

POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Aerospaziale

Tesi di Laurea

**Controllore adattivo MRAC per
quadrirotori in presenza di
incertezze e disturbi**



Relatore
Dr. Stefano Primatesta
Correlatore:
Dr. Riccardo Enrico

Laureando
Luca FIORE
matricola 305961

ANNO ACCADEMICO 2024 – 2025

Sommario

Il presente elaborato di tesi ha come obiettivo lo studio di un controllore adattivo per un drone quadrirotore. Lo studio dei controllori adattivi nasce dalla necessità di fornire un controllo efficiente in caso di disturbi. Negli ultimi decenni la diffusione dei droni e i relativi utilizzi sempre più complessi anche in ambiente ostili, fanno sì che sorga la necessità di sviluppare controllori maggiormente efficaci, in grado di adattarsi a molteplici condizioni senza comportare aggravio di complessità computazionale e garantendo stabilità. Nella prima parte della tesi si introducono dei concetti preliminari per fornire un'idea chiara della modellazione del drone, che viene poi espressa in forma matematica ricavando le equazioni della dinamica complete. Successivamente si analizzano i principali tipi di controllori e lo sviluppo di un Model Reference Adaptive Control (MRAC) per la stabilizzazione del drone sui tre assi in presenza di disturbi lineari e turbolenti. L'analisi dei controllori viene affrontata differenziando l'approccio Model Based da quello Data Driven. Al termine di questa sezione si introduce il concetto di adattamento fornendo uno sguardo su modelli di predizione dei parametri e leggi di controllo in grado di adattarsi alle condizioni non note. Questo approccio permette di tenere in considerazione disturbi e incertezze sia di natura nota e prevedibile, sia di natura non nota. A questo scopo si definisce innanzitutto il modello di riferimento del drone e secondariamente una legge di controllo del tipo Model Reference Adaptive Control (MRAC). Il modello di riferimento viene ricavato tramite la linearizzazione delle equazioni della dinamica complete. Il sistema così ottenuto sarà il Model Reference della struttura MRAC. L'obiettivo è far sì che la dinamica completa del drone tenda a coincidere con quella linearizzata in ogni condizione di volo. Per ottenere ciò si controlla il Model Reference con un controllore di tipo LQR, mentre la legge di controllo adattiva viene ricavata tramite analisi di Lyapunov. Il modello MRAC così ottenuto viene irrobustito con alcune modifiche quali: Dead-band per evitare oscillazioni indesiderate, Sigma Modification per fornire "damping" alla predizione del parametro, Parameter Projection per evitare parametri fuori range, Osservatore e componente Composite per affinare l'adattamento. L'architettura del controllore completo viene infine testata su un modello ad alta fedeltà del drone Parrot Mambo fornito dalla piattaforma Simulink disaccoppiando la logica di controllo del drone Parrot

già presente nel pacchetto e sostituendola con il controllore MRAC. Diverse simulazioni tramite Simulink vengono presentate al termine della tesi. In particolare si evidenzia la capacità del drone di eseguire un tracking del comando di riferimento efficace. Tali simulazioni vengono confrontate con un controllore PID a cascata di tipo classico, evidenziando la bontà del MRAC nel caso di disturbi costanti o turbolenti di intensità elevate. Tale progetto contribuisce alla sperimentazione sui controllori adattivi mostrandone le potenzialità nel caso dei droni e fornendo un metodo efficace per bilanciare i disturbi con la possibilità di un ulteriore sviluppo per il passaggio su hardware reale.

Indice

Elenco delle tabelle	7
Elenco delle figure	8
1 Introduzione	10
1.1 L'avvento dei droni	10
1.2 Contesto della tesi	10
1.3 Outline tesi	11
1.4 Storia dei droni	12
1.4.1 Ambito Militare	12
1.4.2 Ambito Civile	14
2 Aspetti preliminari	17
2.1 Tipologie di droni	17
2.2 Struttura del drone	19
2.2.1 Comandi	20
2.3 Sistemi di Riferimento	21
2.3.1 Earth Centered Earth Fixed	21
2.3.2 Earth Centered Inertial	21
2.3.3 North, East &Down	22
2.3.4 Body	22
2.3.5 Wind Axes	22
2.3.6 Angoli di Eulero	23
2.4 Rotazioni Elementari	24
2.4.1 Rotazione da Body a Wind Axes	25
2.4.2 Rotazione da NED a ECEF	26
2.5 Quaternioni	26
3 Dinamica completa del drone	29
3.1 Dinamica non lineare	29
3.1.1 Equazioni della Cinematica	29
3.1.2 Equazioni della Dinamica traslazionale	29

3.1.3	Equazioni della Dinamica rotazionale	30
3.1.4	Equazione dinamica assi NED	31
3.1.5	Equazione Forze e Momenti di controllo	31
3.1.6	Equazioni dinamica complete	32
4	Stato dell'arte teoria dei controlli	33
4.1	Introduzione	33
4.2	Data Driven Control DDC	34
4.2.1	SPSA (Simultaneous Perturbation Stochastic Approximation)	35
4.2.2	MFAC (Model Free Adaptive Control)	36
4.2.3	Unfalsified control methodology (UCM)	37
4.2.4	Reinforcement Learning (RL) e Neural Network (NN)	38
4.3	Model Based Control (MBC)	43
4.3.1	Controlli classici	44
4.3.2	Controlli Adattivi	51
4.3.3	Esempi architetture adattive	59
5	Controllore MRAC per drone	63
5.1	Drone Parrot Mambo	63
5.2	Architettura generale	64
5.3	Plant non lineare	65
5.4	Modello di riferimento - Plant Lineare	67
5.5	Stabilizzazione Model Reference LQR	70
5.5.1	Esempio 1: Rapporto $\frac{Q}{R}$ alto	70
5.5.2	Esempio 2: Rapporto $\frac{Q}{R}$ basso	71
5.5.3	Modello di close loop Model Reference	72
5.6	Legge di controllo e legge adattiva	74
5.7	Disturbi	77
5.7.1	Disturbi Parametrici	77
5.7.2	Disturbi non parametrici	78
5.8	Simulazioni Proposte	79
5.8.1	Condizioni iniziali	80
5.8.2	Simulazioni senza disturbi	80
5.8.3	Simulazioni con disturbi a raffica costante	85
5.8.4	Simulazioni con disturbi parametrici non lineare	86
6	Modifiche controllore MRAC	89
6.1	σ - Modification	89
6.1.1	Simulazione 1 - Sigma mod.	91
6.1.2	Simulazione 2 - Sigma mod.	92
6.2	Dead Band	94
6.2.1	Simulazione 1 - Dead Band	96

6.3	Parameter Projection	98
6.3.1	Simulazione 1 - Parameter Projection	100
6.3.2	Simulazione 2 - Parameter Projection	101
6.4	C-MRAC (Composite MRAC)	103
6.5	Osservatore su Model Reference	106
6.5.1	Sistema Aumentato	106
6.5.2	Model Reference con osservatore	107
6.5.3	Simulazione 1 - Osservatore	108
6.5.4	Simulazione 2 - Osservatore	110
7	Pacchetto Parrot Minidrone Simulink	113
7.1	Descrizione Modello	114
7.1.1	Simulation Model - Modello del drone	115
7.1.2	Simulation Model - Enviroment	122
7.1.3	Simulation Model - Sensoristica	123
7.1.4	Flight Simulation - Visualizzazione risultati	123
7.1.5	Flight Command	124
7.1.6	Flight Control System	125
8	Controllore baseline PID a cascata	127
8.1	Architettura PID	128
8.1.1	Tuning PID	128
9	Simulazioni di confronto	133
9.1	Simulazione ad 8 senza disturbi	133
9.2	Simulazione ad 8 con disturbi intensivi	137
9.3	Simulazione Spirale con disturbi	141
10	Conclusioni e sviluppi futuri	145
A	Derivazione legge controllo	147
A.1	Legge di controllo base	147
A.2	σ - modification	148
A.3	Algoritmo Simulink per calcolo di Forze e Momenti	148
	Bibliografia	151

Elenco delle tabelle

5.1	Parametri fisici Parrot Mambo [13]	63
5.2	Autovalori, Smorzamento e frequenza naturale sistema Close Loop .	73
8.1	Parametri del metodo di Ziegler-Nichols per diversi tipi di controllo	129
8.2	Coefficienti controllore PID	131

Elenco delle figure

2.1	Esmpi angolo di Eulero per drone	24
3.1	Forze agenti sul drone	31
4.1	Model Based vs Data Driven	33
4.2	Data Driven Control	35
4.3	UCM	37
4.4	Figura (a) sigmoide, Figura (b) tangente iperbolica	41
4.5	Figura (a) ReLU, Figura (b) Binary Step	41
4.6	MPC Esempio	45
4.7	Sliding Surface	46
4.8	Schema H_∞	47
4.9	Schema PN	49
4.10	Esempio potenziale APF	50
4.11	Principali architetture per Parameter Estimation (PE)	52
4.12	Rappresentazione grafica Gradiente	53
4.13	Architettura L_1	59
4.14	Architettura MRAC	61
5.1	Parrot Mambo	63
5.2	Architettura MRAC Completa	64
5.3	Plant Simulink	66
5.4	Model Reference Simulink	69
5.5	Dinamica traslazionale Q/R alto	70
5.6	Dinamica rotazionale Q/R alto	71
5.7	Dinamica traslazionale Q/R basso	71
5.8	Dinamica rotazionale Q/R basso	72
5.11	Legge adattiva per \hat{K}_x	76
5.12	Legge di Controllo (5.50)	76
5.15	Simulazione 3D	81
5.18	Simulazione 3D	83
5.21	Simulazione 3D	84
6.1	Sigma modification	90
6.2	Sigma modification	90
6.9	Dead Band	94

6.10	Dead band	95
6.11	Dead band	95
6.16	Proiezione del Parametro	99
6.17	Funzione parameter projection	99
6.24	C-MRAC	103
6.25	C-MRAC	103
6.26	Filtro	104
6.27	Y cappello	104
6.28	Y	104
6.29	Osservatore su Model Reference	108
7.1	Modello Lineare - Non Lineare	115
7.2	Modello non lineare	115
7.3	Blocco AC modedel	117
7.4	Blocco forza di gravità	118
7.5	Blocco Resistenza Aerodinamica	118
7.6	Blocco resistenza aerodinamica 2	119
7.7	Forze e Momenti motore	119
7.8	Blocco MotorsToW	119
7.9	Blocco For Each Subsystem	120
7.10	Bus Creator	121
7.11	Position on Earth	121
7.12	Blocco Enviroment	122
7.13	Blocco Enviroment Constante	122
7.14	Blocco sensori	123
7.15	Blocco sensori	123
7.16	Blocco Flight Simulation	124
7.17	Blocco Visualization System	124
7.18	Blocco Command	125
7.19	Blocco FCS	126
7.20	Blocco FCS	126
8.1	PID a cascata	127
A.1	Algoritmo per Forze e Momenti	149

Capitolo 1

Introduzione

1.1 L'avvento dei droni

Negli ultimi decenni l'avanzamento tecnologico ha reso possibile uno sviluppo rapido del settore aerospaziale aprendo nuove frontiere in modo trasversale in settori della vita economica, sociale e militare. Una rilevanza primaria è stata assunta con il tempo dai droni, anche detti velivoli a pilotaggio remoto o più comunemente UAV (Unmanned Aerial Vehicles). Questa categoria particolare di velivoli ha la caratteristica di volare in autonomia senza la necessità di un operatore presente fisicamente a bordo. Questa particolarità rende possibile la riduzione sostanziale di peso data dall'assenza di persone a bordo e dei relativi sistemi ad essi legati, come ECS (Environmental Control System) e tutta l'avionica d'interfaccia velivolo-pilota. Ciò ha diverse implicazioni come la riduzione dei volumi. Il velivolo non dispone di baie adibite ai sistemi non più utilizzati e tanto meno della cabina di pilotaggio. Ciò rende possibile la creazione di droni piccoli con un'integrazione strutturale più semplice. Spazi ridotti consentono infatti guadagni notevoli a livello di progettazione strutturale, le cui componenti fisiche necessitano di aree e spessori inferiori avendo carichi distribuiti da sopportare più contenuti. Ciò rende il drone più facile e veloce da progettare con minor costi e tempi di sviluppo e produzione inferiori rispetto ai velivoli standard.

1.2 Contesto della tesi

Il seguente progetto di tesi ha l'obiettivo la realizzazione di un controllore adattivo per quadrotore in Simulink. La necessità di tale sviluppo nasce dalle svariate funzionalità che questa tecnologia richiede attualmente. E' infatti imprescindibile ottenere dei sistemi di controllo più efficaci per poter ampliare lo spettro di missione del drone, riuscendo ad operare in ambienti più complessi e in presenza di disturbi. E' questo aspetto in particolare che viene maggiormente affrontato in

questo progetto, ricavando diversi modelli di disturbi aerodinamici per poter simulare caratteristiche della dinamica differenti. Nel moderno scenario di sviluppo tecnologico si propone codesto lavoro al fine di ispezionare il mondo dei controllori adattivi.

La seconda parte del progetto di tesi si occupa dello sviluppo di un controllore di tipo MRAC. Quest'ultimo basa la sua funzionalità sul tendere a far coincidere la dinamica reale del drone con quella di un modello di riferimento. Nel caso presente si utilizza come modello di riferimento la dinamica linearizzata del drone. La legge adattiva che caratterizza i parametri del controllore viene ricavata tramite analisi di Lyapunov. A seguire si indicano le principali modifiche apportate alla legge di controllo per aumentarne la robustezza e migliorarne le prestazioni. Infine si studia un controllore PID a cascata da usare come riferimento e si propongono diverse simulazioni di confronto.

1.3 Outline tesi

La tesi viene organizzata in otto capitoli differenti, in cui i primi 4 si riferiscono all'architettura generale di un drone e dei controllori. La seconda parte tratta l'architettura MRAC e il modello Simulink del controllore. In particolare si evidenziano le sezioni:

- **Storia dei droni**, con uno sguardo sia all'ambito civile che militare.
- **Aspetti Preliminari**, contiene le informazioni base sull'architettura di un drone, i comandi, i sistemi di riferimento e gli angoli d'assetto.
- **Dinamica**, è la sezione che contiene tutte le equazioni usate nello sviluppo del modello di riferimento per MRAC. In particolare le equazioni vengono prima ricavate non lineari e poi linearizzate durante la trattazione del model reference.
- **Teoria dei controlli**, tratta i due approcci principali alla controllistica del tipo Model Based o Data Driven, fornendo i principali spunti di riferimento per avere una visione d'insieme sulle principali architetture controllistiche. In particolare si conclude tale sezione con i controlli adattivi con introduzione all'architettura MRAC.
- **Controllore MRAC**, in questa sezione si presenta l'architettura MRAC sviluppata in Simulink con il relativo modello matematico. Si introduce inoltre un'analisi dei disturbi che verranno poi utilizzati in simulazione.
- **Modifiche Controllore**, si propongono le modifiche principali fatte alla legge di controllo iniziale per migliorarne le prestazioni. In particolare si mostra Sigma Modification, Dead-band, Parameter Projection, Osservatore e C-MRAC.

- **Pacchetto Parrot Minidrone**, mostra i principali blocchi utilizzati per il plant, tratti dal progetto Simulink Parrot Minidrone, per ottenere simulazioni high fidelity.
- **Controllore PID**, la sezione è dedicata al controllore PID a cascata utilizzata per il confronto.
- **Simulazioni e Conclusione** in cui si mostrano i principali risultati ottenuti.

1.4 Storia dei droni

1.4.1 Ambito Militare

Uno dei settori in cui questa tecnologia ha raccolto maggior interesse è l'ambito militare. Fin dai primi anni del '900, in particolare con la prima guerra mondiale e poi la seconda, era diventato chiaro come un velivolo "autonomo" in grado di essere controllato a distanza potesse fare la differenza sul campo di battaglia. L'idea a cui fin da subito si è incominciato a pensare era caricare su una sorta di drone un ordigno esplosivo, in grado di detonare all'impatto. L'assenza di un pilota infatti avrebbe ridotto al minimo le perdite a discapito del nemico che avrebbe subito un attacco da cui, soprattutto all'epoca, era particolarmente difficile difendersi. La semplicità concettuale e la possibilità di produrre in serie tali armi rendevano il drone un'arma potenzialmente letale e facilmente sfruttabile su larga scala.

Fu così che si incominciò a progettare velivoli sperimentali autonomi come il Kettering Bug, un antenato dei moderni missili da crociera, progettato a partire dalla prima metà degli anni '10 al fine di colpire obiettivi a 120 km di distanza. Il progetto era concettualmente semplice: creare un drone ad ala fissa in grado di trasportare circa 80 kg di esplosivo lungo la tratta voluta. Per fare questo il progettista Charles Kettering inventò per conto dell'esercito americano un velivolo a doppia ala, con un motore ad elica. L'idea era di utilizzare un motore con 4 cilindri, a 2 tempi da 40 cavalli di produzione Ford con un costo esiguo, essendo questo già prodotto per automobili in larga scala. La parte più critica era far sì che l'arma arrivasse a destinazione. Per far ciò si implementò un sistema rudimentale di navigazione basato su giroscopi meccanici con sistema di controllo pneumatico. Prima della partenza i tecnici calcolavano il numero di rivoluzioni che il motore avrebbe dovuto fare per arrivare a destinazione, tenendo in considerazione anche fattori atmosferici come la presenza di vento. Una volta effettuati i giri predisposti, il motore si bloccava, le ali si sganciavano e la piccola fusoliera con a bordo l'esplosivo incominciava una traiettoria balistica andando a colpire il suolo sottostante. Questo velivolo autonomo aprì gli studi per gli albori della tecnologia di volo autonomo, ma non venne utilizzato in battaglia non essendo sufficientemente sicuro per poter trasportare esplosivi al di sopra delle linee alleate.

Di tutt'altra taratura si dimostrò essere la V1 (Vergeltungswaffe 1) progettata dagli ingegneri nazisti per la Luftwaffe nella seconda guerra mondiale. Questo ordigno volante era anch'esso un piccolo aereo fabbricato in lamiera d'acciaio con una struttura semplificata all'osso. Grazie a una rampa di lancio era in grado di prendere il volo per tragitti fino a 250 km e con la capacità di carico di 850 kg. L'aereo aveva un sistema di controllo autonomo basato su un giroscopio e un altimetro. Anche in questo caso (come per il Kettering Bug) un contatore meccanico aveva la funzione di interrompere l'alimentazione al motore una volta raggiunto il numero di giri preimpostato, fornendo la possibilità ai tecnici di terra di impostare prima del lancio la distanza voluta. Il motore era atipico per i velivoli dell'epoca. Esso infatti sfruttava la tecnologia a pulsogetto, con impulsi creati dalla combustione del carburante iniettato a intermittenza, creando il tipico rumore pulsante. Questa soluzione ingegneristica era particolarmente sfavorevole per il volo pilotato, in quanto estremamente difficile da regolare ma risultò particolarmente efficace per la V1 in quanto aveva come obiettivo un volo perlopiù a velocità costante prima dello schianto. La V1 ebbe un vasto impiego da parte dell'Asse contro gli Alleati, in particolare nel bombardare Londra. Gli studi su questo velivolo vennero ben presto soppiantati dallo sviluppo delle V2, un vero e proprio missile balistico a lunga gittata. Arrivando agli anni '50 e '60 si assiste all'avvento delle moderne tecnologie per droni a pilotaggio remoto, soprattutto a causa della guerra fredda. Obiettivo principale era quello di creare un prototipo di jet a pilotaggio remoto da usare come target per addestrare i nuovi piloti e militari di terra. A tal fine incomincia la realizzazione del Firebee da parte della Ryan Aeronautical Company per conto dell'Aeronautica degli Stati Uniti. Il primo modello fu il Q-2 Firebee completato nel 1951. Il drone aveva propulsione jet e poteva essere immesso in volo attraverso un razzo ausiliario, rampa di lancio o tramite un aereo d'appoggio a cui veniva agganciato al ventre dell'ala e successivamente rilasciato in volo. Le caratteristiche rivoluzionarie per l'epoca furono la capacità di raggiungere i 900 km/h e la possibilità di essere radiocomandato da terra o di percorrere un percorso prestabilito. La versatilità delle sue configurazioni portò alla realizzazione di numerose versioni di questo velivolo, che perdurarono nei decenni rendendolo uno tra i droni più utilizzati al mondo in campo bellico. Le successive iterazioni portarono allo sviluppo di nuovi sistemi di controllo, fino al GPS moderno, con una larga categoria di configurazioni con differenti strumenti di ricognizione, raccolta dati, sensoristica. Il suo impiego fu largamente apprezzato anche durante la guerra del Vietnam grazie alla sua capacità di impieghi in missioni rischiose per ricognizione a bassa quota.

1.4.2 Ambito Civile

Bisogna però aspettare gli anni '90 per veder crescere un interesse per l'ambito civile. L'avanzamento tecnologico permette di pensare e ideare droni per nuovi impieghi quali agricoltura e riprese. In particolare con l'avvento del GPS si incomincia ad ipotizzare un utilizzo in campo agricolo al fine di trattamenti da prima generici e successivamente localizzati. Fino agli anni 2000 era infatti appannaggio di piccoli elicotteri la funzione di spargere, tramite appositi diffusori, trattamenti di prevenzione malattie su campi agricoli. Questa soluzione permetteva di abbattere i tempi necessari allo spargimento del prodotto, anche in zone difficilmente praticabili quali colline ad alta pendenza, a discapito di una precisione assai limitata con una buona percentuale di spreco del trattamento. L'avvento dei droni permise di abbattere i costi di mantenimento, inferiori a quelli di un elicottero, con ancora una precisione e autonomia limitata. Quest'ultimo punto in particolare è tuttora uno dei principali ostacoli all'utilizzo dei droni in modo intensivo in campo agricolo. Altri settori in cui ha avuto nettamente più successo fin dai primi anni del 2000 sono le riprese. La riduzione delle dimensioni e pesi di videocamere e l'aumento della risoluzione ha reso possibile riprendere a basso costo in volo, andando a salvare i filmati su dispositivi di archiviazione sempre più piccoli e prestanti. E' così che incominciarono a prendere piede le riprese con droni per scopi pubblicitari, cinematografici e amatoriali. La versatilità dei droni multicopter rese possibile portare in volo telecamere di stampo cinematografico, le quali fino a pochi anni prima erano montate esclusivamente su piccoli elicotteri con tutti i problemi che ciò comportava (soprattutto a livello di costi e di addestramento personale). Ad oggi la varietà di telecamere a disposizione permette non solo semplici riprese ma anche un monitoraggio puntuale di siti di interesse al fine di sorveglianza, prevenzione e manutenzione. I droni vengono infatti sfruttati per sorveglianza di aree estremamente affollate, accoppiati con software di intelligenza artificiale in grado di individuare tempestivamente eventuali pericoli. E' inoltre possibile monitorare impianti industriali, sfruttando telecamere termiche e sensori di movimento avanzato per evidenziare vibrazioni indesiderate che con il tempo potrebbero provocare danni a macchinari e strutture. Vi sono inoltre applicazioni in ambito "ritrovamento e salvataggio" per protezione civile e settori sanitari, ad esempio in caso di disastri ambientali. Le funzioni di mappatura sono largamente utilizzate ormai da anni da figure quali geometri e architetti, per misurare e modellare in poco tempo vaste aree di territorio. Altri ambiti di particolare importanza e tutt'ora in sviluppo sono delivery autonomo e agricoltura di precisione. Entrambi hanno la grossa problematicità di dover volare in modo autonomo in ambienti estremamente complessi. Devono infatti essere in grado di prevedere e evitare collisioni con strutture fisiche e persone, oltre ad essere in grado di raccogliere e analizzare in tempo reale grandi quantità di informazioni relative all'ambiente circostante (ad esempio comprendere quali sono le piante malate in

modo da agire selettivamente su di esse). Questi ambiti sono ad oggi fortemente in via di sviluppo grazie anche all'avvento di tecnologie predittive sempre più accurate.

Capitolo 2

Aspetti preliminari

2.1 Tipologie di droni

Modernamente i droni possono essere categorizzati in diversi modi, principalmente in base all'utilizzo, all'architettura o alla missione. In questo caso ci si concentra sui droni volanti, andando perciò ad escludere tutte le forme di veicoli autonomi da terra e acqua. Una delle classificazioni più esaustive è quella basata sull'architettura, ovvero sulla geometria del drone ed in particolare sul suo modo di ottenere sia la propulsione che le forze di sostentamento quali la portanza. A tale fine si individuano i droni: ad ala rotante, ad ala fissa, ibridi e ad ala battente.

I droni ad **ala rotante** moderni sono formati da 4 o più rotori rivolti verso l'alto. Questa configurazione permette una certa libertà progettuale oltre che a un' intrinseca manovrabilità, maggiore rispetto ad altre architetture. In genere i modelli più prodotti e diffusi sono i quadricotteri, formati da 4 rotori con geometria a croce. Le eliche ruotano in verso opposto in modo incrociato, ovvero due in senso orario e due antiorario, garantendo in casi nominali lo stazionamento in hover. Qualora fosse necessario aumentare le dimensioni o il carico è possibile realizzare droni multicottero con 6 o 8 eliche, in grado di esercitare forze maggiori e conseguentemente rendendoli più adatti su lunghi percorsi o per brevi tratte ma con payload elevati. Questo tipo di architetture richiedono però un dimensionamento opportuno anche delle componenti strutturali, spesso realizzate in fibre di carbonio, con il relativo aumento di costi. Per quanto concerne i quadricotteri è di notevole rilevanza l'utilizzo bellico che si è imposto di recente a causa dei numerosi conflitti. Droni di tipo FPV (First Person View) permettono a bassissimi costi di montare telecamere connesse a stazioni di terra integrate in semplici visori e joystick. In questo modo, caricando un piccolo ordigno esplosivo, è possibile colpire un target fisso o in movimento a distanze piccole-medie, garantendo comunque un campo di sicurezza eccellente per l'operatore a terra. Inoltre tali dispositivi, qualora pilotati da persone esperte, diventano estremamente difficili da evitare o neutralizzare, rendendoli perciò efficaci.

Per droni ad **ala fissa** si intende invece velivoli la cui portanza viene ottenuta, come negli aerei, tramite l'interazione dell'aria relativa al velivolo con un'ala fissa, che crea portanza. Questa tipologia ha una complessità e un costo potenzialmente inferiori rispetto al multirotores, con un'efficienza di volo migliore, adibito alla percorrenza di lunghe distanze. In questo caso è solitamente sufficiente un singolo motore collegato ad un' elica. Il drone viene spesso lanciato tramite rampa, agevolando notevolmente lo spazio di decollo e portandolo in volo con una velocità sostenuta senza dispendio di energia. Successivamente sarà l'elica insieme alle ali a fornire la spinta e portanza necessarie per il suo sostentamento per lunghe tratte. Questa tecnologia ha il grosso limite di impedire lo stazionamento a volo fisso (hover) nè tanto meno il volo all'indietro, e in generale manovrabilità in spazi ristretti. Gli utilizzi principali sono incentrati su delivery, avendo un'efficienza aerodinamica maggiore del multirotores, ma anche e soprattutto sorveglianza e altri scopi bellici. Di notevole rilevanza sono i droni Shahed di fabbricazione iraniana utilizzati nel conflitto Russo-Ucraino e i droni più costosi e sofisticati Bayraktar di fabbricazione turca. I primi sono particolarmente economici, dotati di rampa di lancio non necessitano di spazi adibiti a runway per il loro decollo e sono in grado di trasportare testate esplosive fino a 50 kg per 2500km. I secondi vengono invece utilizzati principalmente per scopo di sorveglianza in alta quota, essendo dotati di apparecchiature sofisticate per ricognizione.

I droni **ibridi**, anche definiti V-TOL (Vertical Take Off Landing) sfruttano una tecnologia nata dalla commistione delle due architetture precedenti. Un'ala fissa, che garantisce efficienze aerodinamiche alte, è accoppiata a dei tilt rotor, ovvero rotori in grado di ruotare rispetto l'asse trasversale. In questo modo è possibile indirizzare le eliche per ottenere un decollo verticale e successivamente inclinarle gradualmente lungo l'orizzontale ottenendo la conversione ad un vero e proprio drone ad ala fissa. Questi velivoli hanno una notevole complessità data sia dall'aggravio di peso e spazio occupato dal cinematismo predisposto alla rotazione, sia dalle complesse interazioni aerodinamiche della struttura durante le fasi intermedie di rotazione. Per questo motivo non sono diffusi in ambito civile.

Infine un ultimo tipo di architettura che vale la pena nominare è il drone ad **ala battente**. Quest'ultimo nasce dall'imitazione delle ali battenti presenti in natura, prendendo come esempio insetti e volatili. Il loro sviluppo è agli albori, a causa dell'elevata complessità cinematica dell'ala e dallo studio delle interazioni aerodinamiche. Le caratteristiche principali di questi droni sono le dimensioni ridotte, con capacità di carico veramente esigue, ma con una propulsione estremamente silenziosa. Per questo motivo l'unico settore su cui attualmente ci si sta focalizzando è il campo dello spionaggio, in cui piccole dimensioni e rumori contenuti sono requisiti fondamentali della missione di volo.

2.2 Struttura del drone

Si vogliono ora analizzare le principali componenti di un drone quadricotore, le quali in linea generale, possono essere ricondotte ad un'architettura più comune anche per altri tipi di drone. La componente strutturale è denominata Frame (Telaio), solitamente realizzata in materiale plastico per piccoli droni, in composito come la fibra di carbonio per strutture più performanti, con componenti in lega di alluminio per conservare alte prestazioni e pesi contenuti. Al telaio vengono fissate tutte le componentistiche necessarie al volo:

- **Flight controller**, svolge le principali funzioni di calcolo, ricezione e distribuzione dei dati funzionali al volo. Al controllore di volo vengono collegati i sensori, le antenne, i motori tramite ESC, e la scheda di distribuzione di potenza (PDB). Quest'ultima può essere già integrata nella scheda di controllo ma sovente viene collegata separatamente per garantire una componente hardware intermedia tra la batteria e le restanti componenti del drone. Compito del Flight controller è raccogliere i dati provenienti dai sensori, antenne ed eventuali telecamere e calcolare i comandi da mandare ai motori per garantire la dinamica di volo desiderata. Hardware comuni sono prodotti dalla Holybro come la serie Pixhawk o Kakute o ancora da DJI e Matek, in particolare per droni FPV. Firmware comuni sono invece Ardupilot, PX4 o Betaflight. Alcuni di questi come PX4 sono open source e perciò particolarmente utilizzati soprattutto nell'ambito di ricerca e sviluppo.
- **Motori**, sono solitamente di tipo brushless. Questa tipologia permette di avere rendimenti maggiori a fronte dell'assenza di spazzole. Sono inoltre più leggeri, meno soggetti a manutenzione ed in generale più affidabili.
- **Eliche**, realizzate in plastica o fibra di carbonio possono avere geometria leggermente differente a seconda delle prestazioni aerodinamiche volute. In studio sono forme atipiche come i toroidi, al fine di ridurre l'inquinamento acustico rendendo il drone più silenzioso.
- **Power Distribution Board (PDB)**, ha il compito di distribuire l'alimentazione in modo consono a tutti i componenti del drone. Può contenere fusibili per garantire la necessaria sicurezza di tutte le componenti.
- **Electronic Speed Controller (ESC)**, ha la funzione di regolare la velocità dei motori brushless. Per questo motivo ha 3 attacchi, uno ai motori, uno alla PDB e uno al controllore che regolerà la tensione per ogni singolo motore in modo opportuno.
- **Sistemi di comunicazione**, classicamente composto da un semplice radio-comando (RC) con antenna a 2.4 GHz o 5.8 GHz a seconda degli standard.

Modernamente esistono diversi tipi di antenne in grado di funzionare anche da radio telemetrica, in grado di comunicare alla ground station dati relativi alle condizioni di volo e allo stato del drone, e altre radio ai fini di riconoscimento di tipo broadcast, fondamentali per droni di grandi dimensioni che operano in spazi aerei complessi (simile all'ADS-B dei velivoli classici).

- **Sensoristica:** la maggior parte dei sensori utili al volo sono solitamente integrati nel controllore di volo. Essi sono giroscopio (per valutare l'assetto), accelerometro e barometro (per calcolare la quota). Sensori aggiuntivi sono il GPS per la navigazione o sensori ottici quali il Lidar per Obstacle Avoidance. In generale un gran numero di sensori aggiuntivi possono essere integrati nel drone a seconda della missione di volo, come sensori infrarossi per ambienti notturni o termici per ambiti di cerca persone e soccorso.
- **Batteria,** le più utilizzate di tipo LiPo (Lithium Polymer) o agli ioni di litio. Le batterie sono in generale una componente il cui dimensionamento è critico perchè inficiano pesantemente le performance del drone. Esse infatti hanno un peso molto rilevante e vincolano i range massimi di percorrenza e le funzionalità delle componenti hardware a bordo.
- **Stazione di Terra (Ground Control),** è componente fondamentale per l'interfaccia tra operatore e drone. In generale può essere sia un computer che un dispositivo fisso o portatile dedicato (come ad esempio un tablet o simili). La stazione di terra deve avere un trasmettitore e ricevitore che si interfaccia con il controllore di volo a bordo del drone. In questo modo è possibile visionare dati di telemetria, relativi a stato e posizione del drone, batteria ecc.. Inoltre è possibile comandare il velivolo online in tempo reale tramite joystick o predisporre un percorso prefissato tramite navigazione GPS. I software di interfaccia più utilizzati sono Qgroundcontrol, Mission Planner o software di produzione DJI come DJI GS Pro.

2.2.1 Comandi

Il drone può essere comandato manualmente e i comandi principali sono 4, perciò facilmente implementabili anche su un semplice joystick da gaming:

- **Collettivo,** permette di comandare contemporaneamente le velocità dei 4 motori elettrici. Variando i giri al secondo di ogni rotore allo stesso modo, si ottiene maggior o minor spinta verso l'alto, permettendo al drone di spostarsi verso l'alto o verso il basso mantenendo invariato l'assetto.
- **Roll,** permette lo spostamento laterale tramite attuazione simmetrica dei rotori (Spinta maggiore rotori a destra comporta spostamento a sinistra e viceversa)

- **Pitch**, permette lo spostamento longitudinale tramite attuazione simmetrica dei rotori (Spinta rotori retrostanti comporta spostamento positivo in avanti e viceversa)
- **Yaw**, permette la rotazione lungo l'asse verticale Z tramite l'attuazione simmetrica dei rotori opposti, ovvero aumentando la velocità dei rotori in senso orario comporta una rotazione in senso antiorario e viceversa. E' da sottolineare infatti che i quattro rotori ruotano a due a due con formazione ad "X" rispettivamente in senso orario (CW) o antiorario (CCW) per garantire stabilità.

Per una trattazione più puntuale si fa riferimento ai capitoli successivi inerenti agli angoli di Eulero.

2.3 Sistemi di Riferimento

In questo capitolo si vuole arrivare alla definizione delle equazioni della dinamica tipiche del drone. Per fare ciò è inanzitutto fondamentale definire l'approccio logico con cui si affronta tale problema. Intuitivamente si vogliono calcolare posizioni e assetto con relative velocità al fine di poterle integrare numericamente nel tempo ed ottenere perciò dei valori puntuali per ogni istante di volo. Per fare questo è necessario definire i sistemi di riferimento in gioco, rispetto cui si determinano le forze che agiscono sul drone e le posizioni. Un riferimento deve essere vincolato a terra al fine di navigazione. Per questo si definiscono diversi sistemi di riferimento Terra che possono essere utilizzati a seconda del velivolo in questione e della sua missione di volo.

2.3.1 Earth Centered Earth Fixed

Il sistema ECEF (Earth Centered Earth Fixed) è un sistema vincolato alla Terra e solidale con esso durante il moto di rotazione. L'origine è il centro della terra, l'asse z punta verso il polo nord, l'asse x si ottiene dall'intersezione del piano equatoriale e dal meridiano di Greenwich mentre l'asse y completa la terna destrorsa. Questo sistema di riferimento viene usato per tutti i velivoli che volano in atmosfera, solidali con la Terra. In questo caso infatti la rotazione terrestre non può essere trascurata ai fini di navigazione. Dal sistema ECEF è possibile ricavare rispettivamente latitudine e longitudine con gli angoli ϕ e λ .

2.3.2 Earth Centered Inertial

Il Sistema ECI (Earth Centered Inertial) a differenza del precedente non è solidale alla rotazione terrestre ma fisso. Per questo motivo viene solitamente utilizzato come riferimento per i velivoli orbitali, la cui posizione nello spazio è ininfluente

rispetto il moto rotativo terrestre. L'origine degli assi è il centro della Terra, l'asse z punta al polo nord, l'asse x giace sul piano equatoriale e punta all'equinozio di primavera mentre l'asse y completa la terna destrorsa, complanare con x sul piano equatoriale.

I due sistemi sopra citati sono quelli comunemente usati, fissi a Terra, come riferimento. In seguito si valutano invece i riferimenti vincolati al baricentro del velivolo. E' chiaro che la posizione reciproca tra il sistema di riferimento vincolato al corpo e quello vincolato alla Terra, permettono di avere a disposizione uno schema univoco entro cui andare a calcolare tutti i termini di posizione, velocità e accelerazioni necessarie per la navigazione e il controllo.

2.3.3 North, East & Down

Il sistema NED (North, East, Down) ha come origine il baricentro (CoG ovvero Center of Gravity) del velivolo. Come consiglia il nome l'asse z ha direzione down, ossia la direzione del vettore locale della forza di gravità e verso diretto al centro della terra. X e y giacciono invece sul piano tangente alla superficie terrestre. L'asse x punta in direzione nord mentre y completa la terna destrorsa in direzione est. Questo sistema di riferimento è il più comune utilizzato come fisso (inerziale) per il volo, compreso l'ambito droni. Usando come sistemi di riferimento il NED e l'ECEF è infatti possibile passare da uno all'altro semplicemente con due rotazioni, una intorno a Z_{NED} per allineare Y_{NED} a Y_{ECEF} e la seconda intorno a Y_{NED} che permette di allineare gli altri due assi.

2.3.4 Body

Il sistema body reference frame è un sistema vincolato al corpo del velivolo e cambia perciò orientamento con esso. L'origine del sistema coincide con il CoG, l'asse $X_{body}(x_b)$ coincide con l'asse longitudinale del velivolo, la y_b con l'asse laterale e z_b chiude la terna destrorsa, rivolto verso il basso. Ne consegue che dal sistema body è possibile ottenere il NED con tre rotazioni rispetto a z_b , y_b e x_b per ottenere il frame di navigazione. Nel caso del drone l'asse x_b determina la direzione della velocità in "avanti", ed è contenuto nel piano di simmetria. Tali assi sono inoltre diretti lungo le principali direzione d'inerzia rendendo la matrice d'inerzia un tensore diagonale. Si osserva che tra il body frame e il NED frame intercorrono i tre angoli di Eulero specificati in seguito.

2.3.5 Wind Axes

Un altro sistema particolarmente importante, soprattutto per i droni ad ala fissa, è il sistema di riferimento ad assi vento (wind axes). In questo caso l'origine degli assi rimane il CoG ma la direzione e verso non è vincolata alla rotazione della struttura

velivolo, bensì alle direzioni in cui si esercitano le principali forze aerodinamiche determinate dal vento relativo. In particolare la $x_{wind}(x_w)$ è la direzione lungo cui si manifesta la resistenza aerodinamica (drag), la z_w ha direzione lungo la portanza (lift) e verso opposto, mentre la y_w chiude la terna destrorsa, coincidente con y_b . Questo sistema di riferimento è utilizzato per valutare le forze aerodinamiche ma non la dinamica, che solitamente viene valutata in assi body essendo quest'ultimi vincolati al velivolo. Nel caso dei droni ad ala rotante tale riferimento non viene utilizzato in quanto non vi è la classica lift esercitata sull'ala, ma una lift esercitata dal rotore, la cui direzione è parallela a Z_b . Per la rotazione da body axes a wind axes sono sufficienti due rotazioni rispettivamente lungo Z_b e y_b . Questo sistema di riferimento individua 2 angoli notevoli tra il riferimento body e vento: il primo angolo è β , chiamato Side Slipe Angle o angolo di derapata, ed è l'angolo di imbardata che allinea i due frame lateralmente; il secondo angolo è α ossia angolo di incidenza, una rotazione di beccheggio che allinea i due sistemi di riferimento longitudinalmente.

2.3.6 Angoli di Eulero

Per definire univocamente l'assetto e le rotazioni di un drone si possono comunemente utilizzare gli angoli di Eulero o i quaternioni. Si affronta per primo il caso degli angoli di Eulero specificando successivamente le problematiche relative a questo approccio e per quale motivo, soprattutto in casi di angoli di rotazione elevati, sia preferibile utilizzare i quaternioni. Gli angoli di Eulero sono 3 angoli che descrivono l'orientamento di un corpo rigido. Nel caso del drone quadrotore le tre rotazioni intercorrono tra il riferimento inerziale NED e il riferimento body e sono:

- **Rollio** (Roll), è la rotazione rispetto all'asse body x_b di un angolo chiamato angolo di rollio ed indicato con la lettera ϕ . Nel drone la rotazione rispetto a ϕ è direttamente legata allo spostamento laterale. L'aumento di velocità dei rotori di sinistra porta il velivolo ad inclinarsi con angolo ϕ positivo ottenendo uno spostamento positivo lungo l'asse y. Il viceversa avviene aumentando la velocità dei rotori di destra.
- **Beccheggio** (Pitch), è la rotazione rispetto all'asse y_b di un angolo definito angolo di beccheggio ed indicato con la lettera θ . La rotazione rispetto quest'angolo è legata allo spostamento longitudinale in "avanti" o "indietro". In particolare l'aumento della velocità di rotazione dei due rotori posteriori comporta la creazione di un angolo θ negativo con il conseguente movimento in "avanti" lungo x positivo. Attuando viceversa i rotori frontali si ottiene il movimento all'indietro.

- **Imbardata** (Yaw), è la rotazione lungo l'asse z_b di un angolo definito come angolo di imbardata e rappresentato dalla lettera ψ . L'attuazione contemporanea dei soli rotori CW o CCW consente di ottenere una rotazione antioraria o oraria rispettivamente.

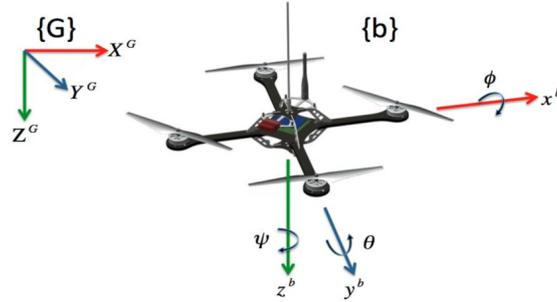


Figura 2.1: Esmpi angolo di Eulero per drone

2.4 Rotazioni Elementari

Per passare da un sistema di riferimento ad un altro si utilizzano classicamente 3 rotazioni: una su ψ , una su θ e una su ϕ . Così facendo si passa attraverso due sistemi di riferimento intermedi che vengono in seguito definiti un apice (‘) o due apici (‘’) se si indica rispettivamente il primo sistema transitorio o il secondo. Le rotazioni vengono intese positive seguendo la regola della mano destra (ovvero rotazioni antiorarie positive). Partendo dal generico sistema di riferimento 2 (SR2) per giungere al sistema di riferimento 1 (SR1) si eseguono le seguenti rotazioni:

- Rotazione di un angolo ψ lungo Z_{SR2} ottenendo il nuovo sistema SR2’ in cui per costruzione Z_{SR2} coincide con $Z_{SR2’}$ (eq. 2.1);
- Rotazione di un angolo θ lungo $Y_{SR2’}$ ottenendo il nuovo reference frame SR2’’ in cui $Y_{SR2’}$ coincide con $Y_{SR2’’}$ (eq. 2.2);
- Rotazione di un angolo ϕ lungo $X_{SR2’’}$ ottenendo il reference frame finale SR1 in cui $X_{SR2’’}$ coincide con X_{SR1} (eq. 2.3);

Si presentano in seguito le tre matrici di rotazione e le rispettive inverse.

$$[\Psi] = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow [\Psi]^{-1} = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

$$[\Theta] = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \rightarrow [\Theta]^{-1} = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.2)$$

$$[\Phi] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \rightarrow [\Phi]^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \quad (2.3)$$

Svolgendo la sequenza di rotazione 3-2-1 si ottiene:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{SR2} = [\Psi] \cdot [\Theta] \cdot [\Phi] \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{SR1} = [L_{21}] \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{SR1} \quad (2.4)$$

con $[L_{21}]$ che è matrice di rotazione globale da SR1 a SR2 pari a:

$$[L_{21}] = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \psi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \psi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \psi \end{bmatrix} \quad (2.5)$$

Si osserva che per passare dal sistema SR2 a SR1 è possibile utilizzare $[L_{12}] = [L_{21}]^{-1} = [L_{21}]^T$. La rotazione 3-2-1 può essere utilizzata per allineare i sistemi di riferimento Body al sistema di riferimento NED a scopi di navigazione. In questo modo integrando numericamente le posizioni North, East e Down si riesce ad ottenere la traiettoria facilmente interpretabile rispetto la superficie terrestre.

2.4.1 Rotazione da Body a Wind Axes

Per allineare il reference frame ad Assi Vento con quello body sono sufficienti due rotazioni:

- Prima rotazione attorno a Z_b di yaw, con un angolo $\psi = -\beta$ angolo di side slip. In questo modo si allinea l'asse Y_{wind} con Y_b .
- Seconda rotazione attorno a Y_b di un angolo di beccheggio $\theta = \alpha$ angolo di incidenza. In questo modo si allineano i restanti due assi tra il reference frame body e wind.

Le matrici che compongono queste due rotazioni sono le stesse presentate precedentemente, ottenendo così una rotazione 3-2. Si ottiene perciò la matrice di rotazione completa (eq. 2.6):

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix}_{BODY} = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ -\sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \cdot \begin{bmatrix} V \\ 0 \\ 0 \end{bmatrix}_{WIND} \quad (2.6)$$

con $\tan \alpha \simeq w/u$ e $\sin \beta \simeq v/V$.

2.4.2 Rotazione da NED a ECEF

Similmente è possibile passare dal sistema di riferimento NED a ECEF. Anche in questo caso infatti sono sufficienti due rotazioni:

- Prima rotazione intorno a Z_{NED} di un angolo $\psi = \tau$ (latitudine) per allineare gli assi Y
- Seconda rotazione attorno a Y_{NED} di un angolo $\theta = -(90 + \lambda)$ per allineare i restanti assi.

Questa rotazione corrisponde ad una rotazione 3-2 ottenendo la matrice completa:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{NED} = \begin{bmatrix} -\sin \lambda \cos \tau & -\sin \tau & -\cos \lambda \cos \tau \\ -\sin \lambda \sin \tau & \cos \tau & -\cos \lambda \sin \tau \\ \cos \lambda & 0 & -\sin \lambda \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ECEF} \quad (2.7)$$

2.5 Quaternioni

L'approccio basato su angoli di Eulero è adatto per rotazioni ad angoli piccoli, minori di 90 gradi. E' infatti verificabile come per angoli retti il coseno dell'angolo di rotazione tenda a zero rendendo la matrice di rotazione indeterminata (problema denominato Gimbal Lock). Per questo motivo, per applicazioni spaziali che prevedono una possibile rotazione maggiore di un angolo retto è bene utilizzare un metodo di rappresentazione alternativo, come ad esempio i quaternioni. Quest'ultimi sono stati introdotti da Hamilton ed hanno la caratteristica di esprimere l'assetto, di per sé determinato da 3 angoli, con un vettore di 4 elementi. Questo perché l'assetto può essere identificato da una rotazione di un angolo θ intorno ad un asse, rispetto al sistema di riferimento (i,j,k). Tale asse prende la forma 2.8:

$$\mathbf{u} = u_x \mathbf{i} + u_y \mathbf{j} + u_z \mathbf{k} \quad (2.8)$$

Utilizzando la notazione di Eulero si definiscono i quaternioni come:

$$\mathbf{q} = e^{\theta/2(u_x \mathbf{i} + u_y \mathbf{j} + u_z \mathbf{k})} = \cos \frac{\theta}{2} + (u_x \mathbf{i} + u_y \mathbf{j} + u_z \mathbf{k}) \sin \frac{\theta}{2} = \cos \frac{\theta}{2} + \mathbf{u} \sin \frac{\theta}{2} \quad (2.9)$$

Con questa notazione è chiaro che il quaternion unitario è quello che rappresenta una rotazione nulla ed esprime l'alinamento dell'assetto con il sistema di riferimento iniziale. Utilizzando opportuni passaggi matematici è possibile dimostrare la relazione tra i 3 angoli di eulero ϕ , θ , ψ e l'assetto espresso dai quaternioni come (eq. 2.10):

$$\begin{cases} q_0 = \pm(\cos \phi/2 \cos \theta/2 \cos \psi/2 + \sin \phi/2 \sin \theta/2 \sin \psi/2) \\ q_1 = \pm(\sin \phi/2 \cos \theta/2 \cos \psi/2 - \cos \phi/2 \sin \theta/2 \sin \psi/2) \\ q_2 = \pm(\cos \phi/2 \sin \theta/2 \cos \psi/2 + \sin \phi/2 \cos \theta/2 \sin \psi/2) \\ q_3 = \pm(\cos \phi/2 \cos \theta/2 \sin \psi/2 - \sin \phi/2 \sin \theta/2 \sin \psi/2) \end{cases} \quad (2.10)$$

$$\begin{cases} \theta = \sin^{-1}[-2(q_1 q_3 - q_0 q_2)] \\ \psi = \tan^{-1} \left[\frac{2(q_1 q_2 + q_0 q_3)}{q_0^2 + q_1^2 - q_2^2 - q_3^2} \right] \\ \phi = \tan^{-1} \left[\frac{2(q_2 q_3 + q_0 q_1)}{q_0^2 - q_1^2 - q_2^2 + q_3^2} \right] \end{cases} \quad (2.11)$$

La variazione nel tempo dei quaternioni è legata dalle velocità di rollio (p), beccheggio (q) e imbardata (r) tramite la relazione (eq. 2.12):

$$\frac{d}{dt} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} 0 & p & q & r \\ -p & 0 & -r & q \\ -q & r & 0 & -p \\ -r & -q & p & 0 \end{bmatrix} \cdot \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (2.12)$$

E' inoltre possibile valutare l'errore di assetto utilizzando il prodotto Hamiltoniano ottenendo l'errore in quaternioni come:

$$q_e = q_d \otimes q_{true} = \begin{bmatrix} q_{e0} \\ q_{e1} \\ q_{e2} \\ q_{e3} \end{bmatrix} = \begin{bmatrix} q_{d0} & -q_{d1} & -q_{d2} & -q_{d3} \\ q_{d1} & q_{d0} & -q_{d3} & q_{d2} \\ q_{d2} & q_{d3} & q_{d0} & -q_{d1} \\ q_{d3} & -q_{d2} & q_{d1} & q_{d0} \end{bmatrix} \cdot \begin{bmatrix} q_{true0} \\ q_{true1} \\ q_{true2} \\ q_{true3} \end{bmatrix} \quad (2.13)$$

Da osservare che q_d si ottiene con l'espressione:

$$q_d = \frac{(q_{desiderato})^T}{\|q_{desiderato}\|_2} = \frac{[q_0 - q_1 - q_2 - q_3]^T}{(q_0^2 - q_1^2 - q_2^2 - q_3^2)}$$

Questa rappresentazione è largamente utilizzata per descrivere l'assetto di satelliti e velivoli in generale evitando il problema del Gimbal Lock. Integrando inoltre l'errore d'assetto è possibile sfruttare tale formulazione per la definizione di leggi di controllo efficaci proporzionali all'errore in questione.

Capitolo 3

Dinamica completa del drone

3.1 Dinamica non lineare

Si vuole presentare in seguito una veloce descrizione dei passaggi per giungere alle equazioni del moto non lineari. Nei capitoli successivi si determinano le opportune semplificazioni per linearizzare il modello da usare come Model Reference nell'architettura MRAC.

3.1.1 Equazioni della Cinematica

In questa sezione si presenta la relazione tra le velocità angolari p , q ed r nel reference frame inerziale e $\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$ nel sistema di riferimento body [26]. Tale relazione si ottiene considerando tre rotazioni consecutive del tipo 3-2-1 (eq. 3.1):

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = [\Phi] \begin{bmatrix} \dot{p} \\ 0 \\ 0 \end{bmatrix} + [\Phi][\Theta] \begin{bmatrix} 0 \\ \dot{q} \\ 0 \end{bmatrix} + [\Phi][\Theta][\Psi] \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (3.1)$$

Da cui è possibile ricavare la relazione rispetto al riferimento body in cui verranno valutate tutte le equazioni ottenendo:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (3.2)$$

3.1.2 Equazioni della Dinamica traslazionale

Le equazioni della dinamica di corpo rigido possono essere ottenute tramite la prima legge di Newton (eq. 3.3), sfruttando la legge di Coriolis (eq. 3.4):

$$m \cdot \frac{d\mathbf{v}}{dt} = \mathbf{f} \quad (3.3)$$

Con m massa del drone e \mathbf{f} forze applicate. In generale è possibile applicare la relazione:

$$\frac{d}{dt_i} \mathbf{p} = \frac{d}{dt_b} \mathbf{p} + \omega_{b/i} \times \mathbf{p} \quad (3.4)$$

con \mathbf{p} generica grandezza vettoriale di cui si vuole calcolare la derivata temporale tra sistema di riferimento inerziale e body, mentre $\omega_{b/i}$ rappresenta la velocità di rotazione tra i due sistemi e determina la variazione del verso della grandezza \mathbf{p} . Grazie a queste relazioni è possibile ottenere:

$$m \frac{d\mathbf{v}}{dt_i} = m \left(\frac{d\mathbf{v}}{dt_b} + \omega_{b/i} \times \mathbf{v} \right) = \mathbf{f} \quad (3.5)$$

considerando $\mathbf{v} = (u, v, w)^T$ il vettore delle velocità lineari body e $\omega_{b/i} = (p, q, r)^T$ vettore delle velocità angolari tra inerziale e body.

Si osserva che $\frac{d\mathbf{v}}{dt_b}$ corrisponde per definizione alla derivata del vettore in riferimento body rispetto al tempo nel medesimo riferimento ovvero $\frac{d\mathbf{v}}{dt_b} = (\dot{u}, \dot{v}, \dot{w})^T$. Raggruppando tale termine a destra e svolgendo il prodotto vettoriale in eq. 3.6 si ottiene:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \frac{1}{m} \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} \quad (3.6)$$

3.1.3 Equazioni della Dinamica rotazionale

Per derivare le equazioni della dinamica rotazionale è possibile partire dalla legge di conservazione del momento angolare con $\mathbf{m} = (\tau_\phi, \tau_\theta, \tau_\psi)^T$ vettore dei momenti applicato di roll, pitch e yaw. Si ottiene:

$$\frac{d\mathbf{h}}{dt_i} = \mathbf{m} \quad (3.7)$$

Sfruttando la relazione di Coriolis 3.4 è possibile giungere all'equazione:

$$m \frac{d\mathbf{h}}{dt_i} = m \left(\frac{d\mathbf{h}}{dt_b} + \omega_{b/i} \times \mathbf{h} \right) = \mathbf{m} \quad (3.8)$$

Si considera $\mathbf{h} = \mathbf{J}\omega_{b/i}$ con \mathbf{J} la matrice di inerzia. In particolare per il drone simmetrico con architettura ad x è possibile ricavare una matrice d'inerzia simmetrica nella forma:

$$\mathbf{J} = \begin{pmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{pmatrix} \quad (3.9)$$

Andando a sostituire la matrice così dichiarata è possibile risolvere il prodotto vettoriale e raccogliere a destra le velocità angolari (come già fatto per le velocità lineari in 3.6) ottenendo:

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{pmatrix} + \begin{pmatrix} \frac{1}{J_x} \tau_\phi \\ \frac{1}{J_y} \tau_\theta \\ \frac{1}{J_z} \tau_\psi \end{pmatrix} \quad (3.10)$$

3.1.4 Equazione dinamica assi NED

Infine è possibile esprimere la dinamica in assi NED tra $(\dot{N}, \dot{E}, \dot{D})$ e (u, v, w) . Tra i due sistemi di riferimento entra in gioco la matrice di rotazione 3-2-1 $[L_{21}]$ (eq. 2.5). Per semplicità si indicherà $(x, y, z)^T = (N, E, D)^T$. Si sottolinea inoltre che z originariamente ha verso opposto rispetto a D perciò è necessario cambiare di segno l'ultima riga della matrice ottenendo:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \psi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \psi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ \sin \theta & -\cos \theta \sin \phi & -\cos \theta \cos \psi \end{bmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (3.11)$$

3.1.5 Equazione Forze e Momenti di controllo

Si vuole ora determinare l'espressione completa delle forze e momenti agenti internamente al drone attraverso la rotazione delle pale. Si introduce inoltre il contributo della gravità. Per far ciò si fa riferimento allo schema in figura 3.1. Nella figura l'architettura è a croce invece che a X ma le considerazioni generali che verranno fatte sono analoghe.

La prima equazione da considerare riguarda le forze esercitate verticalmente. Esse possono essere espresse con un termine pari a:

$$F = F_f + F_r + F_b + F_l = K_t(\Omega_f + \Omega_r + \Omega_b + \Omega_l) \quad (3.12)$$

Con $(\Omega_f, \Omega_r, \Omega_b, \Omega_l)$ le quattro velocità di rotazione dei quattro rotori e K_t è il Thrust Factor. Dopodichè è possibile esprimere altri 4 termini inerenti ai momenti esercitati dalle pale per ricavare i momenti globali di rollio, beccheggio e imbardata

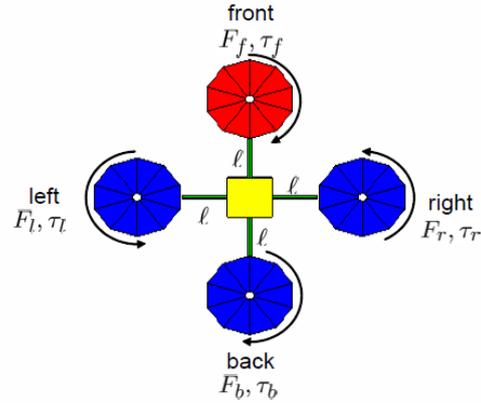


Figura 3.1: Forze agenti sul drone

nel seguente modo:

$$\begin{cases} \tau_\phi = K_t(\Omega_l^2 - \Omega_r^2) \\ \tau_\theta = K_t(\Omega_b^2 - \Omega_f^2) \\ \tau_\psi = K_q(-\Omega_l^2 - \Omega_b^2 + \Omega_b^2 + \Omega_f^2) \end{cases} \quad (3.13)$$

Con K_q Torque Factor del drone.

Infine si vuole definire il contributo della forza di gravità come:

$$\mathbf{f}_g = [L_{21}] \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} = \begin{pmatrix} -mg \sin \theta \\ mg \cos \theta \sin \phi \\ mg \cos \theta \cos \phi \end{pmatrix} \quad (3.14)$$

Tale contributo viene diviso per la massa per ottenere le accelerazioni inerziali da sommare a $(\dot{u}, \dot{v}, \dot{w})$ con $[L_{21}]$ ricavata da 2.5.

3.1.6 Equazioni dinamica complete

Si ottengono perciò 12 equazioni costitutive più 4 di comando che rappresentano in modo completo la dinamica del drone [16]:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\psi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\psi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ s\theta & -c\theta s\phi & -c\theta c\psi \end{bmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (3.15)$$

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \begin{pmatrix} -g \sin \theta \\ g \cos \theta \sin \phi \\ g \cos \theta \cos \phi \end{pmatrix} + \frac{1}{m} \begin{pmatrix} 0 \\ 0 \\ F \end{pmatrix} \quad (3.16)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (3.17)$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{pmatrix} + \begin{pmatrix} \frac{1}{J_x} \tau_\phi \\ \frac{1}{J_y} \tau_\theta \\ \frac{1}{J_z} \tau_\psi \end{pmatrix} \quad (3.18)$$

$$\begin{cases} F = K_t(\Omega_f + \Omega_r + \Omega_b + \Omega_l) \\ \tau_\phi = K_t(\Omega_l^2 - \Omega_r^2) \\ \tau_\theta = K_t(\Omega_b^2 - \Omega_f^2) \\ \tau_\psi = K_q(-\Omega_l^2 - \Omega_b^2 + \Omega_b^2 + \Omega_f^2) \end{cases} \quad (3.19)$$

Per la dinamica linearizzata si fa riferimento ai capitoli successivi.

Capitolo 4

Stato dell'arte teoria dei controlli

4.1 Introduzione

Si vuole ora la progettazione di un controllore per drone evidenziando le principali metodologie e l'approccio concettuale da adottare. Il controllore ha la funzione di ricevere in input il riferimento fornito dall'utente (Reference) e calcolare in output i comandi da fornire al drone (Collettivo, momento di rollio, beccheggio e imbardata). Per fare ciò è necessario prevedere un algoritmo opportuno in grado di rendere, tramite una dinamica stabile, gli stati voluti dall'utente raggiungibili. In generale quest'ultimo può scegliere di comandare il drone online, in tempo reale o offline, con un percorso prestabilito. Nel primo caso si parla di drone propriamente pilotato da remoto. E' il caso dei droni FPV in cui l'operatore, tramite un radiocomando, monitora e controlla

costantemente la posizione e assetto in ogni sua fase di volo. Nel secondo caso invece l'operatore comanda il drone offline, ovvero preimposta una traiettoria tramite la ground station e successivamente il drone eseguirà in maniera autonoma il task indi-

cato. Nel progetto che si vuole trattare in seguito si mostrerà questa seconda opzione di volo autonomo, con una reference basata su un vettore contenente i valori

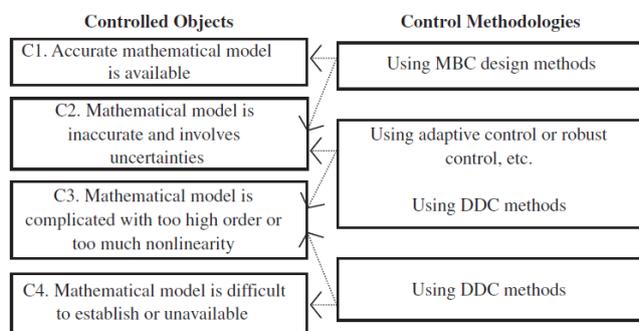


Figura 4.1: Model Based vs Data Driven

(x, y, z, ψ) desiderati in ogni fase di volo. I principali approcci al mondo dei controlli sono Data Driven Control (DDC) [3], Model Based Control (MBC) e modelli ibridi.

4.2 Data Driven Control DDC

Come suggerisce il nome, i Data Driven Control hanno la caratteristica principale di essere controllori che basano la loro funzionalità esclusivamente sui dati. Ciò sta a significare che non è noto a priori il modo in cui l'attuatore/macchina da controllare si comporta, ma si ha a disposizione esclusivamente i dati restituiti dai sensori che mostrano il comportamento postumo, a seguito del comando fornito. Per questo motivo è necessario un processo di apprendimento che porta a vedere la macchina come una black box da controllare. Questo approccio ha alcune conseguenze sia vantaggiose che non:

- **Gestione di problemi complessi.** Non sempre è possibile conoscere a priori un modello matematico che indichi il comportamento di una macchina. Può capitare che il processo da controllare sia di una complessità tale da non renderlo intellegibile. In tal caso si è costretti dalla fisica del sistema a considerare il comportamento come una scatola nera e facendo affidamento esclusivamente sul feedback della sensoristica.
- **Assenza di modello matematico.** Questo può essere considerato un vantaggio qualora si abbia un numero sufficiente di dati a disposizione. Non è infatti necessario creare un modello e ciò può far risparmiare parecchio tempo e ridurre la complessità del controllore.
- **Affidamento sui dati.** Se si hanno a disposizione sufficienti dati è possibile ottenere un controllo ottimale usufruendo solo di questi. Ciò permette di ridurre potenzialmente la complessità del sistema e di mettere in atto algoritmi di learning ottimali. Questo aspetto può trasformarsi però in un potenziale svantaggio, in quanto è necessario prevedere gli apparati corretti per raccogliere ed interpretare moli di dati contemporaneamente.
- **Adattabilità.** Non facendo affidamento ad un modello preconstituito i DDC sono in grado solitamente di adattarsi a condizioni di lavoro non nominali modificando il comportamento in caso di disturbi o comportamenti anomali. Questo punto è particolarmente importante per quei processi da comandare in ambienti estremamente variabili e difficili da prevedere.
- **Disaccoppiamento.** Non avendo necessità di un modello matematico si riesce a disaccoppiare la logica di controllo dalla dinamica ignota del sistema.

In seguito si vuole far riferimento ad una serie di modelli DDC sia in termini di controllo diretto, che di metodologie Data Driven per fare tuning di un controllore

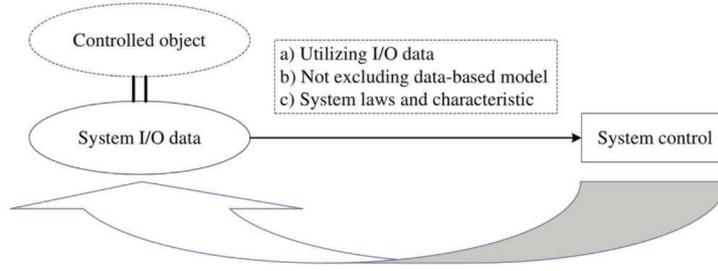


Figura 4.2: Data Driven Control

già predisposto, rendendolo così adattivo (Parameter Estimation che verrà trattato anche per MBC).

4.2.1 SPSA (Simultaneous Perturbation Stochastic Approximation)

Questo metodo prevede il tuning di un controllore con struttura fissata (es PID, NN, SMC ecc..) basandosi esclusivamente su dati di input/output forniti dal sistema tramite un feedback in close loop. Questa caratteristica rende il metodo SPSA sommato ad un controllore generico un DDC, in cui non è necessaria la conoscenza del modello matematico della macchina che si vuole controllare. Come indica il nome, l'algoritmo si basa sull'ottimizzazione stocastica dei parametri in base alla stima e minimizzazione di un gradiente della funzione contenente i parametri stessi e le variabili da controllare. Questo metodo rende di conseguenza il controllore di tipo adattivo, in quanto lo scopo della SPSA è quello di far variare tali parametri per ottenere il minimo del gradiente desiderato. Il gradiente viene calcolato partendo dai valori in output del plant ignoto e dai valori desiderati (ad esempio posizione corrente e posizione desiderata). Si indichi con $y(k)$ e $u(k)$ l'output e l'input del plant controllato e $y_d(k+1)$ il valore desiderato di output al passo $k+1$. Obiettivo di SPSA è trovare il valore di θ che in ogni istante permette di minimizzare il Performance Index:

$$J_k(\theta_k) = E[(y(\theta_k, k+1)^2 - y_d(k+1)^2)] \quad (4.1)$$

Non essendo noto il modello matematico non è possibile calcolare a priori il comportamento dell'output y al variare di θ ovvero non è noto $\frac{\partial y(\theta_k, k+1)}{\partial \theta_k}$. Per questo motivo si usa l'algoritmo SPSA calcolando $\theta(k)$ come:

$$\theta_k = \theta_{k-1} - a_k \cdot g_k(\theta_{k-1}) \quad (4.2)$$

con a_k scaling factor e g_k è la funzione detta a parametri perturbati:

$$g_k(\theta) = \frac{J(\theta + c_k \Delta_k) - J(\theta - c_k \Delta_k)}{2c_k \cdot \Delta_k} \quad (4.3)$$

Con c_k secondo scaling factor. Δ_k rappresenta invece un vettore di perturbazione stocastica, solitamente simmetrica. Forme tipiche di a_k e c_k sono:

$$a_k = \frac{a}{(K+1+A)^n} \quad ; \quad c_k = \frac{c}{(k+1)^\gamma} \quad (4.4)$$

Rendendo i parametri configurabili a, c, α, γ e A . Da sottolineare che tali espressioni non sono univoche, c_k può anche essere costante o comunque con un'espressione simile alla precedente che tende a 0 per l'iterazione k che tende ad infinito. Grazie a questi parametri è possibile con un processo iterativo in due step (uno per $y(k)$ e uno per $u(k)$ ad ogni iterazione) ottenere i parametri desiderati per il controllore voluto. Tale metodo non è però ottimale per macchine il cui stato varia velocemente nel tempo.

4.2.2 MFAC (Model Free Adaptive Control)

Il Model Free Adaptive Control è una tipologia di controllore adattivo Data Driven che fonda il suo funzionamento sull'interpretazione dei dati di input e output insieme alla linearizzazione dei dati in output per valutare la pseudo-derivata parziale che rappresenta il guadagno adattivo con cui calcolare il comando. Si indichino con $y(k)$ il valore in output dal plant non noto al tempo k , $u(k)$ il comando in input al plant e con $\phi(k)$ la pseudo-derivata [15]. In forma compatta è possibile esprimere il valore in output al termine $k+1$ con l'espressione

$$y(k+1) = y(k) + \phi(k)\Delta u(k) \quad (4.5)$$

con $y(k+1) = f(y(k), \dots, y(k-n_y), u(k), \dots, u(k-n_u))$ e n_y e n_u l'ordine non noto di y e u . $\phi(k)$ può essere valutata con diversi metodi di parameter estimation variando quest'ultima lentamente. Si presenta in seguito un metodo iterativo con algoritmo di reset (4.8) per proporre un esempio di una possibile metodologia:

$$u(k) = u(k-1) + \frac{\rho_k \phi(k)}{\lambda + |\phi(k)|^2} (y_d(k+1) - y(k)) \quad (4.6)$$

$$\phi(k) = \phi(k-1) + \frac{\eta_k \Delta u(k-1)}{\mu + |\Delta u(k-1)|^2} (\Delta y(k) - \phi(k-1)\Delta u(k-1)), \quad (4.7)$$

$$\phi(k) = \phi(1), \quad \text{se } |\phi(k)| \leq \varepsilon, \text{ o } |\Delta u(k-1)| \leq \varepsilon, \quad (4.8)$$

Considerando ρ_k, η_k lunghezza degli step, λ, μ sono guadagni e ε è una costante positiva.

Si fa notare che la pseudo-derivata parziale $\phi = \Delta y(t)/\Delta u(t)$ rappresenta il guadagno incrementale e può essere valutata in diversi modi soprattutto a seconda dei dati a disposizione. In linea generale una buona stima porta direttamente alla definizione del controllo ottimale:

$$u(t+1) = u(t) + \phi^{-1} \cdot (y_d(t) - y(t)) \quad (4.9)$$

La principale problematica da tener conto è la necessità di dati online in tempo reale altamente affidabili: rumore o disturbi possono depauperare e rendere inaffidabile il controllore. Inoltre la valutazione della pseudo-derivata parziale è un fattore critico: la variazione impulsiva della dinamica del sistema da controllare potrebbe rendere inefficace la stima di ϕ e pertanto inappropriato il controllore.

4.2.3 Unfalsified control methodology (UCM)

Il metodo Unfalsified control methodology è una tipologia di controllore switch che prevede la scelta tramite un processo decisionale tra un set di controlli candidati. A valle del blocco decisionale viene disposto un performance index in grado di determinare l'adempimento o meno delle specifiche richieste.

La caratteristica importante dell'UCM è che è in grado di falsificare il controllo non funzionante prima che questo venga inserito all'interno del close loop, riuscendo a mantenere ottime prestazioni anche durante i transitori. Si presenta in seguito un tipico schema rappresentato in Figura 4.3 in cui si presentano i set di controllo $C_1 - \dots - C_n$. Con i dati di Input/Output raccolti tra gli intervalli $[0, k-1]$ si aziona lo switch per poi valutare le performance a valle per ogni set distinto. In base alle performance si sceglierà il controllore ottimo al tempo k che andrà a controllare il plant non noto P . Considerando l'indice $j=1,2,\dots,N$ è possibile esprimere il segnale di reference

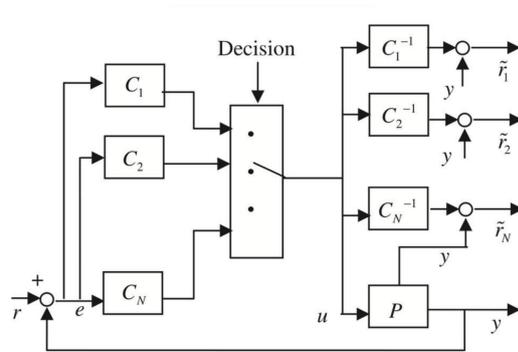


Figura 4.3: UCM

$$\tilde{y}(\tau) = C_i^{-1}(u(\tau)) + y(\tau). \quad (4.10)$$

Il processo decisionale può essere valutato in diversi modi. Si presenta in seguito un classico esempio di funzione quadratica (una metodologia ampiamente utilizzata come parameter estimation che verrà proposta in seguito in forma generale nella sezione MBC):

$$J_j(k) = J(u, y, \tilde{y}, k) = \max_{\tau \in [0, k]} \frac{\|u(\tau)\|^2 + \|\tilde{y}(\tau) - y(\tau)\|^2}{\|\tilde{y}(\tau)\|^2 + \alpha}, \quad \alpha > 0 \quad (4.11)$$

In linea generale si vorrebbe ottenere una funzione con concavità rivolta verso l'alto in modo tale da poter tendere ad un punto di minimo stabile. A tal fine si valuta il gradiente e con un processo iterativo per escludere $N-1$ controllori e inserire nel loop solo il più vantaggioso. In linea generale si può ipotizzare il set di controlli

dipendenti da un parametro θ adattivo. Se $J(0, \tau) \leq 0, \quad \forall \tau \in [0, t]$. il controllore porta la dinamica verso una condizione di equilibrio, viceversa i parametri degli N-1 controllori falsificati vengono aggiornati verso una direzione discendente del tipo:

$$\frac{d\theta}{dt} = -\gamma \Delta J(\theta, t) \quad (4.12)$$

con γ parametro predefinito. Uno dei principali vantaggi di questa metodologia è la potenziale semplicità di applicazione. Tale semplicità però può essere il problema stesso del metodo che, se non trova nel set di partenza un controllore candidato adeguato, potrebbe non riuscire a garantire la stabilità. E' perciò opportuno scegliere un set opportuno a seconda dell'impiego con un relativo proceso decisionale di switch adeguato.

4.2.4 Reinforcement Learning (RL) e Neural Network (NN)

In questo paragrafo si vuole trattare una metodologia moderna, in forte sviluppo, la cui teoria affonda le radici già a partire dalla metà del '900 ma che negli ultimi anni è diventata sempre più applicabile grazie agli avanzamenti in ambito software. Il RL è una metodologia adattiva estremamente potente e variegata che, nella sua base, permette la definizione di un controllore, chiamato "Agente", che interagisce con ciò che gli è esterno, definito come "Ambiente" [9] [10]. In questo caso si è scelto di trattare tale argomento nella sezione DDC perché si considera il plant da controllare come non noto, facente perciò parte dell'ambiente. L'agente interagisce con l'ambiente tramite un' "azione" (a) e riceve da esso uno stato (fornito dall'osservatore) e un reward (r). Il concetto di base dell'RL è definire il reward a seconda della bontà o meno dell'azione, andando così a modificare l'algoritmo di training presente all'interno dell'agente che agisce sulla policy (il vero e proprio "cervello" del controllore, spesso composto da NN) [27]. Quest'ultima è da intendersi come una mappa degli stati, che ricevendo in input la policy update dell'algoritmo di training restituisce un'azione che andrà ad agire sull'ambiente. In questi termini il RL è perciò a tutti gli effetti un DDC basato esclusivamente sui dati di input e output scambiati tra agente e ambiente. Per comprendere i concetti di base è necessario definire alcune caratteristiche e proprietà dei segnali. Il processo rispetta la proprietà di Markov nel momento in cui, per valutare le azioni e reward al tempo $t+1$, si hanno tutte le informazioni necessarie nel tempo t . Si può infatti affermare che se si considerano tutti gli istanti di tempo da $t=1$ fino al t corrente e sono noti i valori dell'azione (a), stato (s) e reward (r) in tali istanti allora è possibile individuare la distribuzione di probabilità per definire il lo stato $S_{t+1} = s'$ e reward R_{t+1} del passo temporale successivo ($t+1$):

$$\Pr\{(R_{t+1} = r, S_{t+1} = s') \mid S_0, A_0, R_1, \dots, S_{t-1}, A_{t-1}, R_t, S_t, A_t\} \quad (4.13)$$

Non sempre però si hanno a disposizione tali valori. Il processo rispetta la proprietà di Markov se:

$$\Pr\{R_{t+1} = r, S_{t+1} = s' | S_t, A_t\} \quad (4.14)$$

ovvero è possibile determinare il reward e lo stato al passo $t+1$ esclusivamente con lo stato e l'azione al passo t . In tal caso lo stato e l'azione al tempo t racchiudono in sé tutte le informazioni necessarie per valutare il processo al passo successivo. Un RL che soddisfa tale proprietà viene definito come Markov Decision Process (MDP). In tal caso è possibile definire gli stati e i reward dei passi successivi definendo due termini espressi in seguito: Il primo è:

$$p(s' | s, a) = \Pr\{S_{t+1} = s' | S_t = s, A_t = a\} \quad (4.15)$$

la probabilità di avere al passo $t+1$ lo stato s' partendo dallo stato corrente s e azione corrente a .

$$r(s, a, s') = \mathbf{E}[R_{t+1} | S_t = s, A_t = a, S_{t+1} = s'] \quad (4.16)$$

Il secondo afferma che l'expected reward al passo $t+1$ dipende dallo stato corrente s , azione corrente a e stato al passo successivo s' . Con queste due espressioni (p ed r) riesco a determinare univocamente il MDC alla base del RL. Per comprendere quale azione intraprendere è necessario definire una funzione in grado di restituire la “bontà” dell'agente nel trovarsi in un determinato stato corrente. Tale bontà dipende dal temine di reward secondo la State-Value Function 4.17:

$$v_*(s) = E_{\pi_*}[G_t | S_t = s] = \mathbf{E}_{\pi_*} \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \middle| S_t = s \right] \quad (4.17)$$

Questa rappresenta il valore atteso se l'agente segue la policy partendo dallo stato s , con γ discount rate, valore di scala, compreso tra 0 e 1. Un'altra importante funzione è quella per determinare la bontà non dello stato corrente bensì dell'azione e si chiama Action-Value Function ed è nella forma:

$$q_\pi(s, a) = E_\pi[G_t | S_t = s, A_t = a] = \mathbf{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \middle| S_t = s, A_t = a \right] \quad (4.18)$$

Essa indica il valore atteso se si svolge l'azione a nello stato s secondo la policy π . v_π e q_π vengono definite con l'esperienza. Se un agente seguendo la policy π ha una media costante di reward indipendentemente dall'azione, ci si aspetta che il valore converga a $v_\pi(s)$. Viceversa se l'agente ottiene diverse medie a seconda dell'azione presa, il valore dipenderà dall'azione e tenderà a $q_\pi(s, a)$. Questi metodi di stima vengono definiti metodi Monte Carlo.

A questo punto è possibile definire la relazione tra una State-Value Function al tempo t e quella al passo successivo considerando il processo a cascata tra azione,

reward, stato arrivando a definire l'equazione di Belmann nel seguente modo:

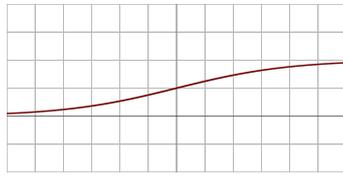
$$\begin{aligned}
 v_\pi(s) &= \mathbb{E}_\pi [G_t | S_t = s] \\
 &= \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s \right] \\
 &= \mathbb{E}_\pi \left[R_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k R_{t+k+2} | S_t = s \right] \\
 &= \sum_a \pi(a | s) \sum_{s'} p(s' | s, a) \left[r(s, a, s') + \gamma \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+2} | S_{t+1} = s' \right] \right] \\
 &= \sum_a \pi(a | s) \sum_{s'} p(s' | s, a) [r(s, a, s') + \gamma v_\pi(s')],
 \end{aligned}$$

Da tale espressione si può affermare che il valore atteso partendo da s e seguendo π dipende dal valore dell'azione "a" che si ottiene da "s", moltiplicato per la probabilità di ottenere s' con "s" e "a" precedente, moltiplicato per il reward atteso passando per "s", "a" ed s' (valutati prima) più il valore atteso tarato dal discount reward. Questo approccio ricorsivo permette di calcolare la probabilità che si ottenga un valore, tarato in base alle possibili azioni e ai conseguenti stati e reward. Questo processo è alla base del RL. L'obiettivo è quello di massimizzare la State-Value Function ottenendo il valore ottimo (*):

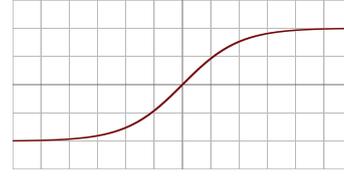
$$v = \max[v_\pi(s)] \quad (4.19)$$

Questo approccio può essere adottato nel caso di reti neurali e per questo motivo si è scelto di trattarle insieme. La policy infatti ha il compito di determinare le azioni dagli input fornitigli e a tal fine è particolarmente funzionale un Neural Network. In questa sezione non si ha la pretesa di voler dare un riferimento completo al lettore in quanto la varietà ed estensione dei temi trattati non sono assolutamente esauribili in poche righe. L'obiettivo a cui si vuol tendere è però quello di dare una visione concettuale di base su cosa sia il RL e NN. Quest'ultime come dice il nome si ispirano al funzionamento dei neuroni umani, schematizzati da nodi che possono assumere al loro interno un range di valori compresi tra 0 e 1. Tali neuroni sono collegati tra di loro su più layer formando una rete che, prendendo in input uno stato iniziale, restituisce attivando o meno i collegamenti tra i nodi, un valore finale. L'attivazione dei nodi è funzione del valore assunto da questi e dipende strettamente da σ chiamata Activation Function. Quest'ultima può essere di diversa natura come un sigmoide, tangente iperbolica, ReLU (Rectified Linear Unit) ecc..

Come intuibile le funzioni di attivazione determinano il passaggio dal neurone spento ad acceso o viceversa, a seconda del valore da questo ottenuto. Gli strati (Layers) di neuroni possono essere di vario numero. Il passaggio da un Layer ad un

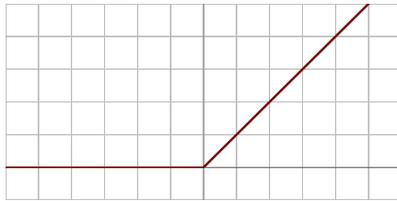


$$(a) \sigma(x) = \frac{1}{1+e^{-x}}$$

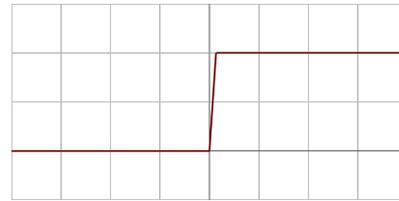


$$(b) \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Figura 4.4: Figura (a) sigmoide, Figura (b) tangente iperbolica



$$(a) \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$



$$(b) \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$

Figura 4.5: Figura (a) ReLU, Figura (b) Binary Step

altro avviene tramite i collegamenti dei singoli neuroni il cui valore viene pesato da una matrice, detta matrice dei pesi ω_i , che andrà a determinare il valore in input di ogni singolo neurone del layer successivo. In questo modo con un processo a cascata si giunge all'ultimo layer i cui nodi determinano il risultato finale.

$$\psi(x) = \sigma\left(\sum_{i=1}^n x_i \omega_i - b\right) \quad (4.20)$$

E' possibile introdurre per ogni layer anche un termine aggiuntivo definito bias che crea un offset a monte della funzione di attivazione variando il comportamento del nodo. Tramite questa struttura ricorsiva è possibile interagire con una varietà notevole di algoritmi di training che hanno la funzione di tarare i valori dei pesi in modo da ottenere in output il valore voluto a seconda dell'input di partenza. E' chiaro che uno di questi metodi generali è proprio il RL. Questi metodi sono in largo sviluppo per le loro potenzialità. Essi infatti permettono di trattare i fenomeni di controllo come scatole nere, di cui non è necessario conoscere il comportamento a priori. Gli algoritmi di training sono inoltre estremamente variegati e versatili. Essi possono essere addestrati in ambienti virtuali mandandoli allenandoli tramite numerosi cicli di addestramento ed ottenendo un tuning puntuale molto efficiente. La versatilità nasce dalla genericità dell'approccio al controllo il che li rende adatti per adempiere ad un enorme quantità di task differenti, che spaziano da controlli in autonomous driving/flying al riconoscimento di immagini ad interpretazione di grandi quantità di dati, in cui sarebbe quasi impossibile districarsi in altro modo.

Il grande svantaggio sono gli alti costi computazionali e di energia ma soprattutto l'aviabilità dei dati. Fare training su dati parziali o inesatti porta infatti la rete neurale ad assumere bias fortemente inesatti rendendo i risultati in uscita inutili o perfino controproducenti. Soprattutto nel caso di estesi algoritmi di training la selezione dei dati diventa perciò il vero problema centrale nello sviluppo di una NN.

4.3 Model Based Control (MBC)

L'approccio Model Based (MBC) si basa sulla conoscenza e realizzazione del modello matematico del sistema fisico che si vuole controllare. Tale approccio, basato perciò su modelli, è largamente utilizzato nel settore industriale e non solo, soprattutto per quei processi la cui dinamica può essere nota in forma completa, o approssimata, a priori. È questo il caso di numerosissime applicazioni industriali in cui l'ambiente e i disturbi giocano un ruolo assai trascurabile rendendo le non linearità imprevedibili pressoché assenti. Questo approccio presenta alcuni vantaggi non indifferenti:

- **Modello matematico.** Sebbene possa risultare un passaggio in più (rispetto DDC) la realizzazione del modello matematico accurato permette la sua riutilizzazione in numerosi campi anche assai diversi l'uno dall'altro. Molti attuatori infatti presentano un comportamento noto a priori e la conoscenza del modello permette la sua diffusione tramite i principali portali di ricerca e sviluppo, rendendoli accessibili a tutte le persone interessate. Non è perciò necessario “ripartire da zero” ogni volta ma sfruttare modelli matematici high fidelity già realizzati da altri per applicazioni differenti, e adattarli alle nuove necessità.
- **Adattabilità dei controlli.** Essendo a conoscenza del modello matematico è possibile prevedere il comportamento dei termini controllati andando a selezionare non solo il controllore più adeguato allo scopo ma effettuando anche una taratura spesso effettuata a monte, riducendo i tempi di sviluppo.
- **Conoscenza degli stati.** Non è necessario avere a disposizione molte quantità di dati sperimentali. E' possibile infatti calcolare gli stati desiderati, invece di prevedere la sensoristica su tutti i valori cercati. Questo permette di conseguenza di realizzare un sistema di feedback ottimale andando a calcolare selettivamente i valori desiderati e manipolandoli tramite la legge di controllo per ottenere i valori in output dal plant voluti.
- **Processo di sviluppo.** Le caratteristiche sopra esposte rendono lo sviluppo per logica Model Based completamente strutturato. E' possibile adattare logiche come il V-model per prevedere fasi e tempi di sviluppo ben strutturate e riducendo i fattori ignoti e possibili problematicità in itinere. Gli svantaggi principali invece sono:
- **Obbligo di un modello matematico.** Per applicazioni classiche note può non essere un problema ricavare il modello matematico ma per applicazioni in via di sviluppo, in presenza di forti non linearità e disturbi, la realizzazione di un modello può essere estremamente problematica. Non riuscire a modellare disturbi importanti significa creare un controllore di fatto inutile o addirittura

dannoso nel momento in cui si verificheranno tali disturbi. Inoltre in alcune applicazioni non è noto a priori la fonte del disturbo e la sua intensità. Ciò potrebbe portare ad uno sviluppo Model Based troppo oneroso e conservativo, o viceversa troppo poco conservativo e perciò inadatto.

- **Costo computazionale.** Per il problema esposto nel punto precedente si può giungere facilmente ad un'onerosità eccessiva del modello che comporta moli di calcolo elevate, con un conseguente inpatto a livello di prestazione sull'intera macchina.
- **Influenza ai disturbi.** Il modello matematico per quanto accurato avrà spesso delle discrepanze dal suo comportamento reale. Qualora quest'ultime siano limitate non è un problema ma spesso è comunque necessario ricorrere a metodologie per aumentare la robustezza del controllo.

Si vuole ora strutturare il lavoro che segue nel seguente modo: in primis si presentano sinteticamente le classiche tipologie di controllo; successivamente si introduce il concetto di controllo adattivo tramite parameter estimation (con cui si può rendere adattivo un controllo classico) e tramite leggi di controllo ad hoc sviluppate sfruttando altri metodi (come feedback linearization o funzioni di Lyapunov). Non si ha qui la pretesa di fornire una visione esaustiva dei vari metodi in questione ma di proporli all'attenzione del lettore per dare una visione di massima di quali approcci possono essere utilizzati nella modellazione del controllore.

4.3.1 Controlli classici

LINEAR QUADRATIC REGULATOR (LQR)

Il controllore lineare quadratico (LQR) è un controllo di tipo ottimo. Esso infatti prevede l'ottimizzazione di una funzione trovando il punto di minimo [21]. Tale funzione è composta da due termini e rappresenta il trade off tra il termine di comando e il termine di stato:

$$J = \int_0^{\infty} (x(t)^T Q x(t) + u(t)^T R u(t)) dt \quad (4.21)$$

La funzione J viene chiamata funzione di costo ed è un integrale sul tempo di simulazione. Il termine $x(t)$ rappresenta lo stato e può essere un valore singolo ma solitamente è un vettore mentre il termine $u(t)$ è il comando. I termini Q ed R sono rispettivamente due matrici che rappresentano il peso degli stati e il peso dei comandi. Solitamente sono matrici diagonali positive e in generale ciò che determina le prestazioni del controllore non sono i singoli valori delle matrici ma il loro rapporto. Avere un rapporto Q/R alto implica dare priorità alla regolazione dello stato, ottenendo perciò controlli più impulsivi e regolazione veloce. Avere rapporti Q/R bassi significa dare priorità al costo del comando diminuendolo. Si ottiene

perciò un comando meno oneroso, con minor dispendio di energia ma transitori più lenti e regolazione meno veloce. Per la taratura delle matrici è possibile partire dal metodo di Bryson come mostrato in seguito (4.22, 4.23):

$$Q(j, j) = \frac{1}{(\text{valore massimo accettabile errore di stato})^2} \quad (4.22)$$

$$R(j, j) = \frac{1}{(\text{valore massimo accettabile errore di input})^2} \quad (4.23)$$

Con $[1, 2, \dots, m]$ e m dimensione massima delle matrici.

Per ricavare la legge di controllo è necessario minimizzare la funzione di costo. E' possibile fare ciò differenziando J e ponendola uguale a 0. Così facendo si ottiene l'equazione di Riccati (4.24):

$$A^T P + P A - P B R^{-1} B^T P + Q = 0 \quad (4.24)$$

con B matrice di comando del sistema stato-spazio. Per la rappresentazione Stato-Spazio si fa riferimento ai capitoli successivi in cui viene linearizzata la dinamica del drone e si ricavano le matrici, tra cui B . La matrice P , soluzione dell'equazione di Riccati, permette di determinare il guadagno K e di ottenere la legge di controllo nel seguente modo (4.25):

$$u = -Kx \quad ; \quad K = R^{-1} B^T P \quad (4.25)$$

Il vantaggio dell'LQR è di poter ottimizzare il costo in modo esplicito ma richiede la formulazione Stato-Spazio per la sua definizione.

MODEL PREDICTIVE CONTROL (MPC)

Il Model Predictive Control è un controllo ottimo, anch'esso basato sull'ottimizzazione di una funzione di costo. A differenza dell'LQR il controllo MP si basa sulla conoscenza puntuale della dinamica che si vuole controllare e permette di calcolare la funzione di costo entro un orizzonte finito di tempo andando a "predire" il suo sviluppo. E' inoltre possibile limitare l'azione sugli stati e controlli tramite delle boundaries. Questo aspetto è particolarmente importante per gli algoritmi di navigazione che devono guidare il processo controllato in un ambiente con ostacoli. La tecnica di ottimizzazione MPC permette di ottenere una stabilità globale inferiore rispetto tutti i termini di stato e di

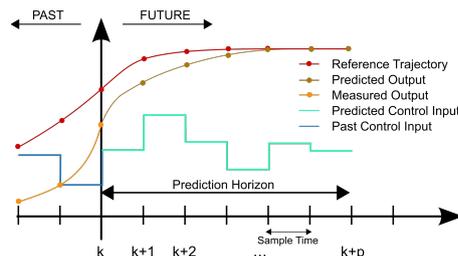


Figura 4.6: MPC Esempio

comando (che vengono invece resi tutti ottimi dall'LQR) ma permette di ottenere un controllo della dinamica molto più puntuale con prestazioni nel transitorio maggiormente controllabili. Un esempio tipico della funzione di costo è (4.26):

$$J = \sum_{i=1}^N \omega_{x_i} (r_i - x_i)^2 - \sum_{i=1}^M \omega_{u_i} \Delta u_i^2 \quad (4.26)$$

con x variabile di stato, r reference, u comando, w_x matrice dei pesi dello stato e w_u matrice dei pesi di comando. In generale essendo un controllo ottimo la legge di controllo prevede che il comando ottimale dipenda dal minimo della funzione di costo (4.27):

$$u_k = \arg[\min_u(J)] \quad (4.27)$$

Come accennato prima il grande vantaggio del MPC è quello di essere “predittivo” e perciò particolarmente adatto alla navigazione e al controllo in ambienti complessi. Lo svantaggio principale è che richiede un modello matematico preciso, pena la previsione errata e perciò possibili transitori non voluti con dinamiche indesiderate.

SLIDING MODE CONTROL (SMC)

Il Sliding Mode Control è una tecnica di controllo la cui logica relativamente facile lo rendono però efficace, soprattutto con sistemi non lineari. L'idea che sta alla base del controllo è quella di fornire un comando impulsivo discontinuo per far sì che lo stato controllato si diriga e permanga nella condizione voluta. Quest'ultima viene identificata come la superficie di riferimento, o sliding surface (4.7). Quando lo stato controllato si trovi “al di sopra” della superficie di riferimento il comando è positivo, viceversa il comando sarà negativo. In questo modo si ottiene aderenza alla superficie di riferimento e un vero e proprio scivolamento dello stato lungo tale superficie. La discontinuità del metodo comporta necessariamente un'oscillazione nell'intorno della sliding surface portando al ben noto fenomeno del chattering. Tali oscillazioni avranno ampiezza tanto minore quanto maggiore sarà la frequenza e intensità del controllo. In linea generale è possibile indicare la superficie di scivolamento come (4.28):

$$\sigma(x) = s_1 x_1 + s_2 x_2 + \dots + s_n x_n \quad (4.28)$$

Considerando il comando come (4.29):

$$u(x, t) = \begin{cases} u^+(x) & \text{se } \sigma > 0 \\ u^-(x) & \text{se } \sigma < 0 \end{cases} \rightarrow u(x, t) = u(x) \text{sign}(\sigma) \quad (4.29)$$

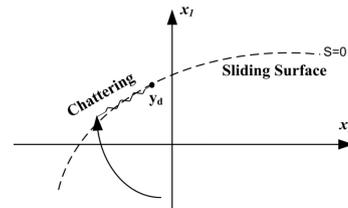


Figura 4.7: Sliding Surface

Il principale vantaggio di questo metodo è l'intrinseca robustezza e stabilità. Esso è in grado di controllare modelli non lineari anche in presenza di incertezze. Il grande svantaggio è il fenomeno del chattering che può rendere problematico questo controllo per alcune applicazioni. In particolare l'oscillazione intorno alla superficie di controllo può comportare un azionamento eccessivo di attuatori in alta frequenza con relative oscillazioni e costi eccessivi (in termini di comando), o viceversa per ottenere attuazioni meno frequenti potrebbe essere necessario accettare un errore sullo stato maggiore. Può essere interessante notare come tale approccio al metodo dei controlli ha aperto la strada ad altri tipi di regolazioni come il PWM.

PROPORTIONAL DERIVATIVE INTEGRATIVE CONTROL (PID)

Il controllo PID è stato sviluppato negli anni '60 da Ziegler e Nichols ed è uno dei metodi di controllo più utilizzati in ambito industriale, grazie anche alla sua facilità interpretativa e di applicazione. Essendo più che noto ci si limiterà a definirlo a grandi linee. Si fa riferimento alle sezioni successive in cui il PID viene implementato a cascata come controllore base del quadrotor. Il PID consta di tre contributi i cui nomi recano l'azione che hanno nei confronti dell'errore: Proporzionale, Integrativo e derivativo (4.30).

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d(\tau) + K_D \frac{de(t)}{dt} \quad (4.30)$$

Il termine proporzionale rende l'azione di comando più intensiva, il termine derivativo rende più o meno accentuate le variazioni nel tempo mentre l'integrale porta solitamente l'errore a regime pari a zero. La giusta taratura dei tre termini è fondamentale per evitare transitori indesiderati, oscillazioni eccessive o overshoot. Per questo motivo si utilizzano metodi ormai ben consolidati come il metodo Ziegler Nichols come punto di partenza per i termini di guadagno voluti. Il principale vantaggio è la semplicità di realizzazione e l'adattabilità a sistemi tra i più disparati. Il principale svantaggio è l'inefficienza in caso di incertezze e problematiche come il Wind up inerente alla saturazione dell'azione integrale. Per questo motivo spesso vengono aggiunte piccole modifiche per limitare tali problemi.

H INFINITO H_∞

Il controllo H infinito prevede la valutazione di un termine di guadagno K tale da minimizzare la norma infinito della funzione di trasferimento proposta in seguito (4.35). Si consideri w come un input esterno al plant, ad esempio dipendente dall'ambiente come un disturbo; u variabile di comando del controllore, K matrice di guadagno del controllore, P funzione di trasferimento del plant, z output del plant e v output parziale del plant

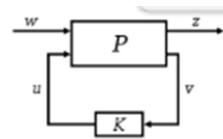


Figura 4.8: Schema H_∞

in input al controllore. Da qui è possibile estrapolare la formulazione matematica seguente (4.31, 4.32):

$$\begin{bmatrix} z \\ v \end{bmatrix} = \mathbf{P}(s) \begin{bmatrix} w \\ u \end{bmatrix} = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \cdot \begin{bmatrix} w \\ u \end{bmatrix} \quad (4.31)$$

$$u = \mathbf{K}(s)v \quad (4.32)$$

Con \mathbf{K} matrice non nota che si vuole calcolare per risolvere il comando. Gli stati importanti che si vogliono controllare al fine della stabilità sono quelli espressi da z in quanto in output verso l'ambiente esterno determineranno la fisica con cui il plant si comporterà.

$$z = F_l(\mathbf{P}, \mathbf{K})w \quad (4.33)$$

Allo stesso modo i valori in input “problematici” sono quelli racchiusi da w perché essendo esogeni, sono difficili da controllare e potrebbero essere anche complessi da prevedere. Per questo motivo si è interessati a ricavare la funzione di trasferimento parziale $F_l = z/w$ che il controllore \mathbf{K} avrà la funzione di stabilizzare. Si procede perciò sostituendo il termine di comando u all'interno dell'espressione precedente e ci si limita a considerare la prima riga della matrice di trasferimento ricavando l'espressione (4.34):

$$F_l(\mathbf{P}, \mathbf{K}) = P_{11} + P_{12}\mathbf{K}(I - P_{22}\mathbf{K})^{-1}P_{21} \quad (4.34)$$

L'obiettivo del controllore H_∞ è quello di determinare \mathbf{K} tale da minimizzare F_l secondo la norma infinito ovvero (4.35):

$$\|F_l(\mathbf{P}, \mathbf{K})\|_\infty = \sup[\sigma(F_l(\mathbf{P}, \mathbf{K})(s))] \quad (4.35)$$

Con σ massimo valore singolare della matrice $F_l(P, K)$.

Come si può intuire il principale problema con questo tipo di controllo è la necessità di conoscere perfettamente la matematica in gioco. Inoltre può essere computazionalmente oneroso dovendo andare a risolvere un problema di minimizzazione non ottimale in partenza. Richiede perciò la capacità di comprendere il metodo ottimale per evitare il rischio di incrementare troppo la complessità computazionale. D'altro canto il controllore H_∞ se strutturato correttamente permette di ottenere robustezza a disturbi e renderlo chiaro dal punto di vista logico e teorico.

PROPORTIONAL NAVIGATION (PN)

Questo tipo di controllo è estremamente settoriale e se ne fa cenno perché è quasi esclusivamente di pertinenza aerospaziale. L'obiettivo principale è l'inseguimento di un oggetto in linea di vista (Line of Sight) tramite un controllo che crea un'accelerazione normale proporzionale alla rotazione angolare dell'oggetto in Line of Sight.

La funzione principale è portare il mezzo comandato a inseguire per poi collidere con l'oggetto in questione e per questo motivo è un metodo ampiamente utilizzato per la navigazione dei missili dotati di sensore ottico. La legge di controllo può essere espressa come (4.36):

$$a_n = N\dot{\lambda}V \quad (4.36)$$

con N guadagno proporzionale, $\dot{\lambda}$ il rateo di rotazione dell'oggetto in Line of Sight e V la velocità del missile.

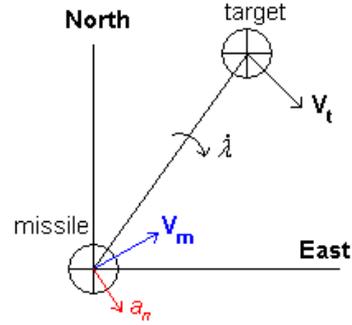


Figura 4.9: Schema PN

ARTIFICIAL POTENTIAL FIELD (APF)

Il metodo Artificial Potential Field non è propriamente una tipologia di controllo diretta ma un metodo di navigazione dal quale è possibile calcolare il comando necessario per raggiungere la posizione desiderata. Il concetto base è quello di creare un “campo artificiale” di tipo attrattivo o repulsivo per guidare verso il target o lontano da un possibile ostacolo [24]. Per fare ciò si crea un campo tramite una funzione con concavità rivolta verso l'alto, il cui punto di minimo coincide con la traiettoria voluta. Una classica formulazione è (4.37):

$$U_{att} = \frac{1}{2}k_{att}\|x - x_{goal}\|^2 \quad (4.37)$$

con x posizione attuale, x_{goal} come posizione voluta e k_{att} come fattore attrattivo. Maggiore è il termine k , maggiore sarà la concavità della parabola e risulterà in una direzionalità maggiore verso il punto di minimo. La forza attrattiva punta verso il valore negativo del potenziale del campo attrattivo, ovvero (4.38):

$$F_{att} = -\Delta U_{att}(x) = -k_{att}(x - x_{goal}) \quad (4.38)$$

Valutando la forza attrattiva è possibile calcolare, in base al tipo di attuazione, le forze dei propulsori da azionare per ottenere l'accelerazione voluta nella direzione desiderata.

Analogamente è possibile calcolare il campo di repulsione, sempre come una funzione con concavità rivolta verso l'alto che però andrà a sottrarsi a quello attrattivo. Una possibile forma è (4.39):

$$U_{rep} = \begin{cases} \frac{1}{2}k_{rep}\left(\frac{1}{d(x,x_{obs})} - \frac{1}{d_0}\right)^2, & \text{se } d(x, x_{obs}) \leq d_0 \\ 0 & d(x, x_{obs}) > d_0 \end{cases} \quad (4.39)$$

Con $d(x, x_{obs})$ è la distanza tra la macchina controllata e l'ostacolo, d_0 è la distanza soglia oltre la quale l'ostacolo non viene considerato e k_{rep} è il guadagno del potenziale repulsivo. Si ottiene così (4.40):

$$F_{rep} = -\Delta U_{rep}(x) \quad (4.40)$$

La forza totale sarà perciò la forza attrattiva meno la forza repulsiva o, in altri termini, si può calcolare il campo totale sommando quello repulsivo e attrattivo ottenendo una vera e propria visualizzazione tridimensionale di conche rivolte verso l'alto che definiscono le direzioni discendenti attrattive ed altre rivolte verso il basso con direzioni repulsive.

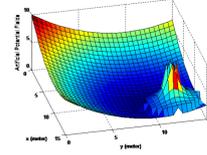


Figura 4.10: Esempio potenziale APF

CONTROLLO GEOMETRICO

Il controllo geometrico si basa sull'analisi geometrica di traiettoria e rotazioni, in spazio tridimensionale, dell'attuatore o macchina da controllare [17]. Tramite matrici di rotazioni appartenenti al gruppo $SO(3)$, ovvero il gruppo di rotazioni secondo gli assi body già presentate precedentemente, è possibile determinare gli errori di posizione e assetto dal sistema inerziale al sistema body. Il controllo geometrico ha come comando un termine dipendente da questi fattori che viene solitamente modificato per forzare dei vincoli intrinseci della macchina. Un esempio può essere caratterizzato da un drone di tipo tilt-rotor in cui le rotazioni dei rotori sono limitate e per tanto il tracciamento geometrico della traiettoria deve prevedere la convergenza a zero dell'errore tenendo conto di questa limitazione [[17]]. Nel paper citato si affronta tale problema andando inanzitutto a definire gli errori geometrici di posizione e velocità (con d valori desiderati)(4.41):

$$e_x = x - x_d \quad ; \quad e_v = v - v_d \quad (4.41)$$

Mentre l'errore di rotazione è definito come la matrice di rotazione R (appartenente a $SO(3)$) che descrive l'assetto del body frame moltiplicata per la matrice di assetto desiderata nel riferimento inerziale $R_e = RR_d^T$ ottenendo R_e , anch'essa appartenente al gruppo di rotazioni $SO(3)$. Derivando quest'ultima è possibile ottenere l'errore sulle velocità di rotazioni $\dot{R}_e = \dot{R}R_d^T + R\dot{R}_d^T$ e si ricava l'errore di navigazione esprimibile come (4.42):

$$\Psi = \frac{1}{2}tr(K_R(I - R_e)) \quad (4.42)$$

con K_r matrice di guadagno diagonale. A questo punto è possibile esprimere la funzione di controllo con un termine del tipo (4.43):

$$f_c = -K_x e_x - K_v e_v \quad (4.43)$$

Con f forza di controllo. Senza volersi addentrare in ogni singolo termine è possibile aggiungere il termine di vincolo come già specificato nella forma $f_c = c(\Psi)R_d^T f_{c,des}$ con $c(\Psi)$ termine di scalatura proporzionale all'errore di navigazione. In questo modo, oltre che ottenere la limitazione dell'escursione massima e minima della rotazione della macchina in questione, si scala anche quest'ultima a seconda dell'errore di

navigazione, portandola al valore massimo consentito per errori elevati e facendola tendere a 0 per errori tendenti a 0. Come si può evincere il controllo è particolarmente efficace per sistemi di navigazione in spazi complessi, con traiettorie curve in 3 dimensioni. Questo lo rende particolarmente adatto in ambito aerospaziale ma ne rende lo studio interessante anche per applicazioni in ambito informatico per grafica 3D e simulazione. Gli svantaggi sono la necessità di conoscere un modello matematico affidabile del problema con cui poter tarare correttamente la legge di controllo. Inoltre il formalismo matematico ha una complessità maggiore di altri metodi e prevede una comprensione profonda della geometria del sistema in modo da poter ottimizzare il controllore al meglio.

4.3.2 Controlli Adattivi

Per controllore adattivo si intende un controllore in grado di modificare le proprie caratteristiche adattandole alla situazione corrente, per ottimizzare il comando da fornire al sistema [12] [10]. A differenza dei metodi proposti precedentemente (fatta eccezione del RL e NN) i parametri non sono fissi ma possono variare secondo leggi matematiche differenti. Il concetto di base è trovare una funzione in grado di effettuare una stima del parametro (Parameter estimation) ad ogni passo temporale, in modo da aggiornare i valori in modo consoni. I controlli adattivi avranno perciò una componente di parameter estimation, che fornisce o direttamente il parametro o la sua derivata del tempo (perciò la direzione in cui varia temporalmente), con a valle una legge di controllo. Quest'ultima ricevendo in input il parametro aggiornato varia a sua volta il comando verso il sistema controllato.

Gli approcci che si possono adottare sono molteplici. In generale è possibile partire da un controllore a parametri fissi e renderlo adattivo in un secondo momento, aggiungendo a monte un parameter estimator che tramite feedback di stato aggiorna i valori desiderati. In secondo luogo è possibile creare da zero un controllore di tipo adattivo prevedendo una legge di controllo già ottimizzata per tale funzione. In seguito si vogliono presentare alcune tecniche consolidate per stimare i parametri insieme a tecniche di controllo generali applicabili in campo adattivo.

Il **Parameter Estimation** (PE) può avvenire secondo tre approcci principali [11]:

- **PE diretto**, anche detto implicito. Il controllore è parametrizzato rispetto ad un set di parametri che vengono direttamente stimati in base al feedback del plant senza intermediazioni.
- **PE indiretto**, anche detto esplicito. Il controllore è parametrizzato rispetto ad un set di parametri (detti parametri del controllore) che vengono calcolati a seguito di una stima dello stato del plant valutato secondo parametri differenti. In altre parole vi è un intermediario a valle del plant che fa una stima del suo stato ed utilizza questi valori per calcolare i parametri effettivi del controllore adattivo.

- **Gain scheduling.** In questo caso il parametro non viene propriamente calcolato ma è fatto variare secondo una tabella pre-determinata che con una logica di scelta prediligerà un valore rispetto ad un altro. In questo caso perciò la legge di adattamento non verte direttamente sul parametro ma sulla logica di scelta tra parametri prefissati. Questo approccio è analogo al metodo descritto in DDC chiamato Unfalsified Control (UCM).

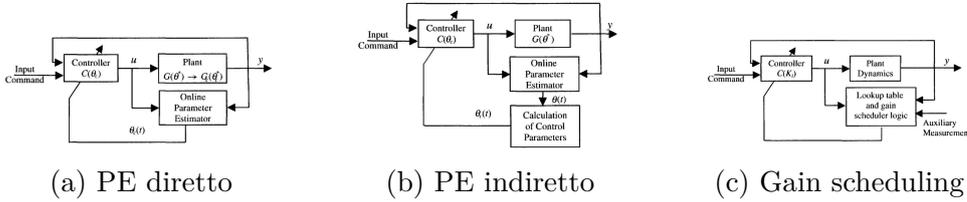


Figura 4.11: Principali architetture per Parameter Estimation (PE)

Si introduce inanzitutto un modello in grado di determinare univocamente i parametri che si vogliono considerare. Ne esistono differenti ma qui si farà riferimento allo Static Parametric Model (SPM) [[12]] nella forma:

$$z = \theta^{*T} \phi \quad (4.44)$$

Con $\phi^* \in \mathbb{R}^n$ vettore dei parametri noti o non noti, $z \in R, \phi \in R^n$ sono invece i segnali misurati in input e output. Questo formalismo permette di determinare i parametri non noti nella legge di controllo creando una dipendenza lineare tra i termini di θ e quelli del relativo stato ϕ . Viceversa la stessa scrittura può essere utilizzata per forzare parametri conosciuti e viene utilizzata in seguito per la definizione dei disturbi parametrici noti.

GRADIENT BASED PARAMETER ESTIMATION

Il modello per per la stima dei parametri basato su gradiente è un metodo che si basa sul calcolo del gradiente di una funzione detta “potenziale”. Questa funzione fittizia ha la finalità di avere concavità rivolta verso l’alto e come punto di minimo un punto di equilibrio stabile in cui l’errore considerato tenda a zero. Questo metodo è già stato accennato per sistemi di navigazione come APF. Si presenta in seguito la sua definizione per sistemi lineari tramite SPM ma può essere applicato anche a sistemi non lineari adattando opportunamente il modello. Si consideri ora la variabile di stato z e \hat{z} la sua stima con SPM:

$$\hat{z} = \theta^T \phi \quad (4.45)$$

E’ possibile definire l’errore normalizzato nel modo seguente (4.46):

$$\epsilon = \frac{z - \hat{z}}{m_s^2} = \frac{z - \theta^T \phi}{m_s^2} \quad (4.46)$$

con $m_s^2 > 1$ è il fattore di normalizzazione del segnale per rendere ϕ bounded. Solitamente il fattore m_s prende la forma $m_s^2 = 1 + n_s^2$ con $n_s >= 0$. Si riesce così a garantire il rapporto ϕ/m_s limitato e perciò il singolo valore dello stato ϕ bounded. Da queste ipotesi si può ricavare la funzione gradiente. Quest'ultimo deve essere con concavità rivolta verso l'alto, essere continuo e derivabile, con un punto di minimo pari alla condizione voluta. Per fare ciò è intuitivo creare una funzione che abbia la sua dipendenza dell'errore e come punto di minimo il termine di errore pari a zero. Per soddisfare queste caratteristiche si sceglie la funzione nella forma (4.47):

$$J(\theta) = \frac{\epsilon^2 m_s^2}{2} = \frac{(z - \theta^T \phi)^2}{2m_s^2} \quad (4.47)$$

Per definizione si vuole che la stima del parametro porti il sistema a stabilizzarsi nel punto di minimo, che è anche di equilibrio. Per tanto assume la forma (4.48):

$$\dot{\theta} = -\Gamma \quad (4.48)$$

Con $\Gamma = \Gamma^T > 0$ matrice di guadagno adattivo. Il gradiente in questo caso assume il valore (4.49):

$$\nabla J = -\frac{(z - \theta^T \phi)\phi}{m_s^2} = -\epsilon\phi \quad (4.49)$$

Ottenendo perciò $\dot{\theta} = \Gamma\epsilon\phi$. Integrando numericamente questa espressione si ottiene la variazione temporale del parametro voluta.

Qualora il sistema sia non lineare il procedimento è analogo. La funzione ha infatti la forma generale $y(t) = f(x(t), \theta)$ con f modello del sistema e $x(t)$ vettore degli stati. L'errore è la differenza tra il valore stimato di prima ipotesi (con cappello $\hat{\cdot}$) e il valore reale ricavando $e(t) = y(t) - \hat{y}(t)$.

La funzione potenziale assume comunque la forma $J(\theta) = \frac{1}{2} \sum_{i=1}^N \frac{e_i(t)^2}{m_s^2}$ ritornando infine all'espressione generale: $\dot{\theta} = -\Gamma \nabla J$.

Come si può osservare il processo teorico è il medesimo. Bisogna però fare attenzione a calcolare il gradiente in modo opportuno. Nel caso di sistemi non lineari infatti questa operazione potrebbe non essere banale. Il metodo del gradiente ricade comunque in una sezione più ampia di algoritmi Least-Squared il cui obiettivo è trovare il minimo di una funzione quadratica che può essere sia continua che discreta. In genere si presta bene all'integrazione numerica essendo la derivata del parametro indice della variazione dello stesso per passo d'integrazione. Da osservare come questo metodo abbia forti implicazioni nei modelli DDC citati nei paragrafi precedenti come SPSA e UCM per la selezione dei parametri ottimali.

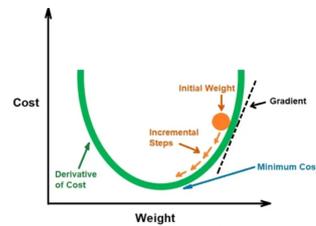


Figura 4.12: Rappresentazione grafica Gradiente

FEEDBACK LINEARIZATION

L'idea del feedback linearization è quella di fare un cambio di coordinate al fine di eliminare le non linearità del sistema per ottenere un modello LTI più facilmente gestibile, soprattutto con il metodo Stato-Spazio. Si propone in seguito la metodologia generale considerando il sistema non lineare nella forma (4.50):

$$\begin{cases} \dot{x} = f(x) + g(x)u \\ y = h(x) \end{cases} \quad (4.50)$$

Con u termine di comando, x stato del sistema e f , g e h tre funzioni infinitamente differenziabili. Derivando l'output y rispetto al tempo si ottiene l'espressione: $\dot{y} = dh/dx \cdot dx/dt$ e sostituendo l'espressione precedente si giunge a (4.51):

$$\dot{y} = \frac{\partial h}{\partial x}(x)f(x) + \frac{\partial h}{\partial x}(x)g(x)u \quad (4.51)$$

Si introduce ora la nozione di derivata di Lie che prende la forma (4.52):

$$\frac{\partial h}{\partial x}f = \frac{\partial h}{\partial x_1}(f_1) + \dots + \frac{\partial h}{\partial x_n}(f_n) = L_f h \quad (4.52)$$

Il grado della derivata è uguale al numero di volte che l'output deve essere differenziato prima che l'input compaia nella sua espressione.

Ad esempio se $\frac{\partial}{\partial x}(\frac{\partial h}{\partial x}(x)f(x))g(x)|_{x=x_0} \neq 0$ allora il grado sarà 2.

Si introduce inoltre la notazione della derivata di Lie i -esima come (4.53):

$$L_f^i h = \frac{\partial}{\partial x} \left(\frac{\partial}{\partial x} \left(\dots \frac{\partial}{\partial x} \left(\frac{\partial}{\partial x} f \right) \cdot f \dots \right) \cdot f \right) \cdot f \quad (4.53)$$

Rispetto la funzione g si definisce la derivata composta come (4.54):

$$L_g L_f h = \frac{\partial(L_f h)}{\partial x} \cdot g \quad (4.54)$$

Ora ipotizzando che il grado della funzione derivata sia pari ad "n" si può effettuare il seguente cambio di variabili:

$$z_1 = y = h(x), \quad z_2 = \dot{y} = L_f h(x), \quad z_3 = \ddot{y} = L_f^2 h(x), \quad \dots \quad z_n = L_f^{n-1} h(x) \quad (4.55)$$

Derivando si ottiene:

$$\dot{z}_1 = z_2, \quad \dot{z}_2 = z_3, \quad \dots \quad \dot{z}_n = L_f^n h(x) + (L_g L_f^{n-1} h(x))u \quad (4.56)$$

Da notare che per il termine di grado "n" il valore della derivata rispetto a $g(x)$ non è più zero. Pertanto esso compare come derivata composta ed insieme a lui

comparare il termine di comando u . Da quest'espressione è così possibile ricavare il comando a valle della linearizzazione del modello come (4.57):

$$u = \frac{1}{L_g L_f^{n-1} h(x)} [v - L_f^n h(x)] \quad (4.57)$$

Ottenendo la nuova variabile di input $v = \dot{z}_n$. Si ricava perciò il nuovo sistema stato spazio linearizzato come:

$$\begin{cases} \dot{z} = Az + Bv \\ y = Cz \end{cases} \quad (4.58)$$

Questo cambio di variabili cancella le non linearità tramite il feedback forzando il sistema di close loop a comportarsi come un LTI. Sebbene questa tecnica non sia di per se una tecnica adattiva, viene soventemente utilizzata a monte di un controllore per poter ottenere un sistema LTI e dunque una parametrizzazione lineare. Questo consente tramite comuni tecniche di parameter estimation come il gradiente di ottenere controllori a parametri variabili adattivi. Il principale vantaggio è la semplificazione del modello che permette di ottenere tecniche di controllo più varie, che nel caso non lineare potrebbero avere problemi di applicazione. In generale però non sempre è applicabile in quanto oneroso dal punto di vista computazionale, dovendo in alcuni casi calcolare derivate di alto ordine per linearizzare tutti gli effetti della dinamica secondaria che viceversa verrebbero persi.

CONTROL LYAPUNOV FUNCTION (CLF)

L'analisi di Lyapunov è largamente utilizzata per provare la stabilità nell'intorno di un punto di equilibrio delle equazioni differenziali. Questo potente metodo viene perciò utilizzato in generale per dimostrare la stabilità di una legge di controllo ma può essere a sua volta utilizzato per creare una nuova legge di controllo partendo dall'analisi matematica del problema e valutando l'evoluzione dell'errore che si vuole annullare nel tempo. Diversi sono i teoremi inerenti alla funzione di Lyapunov che permettono di dimostrare la stabilità di un sistema. In generale è possibile osservare che scegliendo V come funzione candidata (4.59):

$$\dot{V}(x) = \frac{d}{dt} V(x(t)) = \frac{\partial V}{\partial x} \cdot \frac{dx}{dt} = \nabla V \cdot \dot{x} = \nabla V \cdot f(x) \quad (4.59)$$

Si dimostra che nel caso in cui $\dot{V}(x) < 0$ per qualsiasi x nell'intorno del punto di equilibrio allora la dinamica è stabile.

Per la definizione di funzione candidata di Lyapunov di un sistema si fa riferimento alla letteratura [[8], [9], [12]].

Il metodo CLF si basa sul teorema di stabilità e prevede la valutazione di una funzione candidata $V(x)$ partendo dal sistema non lineare $\dot{x} = f(x) + g(x)u$. In

particolare u , essendo adattivo sarà funzione di x nella forma $u=q(x)$. Obiettivo di CLF è valutare u tale che:

$$\frac{\partial V}{\partial x}f(x) + \frac{\partial V}{\partial x}g(x)q(x) \leq -Q(x) \quad (4.60)$$

Con $Q(x)$ funzione definita positiva. E' chiaro che per far si che la disequazione sia rispettata è fondamentale scegliere una funzione di Lyapunov opportuna. In tal caso infatti si riuscirà a ricavare l'espressione di $q(x)=u$ che permette di rispettare la disuguaglianza e garantendo stabilità. Il CFL, se applicato correttamente, è in grado di restituire all'osservatore attento direttamente la legge di controllo adattiva $q(x)$ che stabilizza il sistema. Purtroppo tale approccio non è sempre realizzabile perché non è detto che la funzione candidata che risolve la disequazione esista. Inoltre vi è anche la possibilità di ricavare una funzione più complessa del dovuto, garantendo la stabilità ma con costi computazionali maggiori di altri metodi. Tale metodo verrà esplicitato con la legge di controllo ottenuta per il simulatore MRAC adottato come progetto per questa tesi.

BACKSTEPPING

Il controllo tramite backstepping è un metodo che si basa su un processo iterativo volto a scomporre il sistema di equazione "risolvendone" una alla volta a ritroso. Per trovare la soluzione si utilizza una funzione candidata di Lyapunov per ogni sottosistema in cui si scompone il problema. Si consideri il seguente caso:

$$\begin{cases} \dot{x}_1 = f_1(x_1) + g_1(x_1)x_2 \\ \dot{x}_2 = f_2(x_1, x_2) + g_2(x_1, x_2)u \end{cases} \quad (4.61)$$

Qui la funzione in \dot{x}_1 non dipende dal comando u ma da x_2 . E' la seconda equazione in \dot{x}_2 a dipendere dal comando che influenza a ritroso anche la prima.

Per risolvere tale problema si considera una prima funzione candidata, che per semplicità viene considerata nella forma $V_1(x_1) = 1/2 \cdot x_1^2$. Nel momento in cui viene derivata rispetto il tempo si otterrà $dV/dt = dV/dx * dx/dt$ ovvero: $\dot{V}_1 = x_1(f_1(x_1) + g_1(x_1)x_2)$. E' chiaro che nell'espressione non compare il termine u perché la dipendenza è rispetto ad x_2 . Per questo motivo è necessario introdurre un cambio di variabile $z = x_2 - q(x_1)$ dove $q(x_1)$ è la funzione che potenzialmente stabilizzerebbe l'equazione su \dot{x}_1 se x_2 fosse il comando. In questo modo è possibile introdurre il nuovo sistema con z nel seguente modo:

$$\begin{cases} \dot{x}_1 = f_1(x_1) + g_1(x_1) \cdot (z + q(x_1)) \\ \dot{z} = \dot{x}_2 - \dot{q} \end{cases} \quad (4.62)$$

Ciò che si può notare è la retroazione del valore di \dot{q} che nella nuova variabile compare derivato. Computazionalmente il valore è come se avesse risalito un blocco

integratore ottenendo il fenomeno del Backstepping. A questo punto è possibile passare alla seconda equazione definendo il candidato della funzione di Lyapunov come $V_2(x_1, x_2) = V_1(x_1) + 1/2 \cdot z^2$. Derivando tale espressione si ottiene $dV_2/dt = dV_1/dt + \dot{z}$. Il termine \dot{z} contiene al suo interno il termine \dot{x}_2 che contiene a sua volta u . E' perciò possibile con un passaggio a ritroso calcolare u in funzione di q dalla funzione candidata di Lyapunov V_2 e sostituire tali valori nella prima equazione per ricavare il comando u finale che stabilizza il sistema.

Questo processo iterativo può essere svolto più volte ottenendo nella forma generale:

$$\begin{cases} \dot{x}_1 = f(x_1) + g(x_1)x_2 \\ \dot{x}_2 = x_3 \\ \dots \\ \dot{x}_k = u \end{cases} \quad (4.63)$$

con relative funzioni di Lyapunov del tipo $V_a(x_1, \dots, x_k) = V(x_1) + \frac{1}{2} \sum_{i=2}^k z_i^2$ considerando gli errori z_i come $z_i = x_i - q_{i-1}(x_1, x_2, \dots, x_{i-1})$, $i = 2, 3, \dots, k$.

In questo caso $q_1(x_1)$ è il control input che stabilizza $\dot{x}_1 = f(x_1) + g(x_1)u$.

$q_2(x_1, x_2)$ è il control input che stabilizza $\begin{cases} \dot{x}_1 = f_1(x_1) + g_1(x_1) \cdot x_2 \\ \dot{x}_2 = u \end{cases}$ ed in generale

è possibile esprimere la relazione:

$$\begin{cases} \dot{x}_1 = f_1(x_1) + g_1(x_1) \cdot x_2 \\ \dot{x}_2 = x_3 \\ \dots \\ \dot{x}_r = u \end{cases} \quad (4.64)$$

Tale approccio può essere reso adattivo nel caso in cui si scelga il comando nella forma $u = -\theta^T f(x) - cx$ supponendo che esista un set di parametri θ^* in grado di determinare il sistema non lineare nella forma $\dot{x} = \theta^T f(x) + u$. In questo caso una legge adattiva che porti i parametri θ a tendere ai valori θ^* condurrebbe ad una dinamica di close loop con $c > 0$ stabile. Partendo da questa ipotesi è possibile ricavare sostituendo all'interno della derivata della funzione di Lyapunov la legge adattiva che porta \dot{V} ad essere minore di 0. Tale legge avrà la forma $\dot{\theta}$. Il grosso vantaggio di questo approccio è che si può smontare il problema in diversi sottosistemi da affrontare in modo ricorsivo. Ciò permette di affrontare modelli anche molto complessi ed è garantita la stabilità grazie all'analisi secondo funzione di Lyapunov. Il grande svantaggio è la richiesta computazionale che aumenta con l'aumentare delle equazioni in gioco. Aumenta inoltre anche la complessità matematica rendendo sempre meno leggibile l'insieme delle soluzioni.

ADAPTIVE POLE PLACEMENT CONTROL (APPC)

Le caratteristiche desiderate vengono spesso definite in termini di posizione dei poli localizzati dal controllore. Se il plant è osservabile e controllabile è possibile creare un controllore tale da rispettare le caratteristiche volute. La combinazione del pole placement control unito a parameter estimator conduce all'adaptive pole placement control (APPC) che può essere utilizzato per controllare un'ampia gamma di plant LTI (lineari tempo invarianti). E' possibile dividere questo tipo di controllo in due classi:

- **APPC Indiretto**, dove la legge adattiva genera le stime online dei coefficienti della funzione di trasferimento del plant che vengono utilizzati successivamente per calcolare i parametri di APPC.
- **APPC Diretto**, i parametri di APPC vengono valutati direttamente tramite legge adattiva senza passare per la stima dei coefficienti del plant

In questo caso si vuole fare riferimento ad un semplice esempio di APPC diretto rimandando alla letteratura [[12]] per l'approfondimento in casi più complessi. Si consideri il plant nella forma (4.65):

$$\dot{y} = ay + bu \quad (4.65)$$

In cui a e b sono costanti ma non note. L'obiettivo della legge di controllo è far sì che il controllo u conduca ai poli di close loop con valore $-a_m$ con $a_m > 0$, oltre che a valori di y e u bounded e che $y(t)$ converga a zero per $t \rightarrow \infty$. Per ottenere tale risultato è possibile sommare e sottrarre il termine $-a_my$ arrivando all'espressione (4.66)

$$\dot{y} = -a_my + (a + a_m)y + bu \quad (4.66)$$

Considerando $a + a_m = bk^*$ si ottiene:

$$\dot{y} = -a_my + b(k^*y + u) \quad (4.67)$$

La legge di controllo a cui si vuole arrivare è nella forma seguente:

$$u = -ky \quad (4.68)$$

Sostituendo tale valore in 4.67 si ottiene l'equazione:

$$\dot{y} = -a_my - b\tilde{k}y \quad ; \quad \tilde{k} = k - k^* \quad (4.69)$$

Considerando \tilde{k} l'errore del parametro adattivo.

Scegliendo l'opportuna funzione candidata di Lyapunov $V(y, \tilde{k})$ è possibile calcolare la derivata e ottenere il valore di \dot{k} che soddisfa $\dot{V} < 0$ garantendo stabilità interna. In questo modo si ottiene la legge adattiva voluta che garantisce la posizione dei poli stabili.

4.3.3 Esempi architetture adattive

Controllo L1

Il controllo adattivo L_1 è una metodologia avanzata di controllo progettata per sistemi dinamici incerti, in cui si affrontano sia parametri sconosciuti sia disturbi esterni [18]. Una caratteristica distintiva del controllo L_1 è l'uso di un filtro passa basso che separa le dinamiche del sistema adattivo dalla dinamica del sistema fisico, permettendo di mantenere stabilità e robustezza.

Si consideri il plant del tipo (4.70):

$$\dot{x}(t) = -a_m x(t) + b(u(t) + \theta x(t)) \quad (4.70)$$

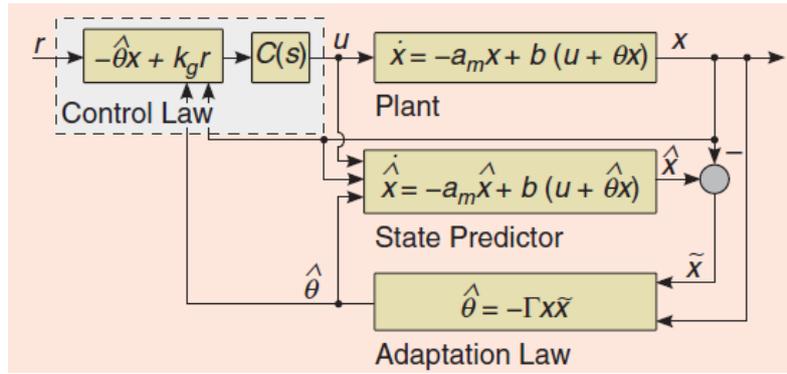


Figura 4.13: Architettura L_1

Si consideri $x(t)$ vettore di stato, $u(t)$ control input, a_m poli del sistema, b gain dell'input e θ disturbi parametrici limitati (bounded).

L'architettura L_1 proposta è formata da uno state predictor (definito dal cappello $\hat{\cdot}$) nella forma (4.71):

$$\dot{\hat{x}}(t) = -a_m \hat{x}(t) + b(u(t) + \hat{\theta} \hat{x}(t)) \quad (4.71)$$

Si specifica inoltre la legge adattiva ottenuta tramite funzione candidata di Lyapunov nella forma $\dot{\hat{\theta}} = -\Gamma x \tilde{x}$ con $\tilde{x} = x - \hat{x}$ errore tra plant e stima.

La legge di controllo viene scelta con due termini, uno dipendente dalla stima del parametro e l'altro alla reference. Prende la forma (4.72):

$$\hat{\eta}(t) = -\hat{\theta}(t)x(t) + k_g r(t) \quad (4.72)$$

Il nome dell'architettura L_1 deriva dalla norma di tipo L_1 che garantisce la limitatezza della risposta del plant. Si definisce norma L_∞ di y come:

$$\|y\|_{L_\infty} = \sup |y(t)| \quad (4.73)$$

Si definisce norma L_1 della funzione di trasferimento $G(s)$ come:

$$\|G(s)\|_{L_1} = \int_0^{\infty} |g(\tau)| d\tau \quad (4.74)$$

Con $g(t)$ risposta ad impulso di $G(s)$ considerando $y(t) = \int_0^t g(t-\tau)u(\tau)d\tau$. Con queste ipotesi si può giungere alla relazione:

$$\|y\|_{L_\infty} \leq \|G(s)\|_{L_1} \|u\|_{L_\infty} \quad (4.75)$$

Questa relazione viene utilizzata in seguito per dimostrare la caratteristica bounded della risposta al controllo L_1 .

In particolare facendo riferimento alla figura 4.13 si può definire il comando $u(s)$:

$$u(s) = C(s)\hat{\eta}(s) \quad (4.76)$$

Con $\hat{\eta}(s)$ la trasformata di Laplace del segnale $\hat{\eta}(t)$. Conoscendo la stima dei parametri si ottiene la forma completa del controllore di riferimento:

$$u_{ref} = C(s) \cdot (-\theta x_{ref}(s) + k_g r(s)) \quad (4.77)$$

Sostituendo $u=u_{ref}$ in 4.70 si ottiene il sistema di riferimento L_1 :

$$x_{ref} = H(s)C(s)k_g r(s) + H(s)(1 - C(s))\theta x_{ref}(s) + x_{in} \quad (4.78)$$

Considerando

$$H(s) = \frac{b}{s + a_m} \quad ; \quad x_{in} = \frac{1}{s + a_m} x_0 \quad (4.79)$$

Considerando $x_{in}(s)$ la trasformata di Laplace della risposta alla condizione iniziale. Il filtro passa basso ha la forma:

$$C(s) = \frac{\omega_c}{s + \omega_c} \quad (4.80)$$

Con $\omega_c > 0$ banda passante del filtro. Sfruttando la relazione 4.75 è possibile ricavare la relazione:

$$\|x_{ref}\|_{L_\infty} \leq \|H(s)C(s)k_g\|_{L_1} \|r\|_{L_\infty} + \|H(s)(1 - C(s))\theta\|_{L_1} \|x_{ref}\|_{L_\infty} + \|x_{in}\|_{L_\infty} \quad (4.81)$$

Si consideri $G(s)=H(s)(1-C(s))$. E' possibile verificare che $\|G(s)\theta\|_{L_1} < 1$ e si può perciò dimostrare il limite (bound) che si risolve per $\|x_{ref}\|$ come:

$$\|x_{ref}\|_{L_\infty} \leq \frac{\|H(s)C(s)k_g\|_{L_1} \|r\|_{L_\infty} + \|x_{in}\|_{L_\infty}}{1 - \|G(s)\theta\|_{L_1}} \quad (4.82)$$

L'analisi matematica di questa architettura mostra l'importanza del filtro passa basso per ottenere la risposta al comando bounded e stabile. In generale l'architettura L_1 permette prestazioni robuste, velocità di adattamento elevate e riduzione del rumore ad alta frequenza. I problemi principali possono essere eventuali complessità di progettazione, sensibilità del filtro alla frequenza di taglio che deve essere selezionata accuratamente e un potenziale costo computazionale elevato.

Controllo MRAC

Il controllo MRAC (Model Reference Adaptive Control) è una tecnica di controllo adattativo in cui il comportamento del sistema da controllare (sistema reale o plant) è forzato a seguire un andamento desiderato definito da un modello di riferimento [4] [22]. Quest'ultimo deve essere controllato in modo tale da ottenere una dinamica verosimile, che il modello reale è in grado di seguire. In caso contrario si osserverà un'incontrollabilità del modello che porterà necessariamente a instabilità. E' perciò necessario conoscere il modello matematico accurato del plant che viene successivamente semplificato in modo adeguato per ottenere il model reference.

La seconda componente dell'MRAC è la legge di adattamento che può essere ottenuta tramite Parameter Estimation con uno dei metodi specificati nei capitoli precedenti. Tale adattamento è funzione dell'errore tra la dinamica reale e quella del modello di riferimento.

Il parametro viene utilizzato dalla legge di controllo che computa il comando adattivo in input al plant [8]. Si mostra in figura uno schema base dell'architettura MRAC (4.14).

Si specifica in seguito in modo più puntuale questo approccio in quanto è quello scelto per la realizzazione del controllore adattivo soggetto della tesi.

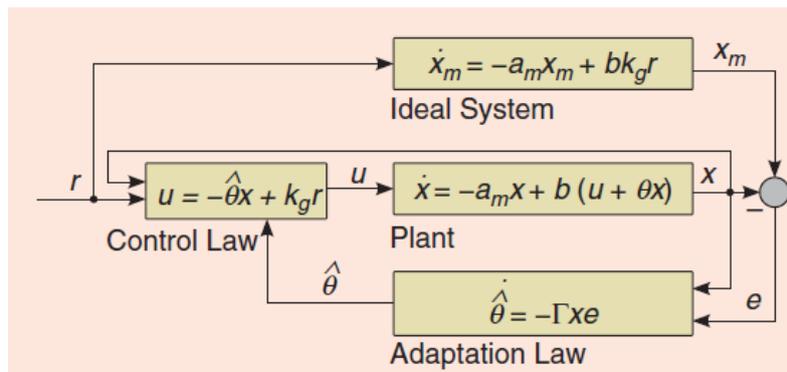


Figura 4.14: Architettura MRAC

Capitolo 5

Controllore MRAC per drone

5.1 Drone Parrot Mambo

Il controllore MRAC realizzato è adattato al Parrot Mambo, un drone di piccole dimensioni progettato per fini ludici e/o didattici. E' estremamente semplice da utilizzare e può essere migliorato con diversi accessori in grado di espandere le sue funzionalità, come un braccio allungabile o uno spara palline a molla. L'autonomia è di circa 10 minuti con batterie LiPo da 660mAh. Il controllo avviene tramite smartphone, tablet o Flypad della Parrot con connessione tramite bluetooth. Il Mambo monta anche una telecamera a bassa risoluzione, adatta per fini ludici ma non riprese professionali.



Figura 5.1: Parrot Mambo

Specifica	Parametro	Unità di misura	Valore
Massa	m	kg	0.063
Lunhezza braccio	l	m	0.0624
Thrust Factor	K_t	Ns^2	0.0107
Torque Factor	K_q	Nms^2	0.78264×10^{-3}
Momento d'inerzia Rollio	I_{xx}	kgm^2	0.0000582857
Momento d'inerzia Beccheggio	I_{yy}	kgm^2	0.0000716914
Momento d'inerzia Imbardata	I_{zz}	kgm^2	0.0001
Momento d'inerzia Eliche Rotore	J_r	kgm^2	0.1021×10^{-6}

Tabella 5.1: Parametri fisici Parrot Mambo [13]

5.2 Architettura generale

In figura 5.2 si mostra il progetto Simulink completo. I principali blocchi sono quello di Reference, Il modello di riferimento, il plant, la legge adattiva, la legge di controllo e il blocco visual per visualizzare i risultati.

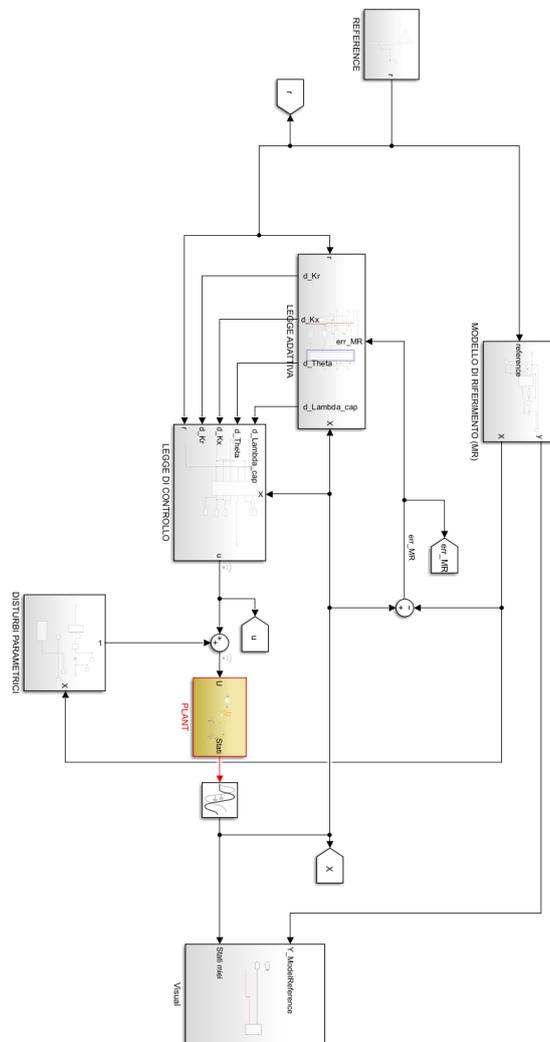


Figura 5.2: Architettura MRAC Completa

5.3 Plant non lineare

Per la realizzazione del Plant si utilizza un primo modello ottenuto dalle equazioni non lineari. In particolare si fa riferimento alle equazioni della dinamica complete. Si deriva il sistema di equazioni 3.16 e si sostituisce in 3.15 escludendo i termini di ordine 2. Inoltre per ipotesi di angoli di rotazione piccoli si equiparano $[\dot{\phi}, \dot{\theta}, \dot{\psi}]$ con $[p, q, r]$. Si deriva poi il sistema 3.17 sostituendo all'interno 3.18. Così facendo si ottiene il sistema non lineare semplificato utilizzato per il plant come segue:

$$\ddot{x} = \frac{F(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi)}{m} \quad (5.1)$$

$$\ddot{y} = \frac{F(\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi)}{m} \quad (5.2)$$

$$\ddot{z} = g - \frac{(\cos \phi \cos \theta)F}{m} \quad (5.3)$$

$$\dot{x} = v_x \quad (5.4)$$

$$\dot{y} = v_y \quad (5.5)$$

$$\dot{z} = v_z \quad (5.6)$$

$$\ddot{\phi} = \frac{I_{yy} - I_{zz}}{I_{xx}} \dot{\theta} \dot{\psi} + \frac{\dot{\theta} J_r \Omega_m}{I_{xx}} + \frac{\tau_\phi}{I_{xx}} \quad (5.7)$$

$$\ddot{\theta} = \frac{I_{zz} - I_{xx}}{I_{yy}} \dot{\phi} \dot{\psi} - \frac{\dot{\phi} J_r \Omega_m}{I_{yy}} + \frac{\tau_\theta}{I_{yy}} \quad (5.8)$$

$$\ddot{\psi} = \frac{I_{xx} - I_{yy}}{I_{zz}} \dot{\phi} \dot{\theta} + \frac{\tau_\psi}{I_{zz}} \quad (5.9)$$

$$\dot{\phi} = \omega_\phi \quad (5.10)$$

$$\dot{\theta} = \omega_\theta \quad (5.11)$$

$$\dot{\psi} = \omega_\psi \quad (5.12)$$

Il vettore di stato che si utilizza è composto da 12 termini come segue (5.13):

$$[x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T \quad (5.13)$$

Considerando:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} ; \quad \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \omega_\phi \\ \omega_\theta \\ \omega_\psi \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (5.14)$$

Tali valori di velocità si ottengono integrando i relativi valori di accelerazione. Si indica infine con Ω_m la velocità media dei rotori come:

$$\Omega_m = -\Omega_1 - \Omega_2 + \Omega_3 + \Omega_4 \quad (5.15)$$

Si condidera il vettore di comando:

$$\mathbf{U} = [F, \tau_\phi, \tau_\theta, \tau_\psi]^T \quad (5.16)$$

Grazie a 5.16 è possibile sostituire i valori noti in 3.19 ricavando l'espressione delle 4 velocità angolari dei rotori come:

$$\omega_1 = \sqrt{\left| \frac{\tau_\theta}{2K_t} - \frac{\tau_\psi}{4K_q} + \frac{F}{4K_t} \right|} \quad (5.17)$$

$$\omega_2 = \sqrt{\left| \frac{1}{4} \left| \frac{F}{K_t} - 2 \frac{\tau_\phi}{K_t} + \frac{\tau_\psi}{K_q} \right| \right|} \quad (5.18)$$

$$\omega_3 = \sqrt{\left| \frac{1}{4} \left| \frac{F}{K_t} - 2 \frac{\tau_\theta}{K_t} + \frac{\tau_\psi}{K_q} \right| \right|} \quad (5.19)$$

$$\omega_4 = \sqrt{\left| \frac{\tau_\phi}{2K_t} + \frac{\tau_\psi}{4K_q} + \frac{F}{4K_t} \right|} \quad (5.20)$$

Con due funzioni Matlab si realizza il blocco Plant (5.3).

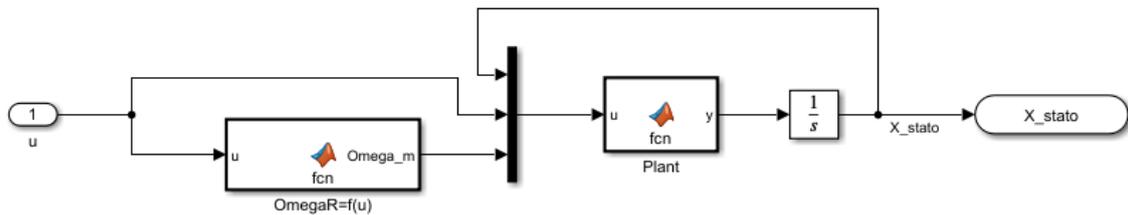


Figura 5.3: Plant Simulink

5.4 Modello di riferimento - Plant Lineare

Come modello di riferimento si sceglie di linearizzare il sistema di equazioni descritto precedentemente. In particolare si fanno le seguenti assunzioni:

- Si trascurano effetti giroscopici e aerodinamici che non vengono considerati nel computo delle forze;
- Essendo la forza verticale F uguale al peso in hover si apporta la seguente approssimazione:

$$\frac{F}{m} \approx \frac{m \cdot g}{m} = g \quad (5.21)$$

Tale approssimazione viene fatta solo per le direzioni x e y mentre per z si mantiene il termine F/m perchè la forza F su tale asse può essere anche molto maggiore e perciò l'approssimazione sarebbe troppo forte;

- Come indicato precedentemente si fa l'ipotesi di piccoli angoli per cui nell'equazione 3.18 la matrice diventa identità ottenendo:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \approx \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (5.22)$$

- Per l'ipotesi di piccoli angoli si può semplificare:

$$\cos \alpha \approx 1 \quad ; \quad \sin \alpha \approx \alpha \quad (5.23)$$

Con α angolo generico di assetto.

- I termini di secondo grado di accoppiamento come la moltiplicazione tra due velocità angolari vengono trascurate:

$$\dot{\theta}\dot{\psi} \approx 0 \quad ; \quad \dot{\phi}\dot{\psi} \approx 0 \quad ; \quad \dot{\phi}\dot{\theta} \approx 0 \quad (5.24)$$

- Si trascurano i termini inerziali delle pale ovvero $J_r = 0$

Così facendo si ottiene il sistema di dodici equazioni linearizzato come:

$$\dot{x} = \dot{x} \quad (5.25)$$

$$\dot{y} = \dot{y} \quad (5.26)$$

$$\dot{z} = \dot{z} \quad (5.27)$$

$$\ddot{x} = -g\theta \quad (5.28)$$

$$\ddot{y} = g\phi \quad (5.29)$$

$$\ddot{z} = \frac{F}{m} \quad (5.30)$$

$$\dot{\phi} = \dot{\phi} \quad (5.31)$$

$$\dot{\theta} = \dot{\theta} \quad (5.32)$$

$$\dot{\psi} = \dot{\psi} \quad (5.33)$$

$$\ddot{\phi} = \frac{\tau_{\phi}}{I_{xx}} \quad (5.34)$$

$$\ddot{\theta} = \frac{\tau_{\theta}}{I_{yy}} \quad (5.35)$$

$$\ddot{\psi} = \frac{\tau_{\psi}}{I_{zz}} \quad (5.36)$$

Il vettore di stato che si utilizza è composto da 12 termini (uguale a 5.13). Il pedice ref sta ad indicare l'appartenenza al modello di riferimento:

$$\mathbf{X}_{\text{ref}} = [x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T$$

In questo modo è possibile esprimere il sistema come Lineare Tempo Invariante (LTI) e trasporlo nella formulazione **State Space** o **Stato Spazio (SS)** del tipo:

$$\begin{cases} \dot{\mathbf{X}}_{\text{ref}} = [A] \cdot \mathbf{X}_{\text{ref}} + [B] \cdot U \\ \mathbf{y} = [C] \cdot \mathbf{X}_{\text{ref}} \end{cases} \quad (5.37)$$

Con $\mathbf{X}_{\text{ref}} \in \mathbb{R}^{12}$ vettore di stato, $y \in \mathbb{R}^4$ vettore di output, $\mathbf{U} \in \mathbb{R}^4$ vettore dei comandi come in 5.16, $[A] \in \mathbb{R}^{12 \times 12}$ matrice di stato, $[B] \in \mathbb{R}^{4 \times 4}$ matrice dei comandi e $[C] \in \mathbb{R}^{4 \times 12}$ matrice di output.

Si specificano in seguito le tre matrici complete:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 & \frac{1}{I_{zz}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.38)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (5.39)$$

Canonicamente nella formulazione State Space è presente anche la matrice [D] che in questo caso viene posta uguale a 0.

In figura si rappresenta il blocco Model Reference.

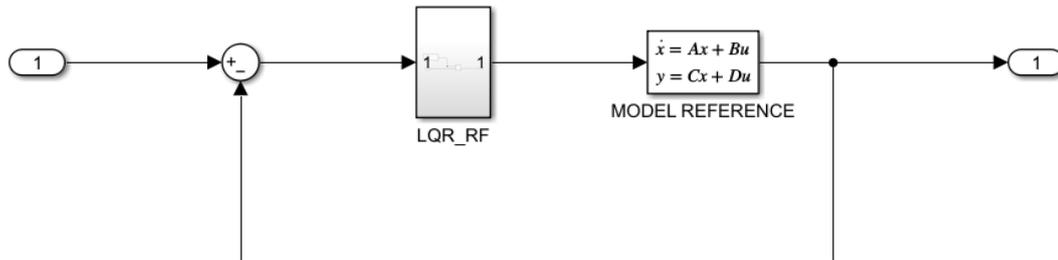


Figura 5.4: Model Reference Simulink

5.5 Stabilizzazione Model Reference LQR

Il moello di riferiento deve essere controllato perciò si sceglie di creare a monte un LQR che permette di soddisfare le Matching Condition espresse in seguito. Inoltre questo tipo di controllore è semplice da implementare e rende la dinamica stabile. Come proposto nell'equazione 4.21 il controllore LQR minimizza la funzione di costo J tramite le matrici Q e R . In generale avere un rapporto $\frac{Q}{R}$ alto comporta un controllo più reattivo mentre un valore basso predilige il "risparmio" in termini di comando rendendo il controllo meno impulsivo.

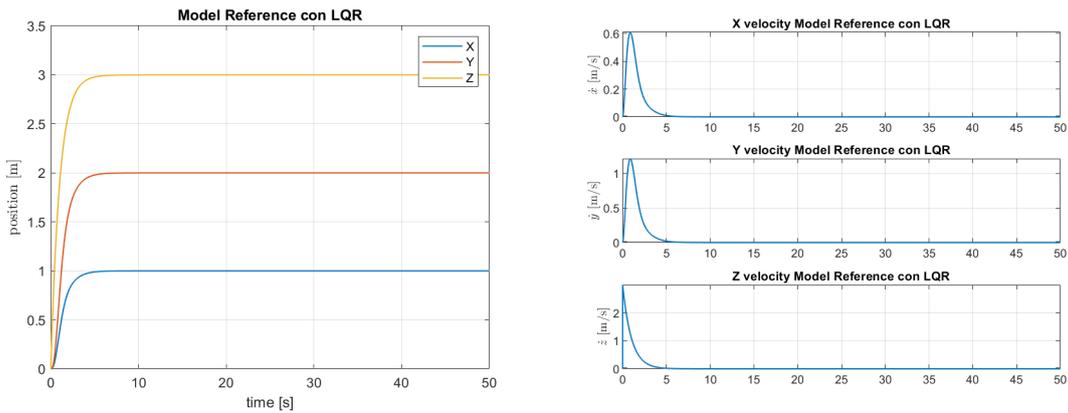
Il comando prende la forma:

$$\mathbf{U} = -[K_{LQR}] \cdot \mathbf{X}_{\text{err}} \tag{5.40}$$

La matrice $K_{LQR} \in \mathbb{R}^{4 \times 12}$ viene computata direttamente tramite comando Matlab. Il vettore $\mathbf{X}_{\text{err}} = \mathbf{X}_{\text{ref}}(\mathbf{t}) - \mathbf{r}(\mathbf{t})$ rappresenta l'errore mentre $\mathbf{r}(\mathbf{t})$ è la reference. La matrice $Q \in \mathbb{R}^{12 \times 12}$ mentre $R \in \mathbb{R}^{4 \times 4}$. Le matrici vengono scelte in entrambi i casi diagonali. Si propongono in seguito 2 esempi che mostrano l'impatto del rapporto $\frac{Q}{R}$. Si sceglie come comando reference i valori $[3,2,1,0]$ rispettivamente per $[x, y, z, \psi]$.

5.5.1 Esempio 1: Rapporto $\frac{Q}{R}$ alto

In questo primo esempio si sceglie la matrice Q con elementi diagonali pari a 100 mentre R con elementi pari a 0.001. Come si può notare il comando è intensivo e



(a) Posizioni Lineari

(b) Velocità Lineari

Figura 5.5: Dinamica traslazionale Q/R alto

comporta un transitorio breve (circa 5 secondi), privo di overshoot ed estremamente stabile.

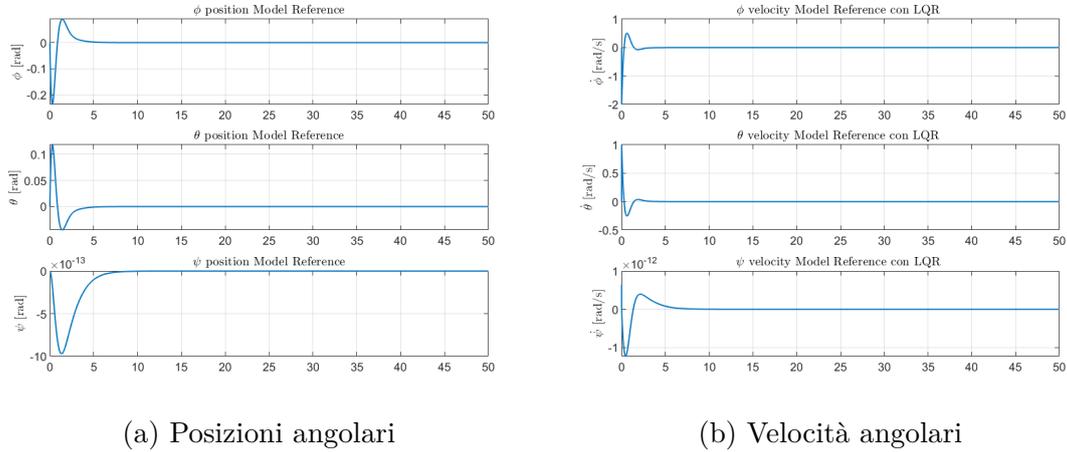


Figura 5.6: Dinamica rotazionale Q/R alto

5.5.2 Esempio 2: Rapporto $\frac{Q}{R}$ basso

In questo caso si sceglie la matrice Q con elementi diagonali pari a 0.001 mentre R con elementi pari a 100.

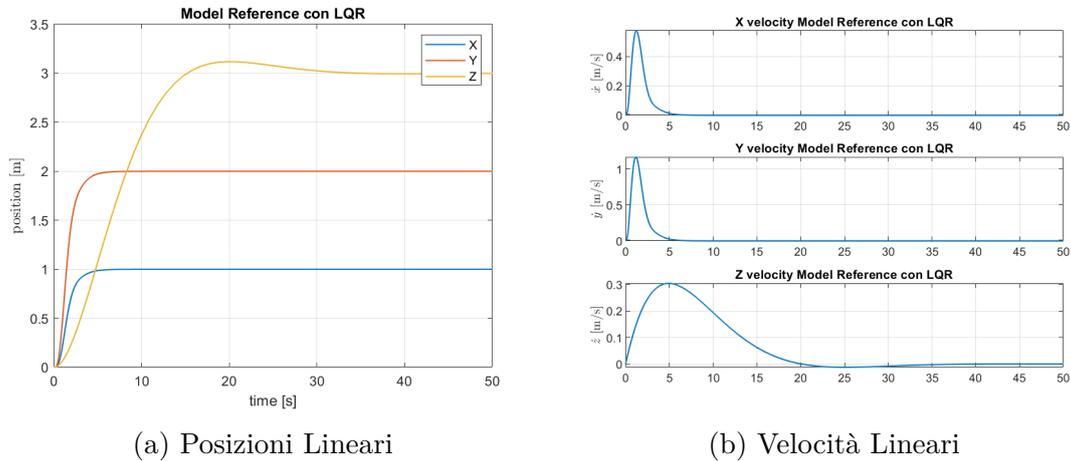


Figura 5.7: Dinamica traslazionale Q/R basso

In questo secondo caso il valore basso del rapporto Q/R fa sì che le posizioni comandate vengano raggiunte in modo più graduale. In particolare si può osservare come i tempi del transitorio su z siano pari a 40 secondi mentre su x e y non siano cambiati in modo considerevole rispetto a prima.

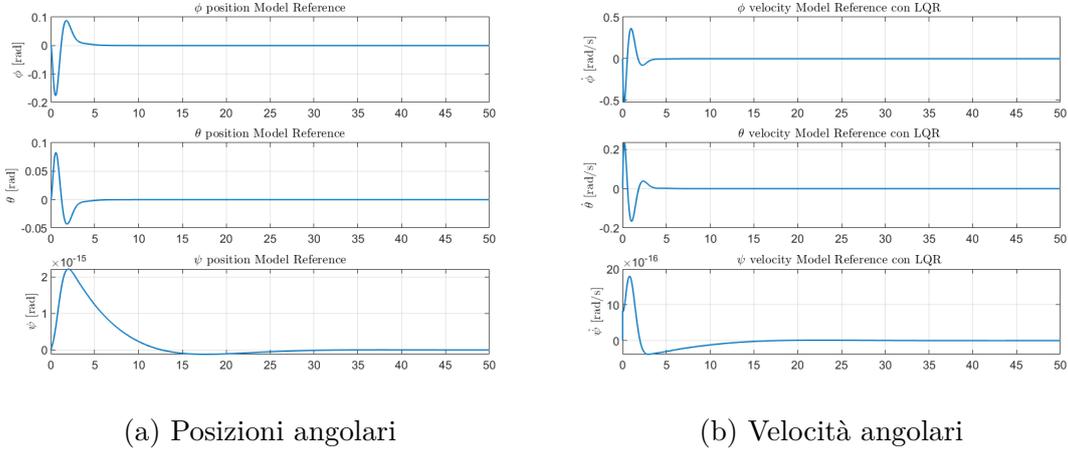


Figura 5.8: Dinamica rotazionale Q/R basso

5.5.3 Modello di close loop Model Reference

Volendo garantire dei transienti bassi senza stressare troppo il controllo si sceglie di attribuire agli elementi di Q valore 1 mentre a R valore 3. Questa scelta porta i risultati estremamente simili all'esempio 1 garantendo transienti all'incirca pari a 5 secondi.

Sostituendo l'equazione 5.40 in 5.37 è possibile ottenere l'espressione:

$$\dot{\mathbf{X}}_{\text{ref}} = \underbrace{A - BK_{LQR}}_{A_{\text{ref}}} \cdot \mathbf{X}_{\text{ref}} + \underbrace{B \cdot K_{LQR}}_{B_{\text{ref}}} \cdot \mathbf{r} \quad (5.41)$$

Ottenendo l'equazione di close loop 5.42:

$$\dot{\mathbf{X}}_{\text{ref}} = A_{\text{ref}} \cdot \mathbf{X}_{\text{ref}} + B_{\text{ref}} \cdot \mathbf{r} \quad (5.42)$$

Ora è possibile calcolare gli autovalori del sistema stabilizzato oltre che i valori di smorzamento e frequenza naturale (5.2).

Come si può osservare gli autovalori sono tutti stabili con 3 di essi in alta frequenza che porta il sistema di close loop ad annullare rapidamente l'errore con una condizione steady state stabile.

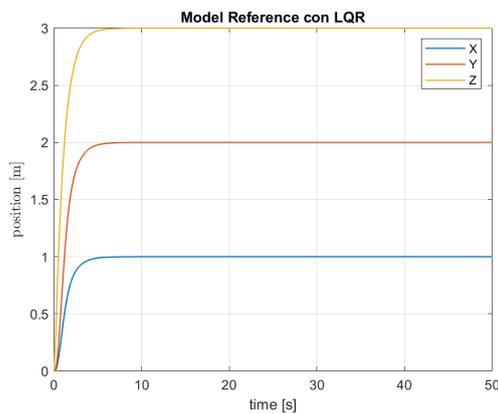
Si presentano in seguito i valori simulati traslazionali e rotazionale con reference:

$$\mathbf{r} = [3, 2, 1, 0] \quad (5.43)$$

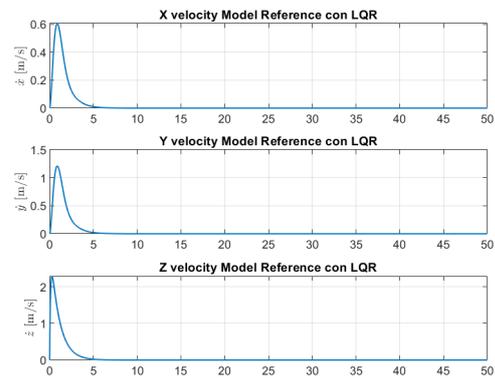
Come già per gli esempi 1 e 2.

Autovalore	Smorzamento	Frequenza naturale
$-5.65 \cdot e^6$	1	$5.65 \cdot e^6$
$-5.02 \cdot e^2$	1	502.52
$-2.21 + i 2.21$	0.70	3.13
$-2.21 - i 2.21$	0.70	3.13
-1.00	1	1
$-6.18 \cdot e^2$	1	618.10
-9.10	1	9.11
$-2.21 + 2.21$	0.70	3.13
$-2.21 - 2.21$	0.70	3.13
-1.00	1	1
-1.00	1	1
-1.00	1	1

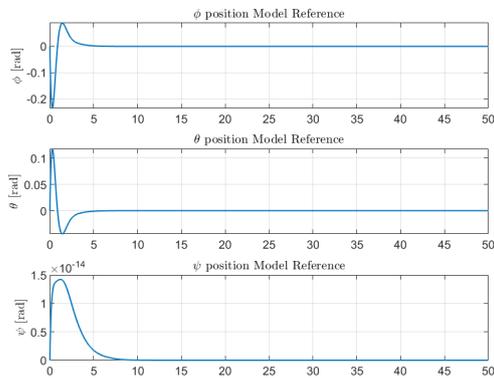
Tabella 5.2: Autovalori, Smorzamento e frequenza naturale sistema Close Loop



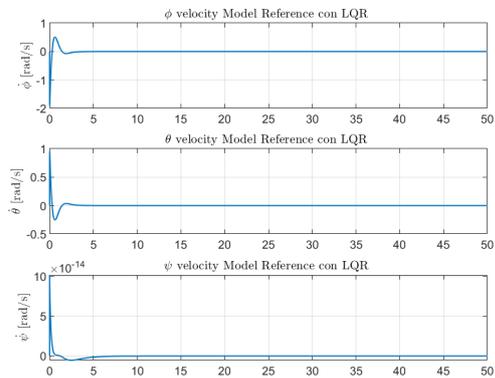
(a) Posizioni Lineari



(b) Velocità Lineari



(a) Posizioni angolari



(b) Velocità angolari

5.6 Legge di controllo e legge adattiva

In questa sezione si affronta il modello matematico che conduce alla formulazione della legge di controllo e della legge adattiva dell'architettura MRAC [9]. In generale si consideri un sistema plant con disturbi del tipo:

$$\dot{\mathbf{X}} = A \cdot \mathbf{X} + B \cdot (U + f(x)) \quad (5.44)$$

Con $X \in \mathbb{R}^{12}$ vettore di stato del sistema, $u \in \mathbb{R}^4$ vettore di comando, $B \in \mathbb{R}^{12 \times 4}$ matrice di controllo nota e $A \in \mathbb{R}^{12 \times 12}$ matrice di stato non nota costante. La funzione:

$$f(x) = \Theta^T \cdot \psi(x) \quad (5.45)$$

E' vettore dei disturbi parametrici con $\psi(x) = (\gamma_1(x), \dots, \gamma_N(x))^T \in \mathbb{R}^N$ vettore delle funzioni di base mentre $\Theta \in \mathbb{R}^{N \times 4}$ matrice dei coefficienti non noti a priori. Questa formulazione viene utilizzata per tenere conto dei disturbi parametrici e si crea una legge adattiva di conseguenza. Successivamente si valuteranno anche altri tipi di disturbi in una sezione dedicata.

L'obiettivo della legge di controllo è quella di ottenere un errore nullo tra la dinamica del sistema reale e quello del sistema di riferimento ovvero:

$$e(t) = x(t) - x_{ref}(t) \quad \text{con} \quad \lim_{t \rightarrow \infty} \|x(t) - x_{ref}(t)\| = 0 \quad (5.46)$$

La legge di controllo ideale è (5.47):

$$u = K_x^T x + K_r^T r - \Theta^T \Psi(x) \quad (5.47)$$

Sostituendo 5.47 in 5.44 si ottiene l'equazione di close loop ideale:

$$\dot{x} = (A + BK_x^T)x + BK_r^T r \quad (5.48)$$

E' così possibile esprimere le **Matching Condition** per il sistema MRAC che garantiscono la possibilità di convergenza teorica tra la dinamica reale e di riferimento:

$$\begin{cases} A + BK_x^T = A_{ref} \\ BK_r^T = B_{ref} \end{cases} \quad (5.49)$$

Se le matching condition vengono rispettate è facile verificare che la dinamica di close loop reale coincide con quella di riferimento. Affinchè ciò sia possibile si è scelto il controllore di tipo LQR per il model reference che tramite le espressioni di close loop 5.41 rispettano queste condizioni.

Il **controllo adattivo reale** è formato da termini stimati (indicati con il simbolo $\hat{\cdot}$) nella forma:

$$u = \hat{K}_x^T x + \hat{K}_r^T r - \hat{\Theta}^T \Psi(x) \quad (5.50)$$

Con $\hat{K}_x \in \mathbb{R}^{12 \times 4}$, $\hat{K}_r \in \mathbb{R}^{4 \times 4}$, $\hat{\Theta} \in \mathbb{R}^{N \times 4}$ matrici di guadagno stimate proporzionali allo stato x , reference r e vettore di basi note Ψ . Sostituendo 5.50 in 5.44 si ottiene la funzione di close loop reale come:

$$\dot{x} = (A + B\hat{K}_x^T)x + B(\hat{K}_r^T r - (\hat{\Theta} - \Theta)^T \Psi(x)) \quad (5.51)$$

Sottraendo 5.51 a 5.44 si ottiene la formulazione dell'errore derivato nel tempo ovvero:

$$\dot{e} = \dot{x} - \dot{x}_{ref} = (A + B\hat{K}_x^T)x + B(\hat{K}_r^T r - (\hat{\Theta} - \Theta)^T \Psi(x)) - A_{ref}x_{ref} - B_{ref}r \quad (5.52)$$

Sostituendo le Matching Condition 5.49 si ottiene:

$$\dot{e} = (A_{ref} + B(\hat{K}_x - K_x)^T)x - A_{ref}x_{ref} + B(\hat{K}_r - K_r)^T r - B(\hat{\Theta} - \Theta)^T \Psi(x) \quad (5.53)$$

$$\dot{e} = A_{ref}e + B[(\hat{K}_x - K_x)^T x + (\hat{K}_r - K_r)^T r - (\hat{\Theta} - \Theta)^T \Psi(x)] \quad (5.54)$$

Si indica $\Delta K_x = \hat{K}_x - K_x$, $\Delta K_r = \hat{K}_r - K_r$, $\Delta \Theta = \hat{\Theta} - \Theta$ gli errori di stima dei parametri ottenendo l'espressione:

$$\dot{e} = A_{ref}e + B[\Delta K_x^T x + \Delta K_r^T r - \Delta \Theta^T \Psi(x)] \quad (5.55)$$

Si introducono i ratei di adattamento come: $\Gamma_x = \Gamma_x^T > 0$, $\Gamma_r = \Gamma_r^T > 0$, $\Gamma_\Theta = \Gamma_\Theta^T > 0$. In questo modo è possibile scrivere la funzione candidata per l'analisi di Lyapunov come:

$$V(e, \Delta K_x, \Delta K_r, \Delta \Theta) = e^T P e + tr[\Delta K_x^T \Gamma_x^{-1} \Delta K_x + \Delta K_r^T \Gamma_r^{-1} \Delta K_r + \Delta \Theta^T \Gamma_\Theta^{-1} \Delta \Theta] \quad (5.56)$$

Con $P = P^T > 0$ soluzione dell'equazione algebrica :

$$P A_{ref} + A_{ref}^T P = -Q \quad (5.57)$$

Tramite opportuna analisi matematica espressa in Appendice è possibile giungere alla formulazione della **Legge Adattiva** come:

$$\begin{cases} \dot{\hat{K}}_x = -\Gamma_x x e^T P B \\ \dot{\hat{K}}_r = -\Gamma_r r(t) e^T P B \\ \dot{\hat{\Theta}} = \Gamma_\Theta \Phi(x) e^T P B \end{cases} \quad (5.58)$$

La derivata temporale di V diventa negativa portando l'errore a tendere a zero per t che tende ad infinito:

$$\dot{V} = -e^T Q e \leq 0 \quad (5.59)$$

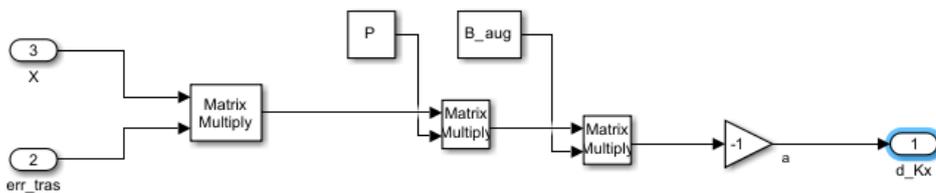


Figura 5.11: Legge adattiva per \hat{K}_x

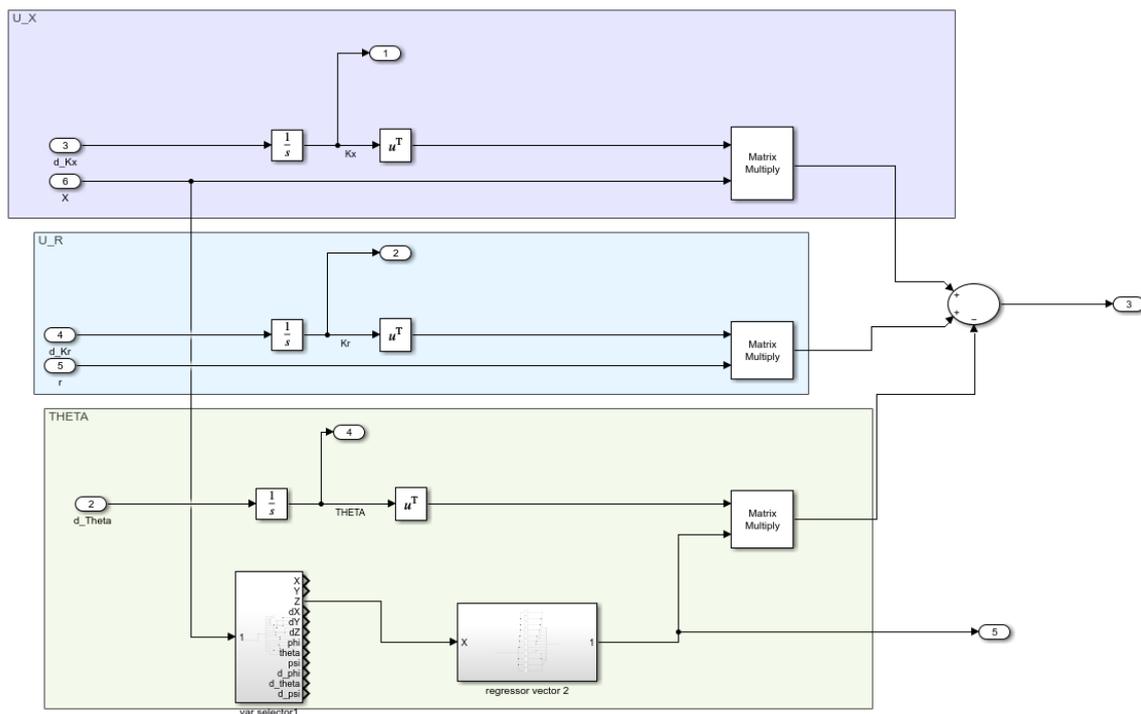


Figura 5.12: Legge di Controllo (5.50)

5.7 Disturbi

In generale è possibile rappresentare e simulare una gran varietà di disturbi. Volendoli indicare rapidamente alcuni di essi sono:

- Vento, in modo:
 - Continuo, come una folata di vento costante;
 - Raffica, vento turbolento rappresentato da sommatoria di sinusoidi;
 - Turbolento non lineare;
- Payload, disturbo creato dall'aumento o la diminuzione improvvisa del payload che comporta una variazione puntuale della massa oltre che una variazione del baricentro;
- efficienza di comando, perdita di efficienza simulata con un fattore correttivo che moltiplica il termine di comando riducendolo di un valore percentuale scelto;
- Incertezze sul plant, derivanti dalla matrice A variabile o con incertezze intrinseche.

In questo caso si farà riferimento esclusivamente a disturbi aerodinamici come il vento che vengono divisi in due categorie sotto esplicate.

5.7.1 Disturbi Parametrici

Per disturbi parametrici si intendono tutti quei disturbi in grado di essere rappresentati tramite la notazione parametrica come già accennato precedentemente. Il riferimento teorico è quello in 4.44 con il modello SPM (Standard Parametric Model) [12]. In questa categoria di disturbi ricadono sia i termini lineari (come una raffica costante) sia i termini non lineari valutati a partire da un vettore delle basi più complesso [14]. In generale si può esprimere il disturbo come un vettore delle basi che moltiplica una matrice dei parametri noti. Tale formulazione corrisponde a:

$$f(x) = \Theta^T \cdot \psi(x) \tag{5.60}$$

Con $\psi(x) = (\gamma_1(x), \dots, \gamma_N(x))^T \in \mathbb{R}^N$ funzione delle basi mentre $\Theta \in \mathbb{R}^{N \times 4}$ matrice dei coefficienti. Il termine N può essere scelto a priori come la tipologia delle funzioni di base. E' però importante che esso sia rappresentativo di un possibile disturbo reale. La matrice dei coefficienti invece è quella che, moltiplicando le funzioni di base, permette di ottenere i disturbi globali sui 4 termini di comando di u.

In questo caso si sceglie un vettore composto da 11 elementi ($N=11$) con vettore ψ composto da RBF (Radial Basis Function) nella forma:

$$\psi(x) = \begin{bmatrix} e^{-\frac{\|z-c_1\|^2}{\sigma_1}} \\ \dots \\ e^{-\frac{\|z-c_i\|^2}{\sigma_i}} \\ \dots \\ e^{-\frac{\|z-c_{11}\|^2}{\sigma_{11}}} \end{bmatrix} \quad (5.61)$$

Con z posizione verticale del drone, c_i vettore dei centroidi equispaziati tra -10 e 10 e σ ampiezza dei RBF pari a 0.023 [[18], [2]].

La matrice Θ viene invece selezionata come:

$$\Theta = \begin{bmatrix} 1 & 2 & 1 & 2 & 1 & 3 & 1 & 4 & 1 & 1 & 1 \\ 1 & 1 & 1 & 5 & -1 & 3 & -1 & 1 & -2 & 1 & 1 \\ 1 & 1 & -3 & 1 & 1 & 5 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 4 & 1 & 5 & 1 & 7 & 1 & 1 & 10 \end{bmatrix} \quad (5.62)$$

Nel modello High Fidelity proposto in seguito il disturbo non viene più imposto a livello di comando ma direttamente nel plant nel blocco di computazione delle forze e momenti agenti sul drone. Tale approccio verrà mostrato in seguito.

5.7.2 Disturbi non parametrici

Nella sezione disturbi non parametrici si segnala il modello matematico di Dryden utilizzato per valutare disturbi turbolenti. In particolare tale modello matematico viene indicato dal Military Specification MIL-F-8785C come prassi per la valutazione di disturbi turbolenti sia in bassa che alta quota.

Questo approccio consiste nel valutare le Power Spectral Density (PSD) delle funzioni di trasferimento (TF) su tre assi vento. Tali TF sono dei filtri che ricevendo in input white noise restituiscono in output i valori di velocità utilizzati per il calcolo delle PSD e i relativi spettri di turbolenza su tutti e tre gli assi. Ad esempio nel caso dell'asse longitudinale (velocità di disturbo $u_{gust} = u_g$) dalle specifiche MIL si ottiene:

$$G_{u_g}(s) = \sqrt{\frac{2\sigma_u^2 V}{L_u}} \cdot \frac{1}{s + \frac{V}{L_u}} \quad (5.63)$$

Dove:

- σ_u è la deviazione standard della turbolenza longitudinale,
- V è la velocità del veicolo rispetto all'aria,
- L_u è la lunghezza di scala della turbolenza longitudinale,

- s è la variabile del dominio di Laplace.

Da questa funzione è possibile ricavare la PSD:

$$\Phi_{u_g}(\omega) = |G_{u_g}(s)|^2 \quad (5.64)$$

Espressioni simili possono essere ricavate per le altre velocità traslazionali e rotazionali.

Si nota come la funzione G sia un filtro passa basso che riceve in input il white noise, il cui comportamento è determinato dalla lunghezza caratteristica L e dalla velocità di volo V .

I valori puntuali per i tre assi vengono computati con l'apposito blocco Simulink e poi ruotati tramite la matrice di rotazione da assi vento ad assi body per poter valutare gli effetti di resistenza indotti dalle componenti di velocità relativa dei disturbi.

5.8 Simulazioni Proposte

Le simulazioni in seguito proposte vogliono mostrare il comportamento del controllore senza un tuning puntuale. Tale affinamento viene svolto nei capitoli a seguire. L'obiettivo di questa sezione è verificare la bontà del comportamento di Disturbance rejection oltre che il raggiungimento e mantenimento delle posizioni comandate con e senza disturbi.

Le simulazioni proposte sono:

- **Senza Disturbi**
 - Reference a gradino costante;
 - Reference ad "otto";
 - Reference ad Elica discendente.
- **Con Disturbi Parametrici lineari - raffica costante**
 - Reference a gradino costante;
- **Con Disturbi Parametrici non lineari**
 - Reference sinusoidale.

I disturbi non parametrici con il modello di Dryden vengono utilizzati per la validazione del controllore in un secondo momento con il modello High Fidelity grazie alla computazione puntuale delle forze di disturbo. Tali considerazioni vengono per ora omesse per il modello non lineare in questione.

5.8.1 Condizioni iniziali

Si considerano le seguenti condizioni iniziali della legge adattiva con i guadagni di cui sotto:

$$\Gamma_x = \begin{bmatrix} d_{x_1} & & \\ & \ddots & \\ & & d_{x_{12}} \end{bmatrix} \quad \text{con} \quad (d_{x_1}, \dots, d_{x_{12}}) = 10 \quad (5.65)$$

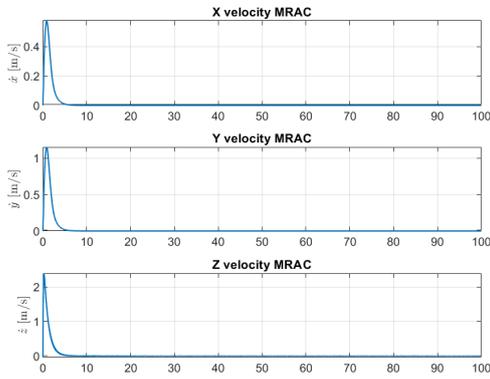
$$\Gamma_r = \begin{bmatrix} d_{r_1} & & \\ & \ddots & \\ & & d_{r_4} \end{bmatrix} \quad \text{con} \quad (d_{r_1}, \dots, d_{r_4}) = 10 \quad (5.66)$$

$$\Gamma_\Theta = \begin{bmatrix} d_{\Theta_1} & & \\ & \ddots & \\ & & d_{\Theta_{11}} \end{bmatrix} \quad \text{con} \quad (d_{\Theta_1}, \dots, d_{\Theta_{11}}) = 10 \quad (5.67)$$

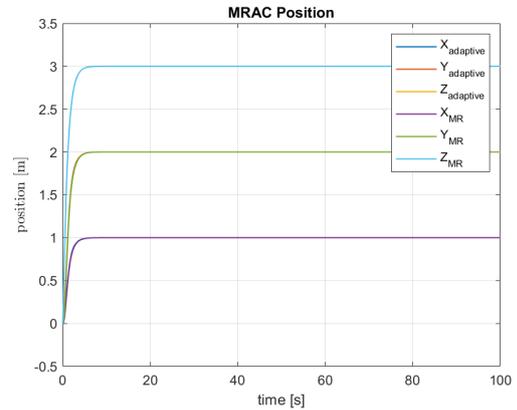
5.8.2 Simulazioni senza disturbi

Reference gradino costante

In questa simulazione si propone una reference di tipo costante come gradino dal valore $r = [x_{ref}, y_{ref}, z_{ref}, \psi_{ref}] = [3, 2, 1, 0]$.



(a) Velocità Lineari



(b) Posizioni Lineari

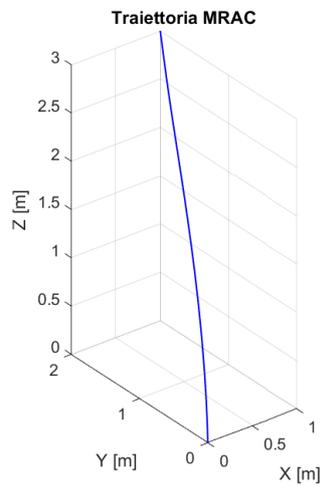
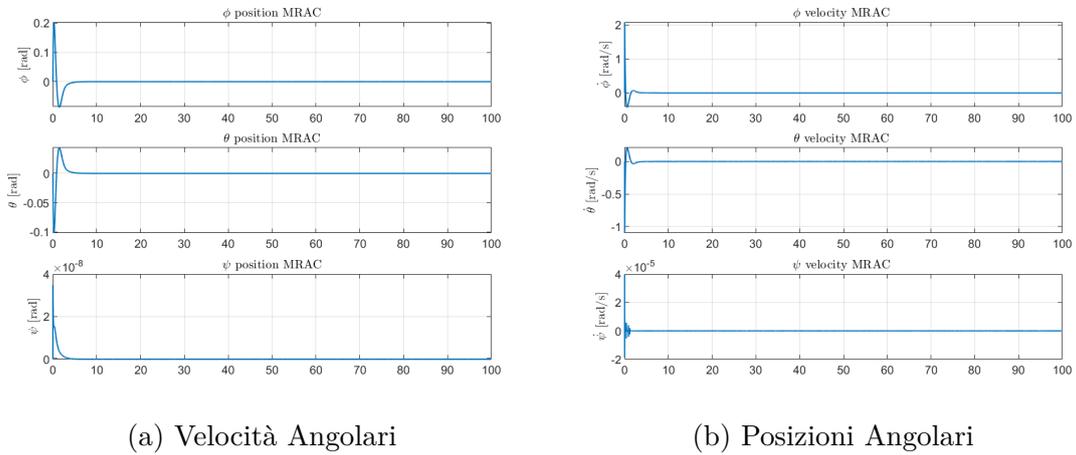
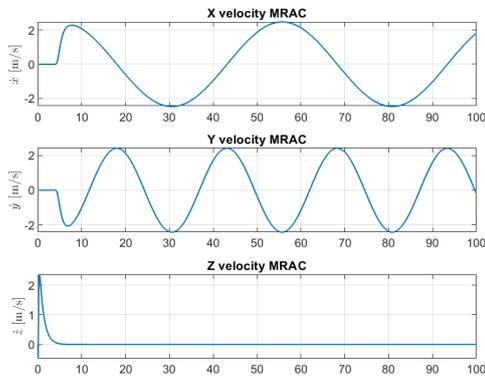


Figura 5.15: Simulazione 3D

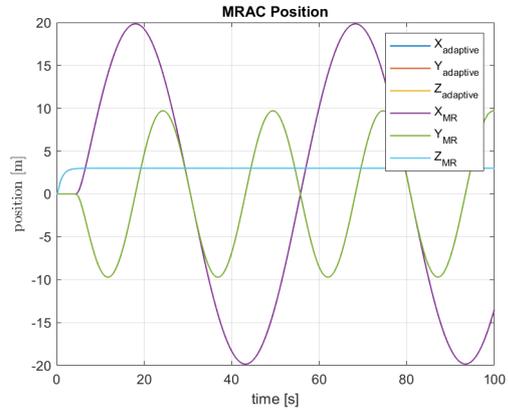
Come si può notare da [5.13b](#) il plant raggiunge i valori di riferimento in modo stabile, senza overshoot e con un transitorio di circa 6 secondi. La dinamica del drone si sovrappone perfettamente a quella del modello di riferimento seguendola, come desiderato.

Reference ad otto

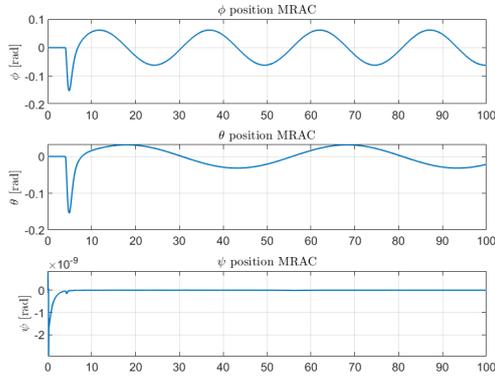
In questa simulazione si vuole portare il drone a 3 metri di quota e fargli compiere un 8 con escursione massima della x pari a 20 metri e escursione massima della y uguale a 10 metri.



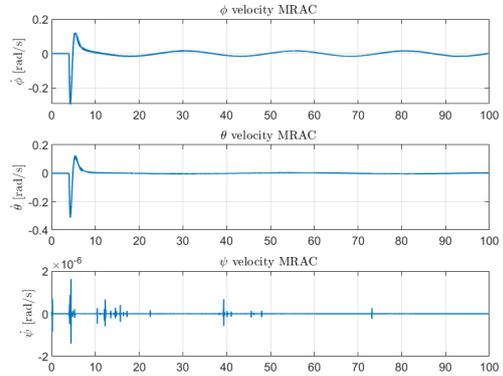
(a) Velocità Lineari



(b) Posizioni Lineari



(a) Velocità Angolari



(b) Posizioni Angolari

Come si può osservare da 5.18 e 5.16b la posizione del plant coincide con quella del modello di riferimento con un errore massimo di 20 cm al vertice del otto quando la y raggiunge 20 metri.

Sia gli angoli che posizioni si intersecano opportunamente con due sinusoidi come desiderato. Vi sono alcune piccole variazioni si ψ date dal modello high fidelity che, accoppiando rollio e imbardata, crea dei piccoli disturbi sull'asse z che devono essere compensati dal controllore.

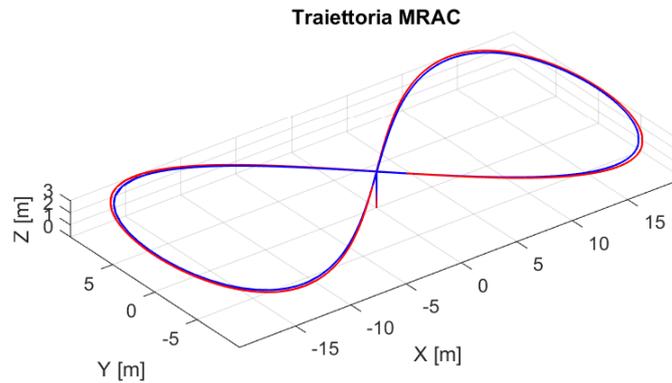
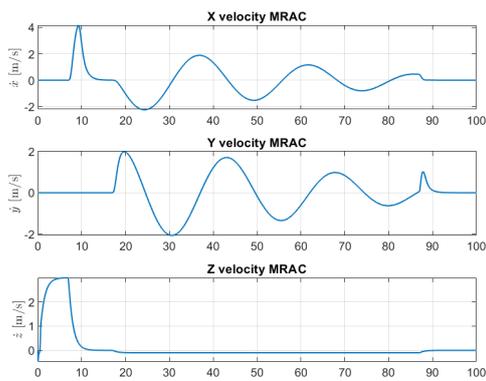


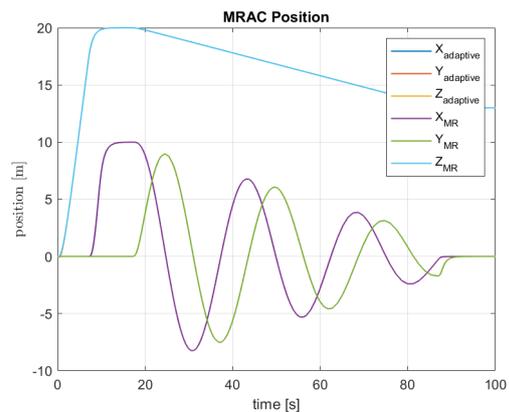
Figura 5.18: Simulazione 3D

Reference ad elica discendente

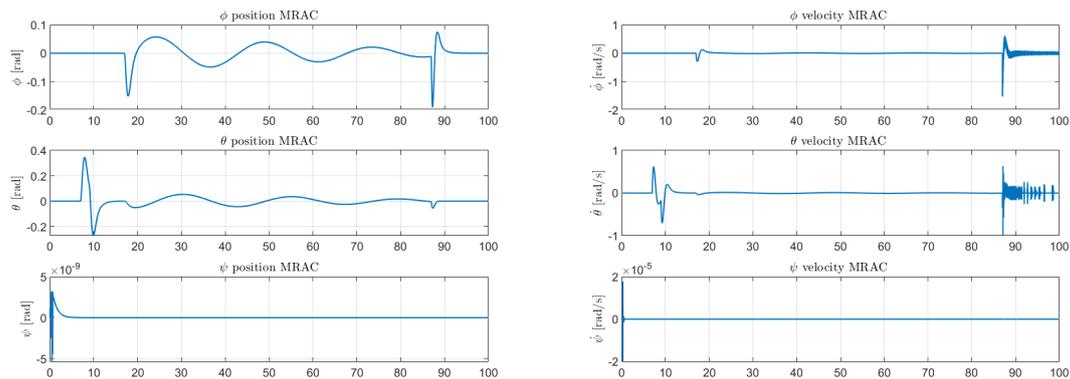
In questa terza simulazione si vuole mostrare la bontà del controllore nel seguire reference più complesse come un elica discendente a raggio variabile. Per questo si porta il drone a venti metri di quota con una rampa e poi si incomincia ad eseguire l'elica di diametro 10 metri partendo dal secondo 17 fino al secondo 87, in cui si impone al drone di ritornare di nuvo in posizione verticale con x e y pari a 0.



(a) Velocità Lineari



(b) Posizioni Lineari



(a) Velocità Angolari

(b) Posizioni Angolari

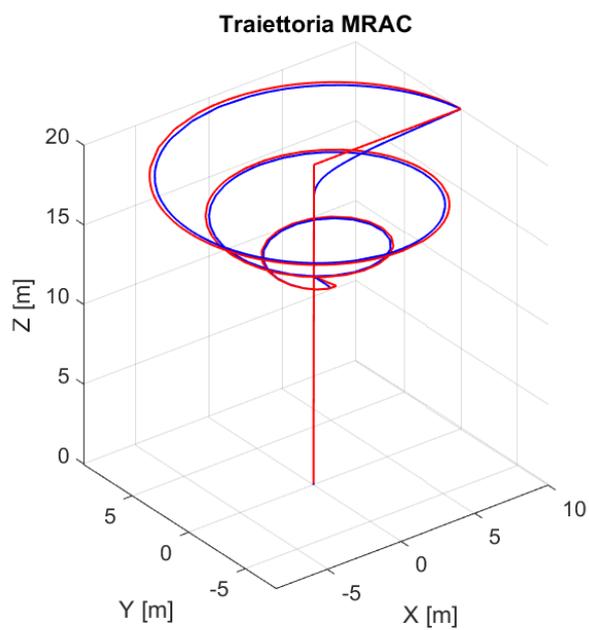
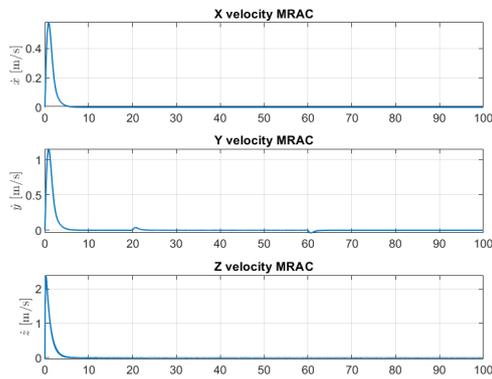


Figura 5.21: Simulazione 3D

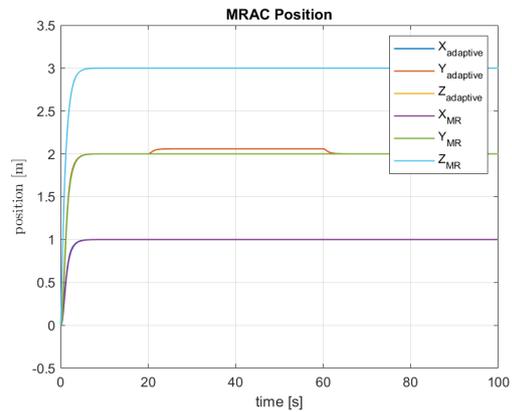
5.8.3 Simulazioni con disturbi a raffica costante

Reference gradino costante

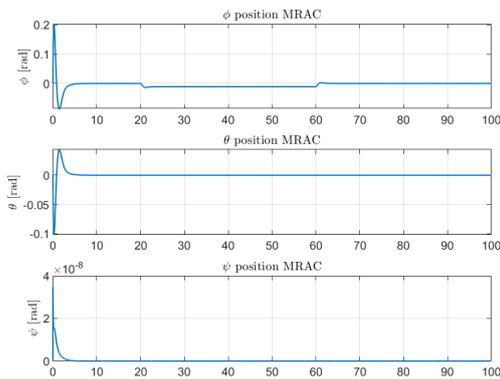
In questa simulazione si vuole utilizzare la stessa reference della simulazione a gradino ma aggiungendo un vento di intensità 10m/s in direzione positiva y_b imponendo il disturbo direttamente sul comando. Come si può osservare nella sezione disturbi



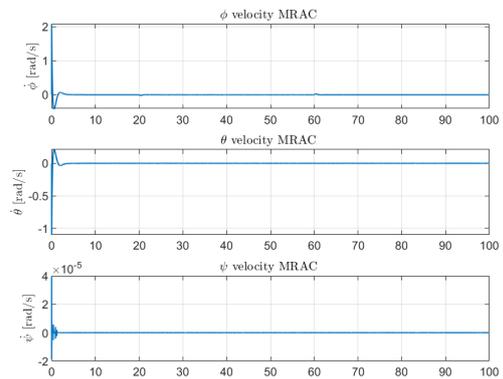
(a) Velocità Lineari



(b) Posizioni Lineari

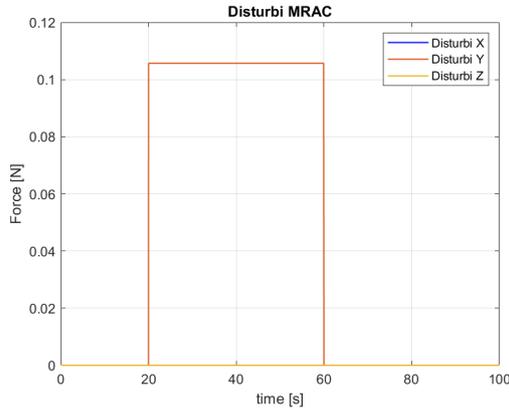


(a) Velocità Angulari

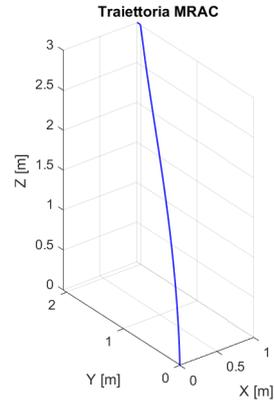


(b) Posizioni Angulari

il vento comporta una forza di circa 0.107 N, ottenuta considerando piena la superficie esposta al vento. A causa di questo contributo si ottiene uno scostamento della posizione lineare in y come dell'angolo di assetto di rollio che tende a contrastare il vento. Al secondo 60 il disturbo cessa e il drone si riporta in una condizione di equilibrio.



(a) Disturbi

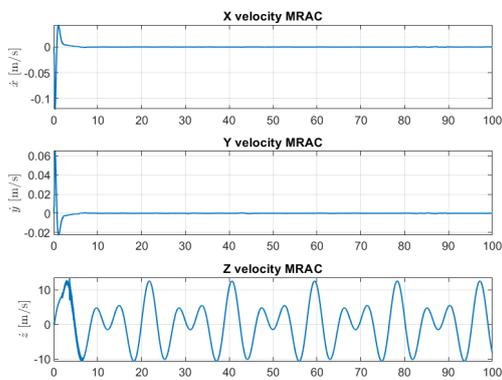


(b) Posizioni 3D

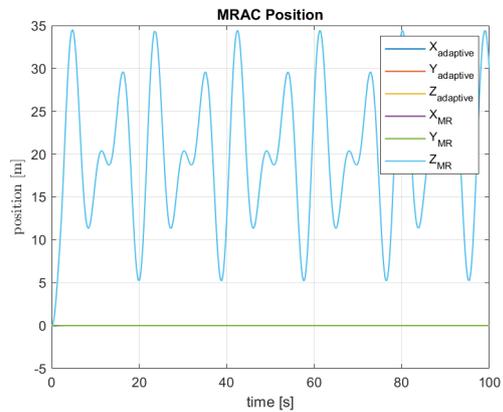
5.8.4 Simulazioni con disturbi parametrici non lineare

Reference Sinusoidale

Viene proposta una simulazione con reference sinusoidale, come mostrato in figura 5.25b. Il disturbo viene valutato in modo parametrico come mostrato nei capitoli precedenti.

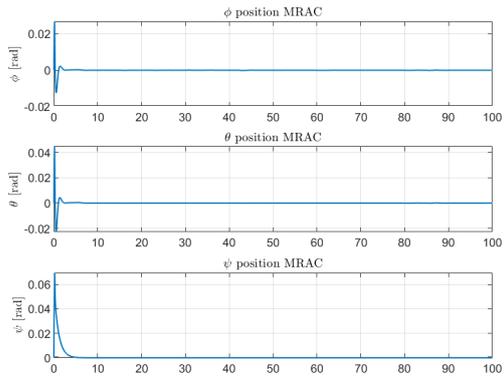


(a) Velocità Lineari

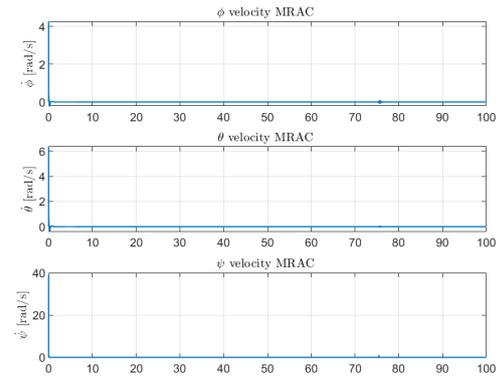


(b) Posizioni Lineari

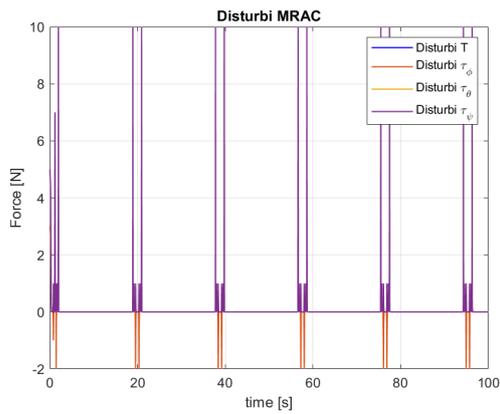
Come si può osservare i disturbi sono ripetitivi e impulsivi, in quanto l'ampiezza della RBF è elevata e la larghezza della banda estremamente ristretta. I disturbi dipendono inoltre dalla quota z , che essendo sinusoidale porterà anche il disturbo a ripetersi. Si nota inoltre come il comando, al termine di un transitorio iniziale, tenda a contrastare puntualmente i disturbi fornendo un comando di intensità uguale ma opposta ad essi.



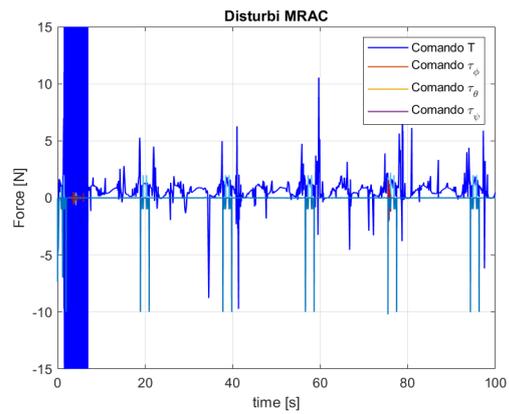
(a) Posizioni Angolari



(b) Velocità Angolari



(a) Disturbi



(b) Comandi

Capitolo 6

Modifiche controllore MRAC

In questo capitolo si espongono le principali modifiche apportate al controllore MRAC per rendere la stima dei parametri adattiva più efficace e la risposta ai comandi più robusta. Si propongono in seguito le seguenti modifiche:

- σ – **mod**, sigma modification per aggiungere damping alla legge adattiva [4];
- **Dead Band**, banda morta per evitare oscillazioni non volute intorno alla posizione comandata [23];
- **Parameter Projection**, proiezione del parametro per rendere la stima adattiva più efficace e limitata nei valori massimi e minimi di guadagno [6];
- **Obs-based con Augmented system**, Osservatore che fa variare il modello di riferimento con sistema stato spazio aumentato [4];
- **C-MRAC**, Composite MRAC varia la legge adattiva rendendola ibrida (diretta-indiretta)[2].

6.1 σ – Modification

L'obiettivo principale della σ -modification è introdurre un termine di "damping" nel meccanismo di adattamento dei parametri. Ciò consente di evitare la crescita incontrollata dei guadagni adattativi (un fenomeno noto come "parameter drift") e di migliorare la stabilità del sistema adattativo. In termini pratici ciò che si ottiene

è la legge adattiva seguente:

$$\begin{cases} \dot{\hat{K}}_x = -\Gamma_x(xe^T PB + \sigma \hat{K}_x) \\ \dot{\hat{K}}_r = -\Gamma_r(r(t)e^T PB + \sigma \hat{K}_r) \\ \dot{\hat{\Theta}} = \Gamma_\Theta(\Phi(x)e^T PB - \sigma \hat{\Theta}) \end{cases} \quad (6.1)$$

La dimostrazione della convergenza della dinamica con tale legge è proposta in appendice.

Si presenano in seguito due esempi che mostrano l'influenza di tale modifica sulla stima dei parametri e sui risultati attesi.

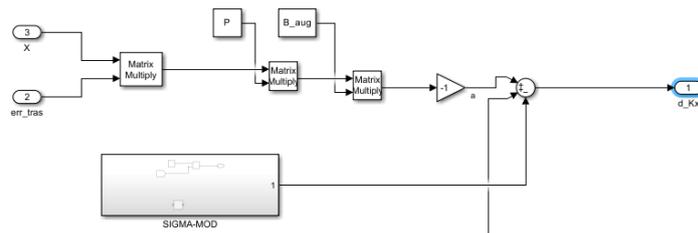


Figura 6.1: Sigma modification

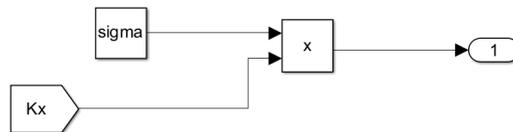
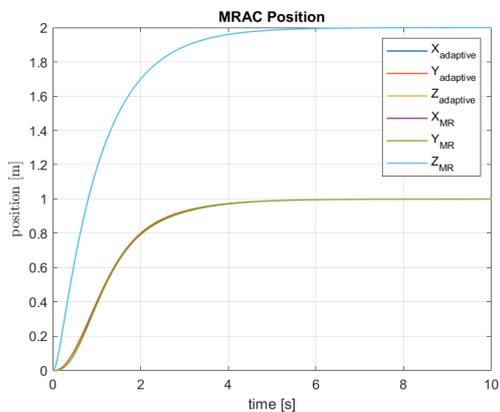


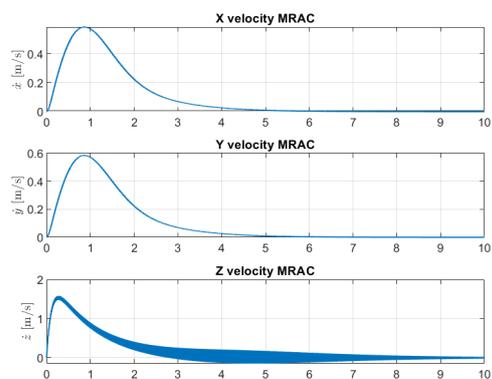
Figura 6.2: Sigma modification

6.1.1 Simulazione 1 - Sigma mod.

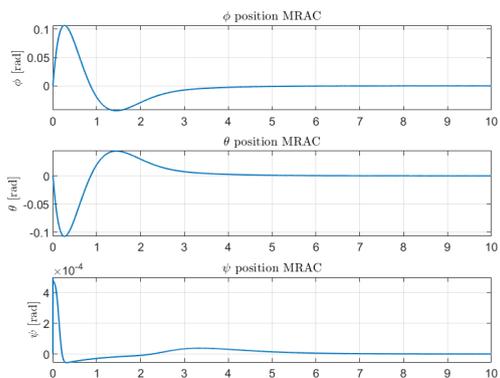
In questa simulazione si sceglie una reference a gradino. Nella simulazione 1 non viene utilizzata la sigma modification mentre nella simulazione 2 si imposta $\sigma=0.1$. Si propongono i commenti al termine della simulazione 2.



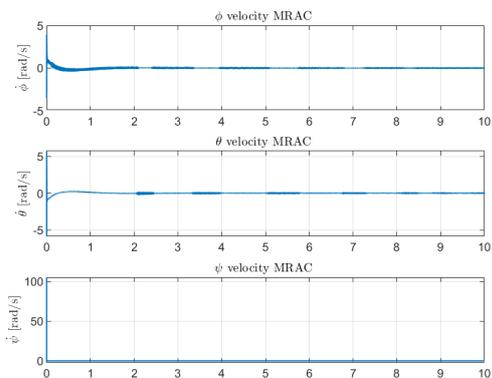
(a) Posizioni lineari



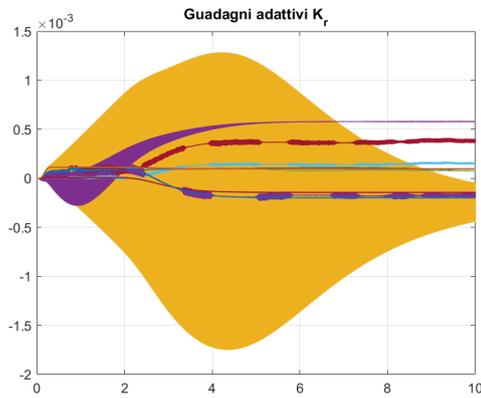
(b) Velocità lineari



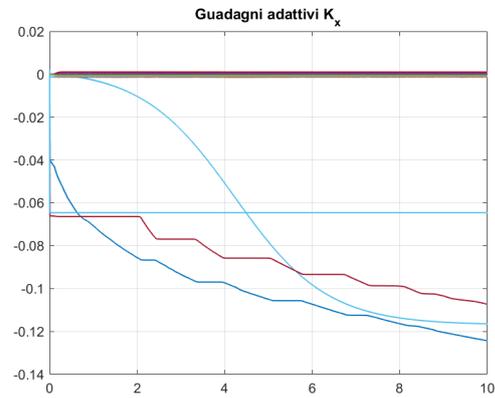
(a) Posizioni angolari



(b) Velocità angolari

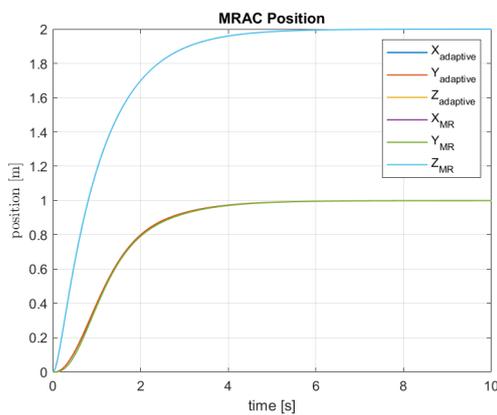


(a) Gaudagno adattivo Kr

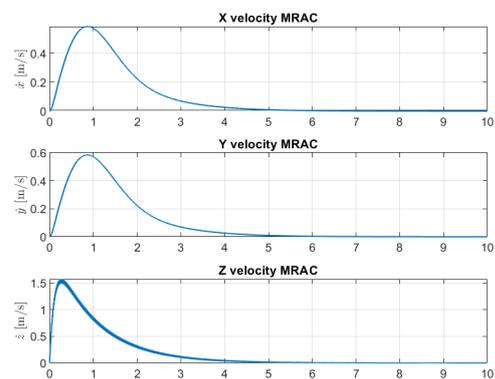


(b) Guadagno adattivo Kx

6.1.2 Simulazione 2 - Sigma mod.



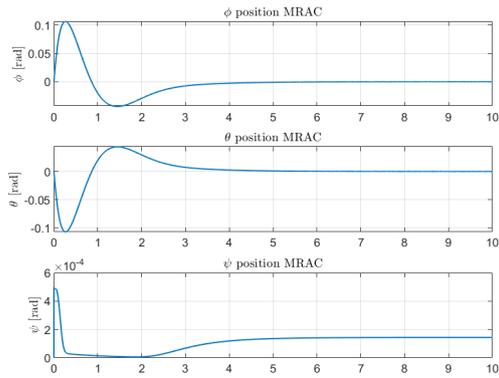
(a) Posizioni lineari



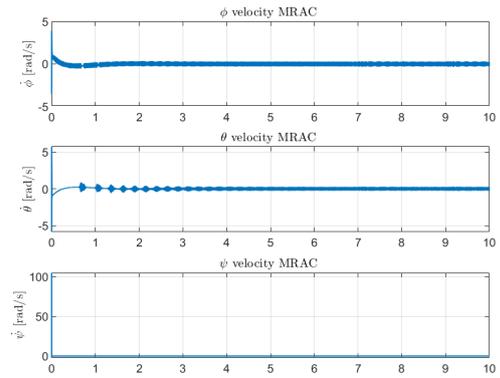
(b) Velocità lineari

Come si può notare dai risultati il termine sigma modification crea un damping ai parametri adattivi. In questo caso si può notare un'adifferenza sostanziale nelle velocità lineari, soprattutto su z, dove nel primo caso le oscillazioni in velocità sono molto superiori a quelle presentate in simulazione 2.

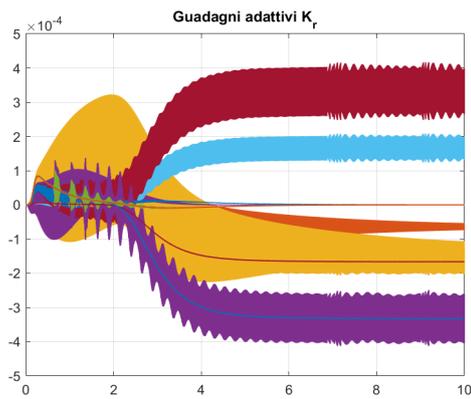
Una seconda influenza importante è quella relativa al calcolo dei parametri adattivi. Nel secondo caso la sigma modification crea damping che porta i parametri K_r e K_x a chiudersi verso un valore piccolo o nullo. Nel primo caso invece alcuni rimangono costanti mentre altri tendono a driftare. La sigma modification si propone però come un valido metodo per ridurre le oscillazioni e limitare il drifting dei parametri.



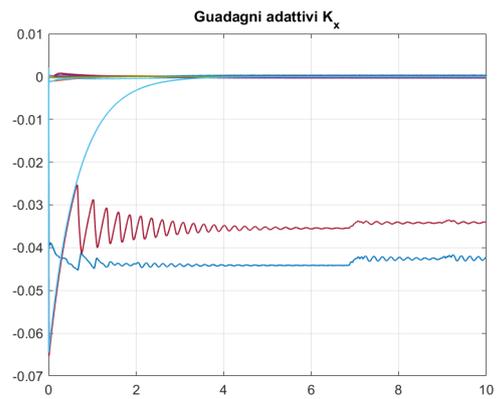
(a) Posizioni angolari



(b) Velocità angolari



(a) Gaudagno adattivo K_r



(b) Gaudagno adattivo K_x

6.2 Dead Band

La dead band o zona morta ha lo scopo di determinare un intervallo nell'intorno dell'errore nullo, per cui la legge adattiva non viene aggiornata, ovvero per cui le derivate dei parametri adattivi nel tempo vengono poste a zero.

Questo approccio ha lo scopo principale di evitare oscillazioni non volute nell'intorno della posizione comandata. Quando l'errore è piccolo infatti, potrebbe accadere che aggiornare la legge di controllo porti i parametri a variare comunque troppo intensamente creando un fenomeno di Chattering intorno alla posizione comandata. Per evitare tale oscillazione si impone la "banda morta" in cui i termini adattivi rimangono costanti fin tanto che l'errore rientra nell'intervallo voluto. La

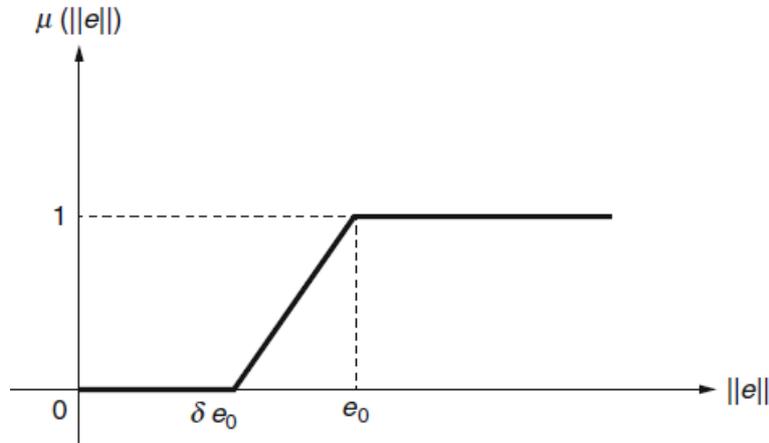


Figura 6.9: Dead Band

formulazione matematica che ne deriva è:

$$\mu(\|e\|) = \max\left(0, \min\left(1, \frac{\|e\| - \delta e_0}{(1 - \delta)e_0}\right)\right) \quad (6.2)$$

Con $0 < \delta \leq 1$ che determina la pendenza della retta mentre e_0 il valore di errore soglia. Nel caso del simulatore in questione si è scelto di forzare la $\delta = 1$ ottenendo uno step.

L'equazione dei parametri adattivi con dead band è perciò:

$$\begin{cases} \hat{K}_x = -\Gamma_x \mu(\|e\|) x e^T P B \\ \hat{K}_r = -\Gamma_r \mu(\|e\|) r(t) e^T P B \\ \hat{\Theta} = \Gamma_\Theta \mu(\|e\|) \Phi(x) e^T P B \end{cases} \quad (6.3)$$

Dove $\mu(\|e\|)$ determina la condizione di "on-off" della legge adattiva

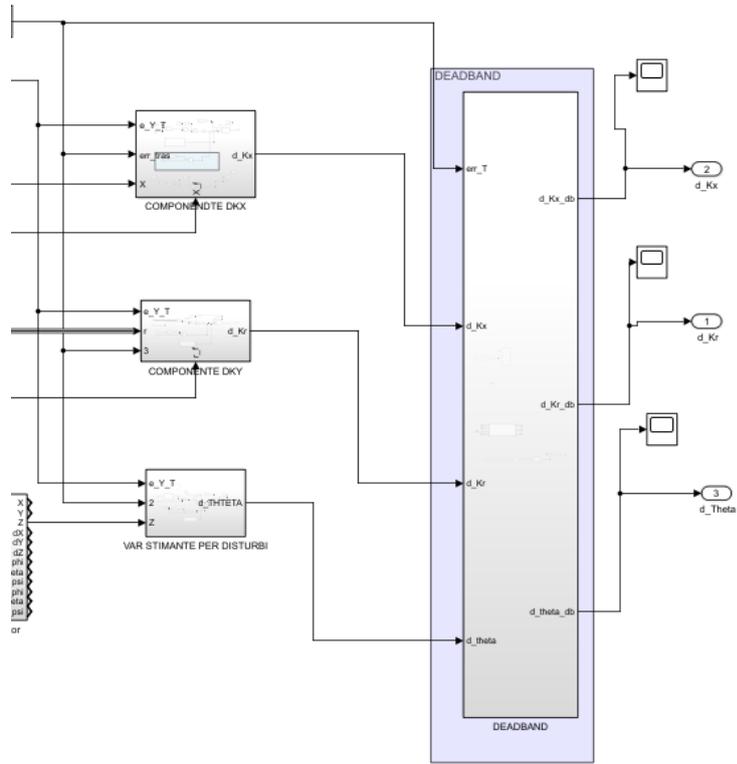


Figura 6.10: Dead band

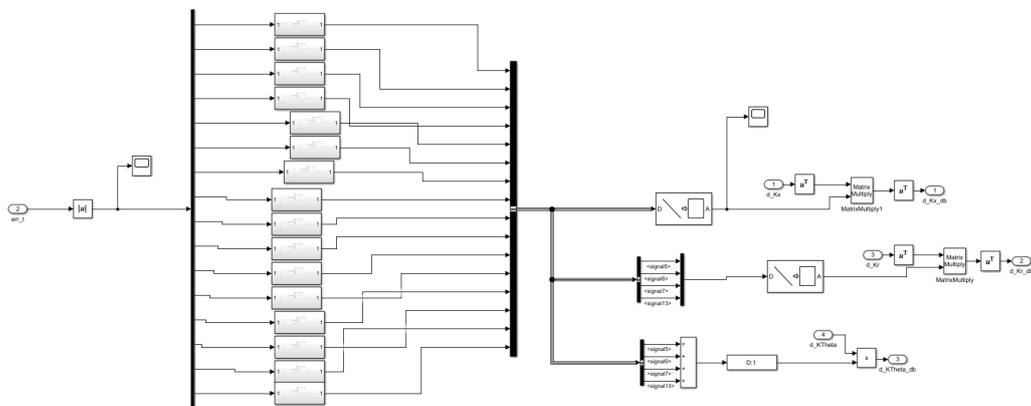
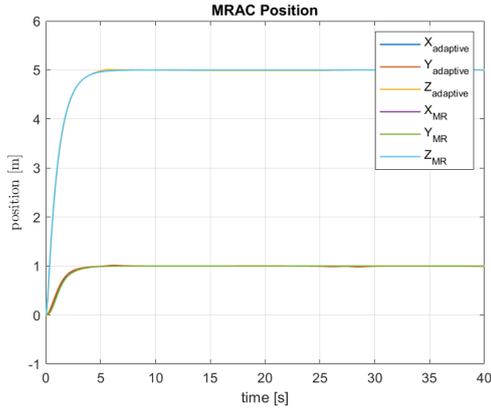
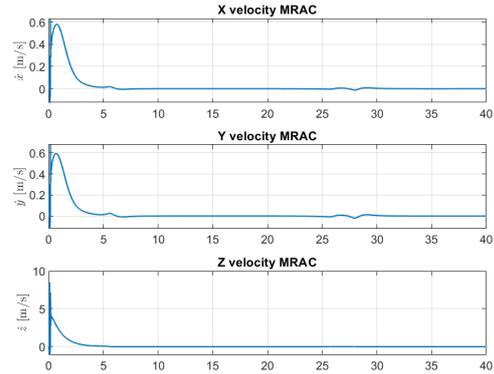


Figura 6.11: Dead band

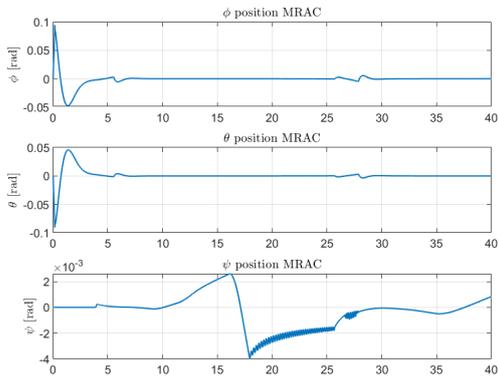
6.2.1 Simulazione 1 - Dead Band



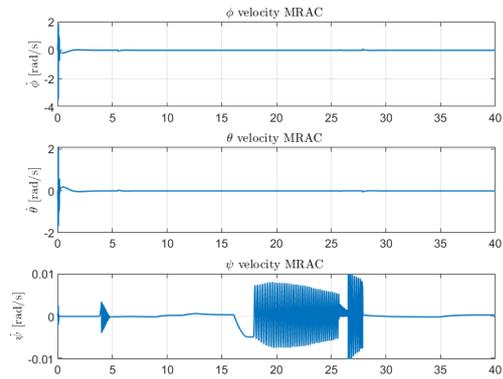
(a) Posizioni lineari



(b) Velocità lineari

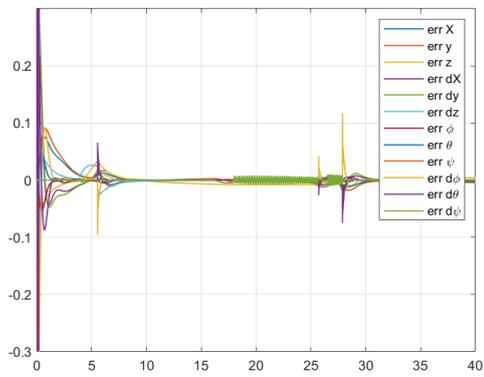


(a) Posizioni angolari

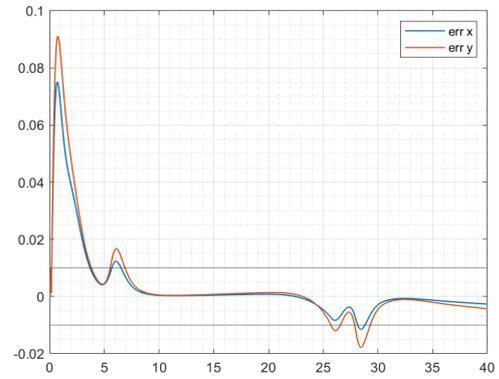


(b) Velocità angolari

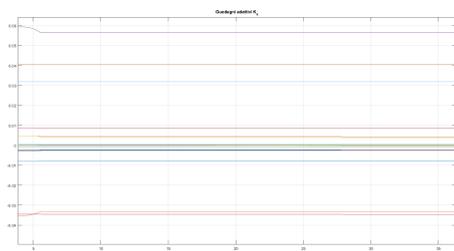
La reference è un gradino costante. Come si può osservare quando le posizioni X e Y superano il valore di dead-band 0.01 metri (6.14b) i parametri K_x e K_r si aggiornano variando leggermente (6.15b, 6.15a). Si notano inoltre oscillazioni indesiderate sulle velocità angolari, in particolare su ψ . Questa dinamica indesiderata è da attribuirsi ad una mancanza di tuning efficace che verrà fatto in seguito proponendo le conclusioni.



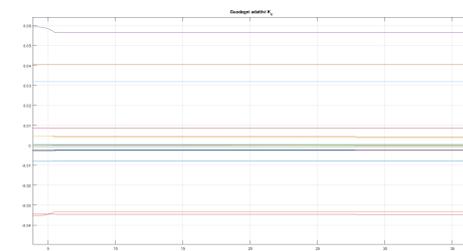
(a) Errore con Modello di Riferimento



(b) Errore rispetto MR posizione x e y



(a) Zoom variazione Kr



(b) Zoom variazione Kx

6.3 Parameter Projection

Il parameter projecton è una modifica alla legge di controllo che permette di ottenere dei parametri limitati. Uno dei problemi dei controllori adattivi è che i guadagni possono divergere ottenendo dei parametri e relativi controlli non fisicamente realizzabili [7]. Per evitare ciò uno dei metodi possibili da adottare è la proiezione del parametro. Si definisce inanzitutto una funzione convessa limitata con smooth boundary del tipo:

$$\Omega_c = \{\theta \in \mathbb{R}^n | f(\theta) \leq c\} \quad ; \quad 0 \leq c \leq 1 \quad (6.4)$$

Con $f(\theta : \mathbb{R}^n \rightarrow \mathbb{R}$ è la funzione convessa dipendente dal generico parametro θ e nella forma:

$$f(\theta) = \frac{(\epsilon_0 + 1)\theta^T \theta - \theta_{max}^2}{\epsilon_0 \theta_{max}^2} \quad (6.5)$$

Si considera θ_{max} come valore massimo del parametro ammissibile e $\epsilon_0 > 0$ la tolleranza di proiezione.

Ora si definisce il termine di proiezione come:

$$\text{Proj}(\Theta, Y, F) = [\text{Proj}(\theta_1, y_1, f_1) \quad \dots \quad \text{Proj}(\theta_m, y_m, f_m)]$$

Con $\Theta = [\theta_1 \dots \theta_m] \in \mathbb{R}^{n \times m}$, $Y = [y_1 \dots y_m] \in \mathbb{R}^{n \times m}$, e $F = [f_1(\theta_1) \dots f_m(\theta_m)]^T \in \mathbb{R}^{m \times 1}$.

$$\text{Proj}(\theta_j, y_j, f_j) = \begin{cases} y_j - \frac{\nabla f_j(\theta_j)(\nabla f_j(\theta_j))^T}{\|\nabla f_j(\theta_j)\|^2} y_j f_j(\theta_j), & \text{if } f_j(\theta_j) > 0 \wedge y_j^T \nabla f_j(\theta_j) > 0, \\ y_j, & \text{altrimenti.} \end{cases}$$

Per $j = 1 \dots m$. Tale espressione indica che qualora il parametro abbia un valore al di fuori del set predisposto e abbia la proiezione che punta verso l'esterno allora viene proiettato verso l'interno andando a sottrarre una componente normale uscente al fine di far ricadere la stima al passo successivo di nuovo all'interno del set. Verosimilmente infatti se il parametro già si trova al di fuori del set e ha come proiezione la direzione uscente, porterà il parametro stimato all'iterazione successiva ad allontanarsi ancora di più dai valori massimi voluti. L'obiettivo è quello di mantenere invece i parametri all'interno del set predisposto ribaltando la proiezione in direzione opposta rispetto la normale uscente. Si propone in figura una rappresentazione grafica di tale operatore.

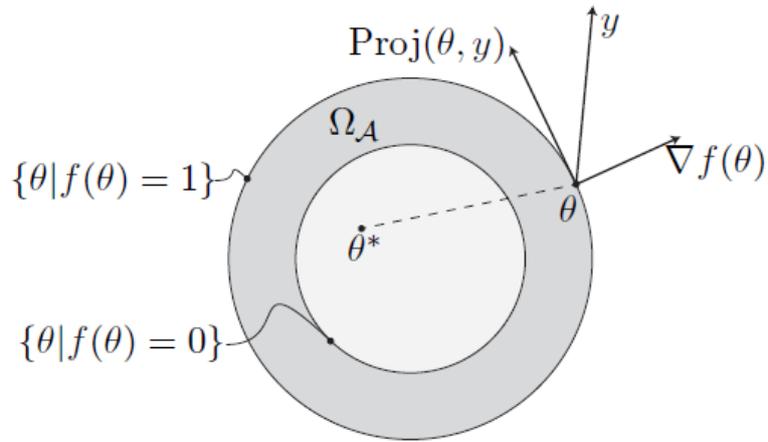


Figura 6.16: Proiezione del Parametro

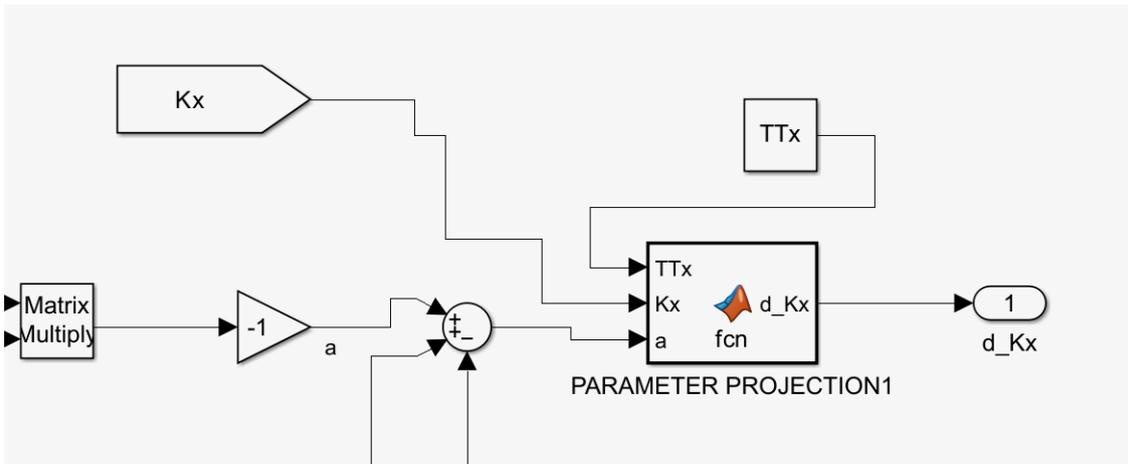
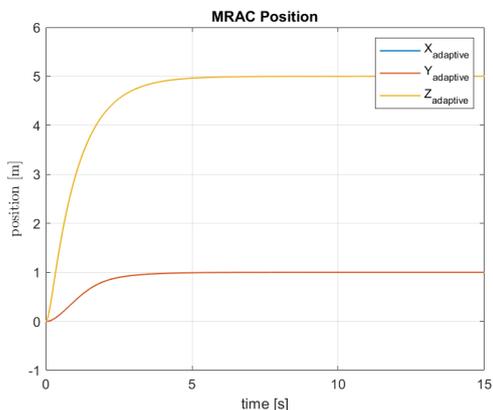


Figura 6.17: Funzione parameter projection

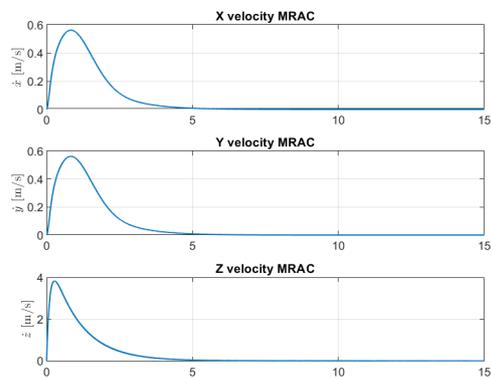
6.3.1 Simulazione 1 - Parameter Projection

In questa simulazione si sceglie una reference costante con z pari a 5 metri e x e y uguale a 1. Nella simulazione 1 la proiezione del parametro è attiva con valore massimo pari a 1 e $\epsilon_0 = 0.1$.

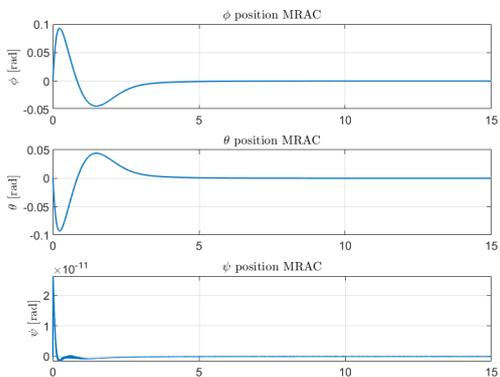
Nella seconda simulazione non viene utilizzata la proiezione mantenendo la medesima reference.



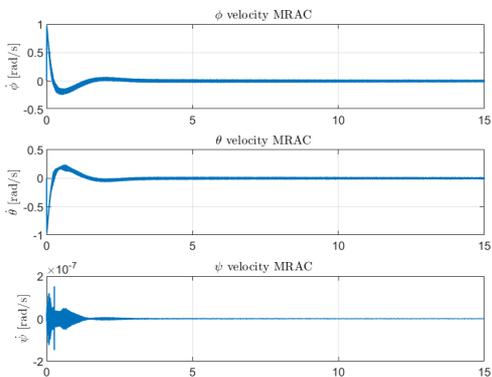
(a) Posizioni lineari



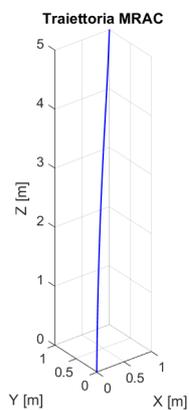
(b) Velocità lineari



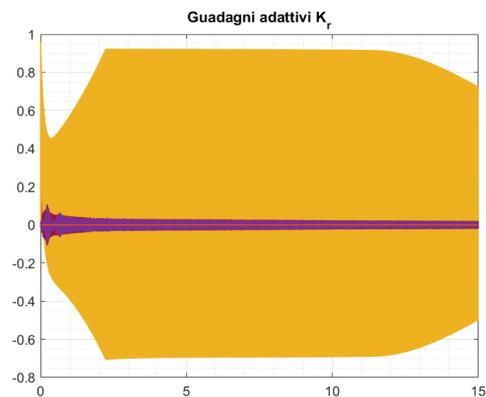
(a) Posizioni angolari



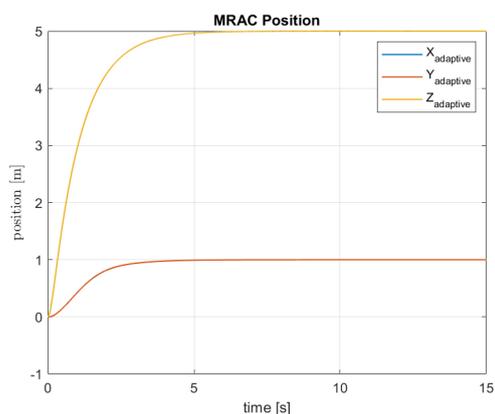
(b) Velocità angolari



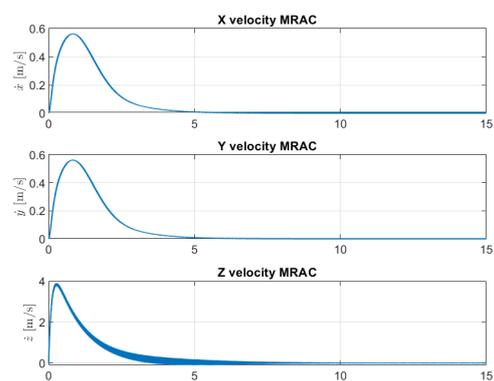
(a) Posizione 3D

(b) Parametro adattivo K_r

6.3.2 Simulazione 2 - Parameter Projection

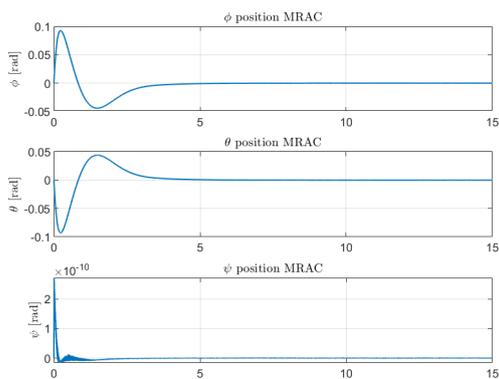


(a) Posizioni lineari

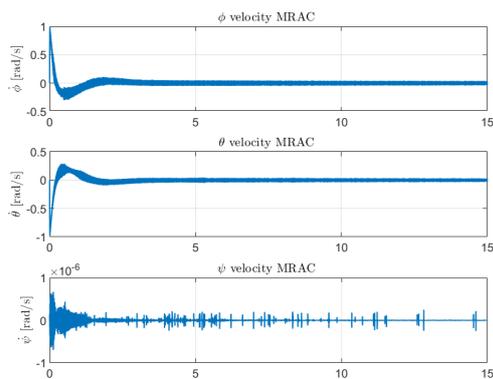


(b) Velocità lineari

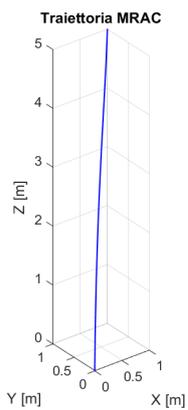
Come si può osservare vi è una restrizione su K_r che ha come valore massimo impostato 1 con $\epsilon_0 = 0.1$. Quando il valore supera il valore $1-0.1=0.9$ il parametro esce dal set. Qualora la proiezione sia anche rivolta verso l'esterno del cerchio si attiva la proiezione che infatti impedisce ai parametri di allontanarsi dal valore 0.9.



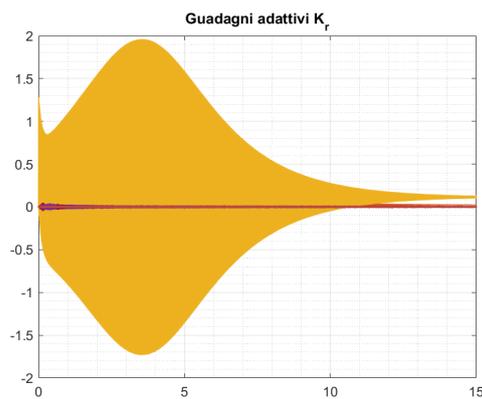
(a) Posizioni angolari



(b) Velocità angolari



(a) Posizione 3D



(b) Parametro adattivo K_r

6.4 C-MRAC (Composite MRAC)

L'architettura Composite ha lo scopo di introdurre un nuovo termine nella legge adattiva dipendente da un termine di previsione in cui il parametro viene stimato a fronte della dinamica filtrata. L'obiettivo del Composite è quello di ottenere a tutti gli effetti un'architettura ibrida in cui il parametro adattivo viene stimato anche in maniera indiretta [1]. (si fa riferimento alla sezione introduttiva sui metodi di controllo diretti e indiretti per maggior chiarezza).

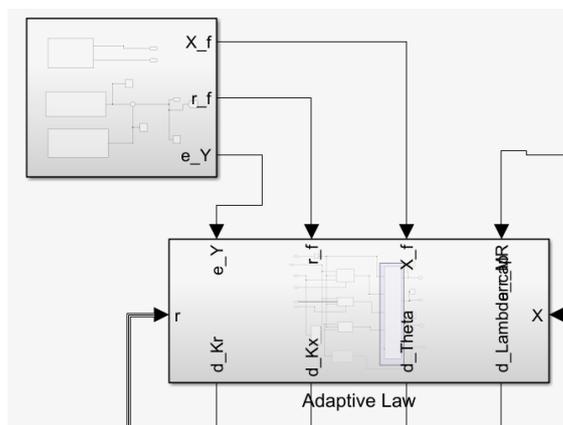


Figura 6.24: C-MRAC

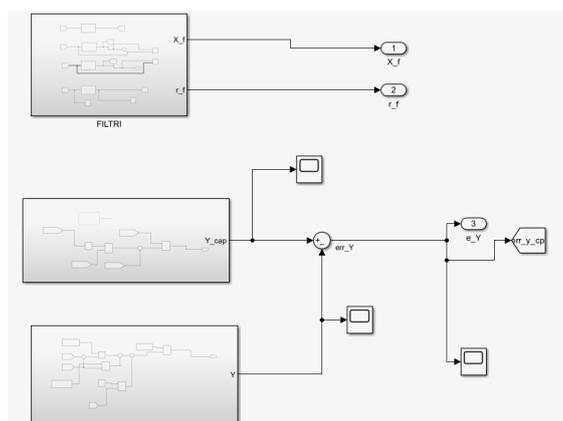


Figura 6.25: C-MRAC

Come accennato precedentemente la dinamica viene prima filtrata con un filtro passa basso evitando l'influenza dei disturbi in alta frequenza. Dopodichè si sfruttano gli stati filtrati per ricavare una stima del contributo Y e \hat{Y} il cui errore determinerà il contributo aggiuntivo alla legge adattiva.

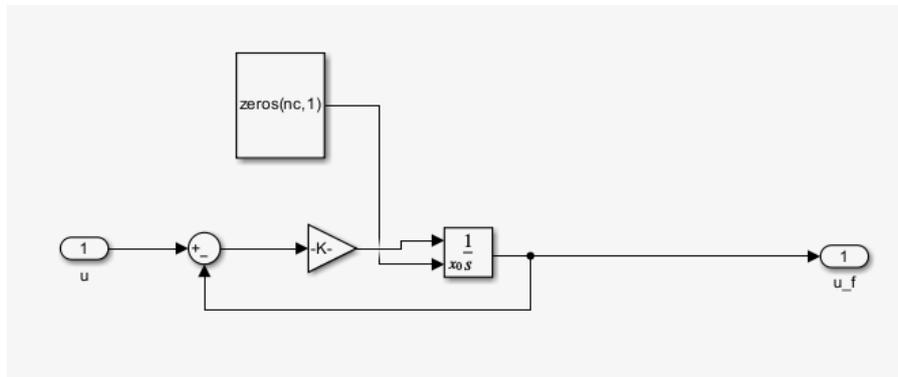


Figura 6.26: Filtro

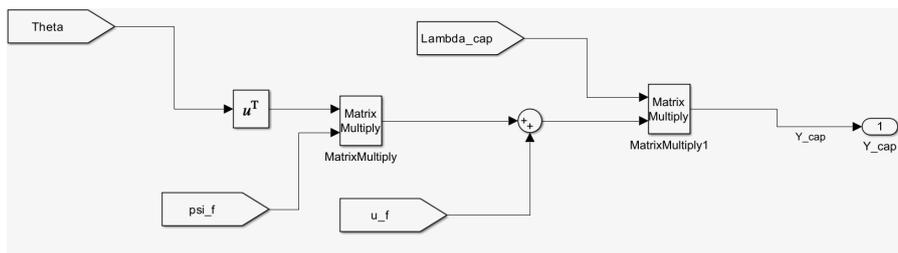


Figura 6.27: Y cappello

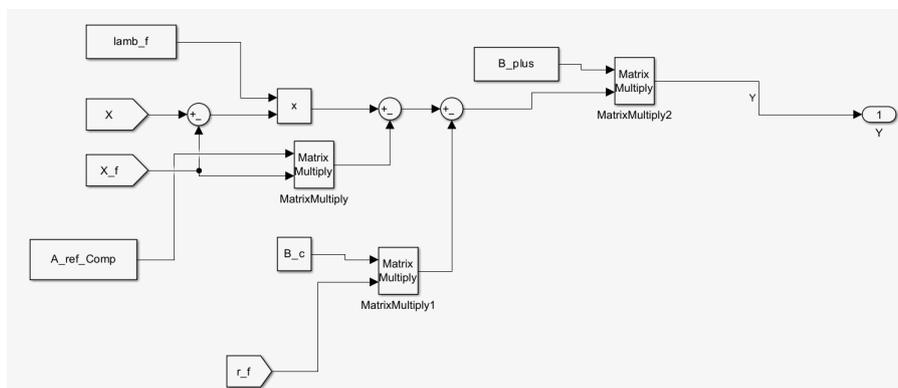


Figura 6.28: Y

In particolare la dinamica filtrata assume la forma:

$$\dot{x}_f = \lambda_f(x - x_f) \quad (6.6)$$

Con $\lambda_f > 0$ l'inverso della costante del filtro. Sostituendo tale espressione nell'equazione della dinamica generale 5.42 si ottiene:

$$\lambda_f(x - x_f) - A_{ref}x_f - B_{ref}r_f = B[u_f + \Theta^T \Psi_f] \quad (6.7)$$

Con u_f vettore dei comandi filtrati contenente i contributi adattivi. Raccogliendo opportunamente è possibile ottenere:

$$\underbrace{\underbrace{(B^T B)^{-1} B^T}_{B^+} (\lambda_f(x - x_f) - A_{ref} x_f - B_{ref} r_f)}_Y = u_f + \Theta^T \Psi_f \quad (6.8)$$

Il termine Y rappresenta la risposta al comando calcolata online e può essere valutato ad ogni istante di tempo con:

$$Y(t) = B^+ (\dot{x}_f(t) - A_{ref} x_f(t) - B_{ref} r_f(t)) \quad (6.9)$$

Il termine stimato $\hat{Y}(t)$ può invece essere valutato come:

$$\hat{Y} = u_f - \hat{K}_x^T x_f - \hat{K}_r^T r_f + \hat{\Theta}^T \Phi_f \quad (6.10)$$

Da queste due espressioni è possibile calcolare l'errore:

$$e_Y = \hat{Y} - Y \quad (6.11)$$

Da cui si ricava la componente adattiva del Composite come:

$$\hat{K}_{x_{comp}} = x_f \gamma_c e_Y^T \quad (6.12)$$

$$\hat{K}_{r_{comp}} = r_f \gamma_c e_Y^T \quad (6.13)$$

$$\hat{\Theta}_{comp} = \Phi_f \gamma_c e_Y^T \quad (6.14)$$

La legge adattiva Composite completa diventa perciò:

$$\begin{cases} \hat{K}_x = -\Gamma_x (x e^T P B - x_f \gamma_c e_Y^T) \\ \hat{K}_r = -\Gamma_r (r(t) e^T P B - r_f \gamma_c e_Y^T) \\ \hat{\Theta} = \Gamma_\Theta (\Phi(x) e^T P B - \Phi_f \gamma_c e_Y^T) \end{cases} \quad (6.15)$$

Tale legge adattiva [2] non viene implementata in quanto i risultati non forniscono un contributo positivo per il controllo di questo quadricotore.

6.5 Osservatore su Model Reference

L'osservatore sul Model Reference ha la funzione di rendere variabile il modello di riferimento a seconda del contributo d'errore tra il vettore di stato e il vettore di stato di riferimento. Per fare ciò si utilizza il modello di Luenberger che permette di ottenere il termine di modifica voluto. Per far ciò è però prima necessario introdurre la dinamica aumentata.

6.5.1 Sistema Aumentato

Il sistema aumentato viene così definito perchè si aumenta il vettore di stato fin ora utilizzato di 12 elementi con 4 termini che rappresentano l'errore integrato rispetto la reference. In particolare si ha che:

$$\begin{cases} \dot{e}_{y_{I_x}} = x - r_x \\ \dot{e}_{y_{I_y}} = y - r_y \\ \dot{e}_{y_{I_z}} = z - r_z \\ \dot{e}_{y_{I_\psi}} = \psi - r_\psi \end{cases} \quad (6.16)$$

Ovvero l'errore integrale derivato coincide con l'errore stesso. Questa formulazione permette di ottenere il nuovo vettore di stato come :

$$\mathbf{X} = [e_{y_{I_x}}, e_{y_{I_y}}, e_{y_{I_z}}, e_{y_{I_\psi}}, x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T \quad (6.17)$$

Così facendo si ottengono le matrici aumentate:

$$A_{aug} = \begin{bmatrix} A_1 & C \\ A_2 & A \end{bmatrix} ; \quad B_{aug} = \begin{bmatrix} B_1 \\ B \end{bmatrix} ; \quad B_c = \begin{bmatrix} B_2 \\ B_3 \end{bmatrix} ; \quad C_{aug} = \begin{bmatrix} C_1 & C \end{bmatrix} \quad (6.18)$$

Con le matrici A, B, C già espresse in 5.38 mentre le altre definite come:

$$A_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} ; \quad A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \quad (6.19)$$

$$B_2 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} ; \quad B_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \quad (6.20)$$

$$C_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (6.21)$$

Il nuovo sistema Stato Spazio aumentato diventa perciò:

$$\dot{X} = A_{aug}X + B_{aug}u + B_c r \quad (6.22)$$

Con r vettore di reference. Tale rappresentazione è utilizzata per il modello di riferimento Open Loop a cui a seguire si somma il contributo dell'osservatore.

6.5.2 Model Reference con osservatore

Il contributo dell'osservatore è calcolato come:

$$L_v(x - x_{ref}) \quad (6.23)$$

Ottenendo l'espressione completa per il modello di riferimento:

$$\dot{X}_{ref} = A_{aug}X_{ref} + B_{aug}u_{lqr} + B_c r + L_v(X - X_{ref}) \quad (6.24)$$

Si evidenzia come L_v sia il guadagno di retroazione dell'errore e viene calcolata come :

$$L_v = P_v R_v^{-1} \quad ; \quad P_v A_{ref}^T + A_{ref} P_v - P_v R_v^{-1} P_v + Q_v = 0 \quad (6.25)$$

Con P_v soluzione dell'equazione di cui sopra e le matrici di riferimento ottenute come già nei capitoli precedenti:

$$A_{ref} = A_{aug} - B_{aug} K_{LQR} \quad ; \quad B_{ref} = B_{aug} \cdot K_{LQR} \quad (6.26)$$

Matrici del sistema di close loop. Le matrici Q_v, R_v vengono definite non come le classiche matrici Q ed R del controllo LRQ bensì come:

$$Q_v = Q_0 + \left(\frac{\nu + 1}{\nu} \right) \mathbb{I}_{16 \times 16} \quad ; \quad R_v = \frac{\nu}{\nu + 1} \mathbb{I}_{4 \times 4} \quad (6.27)$$

con Q_0, ν i due termini da tarare. in generale piccoli valori di ν permettono di ottenere transitori minori. In questo modo si ottiene un osservatore sul model reference che farà variare il modello di riferimento anche rispetto ai disturbi, che porteranno il termine di errore x_{ref} ad essere diverso da 0. Scegliendo tale architettura la matrice P_v diventa la nuova patrice P che determinerà anche i termini adattivi nelle leggi precedentemente esposte.

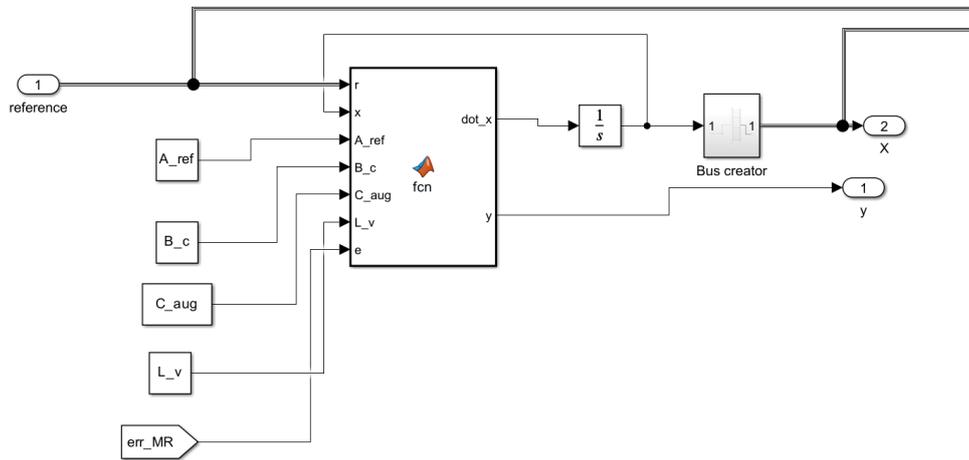
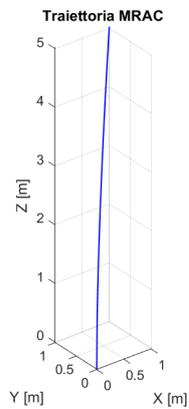


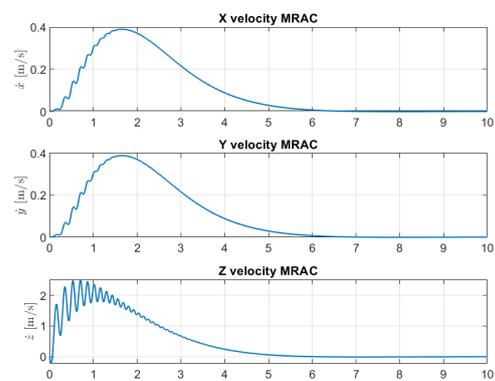
Figura 6.29: Osservatore su Model Reference

6.5.3 Simulazione 1 - Osservatore

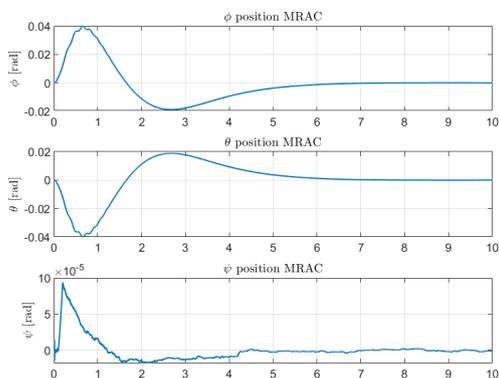
La simulazione 1 è con reference costante senza osservatore. La seconda simulazione viene effettuata con osservatore e si ricavano i confronti.



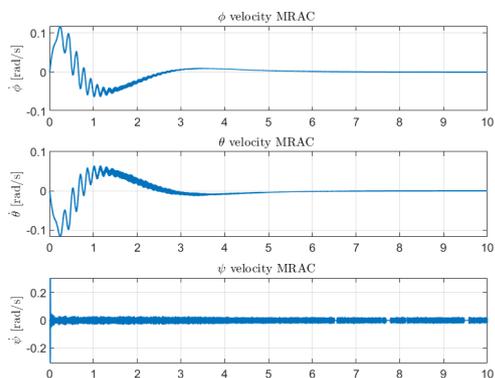
(a) Posizione 3D



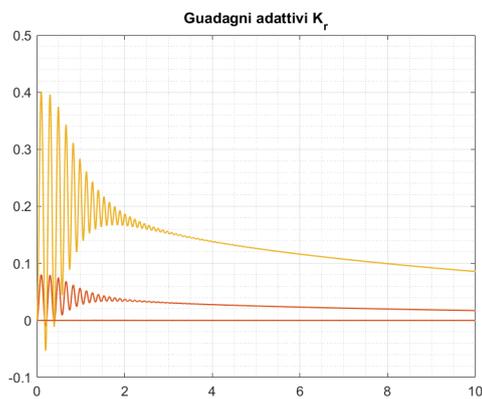
(b) Velocità lineari



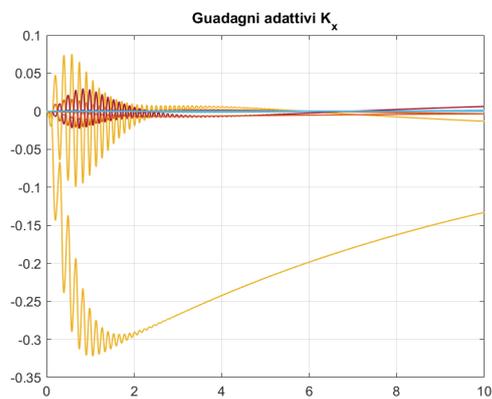
(a) Posizioni angolari



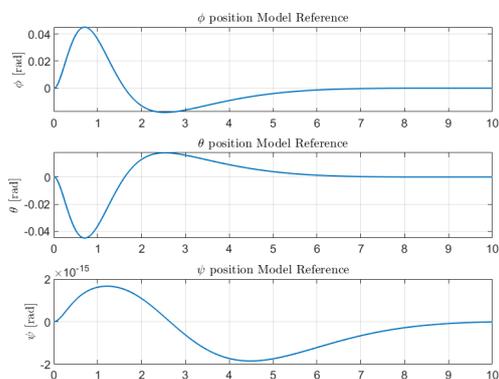
(b) Velocità angolari



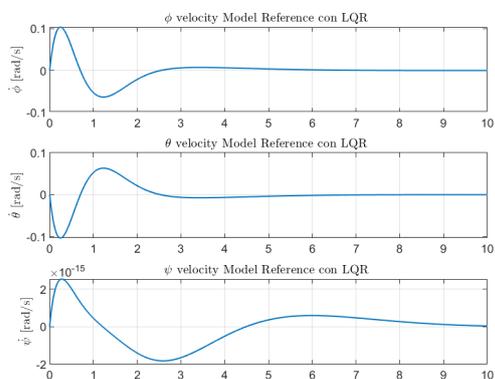
(a) Parametro adattivo K_r



(b) Parametro adattivo K_x

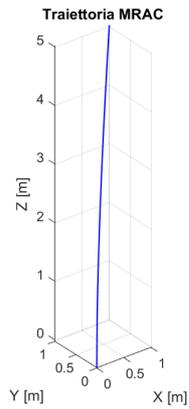


(a) Posizione angolare modello di riferimento

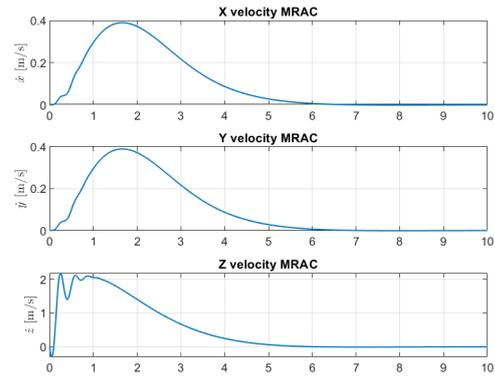


(b) Velocità angolare modello di riferimento

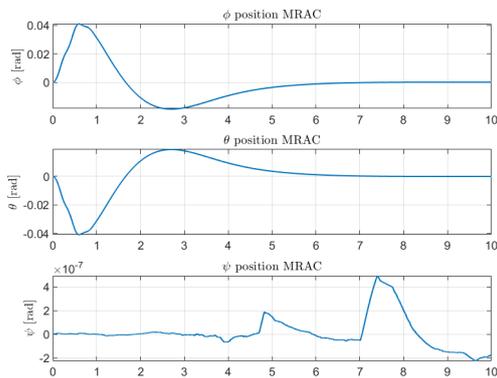
6.5.4 Simulazione 2 - Osservatore



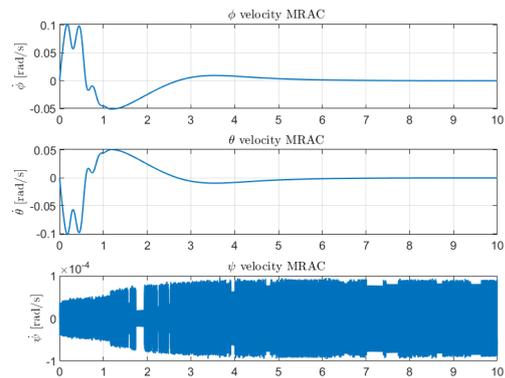
(a) Posizione 3D



(b) Velocità lineari

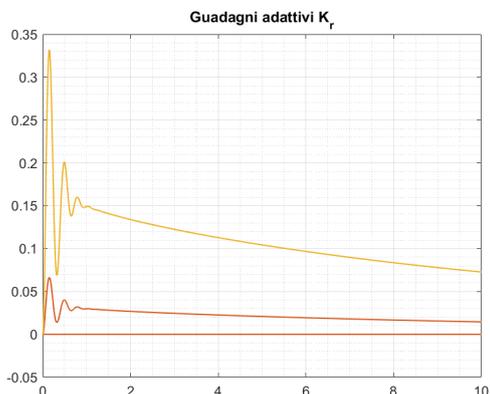


(a) Posizioni angolari

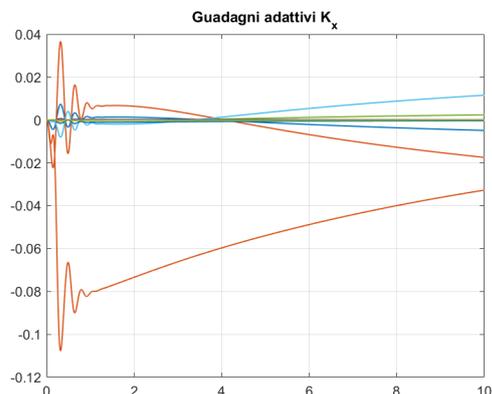


(b) Velocità angolari

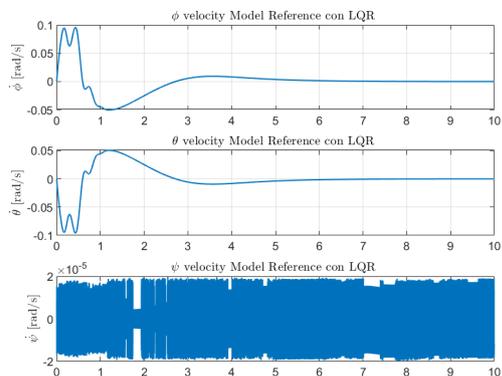
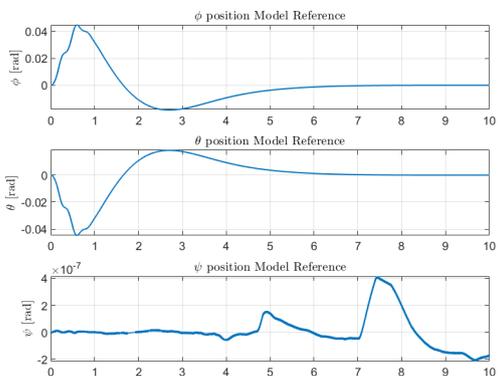
Come si può osservare vi è una differenza sostanziale tra modello di riferimento nelle due simulazioni. Con osservatore il MR cambia in accordo con l'errore diminuendo l'intensità di oscillazione del vettore di stato reale.



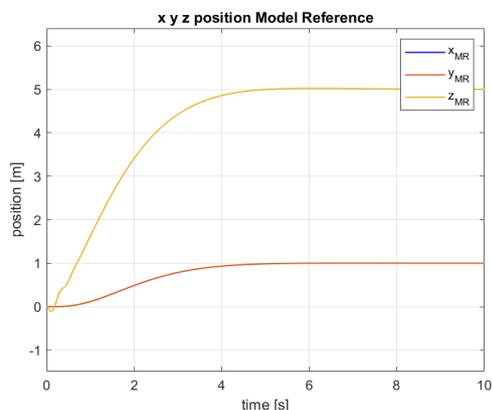
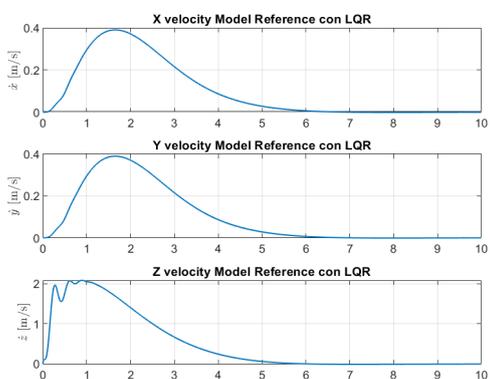
(a) Parametro adattivo K_r



(b) Parametro adattivo K_x



(a) Posizione angolare modello di riferimento (b) Velocità angolare modello di riferimento



(a) Velocità lineari modello di riferimento (b) Posizione lineare modello di riferimento

Capitolo 7

Pacchetto Parrot Minidrone Simulink

In questa sezione si vogliono presentare le principali caratteristiche del progetto Parrot minidrone implementato in Simulink. Tale modello, essendo ad alta fedeltà, viene utilizzato al posto del plant per simulare una dinamica più realistica del drone rispetto le semplici equazioni non lineari utilizzate in precedenza.

In questa sezione non si vogliono affrontare puntualmente tutte le caratteristiche del modello ma una visione d'insieme al fine di presentare i blocchi utilizzati per le simulazioni ad alta fedeltà.

Per avviare il progetto Parrot Minidrones è necessario installare alcuni add-on di Matlab come:

- **Aerospace Blockset**, utilizzato per modellare e simulare sistemi aerospaziali. In generale fornisce blocchi per modellare sistema di propulsione, veicoli spaziali e aerospaziali in diverse condizioni ambientali.
- **Signal Processing Toolbox**, per l'analisi dei segnali.
- **Simulink Control Design**, per modellare sistemi lineari e non lineari
- **Simulink 3D animation** , per visualizzare le simulazioni in 3D fornendo un ambiente grafico versatile e modificabile a proprio piacimento secondo le opportune necessità.
- **Simulink Hardware Support Package for Parrot Minidrones** è il firmware che contiene tutte le applicazioni necessarie per interagire fisicamente con il Parrot Mambo. Fornisce la possibilità di un collegamento bluetooth diretta con Matlab tramite algoritmo integrato. Da qui è possibile modificare interattivamente tutte le componenti del drone al fine di sperimentazione ottenendo una piattaforma unica per il deploy di tali modifiche in tempo reale.

7.1 Descrizione Modello

Per avviare il progetto è sufficiente digitare nella command window di Matlab il comando **parrotMinidroneCompetitionStart**. Così facendo si avvia il progetto che al termine della sua apertura mostra tre elementi:

- Modello ad albero con tutte le variabili necessarie caricate nel workspace di Matlab;
- Modello Simulink del Drone Parrot Mambo;
- Una finestra di Simulink 3D animation per la visualizzazione della dinamica in 3D del drone in ambiente grafico.

Il modello Simulink è diviso in 4 principali sottogruppi:

- **Flight command**, contiene le tipologie di comando con cui controllare il drone;
- **FCS-Flight Control System**, contiene il controllore di volo;
- **Simulation Model**, formato dal modello del drone, dell'ambiente e della sensoristica;
- **Flight Simulation**, fornisce i blocchi necessari alla visualizzazione 3D della dinamica.

Al fine del progetto presentato in questa tesi i blocchi utilizzati sono quelli della dinamica del drone, dell'ambiente e della simulazione in 3D. La parte di sensoristica viene tralasciata mentre la parte di controllo viene implementata tramite blocco MRAC sviluppato come nei capitoli precedenti. In questo modo si ha l'obiettivo di verificare la bontà del controllore custom rispetto al modello alta fedeltà già implementato nel progetto Simulink.

Si procede in seguito a mostrare le principali caratteristiche di ogni blocco, facendo dei brevi accenni per le componenti non utilizzate in seguito perchè sostituite dal controllore MRAC.

7.1.1 Simulation Model - Modello del drone

Nel blocco modello del drone è possibile visualizzare sia il modello lineare che quello non lineare. Nel caso discusso la seconda scelta è quella ottimale in quanto si vuole realizzare una simulazione ad alta fedeltà che rispecchi il più possibile le condizioni reali (7.1, 7.2).

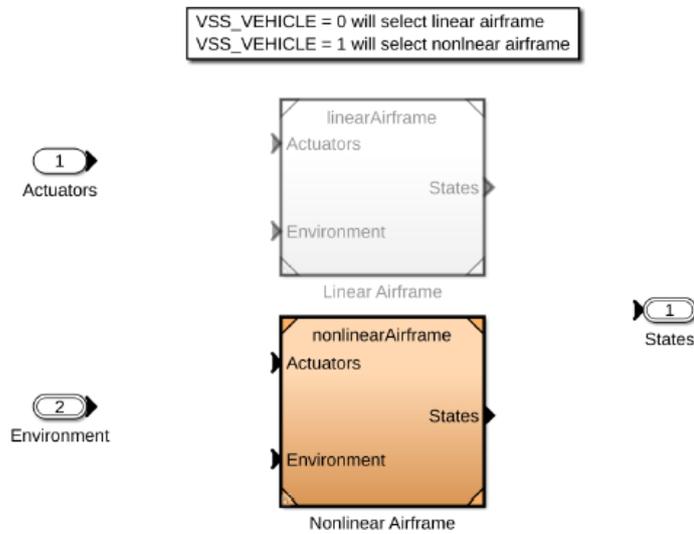


Figura 7.1: Modello Lineare - Non Lineare

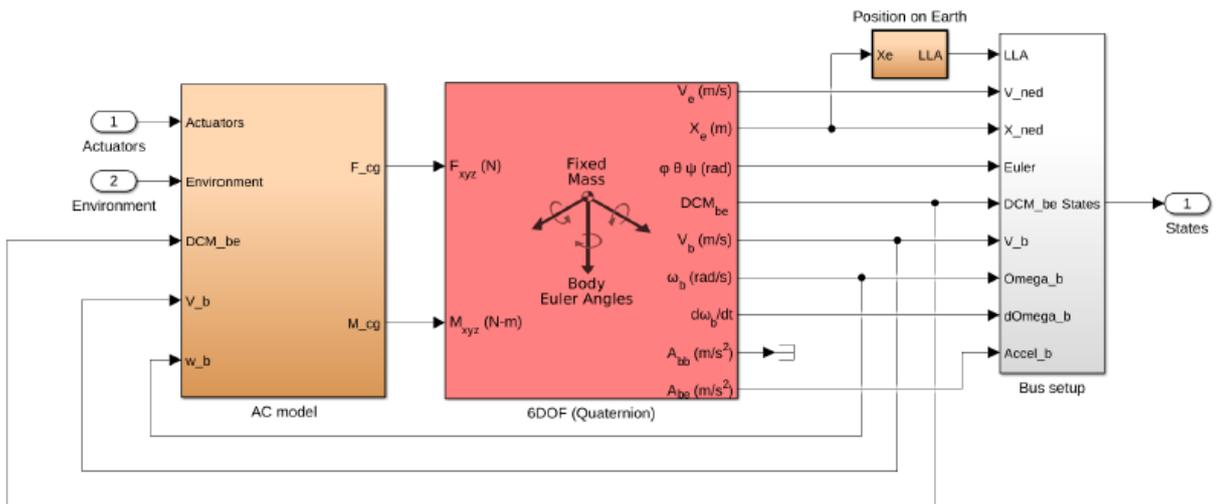


Figura 7.2: Modello non lineare

Il **modello non lineare** è composto dai blocchi:

- AC model, computa la sommatoria delle forze agenti sul drone;
- 6DOF, calcola la risposta alle forze calcolate precedentemente
- Position on Earth, calcola la posizione in termini di latitudine, longitudine e quota sulla terra;
- Bus Setup, per creare il vettore degli stati.

6DOF

Il blocco 6DOF è presente nell'Aerospace Blockset Toolbox di MATLAB e permette di calcolare lo stato complessivo del drone in termini di posizioni velocità e accelerazioni fornendo i valori seguenti:

- \mathbf{X}_e : posizione nello spazio rispetto all'*inertial frame*.
- \mathbf{V}_e : velocità traslazionale rispetto all'*inertial frame* (anche detto *Earth Fixed Frame*).
- \mathbf{DCM}_{b_e} : coefficienti della *Direct Cosine Matrix*, matrice per il cambio di coordinate da inerziali a body.
- ϕ, θ, ψ : angoli di Eulero.
- \mathbf{V}_b : velocità traslazionale rispetto al riferimento body.
- $\boldsymbol{\omega}_b$: velocità angolare rispetto al riferimento body.
- $\frac{d\boldsymbol{\omega}_b}{dt}$: accelerazione angolare rispetto al riferimento body.
- \mathbf{A}_{bb} : accelerazione traslazionale lungo gli assi body rispetto il medesimo sistema di riferimento.
- \mathbf{A}_{be} : accelerazione traslazionale lungo gli assi body rispetto all'inerziale.

Il blocco 6DOF prendendo in input tutte le forze agenti sul drone permette di valutare il vettore di stato in output che viene formato come bus nel blocco Bus Setup.

AC Model

Il blocco AC Model ha lo scopo di valutare le forze agenti sul drone da dare in input al blocco 6DOF. In particolare le forze da valutare sono:

- Forza di gravità;
- Forza di resistenza aerodinamica;
- Forze e momenti agenti sul drone esercitate dai rotori
- Forze di disturbo, aggiunte successivamente, non presenti nel progetto originale.

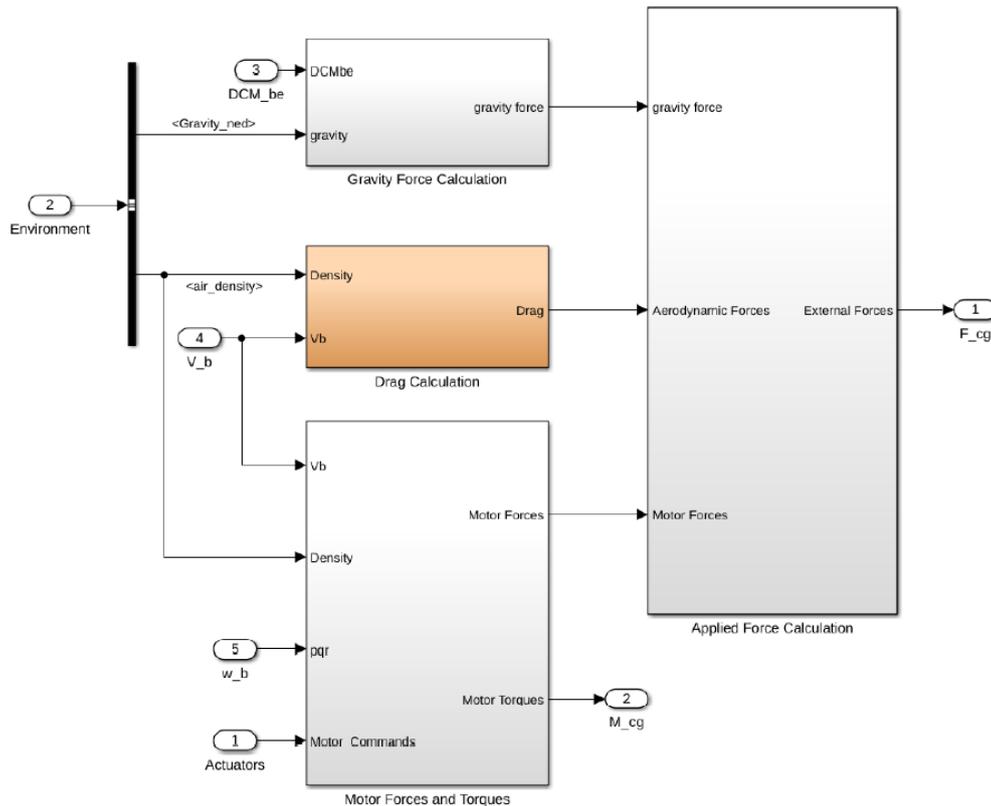


Figura 7.3: Blocco AC model

Forza di Gravità è il blocco che calcola le 3 componenti della forza di gravità rispetto body frame prendendo in ingresso la matrice di rotazione DCM per la conversione da inertial frame a body frame e la costante di gravità g . Moltiplicando la costante di gravità per la massa si ottiene la forza peso, che moltiplicata a sua volta per la matrice di rotazione fornisce le componenti in assi body desiderate (7.4).

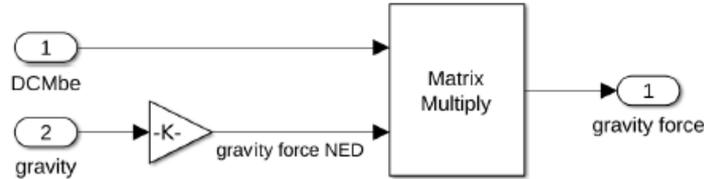


Figura 7.4: Blocco forza di gravità

Resistenza Aerodinamica è il blocco che calcola le componenti di resistenza dovuta all'interazione con l'aria e che si oppone al moto del drone (7.5). L'equazione generale che governa tale fenomeno dipende dalla pressione dinamica e dalle caratteristiche geometriche del drone nel modo seguente:

$$\mathbf{F}_D = \frac{1}{2}\rho\mathbf{V}_b^2SC_d \quad (7.1)$$

Con $\frac{1}{2}\rho\mathbf{V}_b^2$ pressione dinamica, \mathbf{V}_b vettore delle velocità in assi body, S superficie esposta perpendicolare al vento e C_d coefficiente di resistenza aerodinamica del drone. Nel caso considerato si valuta in modo estremamente conservativo la superficie esposta come quella di un cerchio pieno ottenuta considerando l'intero diametro trasversale del drone. Essendo la forza di resistenza in direzione opposta rispetto alla direzione del moto, si moltiplica il segno di V_b per -1 . Il blocco in rosso (7.6) rappresenta il calcolo della pressione dinamica come espresso sopra.

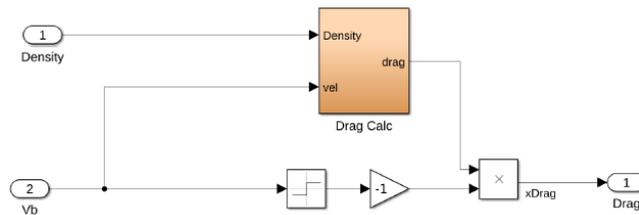


Figura 7.5: Blocco Resistenza Aerodinamica

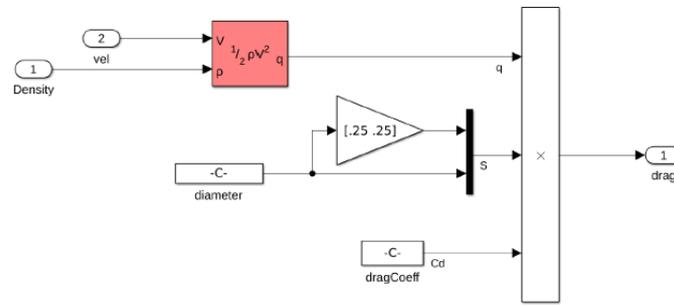


Figura 7.6: Blocco resistenza aerodinamica 2

Motor Forces and Torques è il blocco che calcola tutte le forze applicate al corpo del drone rispetto al riferimento body. Il blocco MotorsToW (7.7) è finalizzato

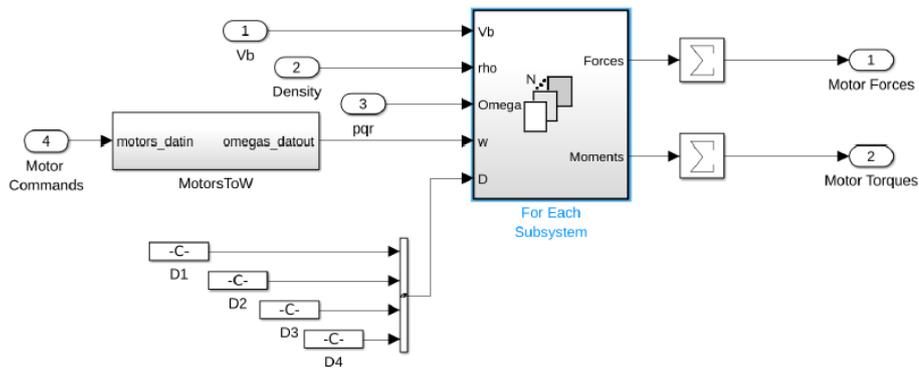


Figura 7.7: Forze e Momenti motore

alla conversione del controllo in un segnale che indichi la velocità angolare che i motori devono raggiungere. Tale conversione è implementata attraverso 3 guadagni, un saturatore ed il calcolo di una radice quadrata (7.8).

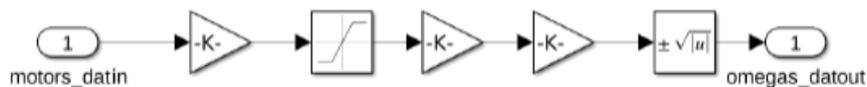


Figura 7.8: Blocco MotorsToW

L'uscita entra in input al blocco For Each Subsystem, assieme alla velocità lineare V_b del drone, alla densità dell'aria, alle velocità angolari p , q e r e ai vettori D_1, D_2, D_3, D_4 rappresentanti ognuno la distanza tra i rotori e il baricentro del drone.

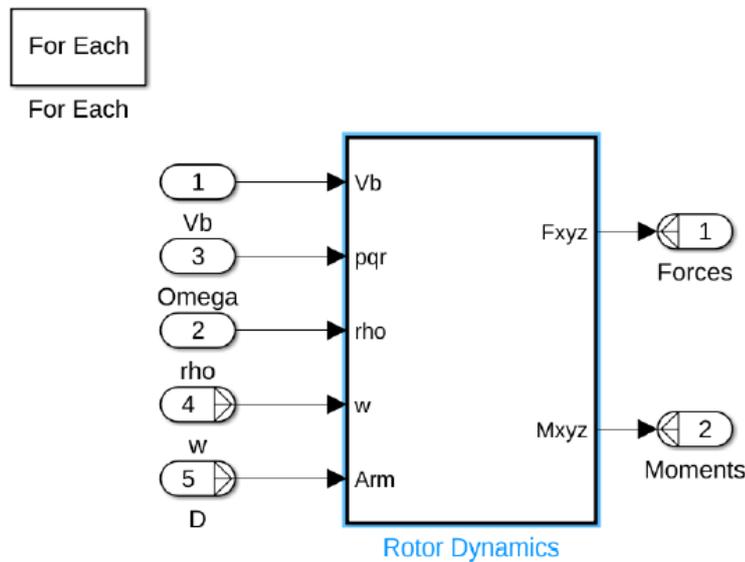


Figura 7.9: Blocco For Each Subsystem

Il blocco For Each Subsystem crea un ciclo 'for each' con cui si calcolano tutte le forze e momenti esercitate da ciascun motore del drone attraverso l'algoritmo rappresentato in Appendice.

Bus setup è il blocco che, date in ingresso tutte le componenti che caratterizzano lo stato del sistema, raggruppa quest'ultime all'interno di un unico bus tale da fornire in feedback al controllore lo stato del drone. Nel caso specifico MRAC tale vettore viene selezionato opportunamente aggiungendo i termini di errore per ottenere il modello Augmented.

Position on Earth è il blocco che, date la posizione X_e del quadricottero nello spazio rispetto all'inertial frame e l'altezza di riferimento che intercorre tra la superficie effettiva della Terra (considerata la sua curvatura) ed il suo sistema di riferimento piatto, stima il valore di latitudine, longitudine ed altitudine.

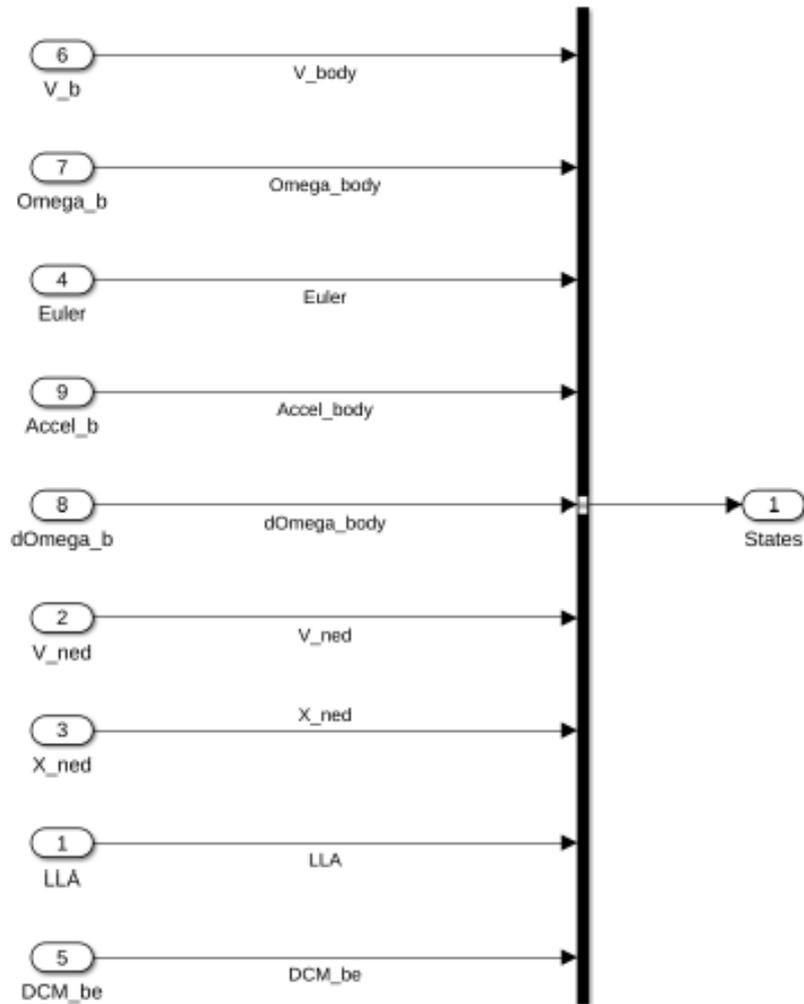


Figura 7.10: Bus Creator

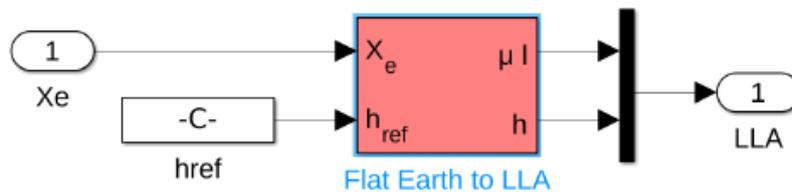


Figura 7.11: Position on Earth

7.1.2 Simulation Model - Enviroment

Il blocco Enviroment calcola le principali variabili ambientali, in particolare:

- Accelerazione di gravità assi NED;
- Temperatura dell'aria;
- Velocità del suono;
- Pressione;
- Densità dell'aria;
- Campo magnetico.

Le opzioni selezionabili (7.12) sono 2: ambiente variabile o costante. Si sceglie l'opzione costante in quanto le escursioni di quota e di spostamento non determinano un cambiamento significativo delle variabili in gioco (7.13).

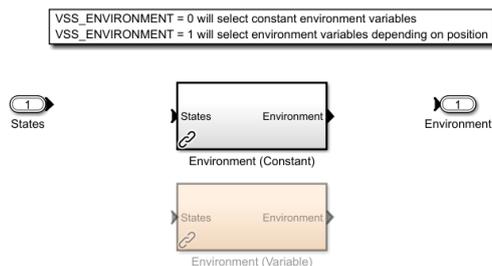


Figura 7.12: Blocco Enviroment

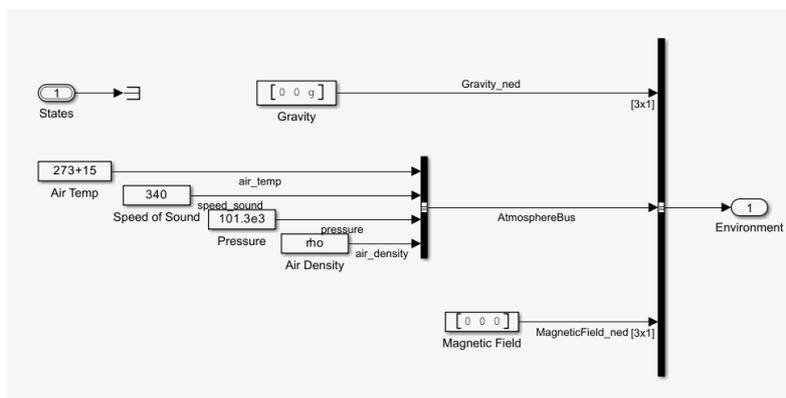


Figura 7.13: Blocco Enviroment Costante

7.1.3 Simulation Model - Sensoristica

Analizzando il blocco dei sensori si possono selezionare due alternative: sensori dinamici con rumore oppure sensori con passaggio diretto (*feedthrough*). All'interno di questo sottosistema sono presenti alcuni dati di calibrazione dei sensori codificati in modo statico e un blocco chiamato "sistema sensori", che contiene i modelli della fotocamera, dell'IMU, del sensore a ultrasuoni e del sensore di pressione. I sensori determinano gli stati del sistema:

- **Un'unità di misura inerziale (IMU):** misura le accelerazioni traslazionali e le velocità angolari.
- **Un sensore a ultrasuoni:** utilizzato per la misurazione dell'altitudine.
- **Una fotocamera:** è rivolta verso il basso per l'elaborazione delle immagini e la stima del flusso di luce .

Il modello può anche essere utilizzato per simulare in sicurezza un guasto e osservare il comportamento del sistema in presenza di bias o altri difetti.

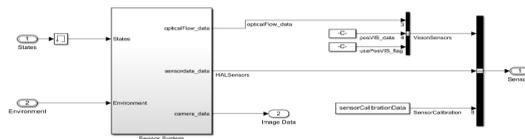


Figura 7.14: Blocco sensori

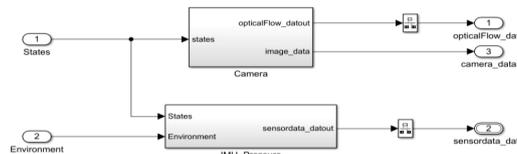


Figura 7.15: Blocco sensori

7.1.4 Flight Simulation - Visualizzazione risultati

Aperto il sottosistema "Visualizzazione del volo", i segnali possono essere visualizzati su un display da cockpit, fornito dal toolbox Aerospace Blockset, con alcuni strumenti di volo standard come l'indicatore di rotta, la percentuale di RPM, l'indicatore di velocità e di velocità di salita, e l'altimetro. Simulink 3D fornisce invece la rappresentazione grafica in 3 dimensioni del drone.

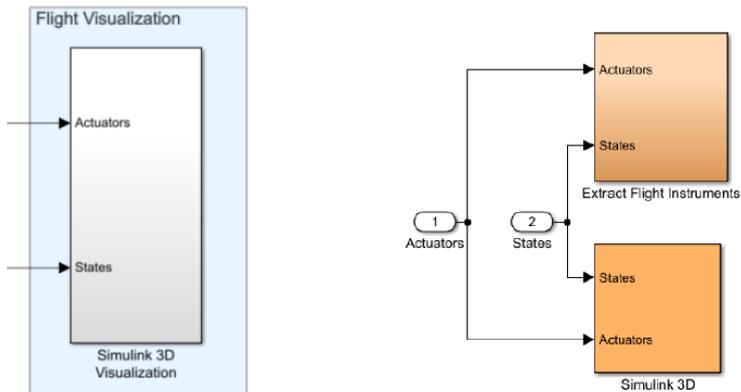


Figura 7.16: Blocco Flight Simulation

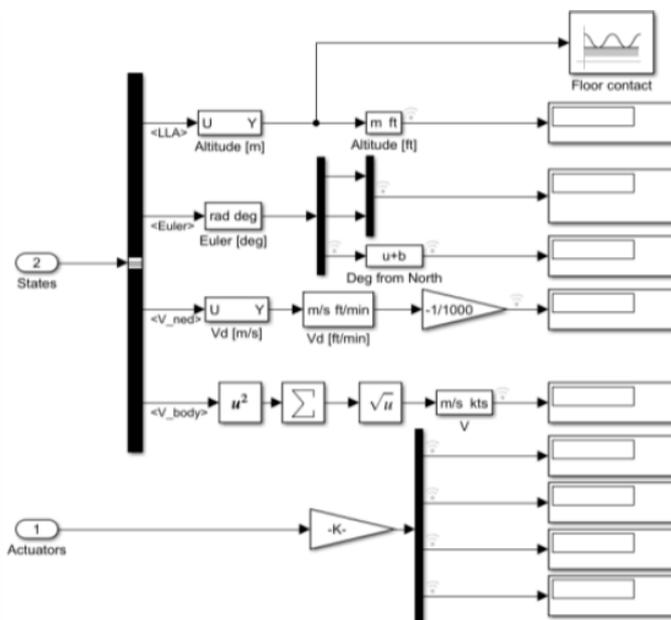


Figura 7.17: Blocco Visualization System

7.1.5 Flight Command

Il blocco **Command** mostra il processo di generazione dei segnali di riferimento, i quali vengono inviati in ingresso al controllore per consentire al drone di seguire la traiettoria desiderata durante il volo.

All'interno di questo blocco si individuano quattro sottoblocchi principali, ciascuno dei quali rappresenta un metodo differente per fornire al drone i riferimenti da seguire. La selezione del metodo avviene attraverso la variabile **VSS_COMMAND**, caricata nel Workspace di MATLAB dal progetto .

I metodi disponibili per la generazione dei riferimenti sono i seguenti:

- **Signal Editor:** consente di impostare i riferimenti utilizzando il blocco predefinito *Signal Editor* di MATLAB.
- **Joystick + Signal Editor:** permette di definire i riferimenti di rollio, beccheggio e imbardata tramite un joystick, con la possibilità di specificare anche le coordinate x , y e z mediante un blocco *Signal Editor*.
- **MAT Data:** utilizza dati memorizzati all'interno di un file MAT (.mat) per fornire i riferimenti di volo.
- **Spreadsheet Data:** consente di caricare i riferimenti da un foglio di calcolo Excel (.xlsx).

Nel caso di progetto MRAC si utilizza un blocco semplificato in cui si fornisce direttamente la reference in termini di x, y, z, ψ .

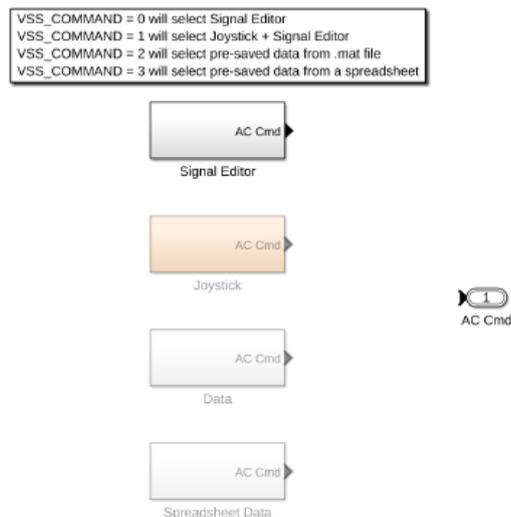


Figura 7.18: Blocco Command

7.1.6 Flight Control System

All'interno del **Flight Control System** sono presenti due sottosistemi:

- **Image Processing System:** contiene la parte di elaborazione grafica .
- **Control System:** include la logica di volo.

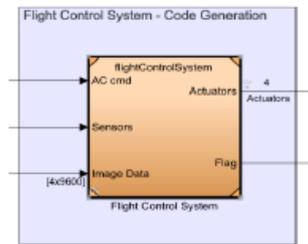


Figura 7.19: Blocco FCS

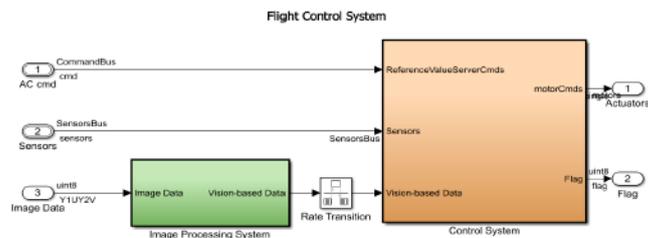


Figura 7.20: Blocco FCS

I due sottosistemi operano a frequenze diverse: l'Image Processing System lavora con frequenza di 60 Hz tipica della telecamera mentre il Control System opera a 200 Hz.

Per garantire il trasferimento dei dati tra sistemi con frequenze diverse, questi sono collegati tramite il blocco "Rate transition".

All'interno del blocco Control System sono presenti ulteriori sottoblocchi:

- **Path planning:** contiene la logica per l'algoritmo di tracciamento della traiettoria.
- **Crash Predictor Flags:** contiene la logica per spegnere il drone in caso di anomalie durante il volo.
- **State estimator:** include l'osservatore di stato.
- **Controller:** Contiene i controllori PID a cascata del controllore.

Si può notare anche un blocco "constant" con valore assegnato pari a 1. Questo blocco serve a definire la modalità di pilotaggio del drone:

- Se il valore è pari a 1: il drone muove la sua posizione in base agli input X e Y .
- Se il valore è pari a 0: il pilotaggio avviene attraverso la variazione dei valori di *Pitch* e *Roll*.

Tale sistema viene sostituito nel progetto dal controllore MRAC.

Capitolo 8

Controllore baseline PID a cascata

In questo capitolo si propone lo schema base di un classico controllore PID a cascata [25]. In particolare viene disaccoppiato il controllo della posizione (riquadro viola) da quella dell'assetto (riquadro azzurro) come mostrato in figura 8.1.

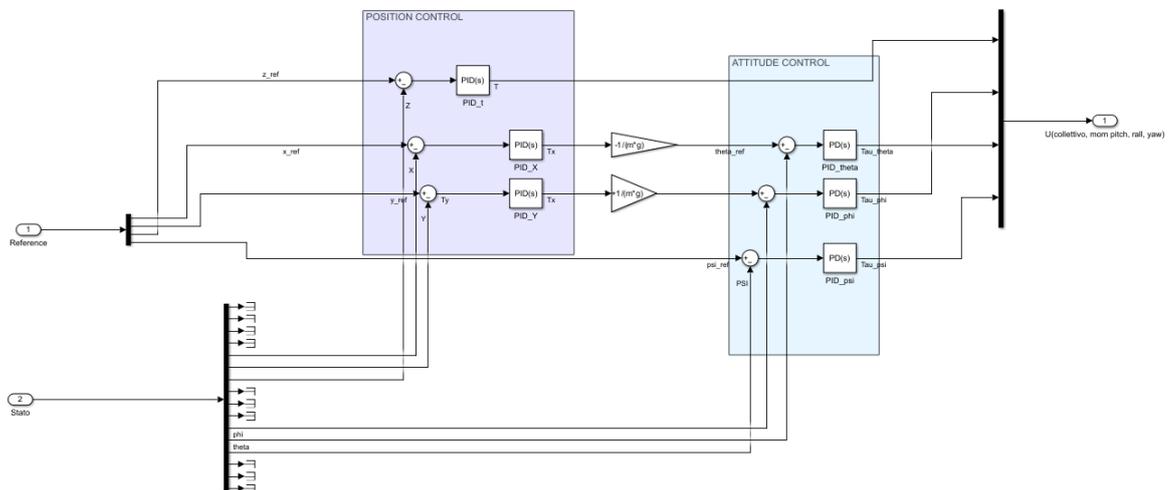


Figura 8.1: PID a cascata

Come si può notare vengono tenuti in considerazione solo i 3 elementi di posizione e 3 di assetto del vettore di stato. I termini di controllo sono infatti proporzionali ai relativi errori come specificato in seguito.

Tale controllore viene implementato al fine di effettuare un confronto in diverse situazioni con il controllore MRAC.

8.1 Architettura PID

Il principio del controllore PID implementato in questa sezione è quello di fornire un segnale di controllo proporzionale all'errore di posizione e di assetto come a seguire. Per il controllo di posizione si implementa un controllo PID (8.1, 8.2, 8.3) mentre per il controllo di assetto un PD(8.4, 8.5, 8.6). Le equazioni implementate sono le seguenti:

$$U(1) = T = e_z \cdot K_{P_z} \left(1 + K_{I_z} \cdot \frac{1}{s} + K_{D_z} \cdot s\right) \quad ; \quad e_z = z_{ref} - z \quad (8.1)$$

$$U(2)' = T_x = e_x \cdot K_{P_x} \left(1 + K_{I_x} \cdot \frac{1}{s} + K_{D_x} \cdot s\right) \quad ; \quad e_x = x_{ref} - x \quad (8.2)$$

$$U(3)' = T_y = e_y \cdot K_{P_y} \left(1 + K_{I_y} \cdot \frac{1}{s} + K_{D_y} \cdot s\right) \quad ; \quad e_y = y_{ref} - y \quad (8.3)$$

$$U(2) = \tau_\theta = e_\theta \cdot K_{P_\theta} (1 + K_{D_\theta} \cdot s) \quad ; \quad e_\theta = \theta_{ref} - \theta \quad (8.4)$$

$$U(3) = \tau_\phi = e_\phi \cdot K_{P_\phi} (1 + K_{D_\phi} \cdot s) \quad ; \quad e_\phi = \phi_{ref} - \phi \quad (8.5)$$

$$U(4) = \tau_\psi = e_\psi \cdot K_{P_\psi} (1 + K_{D_\psi} \cdot s) \quad ; \quad e_\psi = \psi_{ref} - \psi \quad (8.6)$$

I termini $x_{ref}, z_{ref}, y_{ref}, \psi_{ref}$ deriva dal vettore di riferimento mentre i termini θ_{ref}, ϕ_{ref} vengono calcolati dalle equazioni lineari come in (5.28 e 5.29) come in seguito:

$$\theta_{ref} = \frac{T_x}{m \cdot g} \quad ; \quad \phi_{ref} = \frac{-T_y}{m \cdot g} \quad (8.7)$$

Il comando finale è il vettore:

$$\mathbf{U} = [T, \tau_\phi, \tau_\theta, \tau_\psi]^T \quad (8.8)$$

8.1.1 Tuning PID

Il tuning del PID viene fatto passo a passo prima per il controllo posizione e poi assetto. Il metodo di partenza utilizzato è quello di Ziegler-Nichols come mostrato in tabella 8.1. Il modus operandi è di partire con il solo guadagno proporzionale K_p aumentandolo gradualmente fino ad ottenere un'oscillazione di ampiezza costante intorno ad un valore medio. Si definisce a questo punto $K_u = K_p$ valore proporzionale che impone oscillazioni costanti mentre T_u è il periodo di tale oscillazione. Con questi due termini è possibile tramite la tabella individuare dei valori di partenza per ottenere la dinamica desiderata.

Tipo di Controllo	K_p	T_i	T_d	K_i	K_d
P	$0.5K_u$	–	–	–	–
PI	$0.45K_u$	$0.83T_u$	–	$0.54K_u/T_u$	–
PD	$0.8K_u$	–	$0.125T_u$	–	$0.10K_uT_u$
PID classico	$0.6K_u$	$0.5T_u$	$0.125T_u$	$1.2K_u/T_u$	$0.075K_uT_u$
Regola di Pessen	$0.7K_u$	$0.4T_u$	$0.15T_u$	$1.75K_u/T_u$	$0.105K_uT_u$
Qualche sovraelongazione	$0.33K_u$	$0.50T_u$	$0.33T_u$	$0.66K_u/T_u$	$0.11K_uT_u$
Nessuna sovraelongazione	$0.20K_u$	$0.50T_u$	$0.33T_u$	$0.40K_u/T_u$	$0.066K_uT_u$

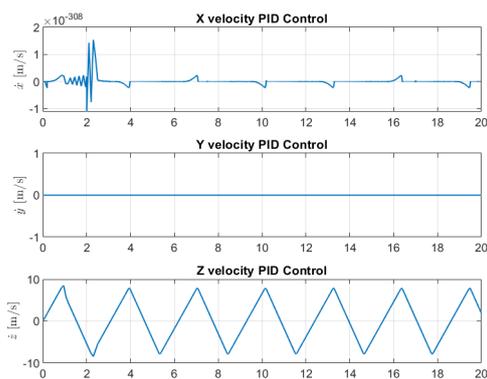
Tabella 8.1: Parametri del metodo di Ziegler-Nichols per diversi tipi di controllo

Tuning Pid su z

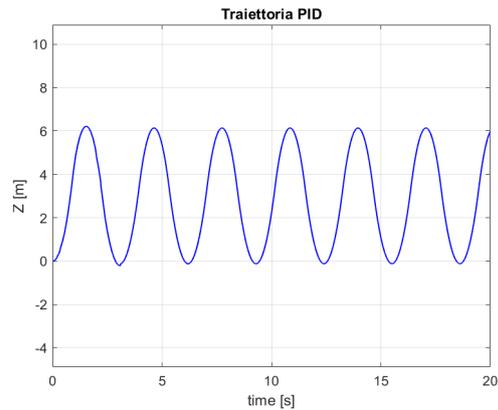
Per fare il tuning del controllore PID si prende ad esempio il controllore su z, usando la tecnica di Ziegler-Nichols.

Inanzitutto si incomincia con un semplice controllore proporzionale aumentando il valore di K_p fino ad ottenere un'oscillazione costante intorno ad un valore medio. Il valore che si ottiene è $K_p=K_u=2$, $T=T_u=3.1$ secondi.

Con il valore di $K_p=K_u$ si ottiene la simulazione in seguito. Si omettono i grafici dell'assetto in quanto è nullo essendo il controllo esclusivamente su z. Sfruttando



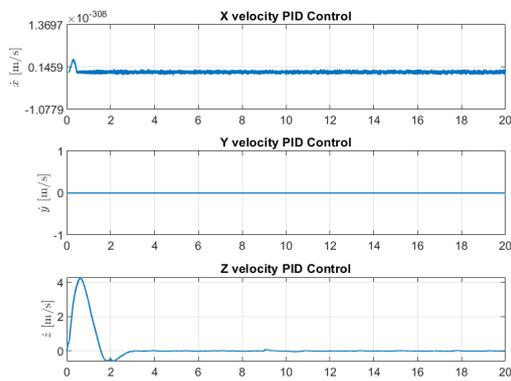
(a) Velocità Lineari controllore P



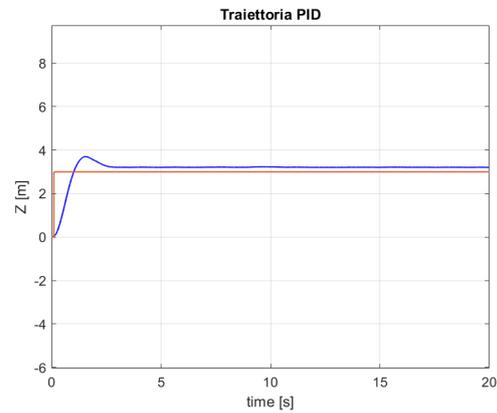
(b) Posizione z controllore P

i valori di K_u e T_u espresse precedentemente si possono ottenere i valori: $K_p=1.2$, $T_i=1.55$, $T_d=0.3875$. Sfruttando il blocco Simulink PID si ricavano i nuovi termini: $P=K_p$, $I=1/T_i$, $D=T_d$.

La simulazione seguente mostra il controllore PD, senza integratore. Come si può

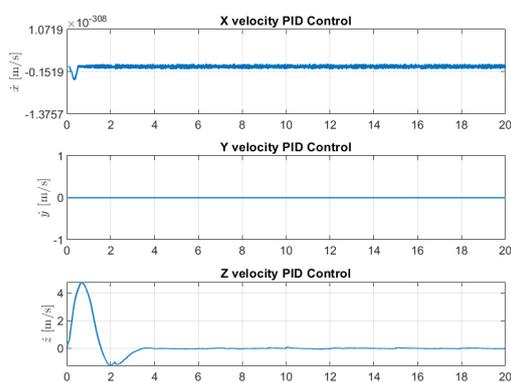


(a) Velocità Lineari controllore PD

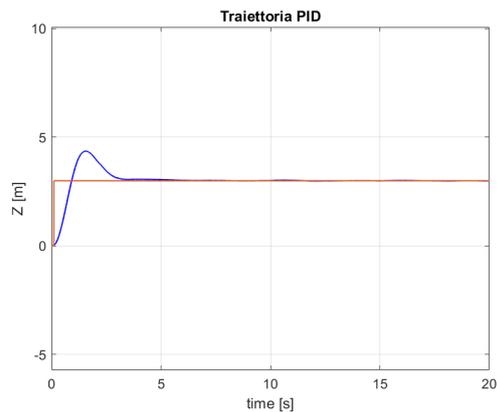


(b) Posizione z controllore PD

osservare il termine integrativo non garantisce un errore a regime nullo. Per ottenere ciò si aggiunge il termine integrativo ottenendo le simulazioni seguenti. Tale approccio



(a) Velocità Lineari controllore PID



(b) Posizione z controllore PID

viene utilizzato anche per il tuning degli altri tre termini di controllo. Si propone una simulazione con reference a step unitario di tutti i termini di comando.

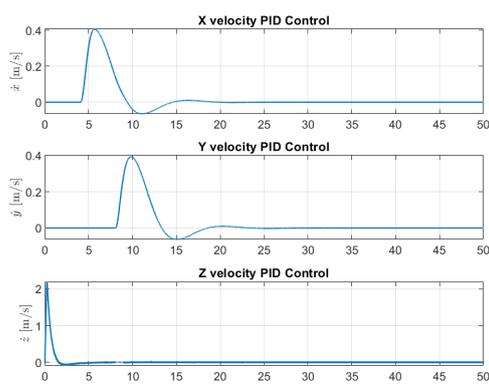
Simulazione Step PID

In questa simulazione si propone la versione finale utilizzata con i termini del controllore PID un po' modificati per ottenere le prestazioni seguenti:

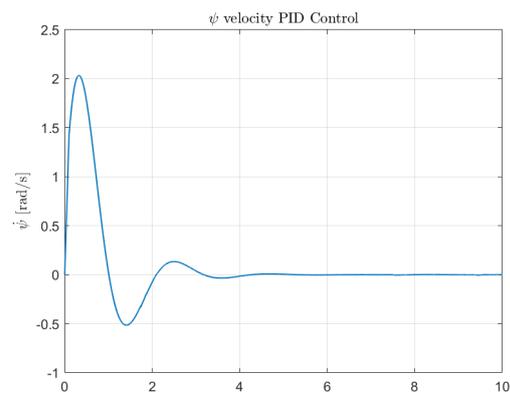
- **Overshoot:** 16% per posizione, 21% per ψ ;
- **Time to peak:** 2 secondi per z e ψ , 5 secondi per x e y;
- **Time to settle:** 10 secondi per z e ψ , 15 secondi per x e y;

	PID_z	PID_x	PID_y	PID_θ	PID_ϕ	PID_ψ
P	15	0.025	0.025	0.0015	0.0015	0.001
I	0.5	0.001	0.001	-	-	-
D	0.387	0.0001	0.0001	0.000236	0.000236	0.0002362

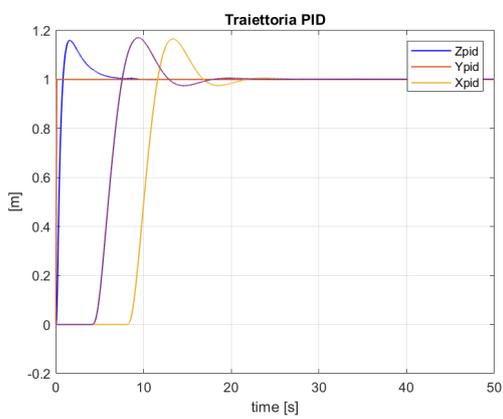
Tabella 8.2: Coefficienti controllore PID



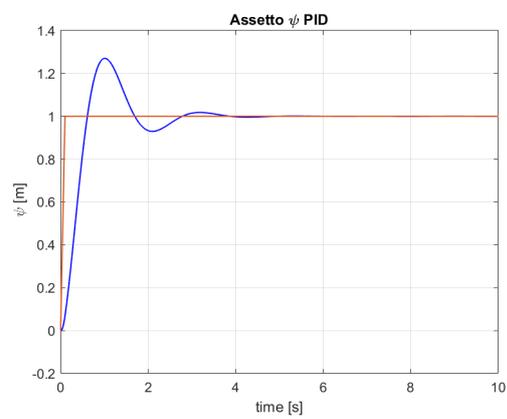
(a) Velocità Lineari



(b) Velocità angolare ψ



(a) Posizioni Lineari



(b) Posizione angolare ψ

Capitolo 9

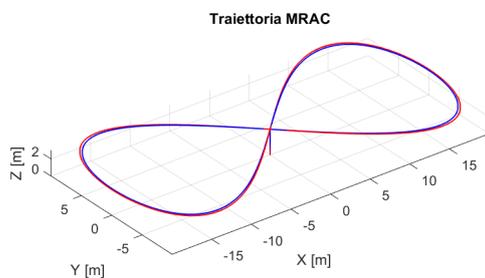
Simulazioni di confronto

Si propongono in questo capitolo diverse simulazioni sia con controllore MRAC con le modifiche proposte precedentemente (eccetto composite) che PID enfatizzando la bontà del controllore adattivo in caso di disturbi.

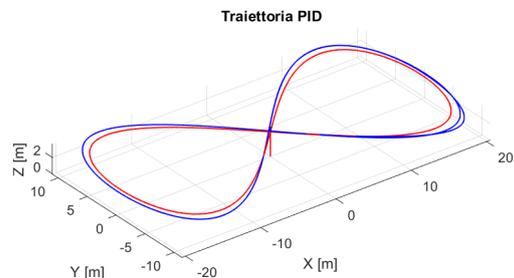
9.1 Simulazione ad 8 senza disturbi

La prima simulazione è una figura ad 8. Inizialmente si porta il drone all'altezza di 3 metri e successivamente si effettua una figura ad 8 intersecando due sinusoidi come reference signal. La distanza massima percorsa sull'asse y è 10 metri mentre su x 20 metri. Si ottiene perciò una figura "schiacciata" all'estremità.

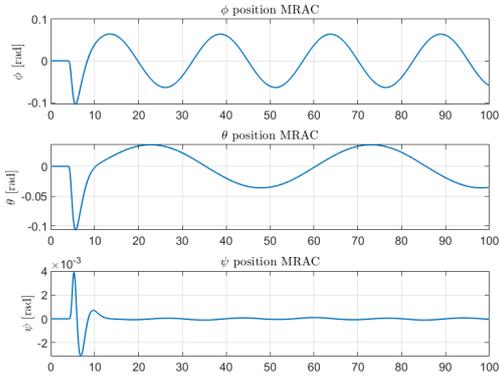
In seguito si vogliono mostrare i principali grafici in termini di angoli di assetto, posizione, velocità lineari e angolari sia per il model reference che per il controllore PID e MRAC.



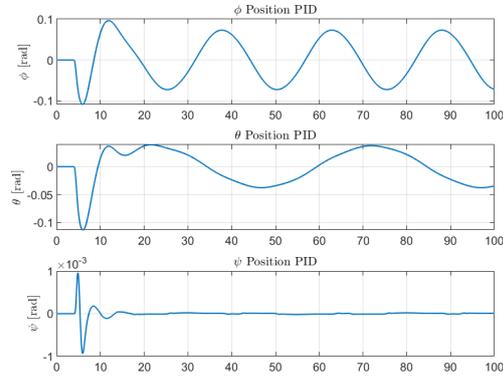
(a) Posizione 3D MRAC



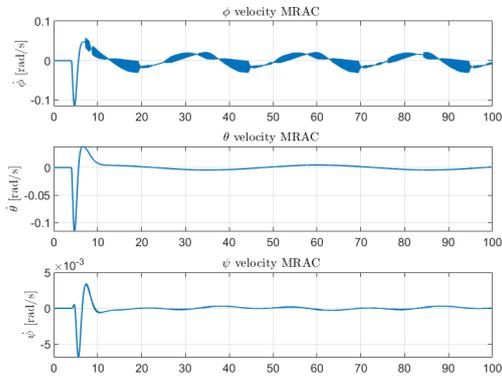
(b) Posizione 3D PID



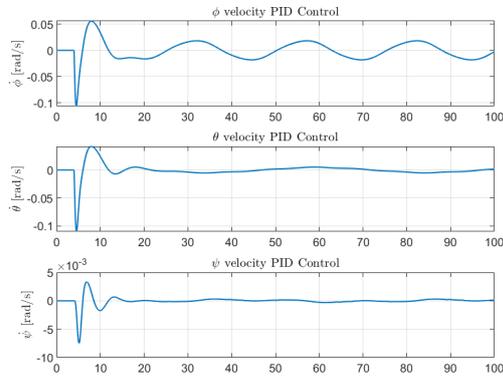
(a) Posizione angolare MRAC



(b) Posizione angolare PID



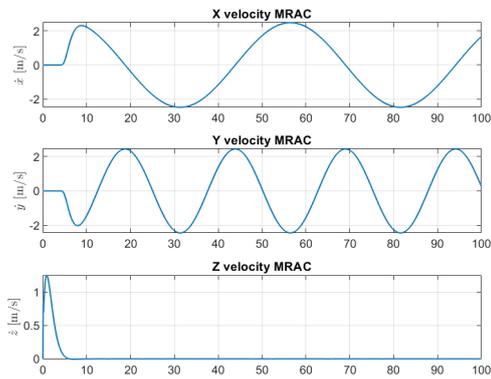
(a) Velocità angolare MRAC



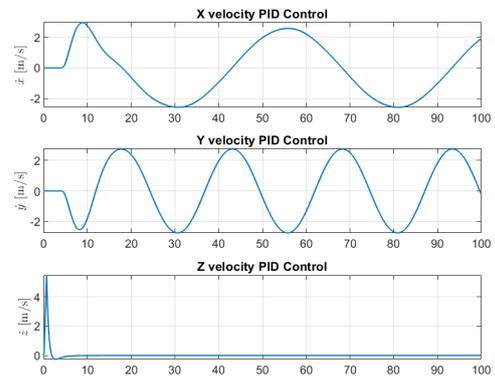
(b) Velocità angolare PID

Come si può osservare il controllore MRAC riesce a mantenere una traiettoria estremamente solidale alla reference. In particolare l'errore più elevato è di 30 cm per un escursione in x di 20 metri. Si mostrano anche i grafici del modello di riferimento. Nelle simulazioni a venire verrà spesso trascurato in quanto il controllore MRAC ha la capacità di combaciare notevolmente con il model reference e diventa perciò rindondante mostrare i tutti i grafici. Si osserva inoltre la capacità del modello di riferimento di variare grazie all'osservatore. Il controllore MRAC in assenza di disturbi mostra alcune oscillazioni sulle velocità angolari che però nella pratica verrebbero filtrate dalla dinamica dei sensori. Tali oscillazioni in alta frequenza infatti non rispecchiano un'oscillazione degli angoli che hanno invece un andamento lineare.

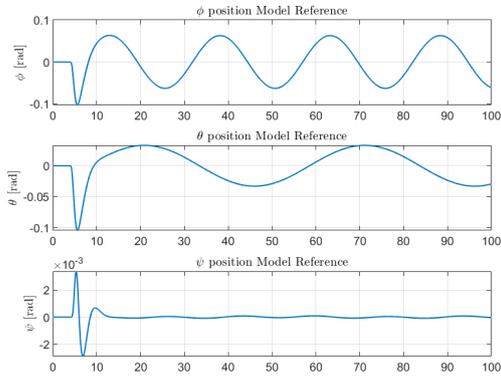
Il controllore PID invece, pur mantenendo andamenti lineari sia in velocità che posizione, mostra un maggior margine di errore che lo rende meno preciso dell'MRAC. Tale errore viene amplificato nel caso di disturbi come mostrato nelle simulazioni a seguire.



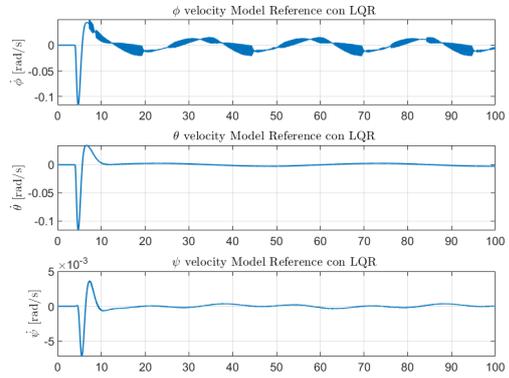
(a) Velocità lineare MRAC



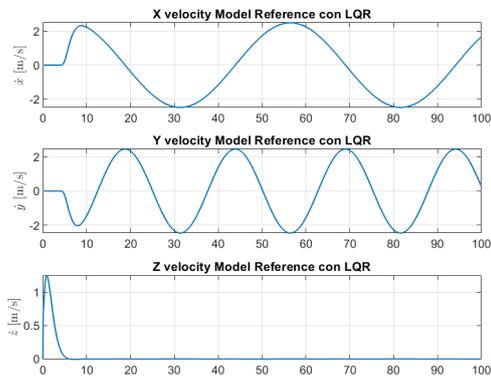
(b) Velocità lineare PID



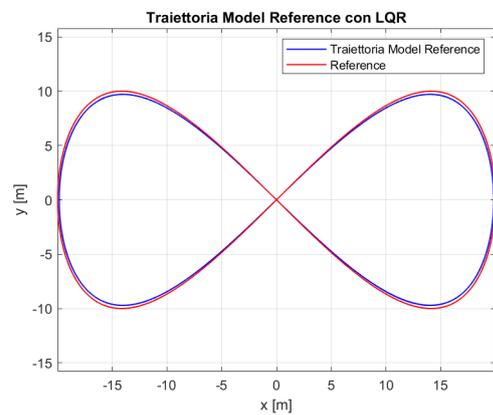
(a) Posizioni angolari modello di riferimento



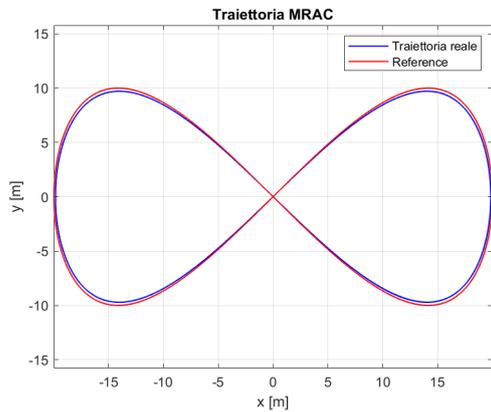
(b) Velocità angolari modello di riferimento



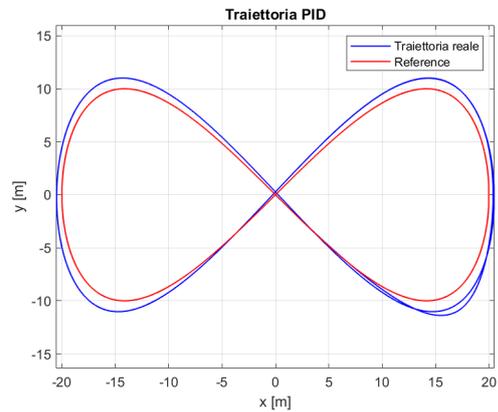
(a) Velocità lineari modello di riferimento



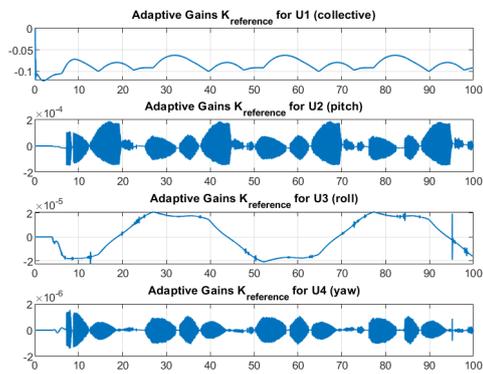
(b) Traiettoria x-y modello di riferimento



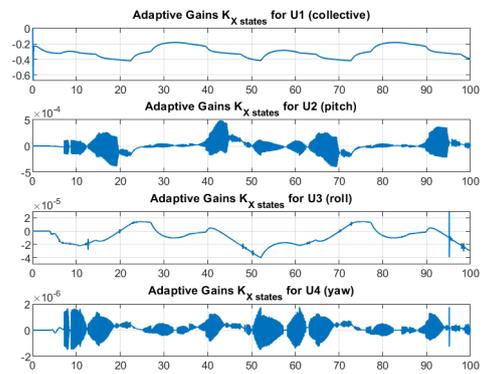
(a) Traiettorie x-y MRAC



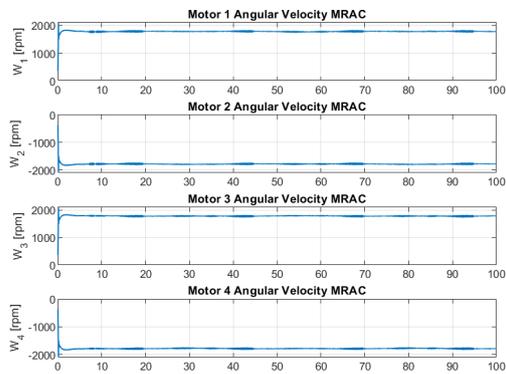
(b) Traiettorie x-y PID



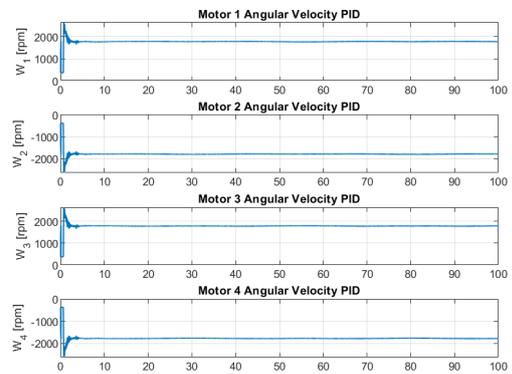
(a) Guadagno adattivo K_r



(b) Guadagno adattivo K_x



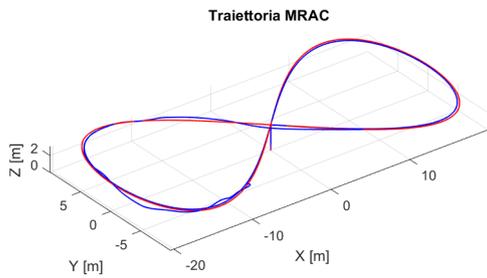
(a) Velocità angolari motori MRAC



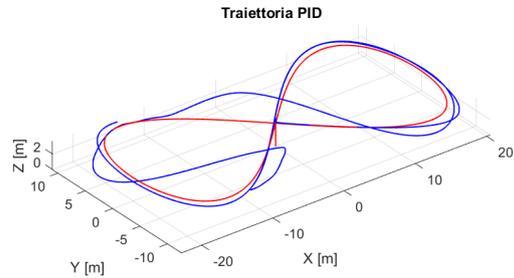
(b) Velocità angolari motori PID

9.2 Simulazione ad 8 con disturbi intensivi

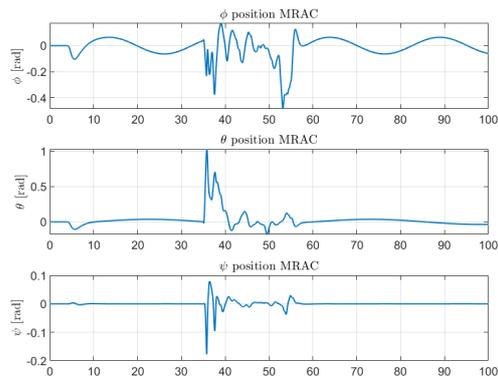
In questa simulazione si utilizza la medesima reference con una forza di disturbo con modello di dryden tra 35 e 55 secondi.



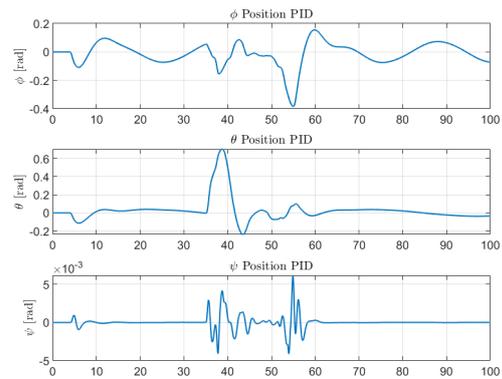
(a) Posizione 3D MRAC



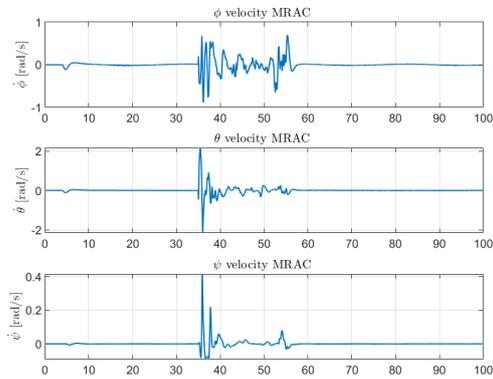
(b) Posizione 3D PID



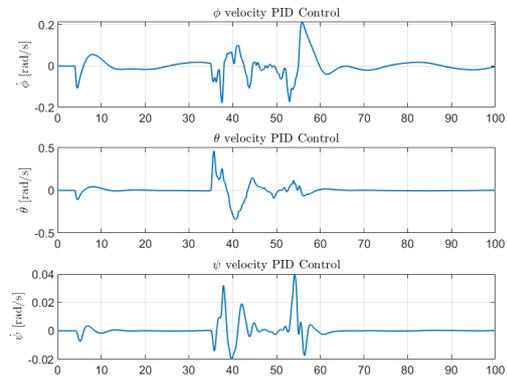
(a) Posizione angolare MRAC



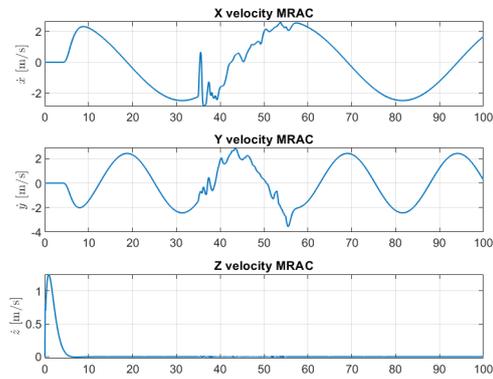
(b) Posizione angolare PID



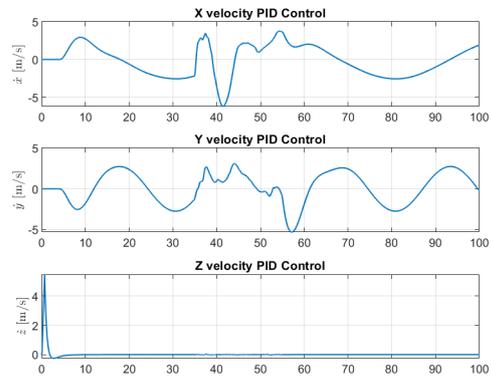
(a) Velocità angolare MRAC



(b) Velocità angolare PID

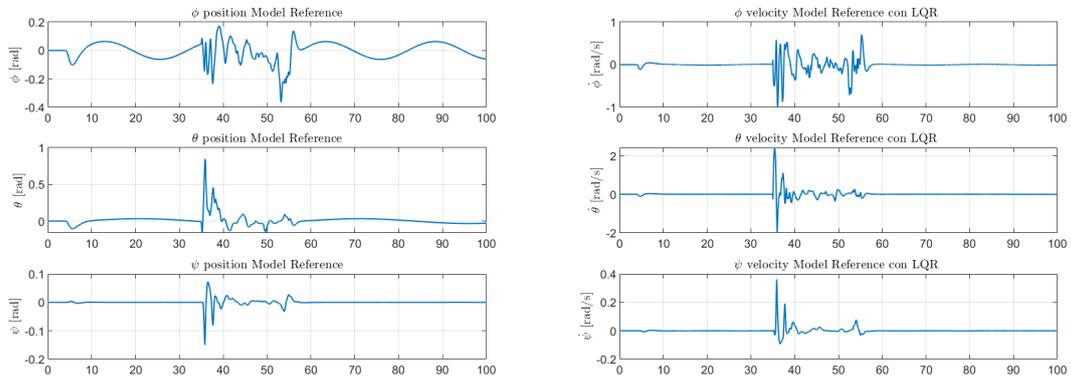


(a) Velocità lineare MRAC

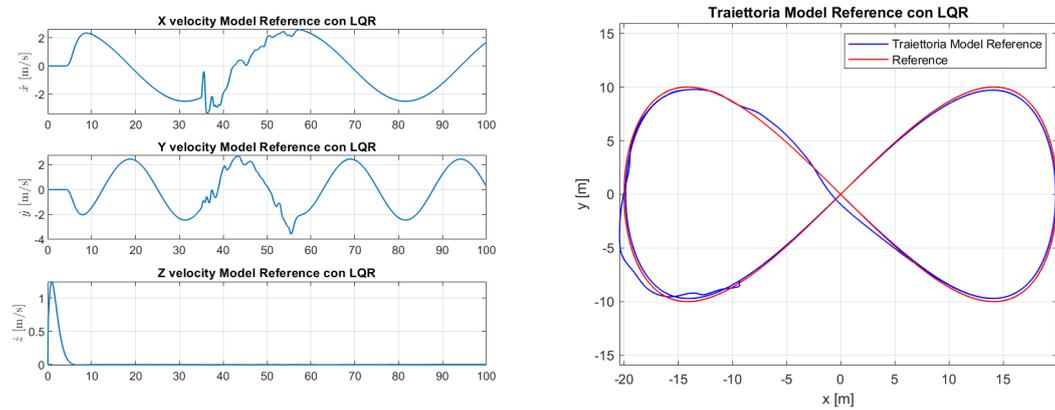


(b) Velocità lineare PID

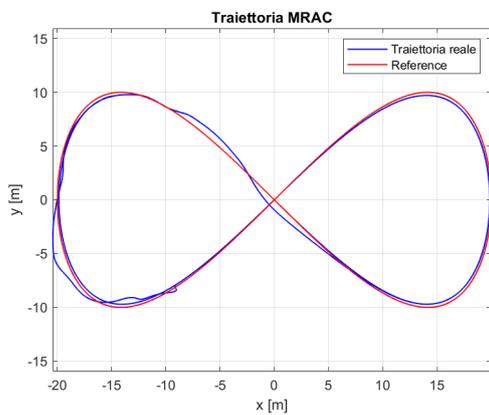
In questa simulazione si può notare come la forza di disturbo raggiunga i 0.45 N pari essenzialmente al peso del drone. Per questo motivo il controllore PID non riesce a compensare il disturbo ottenendo una traiettoria estremamente deformata. Il controllore MRAC invece mostra la sua bontà rimanendo solidale al modello di riferimento.



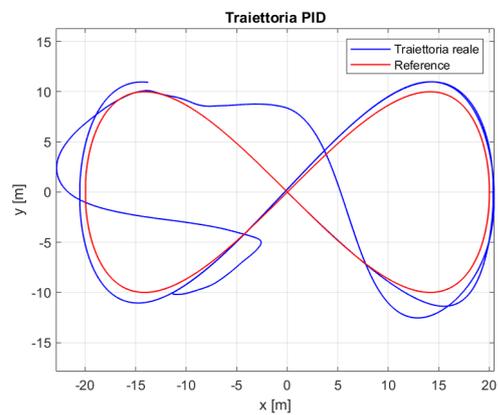
(a) Posizioni angolari modello di riferimento (b) Velocità angolari modello di riferimento



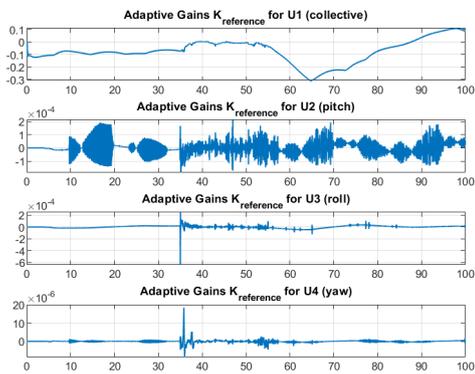
(a) Velocità lineari modello di riferimento (b) Traiettorie x-y modello di riferimento



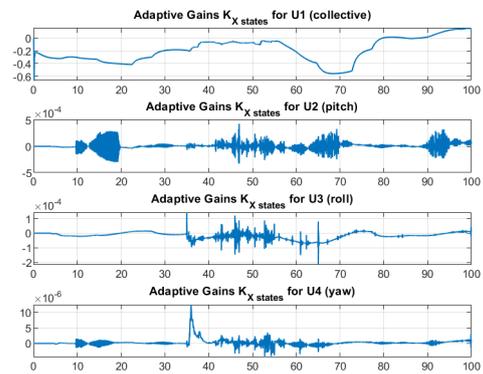
(a) Traiettorie x-y MRAC



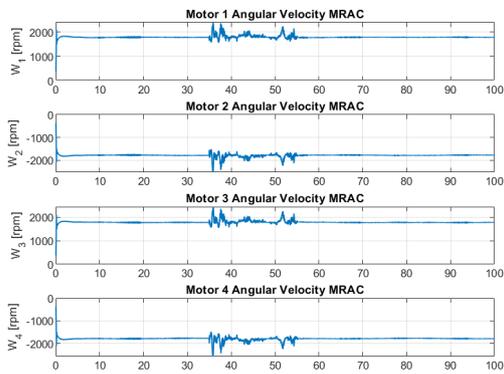
(b) Traiettorie x-y PID



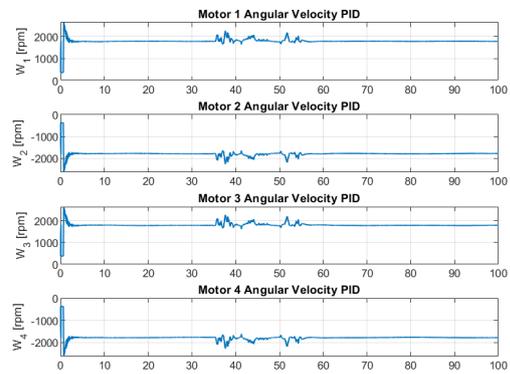
(a) Guadagno adattivo K_r



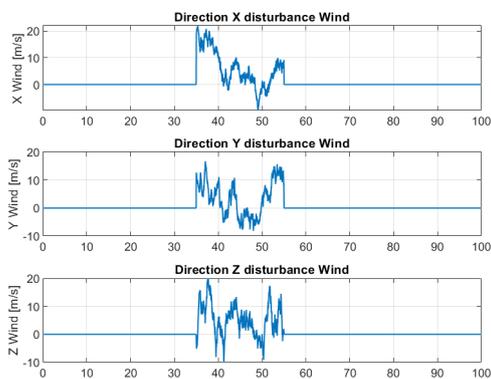
(b) Guadagno adattivo K_x



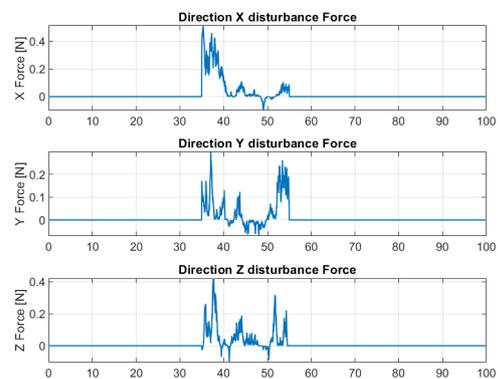
(a) Velocità angolari motori MRAC



(b) Velocità angolari PID



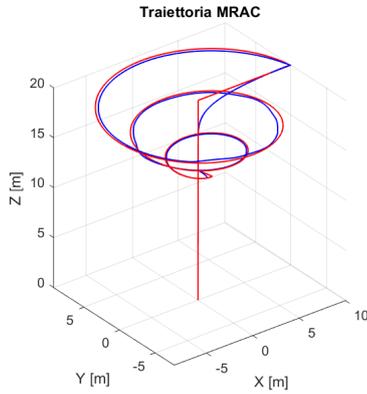
(a) Velocità vento di disturbo



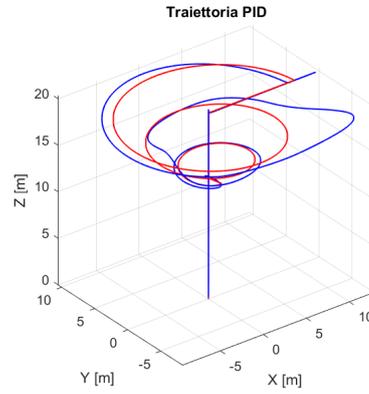
(b) Forza di disturbo

9.3 Simulazione Spirale con disturbi

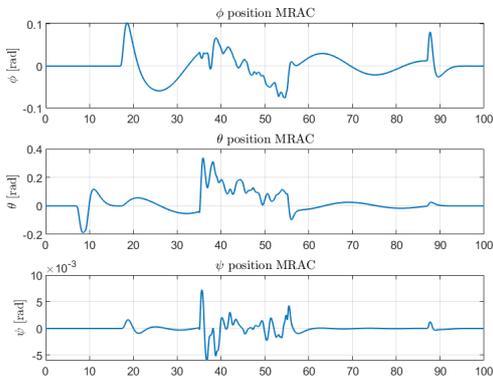
In questa simulazione si propone una reference più complessa. In particolare si porta il drone alla quota di 20 metri e dopodichè si incomincia una spiare discendente con raggio massimo 10 metri. Al termine della spirale il drone ritorna in posizione x e y nulla mantenendosi alla quota raggiunta.



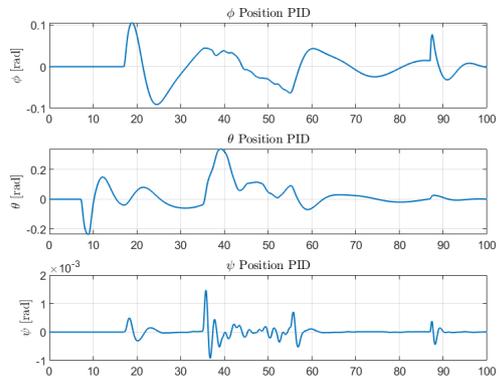
(a) Posizione 3D MRAC



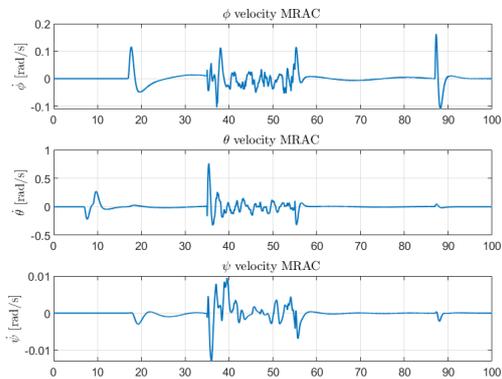
(b) Posizione 3D PID



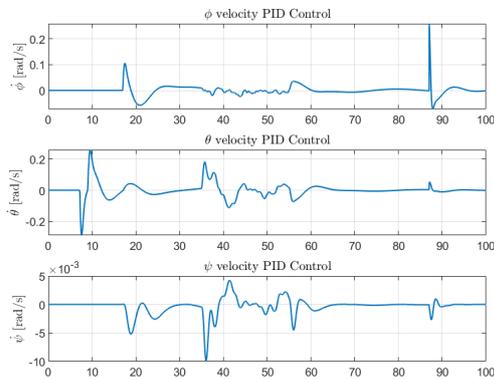
(a) Posizione angolare MRAC



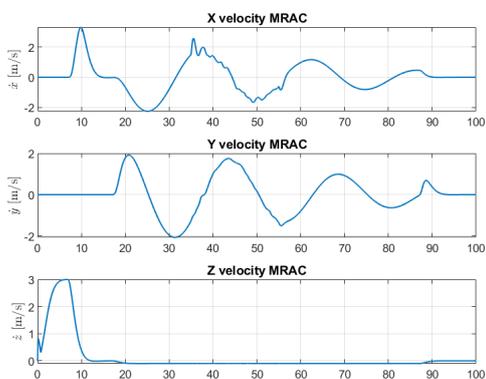
(b) Posizione angolare PID



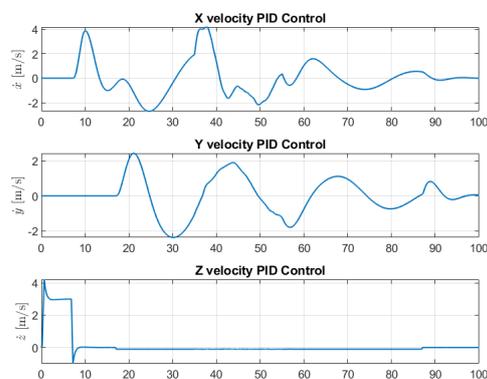
(a) Velocità angolare MRAC



(b) Velocità angolare PID

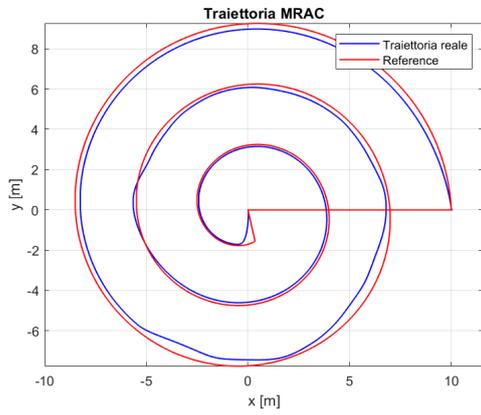


(a) Velocità lineare MRAC

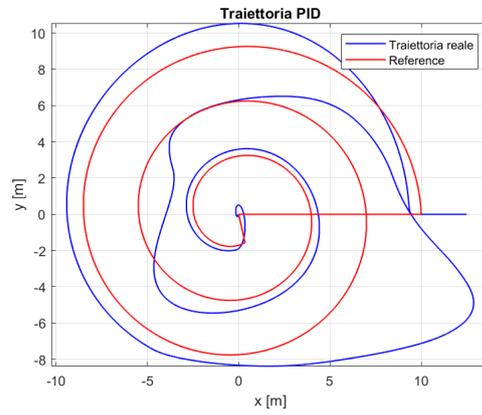


(b) Velocità lineare PID

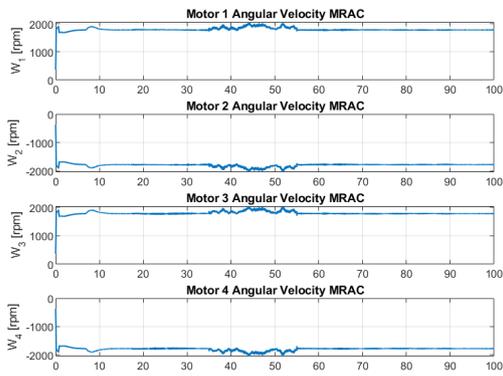
Il vento di disturbo raggiunge i 14 metri al secondo rendendo la dinamica del drone estremamente condizionata. Anche in questo caso osservando i grafici si può notare la bontà del controllore MRAC che continua a mantenere la traiettoria solidale a quella del modello di riferimento che segue a sua volta la reference. Nel modello PID l'errore di posizione arriva a superare il 100% rispetto al modello di riferimento rendendo a tutti gli effetti il controllore inefficace. Si nota inoltre la repentina adattabilità dell'MRAC soprattutto tra i 35 e 55 secondi di simulazione in cui il drone è soggetto ai disturbi.



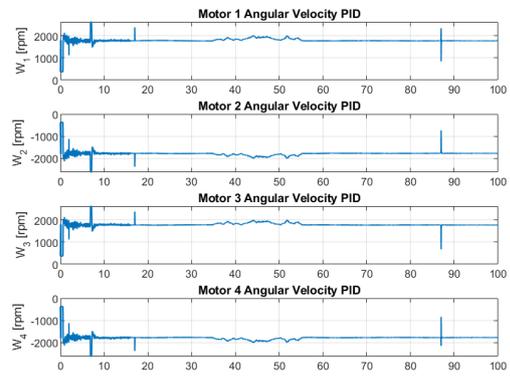
(a) Traiettorie x-y MRAC



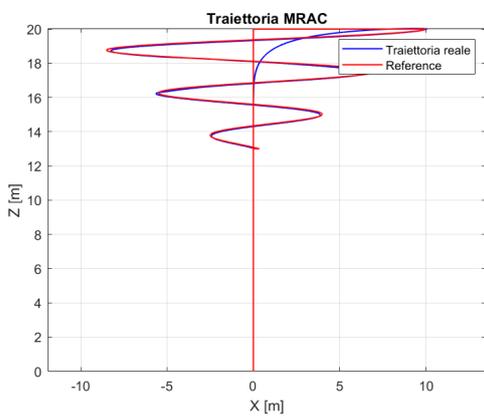
(b) Traiettorie x-y PID



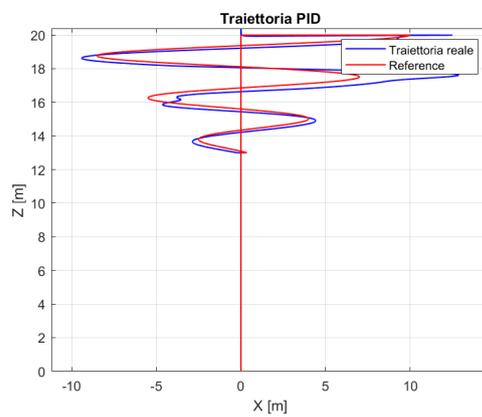
(a) Velocità angolari motori MRAC



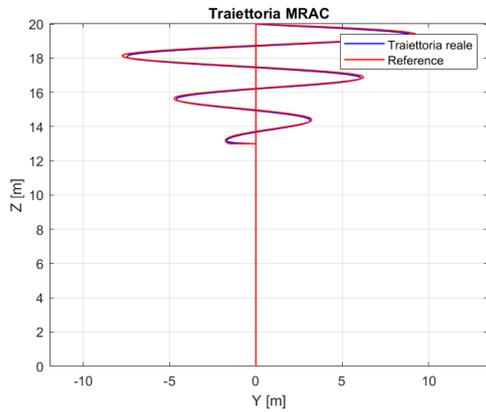
(b) Velocità angolari motori PID



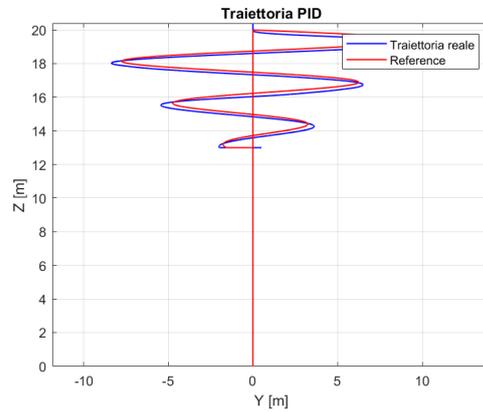
(a) Traiettorie x-z MRAC



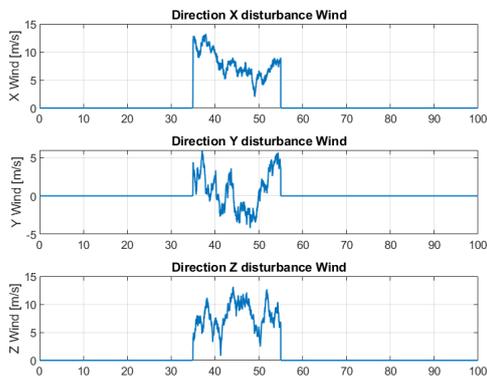
(b) Traiettorie x-z PID



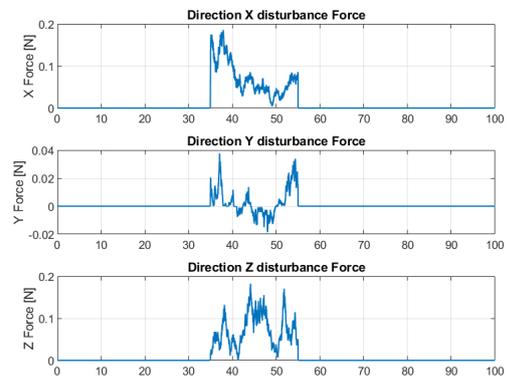
(a) Traiettorie y-z MRAC



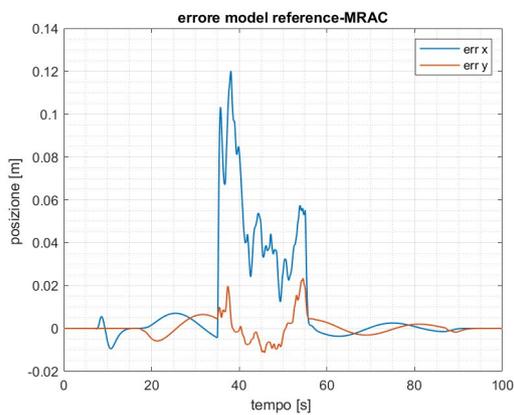
(b) Traiettorie y-z PID



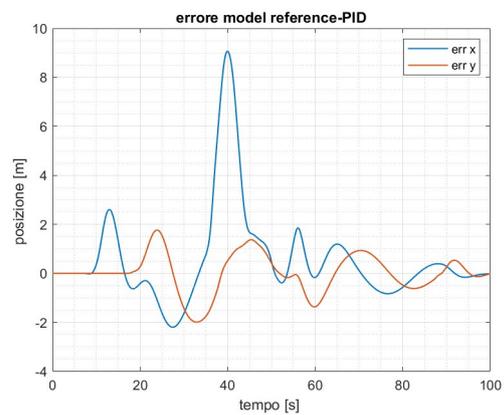
(a) Velocità vento di disturbo



(b) Forza di disturbo



(a) Errore MRAC-Model Reference



(b) Errore PID-Model Reference

Capitolo 10

Conclusioni e sviluppi futuri

Al termine della sezione simulazioni è possibile tratte alcune conclusioni sulla bontà del controllore MRAC rispetto al controllo tradizionale PID.

In primis si evidenzia la capacità del controllore adattivo di rispondere in modo efficace a disturbi anche di elevata intensità. Si nota infatti che si è arrivati a sollecitare il drone con forze di disturbo pari alla sua forza peso, ottenendo risultati comunque soddisfacenti. La stessa cosa non si può dire del PID la cui dinamica diventa incontrollata e il controllo inefficace nel caso di disturbi intensivi.

Dal punto di vista computazionale il controllore MRAC risulta più pesante rispetto al PID, rendendo la complessità di calcolo maggiore. Questo aspetto negativo del controllore adattivo può essere migliorato agendo ulteriormente sull'architettura e analizzando i blocchi Simulink maggiormente coinvolti che ritardano la velocità di calcolo.

In caso di disturbi il controllore adattivo risulta perciò essere la soluzione prestazionalmente più conveniente che permette errori contenuti rispetto al modello di riferimento. Anche le velocità angolari dei motori elettrici risultano avere un andamento meno oscillatorio dei controllori classici con una capacità di controllo più stabile.

Il principale svantaggio dell'architettura MRAC risulta essere un'adattabilità difficile nel caso di assenza di disturbo (assenza di "eccitazione"). E' dimostrabile infatti che MRAC funziona meglio in condizioni di "persistence of excitation".

Il lavoro proposto in questo progetto di Tesi apre alla possibilità di un possibile sviluppo successivo su hardware reale. In particolare è possibile sfruttare la funzionalità intrinseca a Simulink chiamata "code generator" per ottenere una prima versione del codice in linguaggio C che può essere implementato nell'autopilota. In secondo luogo è possibile un test sul campo in cui verificare l'efficacia nel caso reale, in presenza di disturbi. Tale sperimentazione è un passo fondamentale al fine di ottenere una validazione del controllore per il suo utilizzo su un drone reale.

Appendice A

Derivazione legge controllo

A.1 Legge di controllo base

Con analisi di Lyapunov e la selezione della funzione candidata [5.55](#)

$$\begin{aligned}\dot{V} &= \dot{e}^T P e + e^T P \dot{e} + 2 \operatorname{tr} \left(\left[\Delta K_x^T \Gamma_x^{-1} \hat{K}_x + \Delta K_r^T \Gamma_r^{-1} \hat{K}_r + \Delta \Theta^T \Gamma_\Theta^{-1} \hat{\Theta} \right] \right) \\ &= (A_{\text{ref}} e + B (\Delta K_x^T x + \Delta K_r^T r - \Delta \Theta^T \Phi(x)))^T P e \\ &\quad + e^T P (A_{\text{ref}} e + B (\Delta K_x^T x + \Delta K_r^T r - \Delta \Theta^T \Phi(x))) \\ &\quad + 2 \operatorname{tr} \left(\left[\Delta K_x^T \Gamma_x^{-1} \hat{K}_x + \Delta K_r^T \Gamma_r^{-1} \hat{K}_r + \Delta \Theta^T \Gamma_\Theta^{-1} \hat{\Theta} \right] \right) \\ &= e^T (A_{\text{ref}} P + P A_{\text{ref}}) e + 2 e^T P B (\Delta K_x^T x + \Delta K_r^T r - \Delta \Theta^T \Phi(x)) \\ &\quad + 2 \operatorname{tr} \left(\left[\Delta K_x^T \Gamma_x^{-1} \hat{K}_x + \Delta K_r^T \Gamma_r^{-1} \hat{K}_r + \Delta \Theta^T \Gamma_\Theta^{-1} \hat{\Theta} \right] \right)\end{aligned}$$

Ricordando che:

$$P A_{\text{ref}} + A_{\text{ref}}^T P = -Q$$

Si ottiene:

$$\begin{aligned}\dot{V} &= -e^T Q e + \left[2 e^T P B \Delta K_x^T x + 2 \operatorname{tr} \left(\Delta K_x^T \Gamma_x^{-1} \hat{K}_x \right) \right] \\ &\quad + \left[2 e^T P B \Delta K_r^T r + 2 \operatorname{tr} \left(\Delta K_r^T \Gamma_r^{-1} \hat{K}_r \right) \right] \\ &\quad + \left[-2 e^T P B \Delta \Theta^T \Phi(x) + 2 \operatorname{tr} \left(\Delta \Theta^T \Gamma_\Theta^{-1} \hat{\Theta} \right) \right]\end{aligned}$$

Utilizzando la proprietà della traccia:

$$\begin{aligned}
 e^T P B \Delta K_x^T x &= \text{tr} \left(\Delta K_x^T x e^T P B \right) \\
 e^T P B \Delta K_r^T r &= \text{tr} \left(\Delta K_r^T r e^T P B \right) \\
 e^T P B \Delta \Theta^T \Phi(x) &= \text{tr} \left(\Delta \Theta^T \Phi(x) e^T P B \right)
 \end{aligned}$$

Si ottiene così:

$$\begin{aligned}
 \dot{V} &= -e^T Q e + 2 \text{tr} \left(\Delta K_x^T \left[\Gamma_x^{-1} \hat{K}_x + x e^T P B \right] \right) \\
 &+ 2 \text{tr} \left(\Delta K_r^T \left[\Gamma_r^{-1} \hat{K}_r + r e^T P B \right] \right) + 2 \text{tr} \left(\Delta \Theta^T \left[\Gamma_\Theta^{-1} \hat{\Theta} - \Phi(x) e^T P B \right] \right) \quad (\text{A.1})
 \end{aligned}$$

Se le leggi adattive sono selezionate come:

$$\begin{aligned}
 \dot{\hat{K}}_x &= -\Gamma_x x e^T P B \\
 \dot{\hat{K}}_r &= -\Gamma_r r(t) e^T P B \\
 \dot{\hat{\Theta}} &= \Gamma_\Theta \Phi(x) e^T P B
 \end{aligned}$$

allora la derivata temporale di V diventa globalmente semidefinita negativa:

$$\dot{V} = -e^T Q e \leq 0$$

A.2 σ – modification

Si scelga come funzione di Lyapunov candidata la A.1 ottenuta come nella legge di controllo base:

$$\begin{aligned}
 \dot{V} &= -e^T Q e + 2 \text{tr} \left(\Delta K_x^T \left[\Gamma_x^{-1} \hat{K}_x + x e^T P B \right] \right) \\
 &+ 2 \text{tr} \left(\Delta K_r^T \left[\Gamma_r^{-1} \hat{K}_r + r e^T P B \right] \right) + 2 \text{tr} \left(\Delta \Theta^T \left[\Gamma_\Theta^{-1} \hat{\Theta} - \Phi(x) e^T P B \right] \right) \quad (\text{A.1})
 \end{aligned}$$

Selezionando la nuova legge adattiva con σ modification si ha:

$$\begin{aligned}
 \dot{\hat{K}}_x &= -\Gamma_x (x e^T P B + \sigma \hat{K}_x) \\
 \dot{\hat{K}}_r &= -\Gamma_r (r(t) e^T P B + \sigma \hat{K}_r) \\
 \dot{\hat{\Theta}} &= \Gamma_\Theta (\Phi(x) e^T P B - \sigma \hat{\Theta})
 \end{aligned}$$

Sostituendo tale espressione in A.1 si ottiene:

$$\dot{V} = -e^T Q e - 2\sigma \text{tr} \left(\Delta K_x^T \hat{K}_x \right) - 2\sigma \text{tr} \left(\Delta K_r^T \hat{K}_r \right) - 2\sigma \text{tr} \left(\Delta \Theta^T \hat{\Theta} \right) < 0$$

A.3 Algoritmo Simulink per calcolo di Forze e Momenti

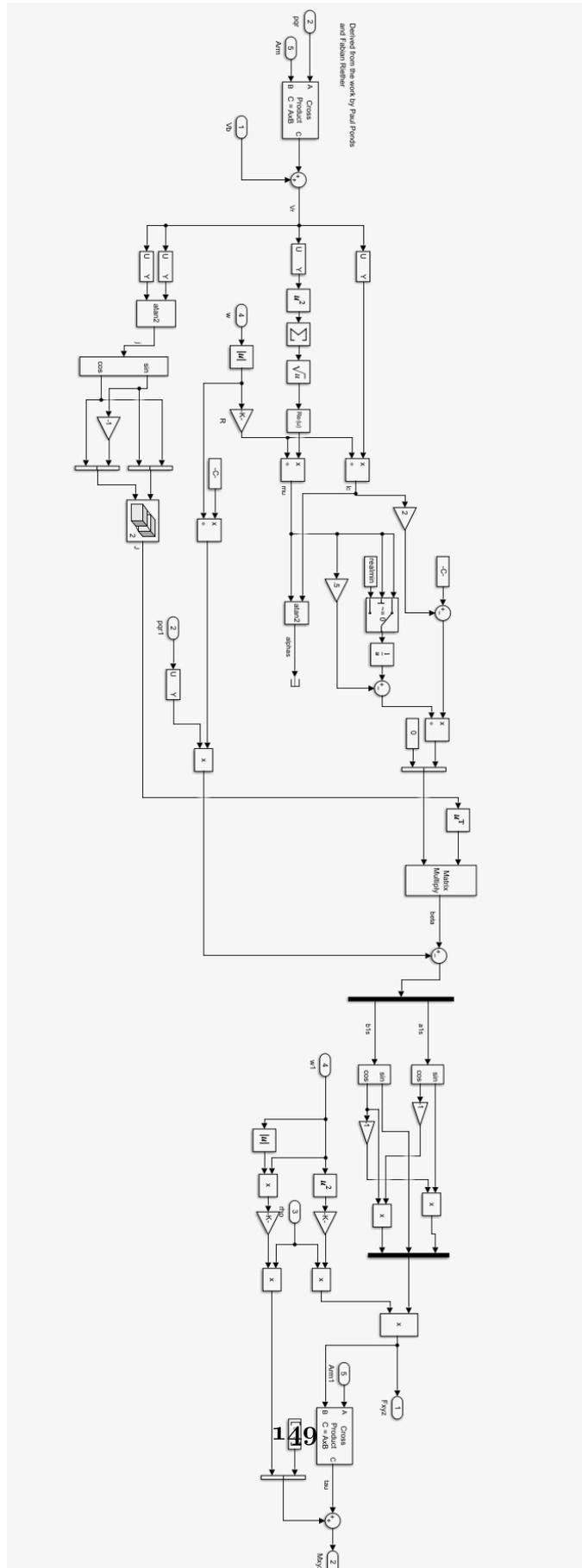


Figura A.1: Algoritmo per Forze e Momenti

Bibliografia

- [1] Zachary T. Dydek, Anuradha M. Annaswamy, Eugene Lavretsky, *Adaptive Control of Quadrotor UAVs: A Design Trade Study With Flight Evaluations*, IEEE Transactions on Control Systems Technology, VOL. 21, NO. 4, July 2013.
- [2] Eugene Lavretsky, *Combined/Composite Model Reference Adaptive Control*, IEEE Transactions on Automatic Control, VOL. 54, NO. 11, November 2009.
- [3] Zhong-Sheng Hou, Zhuo Wang, *From model-based control to data-driven control: Survey, classification and perspective*, Information Sciences, VOL.235 , June 2013.
- [4] Kimon P. Valavanis George J. Vachtsevanos, *Handbook of Unmanned Aerial Vehicles*, Springer, 2015.
- [5] Mouhacine Benosman, *Model-based vs data-driven adaptive control: An overview*, International Journal of Adaptive Control and Signal Processing, January 2018.
- [6] Eugene Lavretsky, Travis E. Gibson, Anuradha M. Annaswamy *Projection Operator in Adaptive Systems*, October 2012
- [7] Petros A. Ioannou, Jing Sun, *Robust Adaptive Control* Prentice-Hall, 1996.
- [8] Eugene Lavretsky, *Adaptive Control: Introduction, Overview, and Applications*, cds.caltech.edu
- [9] Brian L.Stevens, Frank L.Lewis, Eric N. Johnson, *Aircraft Control and Simulation*, John Wiley Sons, 2016
- [10] Anuradha M. Annaswamy, *Adaptive Control and Intersections with Reinforcement Learning*, Annual Review of Control, Robotics, and Autonomous Systems, January 2023
- [11] Ioan Doré Landau, Rogelio Lozano, Mohammed M'Saad, Alireza Karimi, *Adaptive Control Algorithms, Analysis and Applications* , Springer, 2011.
- [12] Petros Ioannou, Baris Fidan, *Adaptive Control Tutorial*, Siam, 2006.
- [13] Aminurrashid BIN Noordin, Zaharuddin Mohamed, Ariffanan Basri, *Simulation and experimental study on PID control of a quadrotor MAV with perturbation*, Bulletin of Electrical Engineering and Informatics, October 2020.

- [14] Santos Miguel Orozco Soto, Jonathan Cacace, Fabio Ruggiero, Vincenzo Lippiello, *Active Disturbance Rejection Control for the Robust Flight of a Passively Tilted Hexarotor* MDPI Journal, 2022.
- [15] Shida Liu, Zhongsheng Hou, Xin Zhang, Honghai Ji, *Model-free adaptive control method for a class of unknown MIMO systems with measurement noise and application to quadrotor aircraft* IET Control Theory Applications, Vol. 14, 2020.
- [16] Ngoc Phi Nguyen, Nguyen Xuan Mung, Ha Le Nhu Ngoc Thanh, Tuan Tu Huynh, Ngoc Tam Lam, Sung Kyung Hong, *Adaptive Sliding Mode Control for Attitude and Altitude System of a Quadcopter UAV via Neural Network*, IEEE, February 2021.
- [17] Davide Invernizzi, Marco Lovera, *Geometric tracking control of thrust vectoring UAVs*, Automatica, March 2017.
- [18] Elisa Capello, Giorgio Guglieri, Daniele Sartori, *Performance Evaluation of an L1 Adaptive Controller for Wing-Body Rock Suppression*, Journal of Guidance, Control and Dynamics, VOL.35, December 2012.
- [19] Naira Hovakimyan, Chengyu Cao, Evgeny Kharisov, Enric Xargay, Irene M. Gregory, *L1 Adaptive Control for Safety-Critical Systems*, IEEE Control Systems Magazine, October 2011.
- [20] Elisa Capello, *Flight Simulation*, Slides 02IIQMT/01PETMT, 2024.
- [21] Luis Martins, Carlos Cardeira, Paulo Oliveira, *Linear Quadratic Regulator for Trajectory Tracking of a Quadrotor*, Instituto Superior Tecnico, Universidade de Lisboa, Portugal.
- [22] J. V. Candy, *Model Reference Adaptive Control (MRAC) for Additive Manufacturing*, Lawrence Livermore National Laboratory, June 2023.
- [23] Xin Lu, Qingbo Geng, Rigen Mo, *New Composite MRAC with Modification for Fixed-wing UAV*, IEEE, 2016.
- [24] Iswanto, Alfian Ma'arif, Oyas Wahyunggoro, Adha Imam Cahyadi, *Artificial Potential Field Algorithm Implementation for Quadrotor Path Planning*, International Journal of Advanced Computer Science and Applications, Vol.10, 2019.
- [25] Aminurrashid Noordin, Mohd Ariffanan Mohd Basri, Zaharuddin Mohamed, *Real-Time Implementation of an Adaptive PID Controller for the Quadrotor MAV Embedded Flight Control System*, MDPI Journal, January 2023.
- [26] Randal W. Beard, *Quadrotor Dynamics and Control* Brigham Young University, 2008.
- [27] Richard S. Sutton, Andrew G. Barto, *Reinforcement Learning: An Introduction* The MIT Press Cambridge, 2012.