con un minor numero di gradi di libertà, ma non per questo meno rappresentativo del fenomeno in studio.

Per questo secondo sistema 2D, si è proceduto in maniera analoga che con il sistema 3D. Vista la forte similitudine tra i due, anzi, essendo il secondo un caso particolare del primo, non è stato necessario ripetere l'intero studio da capo. Una volta particolareggiato la trattazione per il sistema in esame, è stato sviluppato e presentato un modello Simulink del drone, cioè il sistema da controllare.

In un secondo momento si è passati quindi a sintetizzare un opportuno modulo di controllo. L'obiettivo è quello di ottenere un sistema complessivo controllato mediante retroazione. Per ciascuna delle variabili di interesse è stato implementato un opportuno anello di retroazione, contenente un controllore PID. Partendo da un sistema tempo-continuo, si è passati al dominio discreto mediante il metodo di Tustin. Inoltre, vista la modularità del codice, è possibile cambiare metodo di discretizzazione, e confrontare i risultati ottenuti con ciascuna delle tecniche. Per ragioni di compattezza, si è affrontata in maniera dettagliata solo la discretizzazione col metodo di Tustin.

Sono poi stati presentati e discussi i risultati della simulazione. Il sistema complessivo è stato quindi analizzato anche in forma analitica, in modo da dimostrare in maniera esaustiva anche dal punto di vista teorico le considerazioni inerenti stabilità e i relativi margini. L'analisi è stata svolta sia nel dominio continuo, ma soprattutto nel dominio discreto.

Complessivamente si tratta di un sistema semplificato, adatto solo ad alcuni casi particolari di un utilizzo reale. Tuttavia, il lavoro svolto ci ha permesso di apprezzare i fondamenti della modellazione di un sistema reale (*model based approach*) e dell'implementazione di un adeguato sistema di controllo.

Riferimenti

1. Slide del professor Alessandro Pisano del corso di Controllo Digitale dell'Anno Accademico 2020/2021 - https://teams.microsoft.com/_#/school/files/General?threadId=19%3Ae4cf10be6a074c17b3570c6848c20111%40thread.tacv2&ctx=channel&context=Materiale%2520del%2520corso&rootfolder=%252Fsites%252FControlloDigitale-AA2020-21%252FMateriale%2520del%2520corso
2. Lezioni del professor Alessandro Pisano del corso di Controllo Digitale dell'Anno Accademico 2020/2021 - <https://web.microsoftstream.com/channel/97d98058-b0a2-4c41-87e5-89abb3c3071c>
3. Documentazione MATLAB e Simulink - <https://www.mathworks.com/help/matlab/>
4. Chun-Wei Kong - 6-Dof Quadcopter Simulation and Control - <https://jordan787878.github.io/firstweb/6dofQuadcopter/6dofQuad.html>
5. VDEngineering - MATLAB & Simulink Tutorial: Quadrotor UAV Trajectory and Control Design (PID + Cascaded) - <https://www.youtube.com/watch?v=iS5JFuopQsA>
6. MathWorks - Quadcopter Simulation and Control Made Easy - https://www.mathworks.com/videos/quadcopter-simulation-and-control-made-easy-93365.html?s_tid=srchtitle&s_eid=PSM_15028
7. MathWorks - Modelling, Simulation, and Control of a Quadcopter - https://www.mathworks.com/videos/modelling-simulation-and-control-of-a-quadcopter-122872.html?s_eid=PSM_15028
8. Seongheon Lee - AE450_Lec10_Quadcopter_Dynamics_and_Control – Simulating Quad(+)copter Dynamics and PID attitude Controller - https://github.com/SKYnSPACE/AE450/blob/master/Lec10/AE450_Lec10_Quadcopter_Dynamics_and_Control.pdf
9. Sertac Karaman and Fabian Riether – <https://dspace.mit.edu/handle/1721.1/106777>
10. Stack Exchange Mathematics - How to control a nonlinear system with a PID controller – <https://math.stackexchange.com/questions/1509148/how-to-control-a-nonlinear-system-with-a-pid-controller>
11. ResearchGate - Could I use the PID controller for nonlinear systems? - https://www.researchgate.net/post/Could_I_use_the_PID_controller_for_nonlinear_systems
12. Learn to Fly - Foo Fighters Rockin'1000 Official Video - <https://www.youtube.com/watch?v=JozAmXo2bDE>

Drafts' page

Bacheca

magari possiamo spezzare l'introduzione: tenerla più piccola, e spostare i pezzi rimanenti in capitoli dedicati, per presentare bene il topic.

Fare dei live script MATLAB (tipo i notebook di Biggio), nei quali mettere sia il codice MATLAB che il testo delle spiegazioni?

Quella che abbiamo fatto finora è una modellizzazione continua del sistema. Non è che a noi serve una cosa espressamente discreta?

La trattazione che abbiamo fatto noi, per quanto riguarda Simulink, è continua. Ma a noi serve qualcosa di discretooooooooo!

tune PID controller

Per gli ultimi due punti: <https://www.mathworks.com/videos/getting-started-with-simulink-part-8-modeling-continuous-and-discrete-systems-in-one-simulink-model-1508446587027.html>

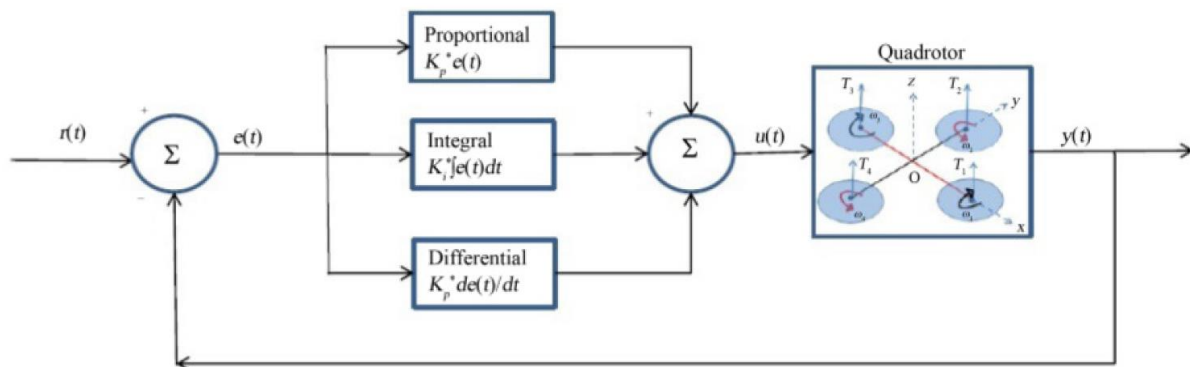
Fare riferimento ai motori DC brushless che nomina negli esempi delle slide (anche nel blocco dell'osservatore)

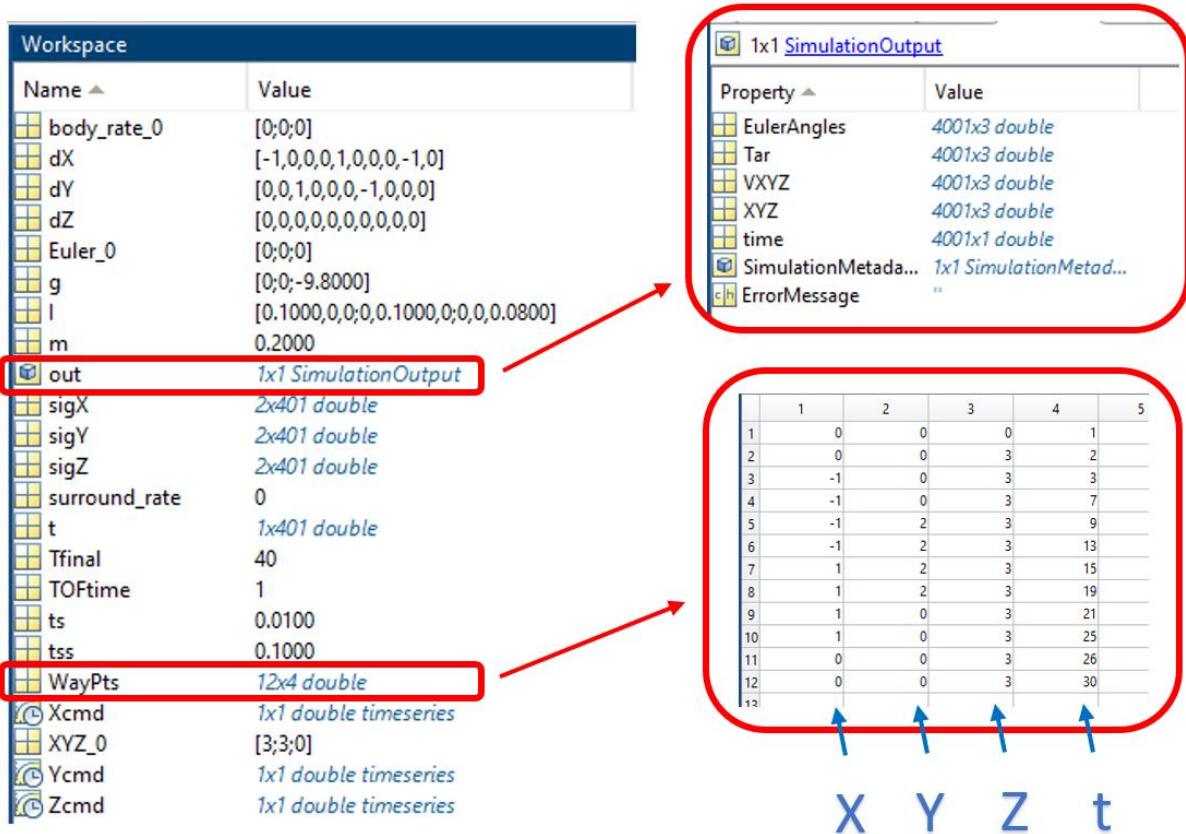
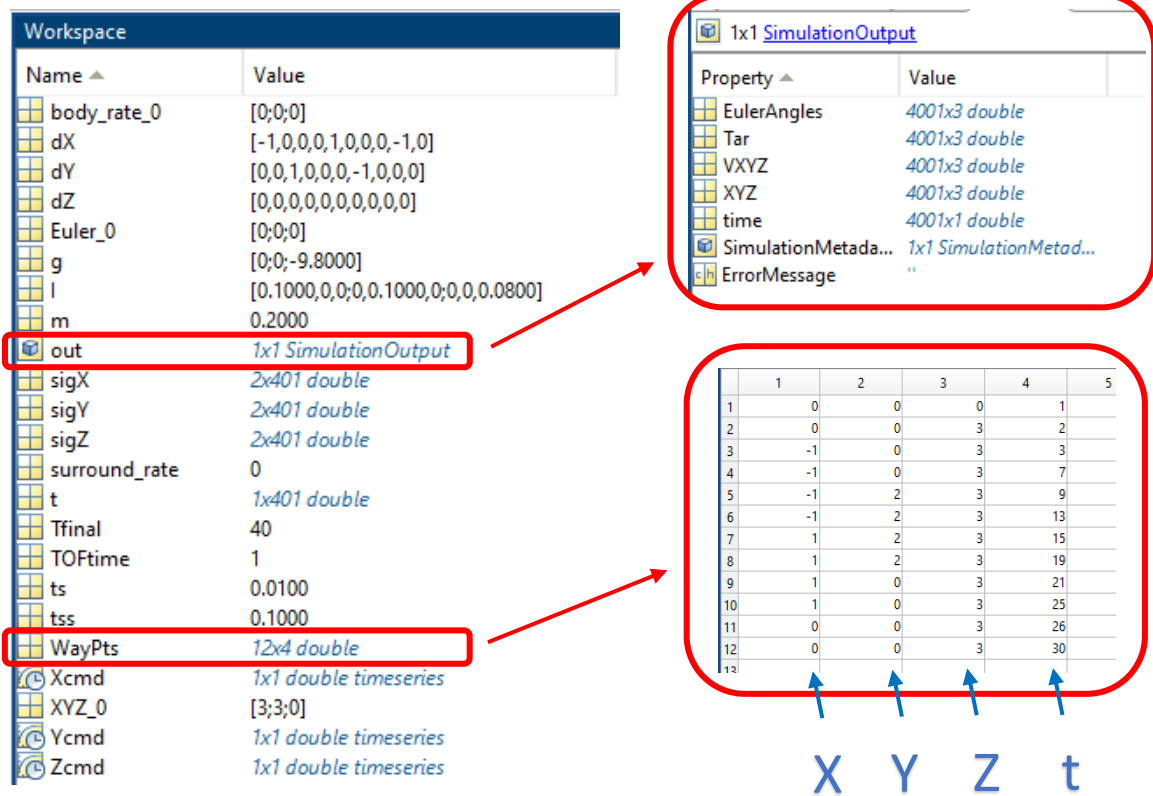
determinare poli e zeri del sistema? Nel caso, cercare qualche procedura automatizzata (tipo: <https://www.mathworks.com/help/slcontrol/ug/polezeroplot.html>)

occhio alla definizione degli angoli di Eulero -> Una particolare variante degli angoli di Eulero, usata in aeronautica e in robotica, è quella degli angoli di Tait-Bryan. In questo caso gli angoli alpha, beta e gamma prendono il nome, rispettivamente, di angoli di imbardata, rollio e beccheggio.



1. **FPV**, acronimo di First Person View. Si indicano con questa sigla i droni generalmente pilotati indossando dei goggles per visualizzare l'anteprima.



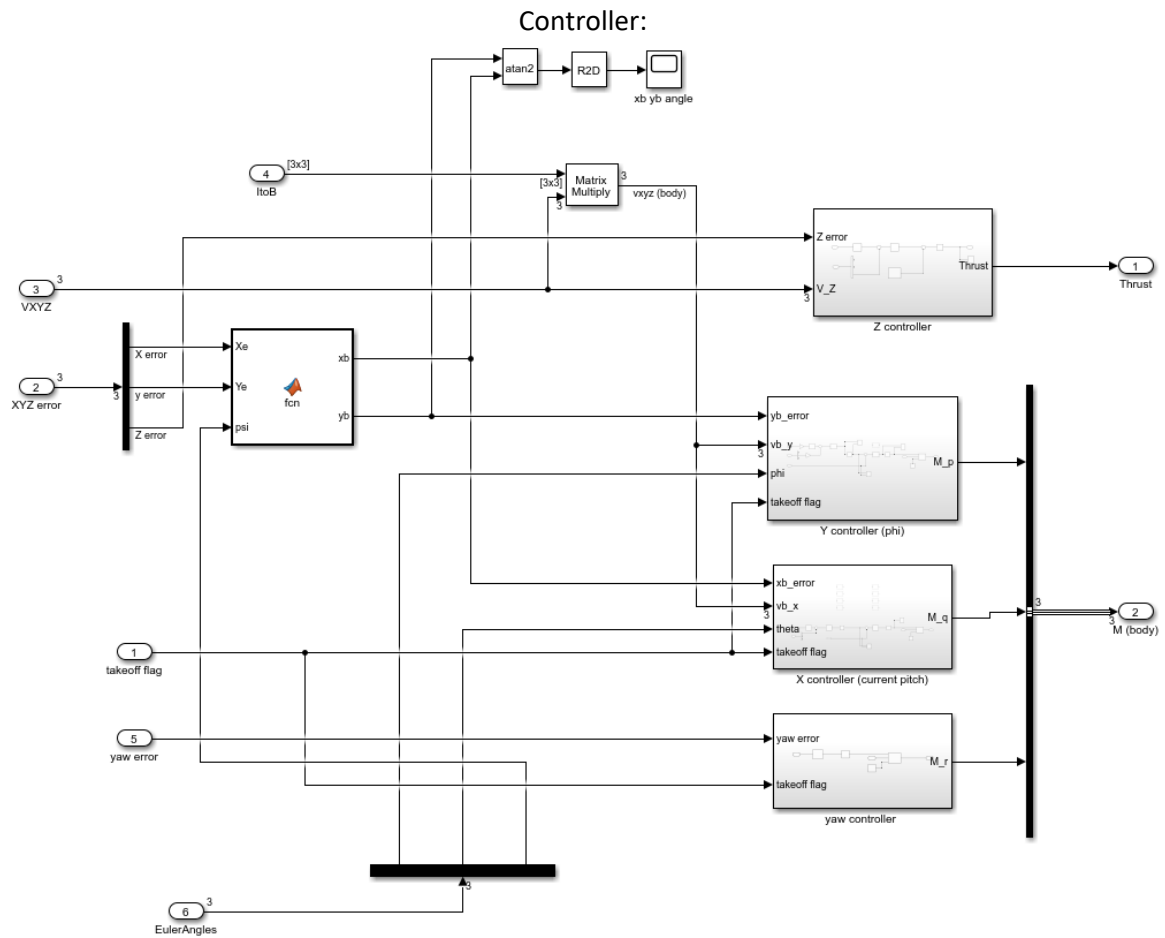


Workspace:

| | |
|----------------|-------------------------------------------------------------------------|
| Body_rate_0 | |
| dX | Tabella delle derivate delle posizioni sulla coordinata X |
| dY | Tabella delle derivate delle posizioni sulla coordinata Y |
| dZ | Tabella delle derivate delle posizioni sulla coordinata Z |
| Euler_0 | Assetto di volo iniziale del drone |
| g | Accelerazione gravitazionale |
| I | Momento di inerzia sui tre assi |
| m | Massa del drone = 200g |
| out | Output di simulazione (spiegato sotto) |
| sigX | Posizioni in X del riferimento nel tempo |
| sigY | Posizioni in Y del riferimento nel tempo |
| sigZ | Posizioni in Z del riferimento nel tempo |
| sourround_rate | Rateo di rotazione attorno al riferimento |
| t | Tempo |
| Tfinal | Tempo di fine simulazione |
| TOFtime | Tempo al decollo |
| ts | Periodo di discretizzazione nel tempo = 0,01s delle uscite di Simulink |
| tss | Periodo di discretizzazione nel tempo = 0,1s degli ingressi di Simulink |
| WayPts | Posizioni del riferimento nel tempo |
| Xcmd | Posizione del riferimento rispetto a X |
| XYZ_0 | Posizione drone rispetto al riferimento |
| Ycmd | Posizione del riferimento rispetto a Y |
| Zcmd | Posizione del riferimento rispetto a Z |

Output:

| | |
|--------------------|------------------------------------------------------------------------|
| EulerAngles | Assetto di volo del drone |
| Tar | Posizioni del riferimento ad ogni istante (100 al secondo, campionate) |
| VXYZ | Posizione relativa al riferimento del drone |
| XYZ | Posizione assoluta del drone |
| time | Tempo discreto (100 sample al secondo) |
| SimulationMetadata | Altri dati di simulazione |
| ErrorMessage | Possibili errori |



Funzione di trasferimento controllore **PID** posizione **Z**:

$$\frac{35.65 s^2 + 65.3 s + 30}{s^2 + 100 s}$$

Funzione di trasferimento anello aperto, **Z**:

$$\frac{178.3 s^2 + 326.5 s + 150}{s^4 + 100 s^3}$$

Sistema di controllo a ciclo chiuso, **Z**:

$$\frac{178.3 s^2 + 326.5 s + 150}{s^4 + 100 s^3 + 178.3 s^2 + 326.5 s + 150}$$

Sistema di controllo a ciclo chiuso, **theta**:

$$\frac{706 s + 600}{s^3 + 100 s^2 + 706 s + 600}$$

Funzione di trasferimento Plant, **Y**:

$$\frac{s^2 + s + 1}{s^4 + s^3 + s^2 + s + 1}$$

9 - Motore DC brushless

Come si accennava nel capitolo dedicato alla descrizione degli attuatori, la maggior parte dei quadricotteri commerciali impiega dei motori DC brushless per erogare la spinta per tenere in volo il mezzo e farlo spostare.

prendere screen progetto motore DC brushless del prof

È quindi possibile implementare un opportuno algoritmo di motor mixing per ricavare la spinta che ciascun motore deve erogare al fine di generare la spinta totale e i momenti torcenti che consentiranno al drone di seguire il set-point fornito al sistema.

Tuttavia, a loro volta, i motori impiegati, di tipo DC brushless, richiedono loro stessi un opportuno sistema di controllo al fine di garantirne una risposta adeguata alle richieste.

vedi progetto motori brushless del prof

Perché implementare un sistema di controllo?

Non basta il pilota per guidare bene il drone e in generale un velivolo?

Serve un osservatore? Forse nope. Anzi, no e basta. Il prof ha detto che andava bene 2d + analisi teorica

Come abbiamo detto prima, alcuni dei sei movimenti non direttamente controllabili. Di questi, abbiamo dimostrato come quattro lo siano. I restanti due sono la traslazione lungo gli assi X e Y (e le loro combinazioni, ottenibili mediante principio di sovrapposizione). Possiamo ovviare a questo problema cambiando l'assetto di volo, in modo tale che la spinta non sia più orientata esclusivamente lungo la verticale (parallelamente a Z), ma che abbia anche una componente non nulla nella direzione orizzontale desiderata. Per fare questo è sufficiente cambiare l'assetto di volo, combinando opportunamente rollio e beccheggio, in modo da inclinare il drone verso la direzione desiderata. Al fine di mantenere costante l'altezza del velivolo, incrementiamo la spinta totale, in modo che la componente lungo Z resti invariata.

Al fine di rendere i risultati della simulazione più facilmente consultabili, apprezzabili e fruibili, è stato realizzato uno script dedicato alla realizzazione di un grafico animato. Questo rappresenta lo spostamento del drone a partire dalla posizione iniziale fino al raggiungimento del setpoint desiderato. Lo stesso script è in grado di mostrare anche l'evoluzione del sistema quando, anziché un setpoint statico, si intende far seguire al drone una traiettoria specifica. In sovrapposizione, si può stampare anche la traiettoria desiderata, in modo da apprezzare come il drone segua fedelmente il percorso richiesto.

Dubbio: definizione di quello che implementiamo

Tipo, noi facciamo "solo" una sorta di stabilizzatore, tipo per tenere la posizione in presenza di alcuni disturbi esterni (tipo vento laterale) (?)

Possiamo poi, in secondo luogo, implementare un cosetto che cerca di seguire al meglio una traiettoria predefinita.

In entrambi i casi, come metrica dell'accuratezza del sistema sviluppato possiamo considerare quanto il risultato della simulazione si scosta dal comportamento desiderato.

plottare quindi traiettoria desiderata e quella ottenuta

magari possiamo anche calcolare una specie di scostamento quadratico medio, per dare una stima di quanto sia accurato

Scelta di tenere il plant lineare -> non è obbligatorio, e non lo abbiamo fatto

Come descritto nella prima parte dell'elaborato, infatti, il drone non può solo traslare orizzontalmente, ma deve prima inclinarsi verso la direzione lungo la quale intende dirigersi. Maggiore tale inclinazione, maggiore sarà la componente della spinta totale orientata in orizzontale, e più veloce sarà tale movimento. Limitare fortemente tale angolo, risulta in una dinamica verticale molto più veloce che non quella orizzontale. Tale fenomeno sarebbe estremamente limitante in un drone reale. Si è pertanto deciso di partire da una **modellazione non lineare del plant**, senza limitare l'angolo. Questo, almeno da un punto di vista strettamente teorico, dovrebbe impedire il corretto utilizzo di tecniche di controllo lineare. D'altra parte, sappiamo che, per quanto un modello possa essere lineare, nella realtà **tutti i sistemi fisici** presentano un **certo grado di non-linearità**. Questo non impedisce però ai regolatori PID, appunto dei controllori lineari, di continuare ad operare con ottimi risultati in un vasto range di sistemi reali. In particolare, grazie soprattutto all'azione integrale del PID, è possibile compensare tali non idealità.

Dal punto di vista numerico invece, interviene Simulink stesso a darci una mano. Il tool stesso, infatti, prima di lanciare la simulazione, effettua una linearizzazione automatica del plant, eventualmente ricalcolando il punto intorno al quale operare tale semplificazione, a mano a mano che il sistema evolve. In questo modo è possibile ottenere risultati molto più soddisfacenti che non operando una linearizzazione statica intrinseca del sistema simulato.

Sui social siamo sommersi da spettacolari fotografie aeree e video mozzafiato con riprese dall'alto. Quando ci siamo chiesti come si realizzano queste immagini, ci hanno sempre risposto: con un drone! Ok, ma, esattamente, che cos'è un drone?

Si potrebbe implementare un algoritmo che restituisca direttamente la velocità angolare di ciascuno dei quattro rotori. Tuttavia, per una questione di chiarezza, sia di funzionalità del sistema di controllo, l'approccio classico è quello di lavorare in termini di movimenti traslazionali e rotazionali (quelli descritti a inizio trattazione), e solo una volta determinata l'azione correttiva tradurre questa in termini di velocità angolare di ciascun singolo rotore. Questa operazione viene svolta da un particolare algoritmo che prende il nome di Motor Mixing Algorithm. Il sistema complessivo risultante è mostrato, almeno concettualmente, di seguito:

7.5 - Linearizzazione del modello - parte 2

Una volta **tarato** il **sistema di controllo** si è proceduto con l'**esportare** tale **configurazione** ad applicarla al **plant linearizzato** secondo le tecniche discusse nel precedente paragrafo dedicato alla linearizzazione. Se il modello così ottenuto risulta ancora rappresentativo del modello originale, allora

applicando gli stessi stimoli esterni, stesso set-point e stessi parametri interni, ci si aspetta che la risposta del secondo non si discosti troppo dal primo. E così è stato. D'ora in poi nel resto della trattazione si farà riferimento al solo modello linearizzato.

Nel caso di spostamenti in direzioni diagonali, inoltre, la sovraelongazione di una delle due componenti può aiutare a raggiungere il setpoint dell'altra in maniera più rapida. Analogamente, il sistema di controllo di una può aiutare a compensare l'overshooting della prima.

→ nope, perché nel modello linearizzato sono disaccoppiate (o almeno, credo che il disaccoppiamento non consenta questo trick, che però resta valido nel modello non linearizzato). Per non incartarci su questo aspetto nel sistema LINEARIZZATO non ne parlo



Argomenti visti a lezione

1. Sistemi tempo discreto
 - a. Forma ingresso uscita
 - i. In avanti
 - ii. All'indietro
 - b. Modi e classificazione
 - c. Evoluzione libera
2. Variabili di stato
 - a. Evoluzione libera
 - i. Sylvester
 - ii. Autovalori e autovettori
 - iii. Matrice di transizione dello stato
3. Classificazione dei modi
 - a. Impulsivi
 - b. Aperiodici
 - c. Pseudoperiodici
 - d. Stabili
 - e. Al limite di stabilità
 - f. instabili
 - g. Regione di stabilità (cerchio unitario)
4. Trasformata Z
 - a. Definizione
 - b. Trasformate notevoli

- c. Proprietà e teoremi
- d. Convoluzione
- 5. Antitrasformata Z
 - a. Lunga divisione
 - b. Scomposizione in fratti semplici (metodo di Heaviside)
 - c. Metodo computazionale
 - i. Symbolic Math Toolbox
- 6. Analisi di sistemi a tempo discreto nel dominio della trasformata Z: modelli ingresso-uscita
 - a. Risposta libera e risposta forzata
 - b. Funzione di trasferimento
 - i. Serie
 - ii. Parallelo
 - iii. Retroazione
 - iv. Luogo delle radici (root locus)
- 7. Analisi di sistemi a tempo discreto nel dominio della trasformata Z: modelli in variabili di stato
 - a. Risposta libera e risposta forzata
 - b. Matrice risolvibile
 - c. Funzione di trasferimento
 - i. Symbolic Math Toolbox
- 8. Campionamento e ricostruzione di segnali
 - a. Campionamento (conversione AD)
 - b. Ricostruzione (conversione DA)
 - i. Ricostruttore di ordine zero (ZOH – zero-order-hold) -> ricostruzione costante a tratti
 - c. Spettro del segnale campionato
- 9. Sistemi a dati campionati e relative funzioni di trasferimento
 - a. Funzione di trasferimento
 - b. Modellizzazione sistema di controllo
 - c. Discretizzazione esatta di un processo a tempo continuo espresso in variabili di stato
 - i. Retroazione dello stato (non solo dell'uscita)
 - ii. Matrice caratteristica del modello
 - iii. Calcolo della matrice caratteristica
- 10. Mapping fra il piano s ed il piano z (corrispondenze tra trasformata di Laplace e trasformata Z)
 - a. Trasformazione di campionamento
 - b. # corrispondenza biunivoca
 - i. Striscia principale
 - ii. Semiasse reale negativo
 - iii. Tempo di assestamento
 - iv. Sovraelongazione percentuale -> retta a smorzamento costante
 - 1. Cardioide
 - v. Tempo di salita
- 11. Stabilità dei sistemi a tempo discreto
 - a. Sistemi dinamici asintoticamente stabili – AS
 - i. tutti i suoi poli sono strettamente contenuti all'interno del disco unitario
 - b. Sistemi dinamici stabili (semplicemente stabili) – SS
 - i. Hanno uno o più poli semplici sul perimetro del disco unitario

- ii. Non hanno poli esterni al disco unitario
- c. Sistemi dinamici instabili – I
 - i. Hanno almeno un polo esterno al disco unitario oppure uno o più poli multipli sul perimetro del disco unitario
- d. Verifica della stabilità
 - i. Criterio di Jury
 - ii. Metodo della trasformazione bilineare
 - 1. Trasformazione di variabile + criterio di Routh Hurwitz
 - 2. -> mappare il disco unitario sul semipiano sinistro
- 12. Comportamento a regime dei sistemi di controllo a tempo discreto – Parte 1
 - a. Attenuazione dei disturbi sull'uscita
 - b. Poli in $z=1$ nel processo o nel controllore (a ciclo aperto) -> reiezione totale dei disturbi (?)
 - c. Sistema a ciclo chiuso: deve essere asintoticamente stabile
 - d. Tipo di sistema di controllo -> numero complessivo di poli in $z=1$ presenti nel controllore e nel processo
 - i. Sistemi di controllo tipo zero -> l'uscita non converge al set-point; la precisione aumenta al crescere del guadagno. Ma questo non può crescere indefinitamente, pena giocarsi la stabilità a ciclo chiuso del sistema
- 13. Comportamento a regime dei sistemi di controllo a tempo discreto – Parte 2
 - a. Sistemi di controllo di tipo uno -> nel regolatore o nel processo è presente un polo in $z=1$ -> l'uscita a regime coincide col set-point
 - i. Disturbo costante: reiezione totale se polo in $z=1$ è nel controllore. Reiezione parziale se è nel processo. Attenuazione maggiore al crescere del guadagno
 - ii. È un regolatore PI. Godono di precisione statica robusta (quello di sopra), per questo sono molto diffusi
 - b. Principio del modello interno -> **Se un sistema a ciclo aperto contiene un disturbo (la sua FdT comprende un termine uguale alla trasformata del disturbo), allora si ha reiezione totale di quel disturbo a regime. # Il sistema a ciclo chiuso deve essere asintoticamente stabile**
 - c. Risposta armonica -> comportamento a regime quando in ingresso al sistema viene applicata una sequenza sinusoidale -> l'uscita ha la stessa frequenza dell'ingresso, ma diversa ampiezza e fase
 - i. In questi casi però fare Bode a mano è molto più vortex che non in continuo -> si usa MATLAB
- 14. Sintesi per discretizzazione
 - a. Approssimazione tramite integrazione numerica
 - i. Metodo delle differenze in avanti
 - ii. Metodo delle differenze all'indietro
 - iii. Metodo di Tustin (o della Trasformazione bilineare)
 - 1. Metodo di Tustin con precompensazione in frequenza (prewarping)
 - b. Approssimazione tramite invarianza della risposta
 - i. Metodo della invarianza della risposta al gradino
- 15. Discretizzazione e controllo digitale di sistemi con ritardo
 - a. # Scegliere il periodo di campionamento come un sottomultiplo intero del ritardo
- 16. Forme canoniche dei modelli in variabili di stato. Assegnamento poli e controllabilità
 - a. forma canonica controllabile

- i. utile per assegnamento poli»
 - 1. Si fa una particolare retroazione dello stato
 - 2. Matrice di controllabilità
 - 3. Individuare il polinomio caratteristico desiderato
 - 4. Descrizione del sistema in forma compagna controllabile
 - b. forma canonica osservabile
 - i. utile per osservatore di stato
- 17. Osservatore di stato
 - a. -> stimare evoluzione temporale delle variabili di stato a partire unicamente dalla misura diretta della variabile di uscita
 - b. Procedura standard per il calcolo della matrice dell'osservatore