



UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA

Corso di Laurea Triennale in Ingegneria
Informatica e dell'Automazione

**STUDIO E SIMULAZIONE DI ALGORITMI DI
CONTROLLO PER DRONI**

*STUDY AND SIMULATION OF CONTROL
ALGORITHMS FOR DRONES*

Relatore: Chiar.mo
Prof. Ippoliti Gianluca
Correlatore: Chiar.mo
Prof. Orlando Giuseppe

Candidato:
Masella Giacinto

A.A. 2018/2019

INDICE

INTRODUZIONE	4
STRUTTURA DI UN QUADRICOTTERO	5
ARCHITETTURA PX4	8
QGROUNDCONTROL	10
AMBIENTE DI SIMULAZIONE	11
JMAVSIM	13
ALGORITMI DI CONTROLLO	14
PID	15
AZIONE PROPORZIONALE	16
AZIONE INTEGRALE	17
AZIONE DERIVATIVA	17
POSITION CONTROLLER	18
ALTITUDE CONTROLLER	18
HORIZONTAL CONTROLLER	19
ATTITUDE CONTROLLER	20
SVOLGIMENTO DEL PROGETTO	22
CREAZIONE MODELLO DEL QUADRICOTTERO	22
REGOLAZIONE DEI PID	24
SIMULAZIONI DI VOLO	27
SIMULAZIONE MANUAL MODE	27
SIMULAZIONE ALTITUDE MODE	31
SIMULAZIONE POSITION MODE	33
CONCLUSIONE E SVILUPPI FUTURI	35
Bibliografia	37

INTRODUZIONE

Il drone è un oggetto volante appartenente alla categoria degli aeromobili a pilotaggio remoto (UAV). Sono dispositivi in grado di volare senza la presenza di un pilota a bordo e possono essere controllati a distanza tramite una stazione di controllo posta a terra oppure attraverso un radiocomando.

I primi esemplari di drone senza pilota furono realizzati durante gli anni della Prima Guerra Mondiale per scopi militari. In particolare, sono da ricordare l'*Aerial Target* e la *Bomba Volante*, realizzati intorno l'anno 1916, e utilizzati con l'unico scopo di bombardare.

Negli ultimi decenni questi mezzi hanno visto una vera e propria esplosione di interesse da parte dei consumatori. Oggi, infatti, i droni vengono utilizzati non solo per scopi militari ma anche in altri ambiti.

Tra i principali impieghi troviamo:

- Riprese e foto aeree
- Fotogrammetria, una tecnica in grado di acquisire posizione e forma di un oggetto in tre dimensioni
- Agricoltura di precisione, grazie al quale attraverso delle camere multispettrali è possibile seguire l'avanzamento e lo stato di salute della vegetazione di interesse
- Ispezione
- Missioni di ricerca e recupero
- Trasporto

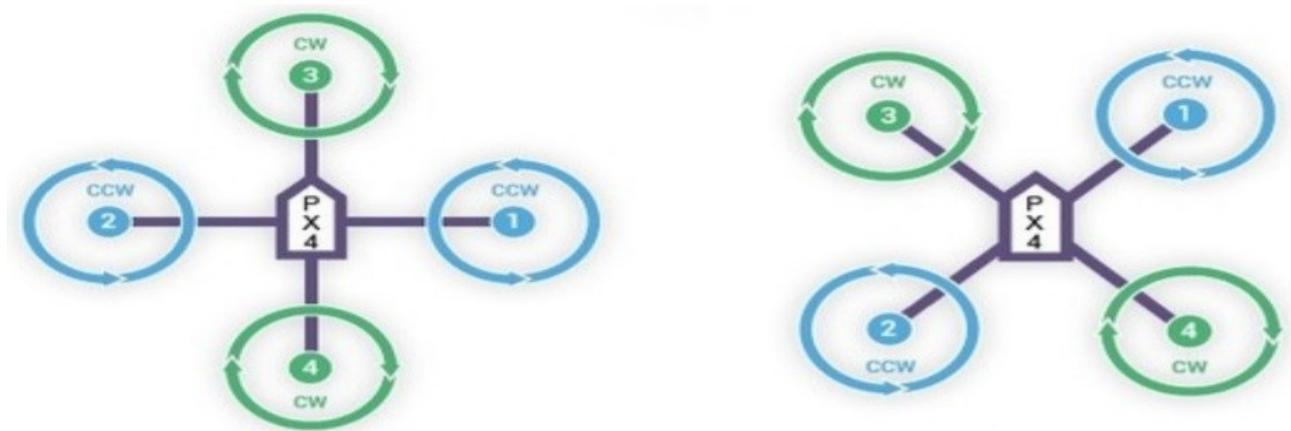
Il termine drone non fa riferimento ad un unico modello ma bensì a una vasta gamma di aeromobili con caratteristiche e forme diverse.

Nel mio lavoro mi sono occupato principalmente di un particolare tipo di drone definito quadricottero, composto come dice il nome stesso da quattro motori.

STRUTTURA DI UN QUADRICOTTERO

Un quadricottero è un multirottore UAV (Unmanned Aerial Vehicle) equipaggiato con quattro rotori, di cui due girano in senso orario (CW) e gli altri due in senso antiorario (CCW).

Ogni motore è collegato ad un'elica che, a seconda del modello, possono essere disposti in configurazione (x) o (+) come mostrato nella figura seguente:



Nella configurazione (+) ogni motore è disposto nella direzione dei rispettivi assi di movimento x-y. Questo implica che ad un aumento di spinta di un singolo motore verrà generata una coppia tale da far spostare il drone in quella specifica direzione.

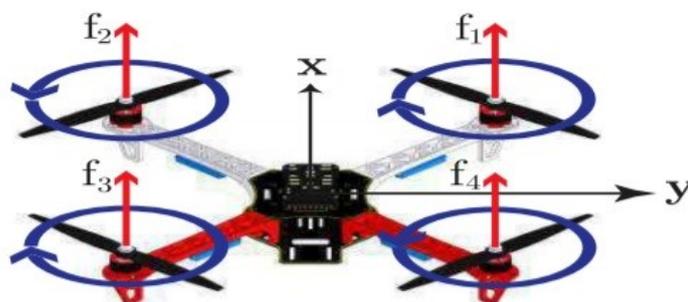
La configurazione (x) risulta essere più complessa in quanto il movimento lungo gli assi x-y è ottenuto dal conseguente aumento di spinta di due motori adiacenti. Per questa ragione tale tipologia di configurazione è ritenuta più stabile.

Il movimento di un quadricottero è controllato variando la velocità dei rispettivi motori per ottenere la spinta desiderata.

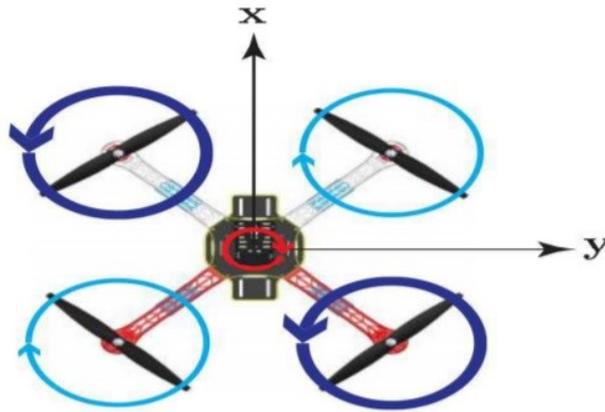
Esso può compiere 4 differenti movimenti nello spazio:

- Movimento sull'asse z (su/giù)
- Movimento sull'asse x (avanti/indietro)
- Movimento sull'asse y (destra/sinistra)
- Rotazione sull'asse z (orario/antiorario)

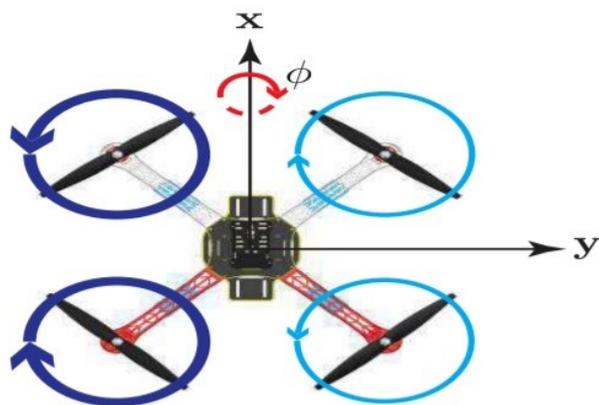
Il movimento verticale è ottenuto aumentando e diminuendo contemporaneamente la velocità di tutti e quattro i motori in maniera tale da vincere la forza peso del drone. Quando la spinta eguaglia esattamente la forza peso, il drone resterà sospeso in aria in una condizione di equilibrio (*hovering*).



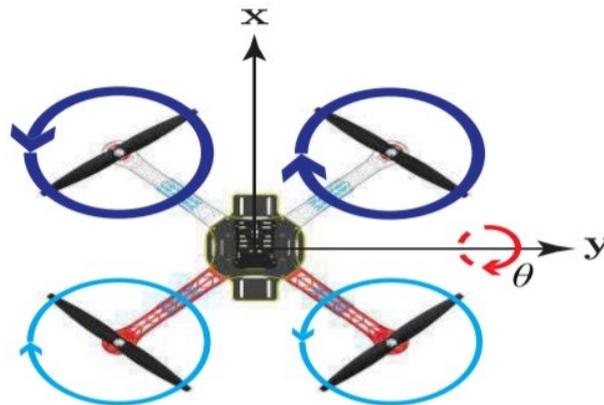
La rotazione antioraria sull'asse z è ottenuta aumentando la velocità dei motori che ruotano in senso orario e diminuendo la velocità dei motori che ruotano in senso antiorario.



Il movimento sull'asse x è ottenuto aumentando la velocità dei motori a destra, generando uno spostamento verso sinistra, oppure viceversa aumentando la velocità dei motori a sinistra per generare uno spostamento verso destra.



Infine, il movimento sull'asse y è ottenuto aumentando la velocità dei motori posteriori e diminuendo la velocità dei motori anteriori o viceversa. Questo permetterà al drone di compiere movimenti avanti/indietro.



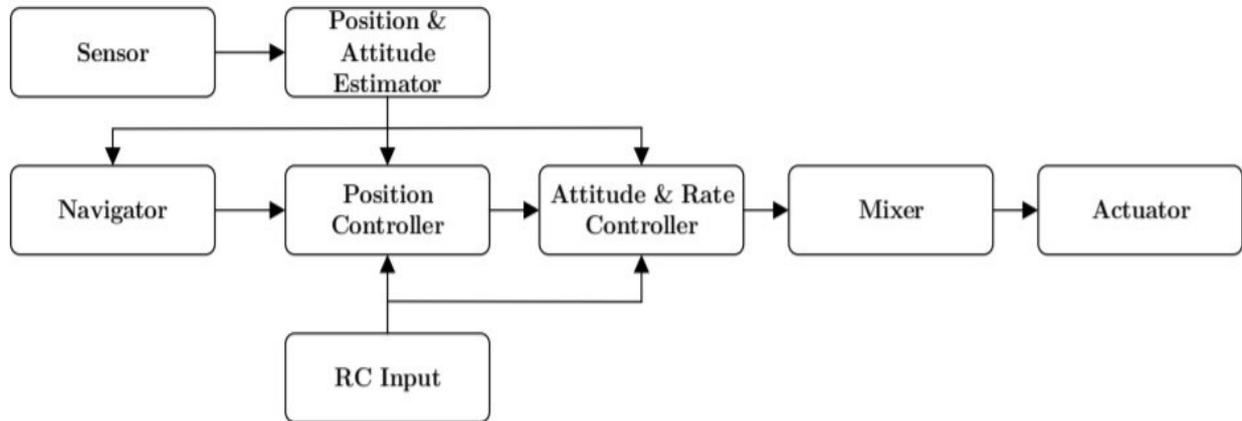
ARCHITETTURA PX4

Il quadricottero per poter effettuare i movimenti discussi nel capitolo precedente necessita di un sistema di controllo. Questo sistema di controllo è implementato nel firmware della PX4 installato sulla flight control board. Il PX4 è uno tra i migliori software di autopilotaggio open source ed è supportato e aggiornato da una comunità molto ampia.

Esso è diviso in due strati principali:

- *Flight Stack*
- *Middleware*

La Flight Stack è suddivisa in moduli come riportato nella figura seguente. Ognuno di essi svolge una funzione ben precisa.



Il modulo *Estimator* riceve in ingresso i dati letti dai sensori, li elabora attraverso un algoritmo, il Filtro di Kalman Esteso (EFK), per produrre una stima dello stato attuale del veicolo che servirà al modulo *Position Controller* e *Attitude Controller*. Attraverso il radiocomando è possibile fornire, al modulo *Position Controller* e *Attitude Controller*, rispettivamente la posizione e l'orientamento desiderato del veicolo oppure tramite il modulo *Navigator*, di solito una GCS (*Ground Control Station*) installata solitamente su di un computer, fornire una serie di comandi di navigazione desiderati. Questi comandi vengono processati dai vari controllori che, calcolano l'azione necessaria per portare il nostro veicolo dallo stato attuale verso quello desiderato. L'azione necessaria per portare il sistema verso lo stato desiderato viene fornita al *Mixer* che traduce questi segnali in comandi per ogni singolo motore.

Il *Middleware*, invece, si occupa della comunicazione interna dei moduli, attraverso l'utilizzo del protocollo uORB, e della comunicazione tra il firmware PX4, la GCS installata sul PC e all'occorrenza un simulatore attraverso il protocollo MAVlink.

QGROUNDCONTROL

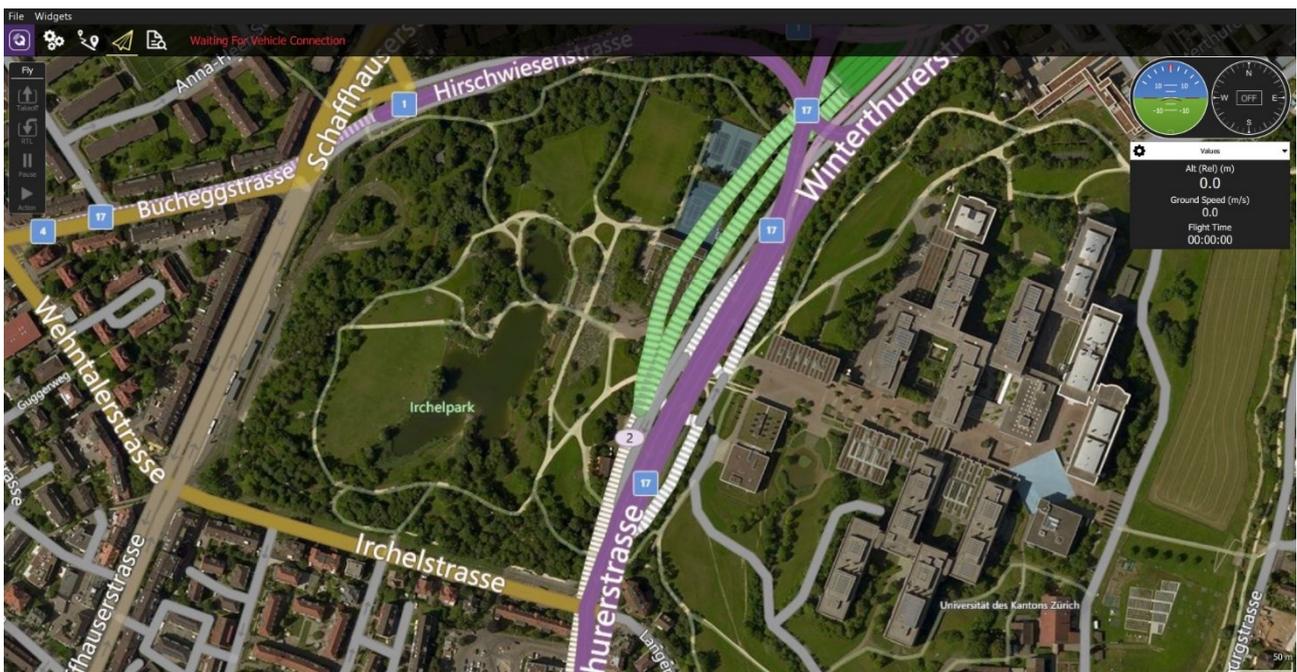
QGroundControl è un software che funge da stazione di controllo a terra.

Il software è compatibile con le principali piattaforme quali Windows, Linux, Mac, Android e IOS.

Supporta molteplici software di autopilotaggio in particolare PX4 e ArduPilot. Essi comunicano tramite protocollo MAVLink.

QGroundControl offre una molteplicità di funzioni. Le più importanti sono:

- Creare o pianificare missioni in maniera autonoma.
- Visualizzare in tempo reale dei dati di volo.
- Possibilità di controllare molteplici droni contemporaneamente.
- Pilotare il veicolo attraverso un joystick.
- Visualizzare in tempo reale le immagini di eventuali videocamere poste sul veicolo.
- Analizzare i cosiddetti log di volo per verificare il corretto funzionamento del firmware.



Nella figura in alto è rappresentato la schermata principale di QGroundControl. In alto a sinistra cliccando sul logo del software colorato in viola è possibile accedere alle impostazioni generali del software; in questa sezione è possibile impostare le unità di misura della distanza, dell'area, della velocità e della temperatura. Inoltre, è possibile modificare l'aspetto della mappa scegliendo la tipologia e il provider che la fornisce. Un'altra funzione molto utile è la possibilità di attivare da questo pannello un joystick virtuale che fungerà da radiocomando per controllare i movimenti del veicolo. Cliccando sul simbolo dell'ingranaggio è possibile visualizzare lo stato dei sensori, della batteria, di un eventuale fotocamera o videocamera installata, inoltre è possibile scegliere la configurazione del nostro veicolo, calibrare i vari sensori, il radiocomando, assegnare le modalità di volo, settare i parametri dei PID e altri parametri di sistema. Cliccando a destra dell'ingranaggio si accede al pannello per pianificare una missione attraverso dei waypoints.

AMBIENTE DI SIMULAZIONE

Per simulazione si intende la possibilità di riprodurre il comportamento di un sistema reale attraverso un software.

Nel nostro caso il firmware PX4 è in grado di controllare il modello di un veicolo tramite il computer in un mondo simulato molto simile al mondo reale. È possibile interagire con il veicolo attraverso il software QGroundControl o tramite un radiocomando/joystick.

La simulazione è un metodo molto facile e rapido per testare le modifiche apportate sul firmware prima ancora di provarle in un volo reale oppure quando non si è ancora muniti di un veicolo.

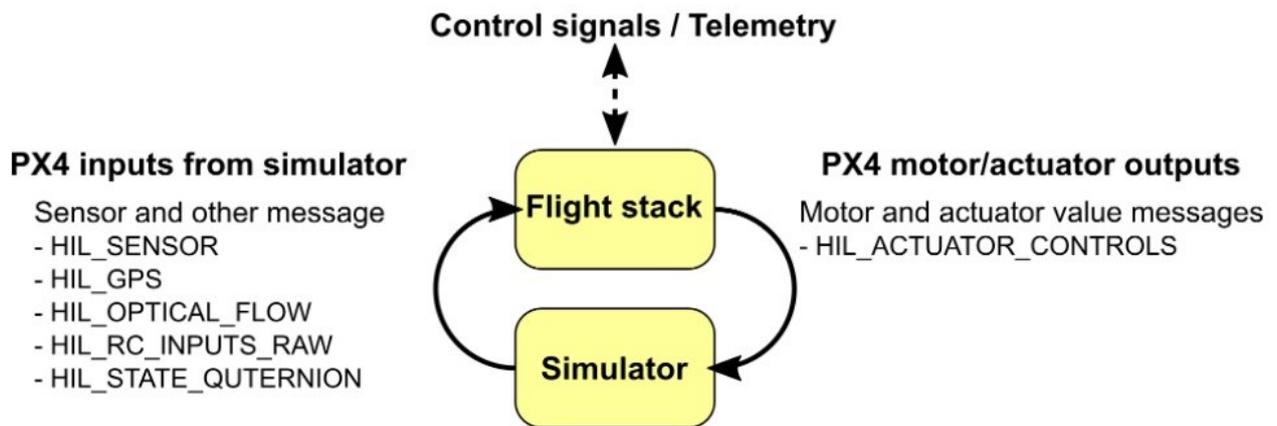
PX4 supporta due tipologie di simulazioni:

- Software in the loop (SITL) in cui tutto il codice della flight stack viene riprodotto sul processore del pc.
- Hardware in the loop (HITL) in cui gli algoritmi di controllo presenti nella flight stack sono riprodotti su una scheda di controllo reale.

Per quanto riguarda i simulatori, molti sono compatibili con il firmware PX4 tra cui: Gazebo, jMAVSim, AirSim, XPlane.

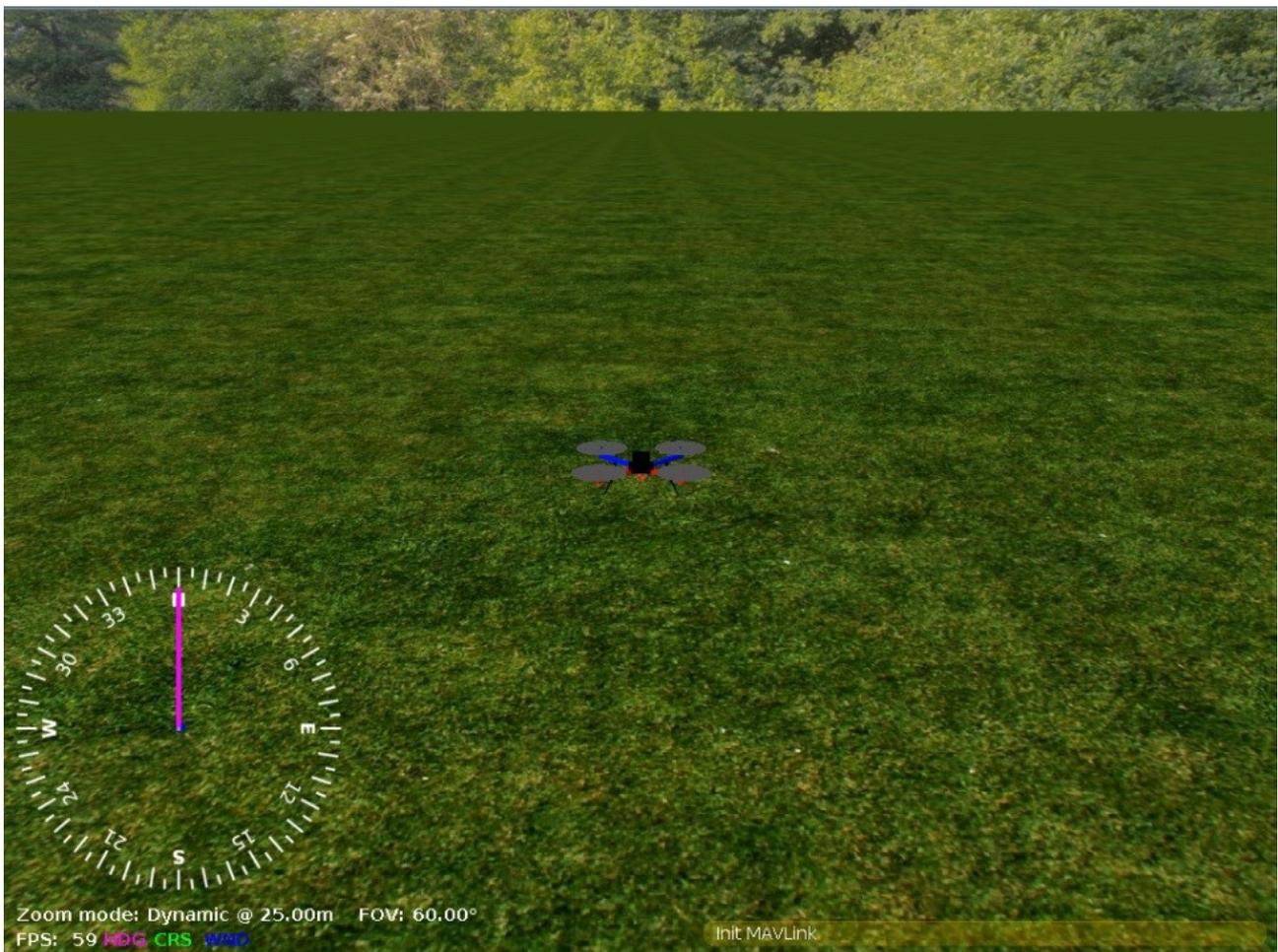
Tutti questi simulatori comunicano con il firmware PX4 attraverso il Simulator MAVLink API. La comunicazione avviene tramite scambio di messaggi tra i dati di tutti i sensori proveniente dal veicolo simulato verso il firmware e restituisce i valori dei motori e attuatori prelevati dal codice dei controllori e li applica al veicolo simulato.

Il meccanismo di questo sistema è rappresentato nell'immagine sottostante.



JMAVSim

JMAVSim è un simulatore per droni molto semplice e intuitivo ed è compatibile con il firmware PX4 e QGroundControl. È possibile testare il veicolo anche in presenza di disturbi. Ad esempio, attraverso alcuni parametri è possibile abilitare la presenza del vento nel mondo simulato specificandone la direzione e l'intensità. È possibile inoltre simulare possibili malfunzionamenti, come la perdita del segnale GPS o del radiocomando, livelli di batteria bassa e tante altre cose.



Questo simulatore è programmato interamente in JAVA utilizzando varie classi che permettono di riprodurre le dinamiche di volo reali.

Tra le classi di maggiore interesse si hanno:

- KinematicObject, classe adibita alla gestione della cinematica.
- DynamicObjet, adibita alla gestione della dinamica.
- Rotor, adibita alla creazione del modello del rotore.
- Quadcopter, adibita alla creazione del modello del quadricottero.
- Simulator, classe principale che richiama tutte le altre per creare l'ambiente di simulazione.

ALGORITMI DI CONTROLLO

Dopo aver dato una panoramica sul funzionamento dinamico e sul firmware installato sul nostro quadricottero possiamo spiegare il funzionamento degli algoritmi di controllo.

Gli algoritmi di controllo presenti nel firmware PX4 sono raggruppati principalmente in 2 categorie. Essi vengono richiamati in base alla modalità di volo selezionata dal pilota o tramite la stazione di controllo posta a terra.

PID

Prima di descrivere gli algoritmi di controllo, dato che essi sono implementati tramite regolatori PID posti in cascata, è doveroso procedere con la spiegazione di questi ultimi.

Il regolatore PID (Proporzionale-Integrativo-Derivativo) è un algoritmo di controllo dotato di una struttura prefissata, e viene tarato in base al suo utilizzo attraverso la modifica di alcuni suoi parametri. Per la sua semplicità di utilizzo ed efficacia viene utilizzato nella maggior parte delle applicazioni industriali.

Dal punto di vista matematico, il regolatore è un sistema dinamico che elabora un segnale in ingresso, detto errore, e fornisce in uscita un segnale di controllo.

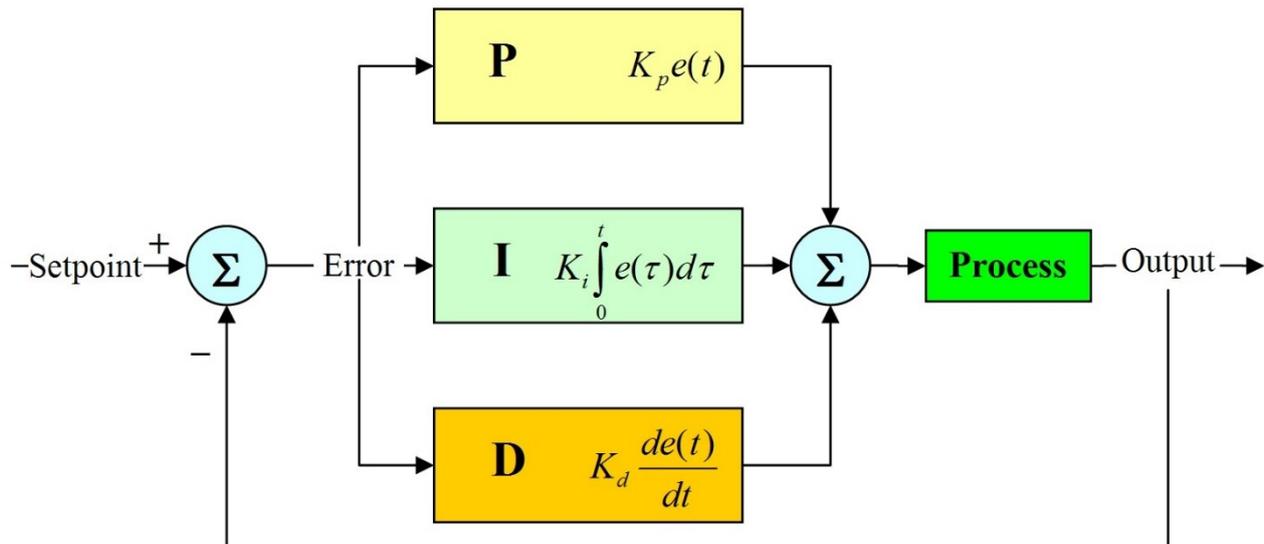
La variabile errore è data dalla differenza tra il riferimento e la variabile controllata.

Il controllore PID nella sua forma base è composto dalla somma di tre termini:

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

I parametri K_P , K_I , K_D sono chiamati gradi di libertà del sistema e sono i coefficienti che devono essere tarati per il corretto funzionamento del sistema di controllo.

La figura sotto riporta lo schema a blocchi di un semplice regolatore PID:



AZIONE PROPORZIONALE

L'azione proporzionale è ottenuta moltiplicando il segnale errore con un'opportuna costante:

$$u_p = K_p \cdot e(t)$$

È possibile regolare un processo instabile tramite questo controllore ma non è possibile garantire che l'errore converga a zero: questo perché un'azione di controllo è possibile solo se l'errore è diverso da zero

AZIONE INTEGRALE

L'azione integrale è ottenuta moltiplicando l'integrale nel tempo dell'errore per la costante K_I :

$$u_I = K_I \cdot \int_0^t e(\tau) \cdot d\tau$$

L'azione integrale fa sì che il regolatore abbia memoria dei valori passati del segnale errore. Questa proprietà dà al regolatore la capacità di portare il processo al punto di riferimento richiesto, dove la sola azione proporzionale risulterebbe nulla.

AZIONE DERIVATIVA

Per migliorare ulteriormente le prestazioni del regolatore si aggiunge l'azione derivativa:

$$u_D = K_D \cdot \frac{de(t)}{dt}$$

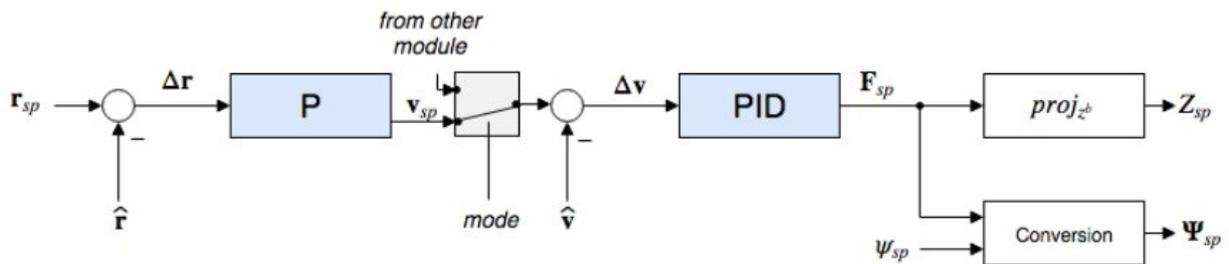
L'idea è quella di compensare le variazioni dell'errore: se notiamo che l'errore aumenta, l'azione derivativa cercherà di compensare questo aumento sulla base della sua velocità di cambiamento senza aspettare che l'errore diventi significativo (azione proporzionale) oppure persista per lungo tempo (azione integrale). Di solito l'azione derivativa è trascurata in quanto brusche variazioni del segnale di riferimento farebbero tendere la derivata del segnale errore all'infinito con il conseguente rischio di rompere gli attuatori fisici.

POSITION CONTROLLER

Il controllore di posizione gestisce separatamente il controllo sull'asse z (altitude controller) e il controllo sul piano x-y (horizontal controller).

Entrambi i controlli sono effettuati attraverso due controllori posti in cascata:

1. un controllore di tipo proporzionale per gestire le posizioni;
2. un controllo di tipo proporzionale, integrativo e derivativo per gestire le velocità.



\mathbf{r} - position	Δ - difference
\mathbf{v} - velocity	Z - body vertical thrust
Ψ - attitude	$\widehat{(x)}$ - estimated value (of x)
\mathbf{F} - thrust	$(x)_{sp}$ - setpoint (of x)
	$proj_{z^b}$ - vector projected onto body Z-axis

ALTITUDE CONTROLLER

Il controllo dell'altitudine avviene prendendo come ingressi la posizione stimata e la posizione desiderata.

La posizione stimata è fornita da GPS, barometro e magnetometro.

La posizione desiderata è data dal radiocomando oppure dalla stazione di controllo a terra.

La differenza tra le due variabili fornisce la variabile errore che dopo essere stata processata dal regolatore di tipo proporzionale restituisce la velocità necessaria che viene confrontata con la velocità attuale. Adesso questa velocità viene processata da un controllore di tipo PID che fornisce in uscita la spinta necessaria per ottenere l'altitudine desiderata.

HORIZONTAL CONTROLLER

Questo controllore si occupa di regolare la posizione e la velocità orizzontale di volo. Esso si divide in un controllore di tipo proporzionale per gestire la posizione e un controllore di tipo proporzionale, integrativo e derivativo per gestire le velocità, entrambi posti in cascata.

La prima parte è un controllo di posizione che prende in entrata la differenza tra la posizione desiderata, imposta dal radiocomando o dalla GCS, e la posizione corrente, ottenuta attraverso lo stimatore, per poi moltiplicarla per un blocco di tipo proporzionale P.

L'uscita di questo blocco sarà un vettore bidimensionale avente le velocità desiderate sugli assi x-y.

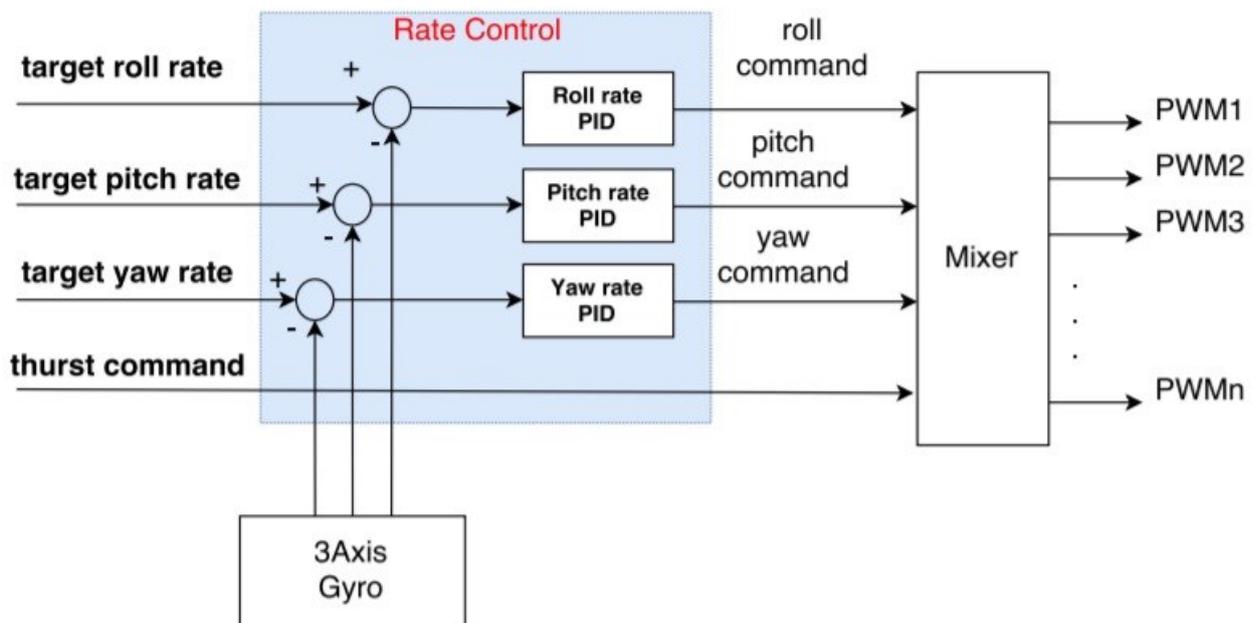
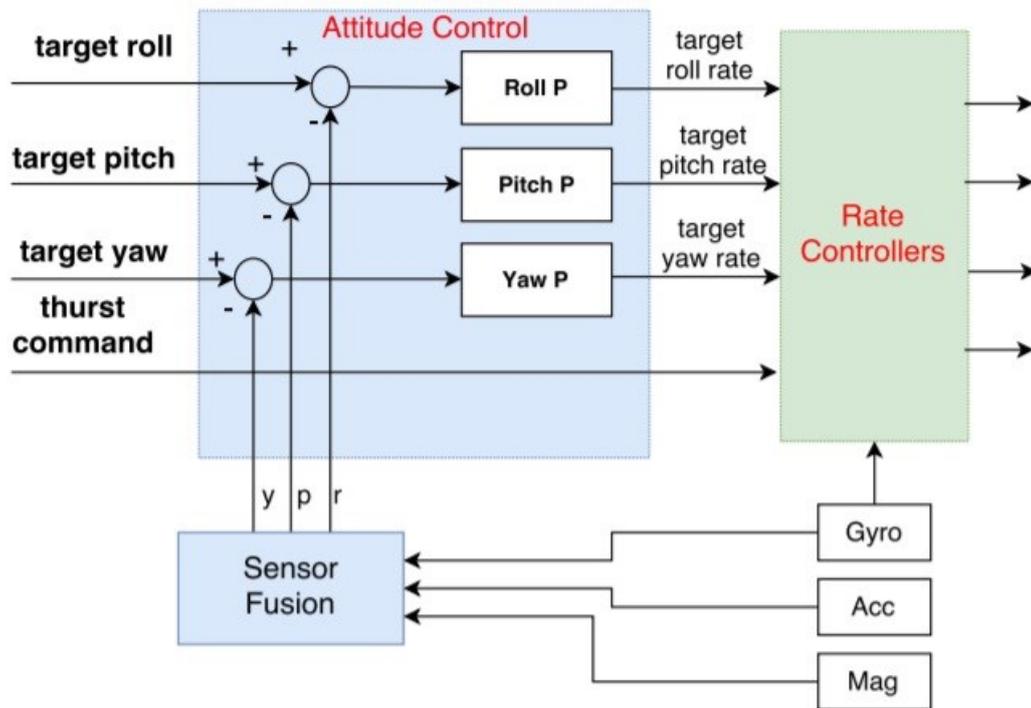
La seconda parte del controllore prende in ingresso la differenza tra la velocità desiderata e quella corrente e la processa attraverso un blocco di tipo proporzionale, integrativo e derivativo per ottenere un vettore contenente le angolazioni necessarie per portare il drone nella posizione desiderata e che saranno utilizzate dal controllore di assetto

ATTITUDE CONTROLLER

Anche il controllore di assetto è diviso in due parti, un controllo sull'angolo e un controllo sulla velocità angolare. Entrambi questi controllori sono rispettivamente di tipo proporzionale e di tipo PID e sono posti in cascata.

Il controllo dell'angolo avviene prendendo come ingresso i dati forniti dal controllore di posizione come variabili desiderate, e i dati forniti dagli accelerometri e giroscopi come variabili attuali. La differenza tra l'angolazione desiderata e l'angolazione stimata verrà processata da un blocco di tipo proporzionale. Il risultato di questo blocco corrisponde a una velocità angolare desiderata che viene processata a sua volta da un blocco di tipo PID. L'uscita è l'accelerazione angolare necessaria sui tre assi di rotazione per portare il veicolo nella posizione e nell'angolazione desiderata. Queste accelerazioni verranno in seguito tradotte dal Mixer in comandi compatibili per i motori. In pratica il Mixer trasformerà questi valori in segnali PWM che attraverso gli ESC regoleranno la rotazione dei motori per far compiere al veicolo il movimento desiderato.

Nelle figure seguenti sono mostrati rispettivamente gli schemi a blocchi del controllo sull'angolo e sulla velocità angolare:



SVOLGIMENTO DEL PROGETTO

Il principale obbiettivo di questa tesi è stato quello di simulare il volo del quadricottero per verificarne l'efficienza degli algoritmi di controllo discussi nei capitoli precedenti e riportandone i risultati finali attraverso una rappresentazione grafica.

Ho utilizzato i software discussi precedentemente quali JMAVSIM, QGROUNDCONTROL e il firmware PX4.

CREAZIONE MODELLO DEL QUADRICOTTERO

Dal momento che il software JMAVSIM carica un modello predefinito di veicolo, il primo passo è stato creare un nuovo modello con i dati del quadricottero che ho utilizzato per la simulazione.

I parametri utilizzati per la creazione:

1. Massa del quadricottero
2. Lunghezza del braccio
3. Momenti di inerzia sui tre assi x, y e z
4. Coefficiente di spinta (thrust)
5. Coefficiente di coppia (torque)
6. Coefficiente di resistenza del fluido immerso (drag)

Per fare ciò ho utilizzato la classe `AbstractMulticopter`, che si occupa della creazione del veicolo.

I dati modificati del drone sono:

Massa	Braccio	Momenti di inerzia			Thrust	Torque	Drag
		x	y	z			
0.78 Kg	0.08 m	0.0030	0.0030	0.0047	5.425 N	0.434 N·m	0.506

Nella figura seguente è presente la parte di codice relativa alla creazione del quadricottero con i dati aggiornati:

```

1 private AbstractMulticopter buildMulticopter() {
2     Vector3d gc = new Vector3d(0.0, 0.0, 0.0); // centro di gravità
3     AbstractMulticopter vehicle = new Quadcopter(world, DEFAULT_VEHICLE_MODEL, "x", "default",
4         | 0.08, 5.425, 0.434, 0.005, gc, SHOW_GUI);
5         | // (braccio) (spinta)(coppia)
6     Matrix3d I = new Matrix3d();
7     // Momenti di Inerzia
8     I.m00 = 0.0030; // X
9     I.m11 = 0.0030; // Y
10    I.m22 = 0.0047; // Z
11    vehicle.setMomentOfInertia(I);
12    vehicle.setMass(0.78); // massa
13    vehicle.setDragMove(0.506); //coefficiente di Drag
14    SimpleSensors sensors = new SimpleSensors();
15    sensors.setGPSInterval(50);
16    sensors.setGPSDelay(200);
17    sensors.setNoise_Acc(0.05f);
18    sensors.setNoise_Gyo(0.01f);
19    sensors.setNoise_Mag(0.005f);
20    sensors.setNoise_Prs(0.1f);
21    vehicle.setSensors(sensors, getSimMillis());
22
23    return vehicle;
24 }
25

```

REGOLAZIONE DEI PID

Prima di passare alla fase di volo è opportuno settare i parametri dei regolatori PID.

Un buon settaggio implica un volo preciso e stabile, in caso contrario il veicolo avrà un comportamento instabile con il serio rischio di schiantarsi.

Tramite la nostra stazione di controllo (QGROUNDCONTROL), accedendo al pannello “tuning” è possibile settare i nostri parametri in maniera relativamente semplice.



Come mostrato in figura, è possibile modificare i vari parametri per ogni singolo asse e in aggiunta è possibile visualizzare dei grafici in tempo reale per verificarne i risultati. Il settaggio dei parametri del regolatore PID è avvenuto in particolare modificando i guadagni del solo controllore della velocità angolare, che si occupa di prendere ingresso le velocità angolari e fornire in uscita le accelerazioni necessarie per far compiere sugli assi di rollio, beccheggio e imbardata il movimento desiderato. Un buon settaggio di questo controllore influenzerà in maniera positiva gli altri controllori.

I parametri in questione sono:

- | | | |
|-----------------|------------------|----------------|
| • MC_ROLLRATE_P | • MC_PITCHRATE_P | • MC_YAWRATE_P |
| • MC_ROLLRATE_D | • MC_PITCHRATE_I | • MC_YAWRATE_I |
| • MC_ROLLRATE_I | • MC_PITCHRATE_D | • MC_YAWRATE_D |

Questi parametri sono stati configurati utilizzando la modalità MANUAL e procedendo a settare i coefficienti proporzionali e derivativi del roll e pitch e in seguito i coefficienti integrativi degli stessi. In seguito, si passa ai coefficienti dello yaw con lo stesso procedimento.

Il coefficiente proporzionale serve a limitare l'errore sul riferimento.

Inoltre, stabilisce prontezza del sistema a cambiamenti di ingresso perciò deve essere impostato a un valore elevato senza però introdurre oscillazioni.

Se a questo termine si attribuisce un valore elevato si noteranno delle oscillazioni con alte frequenze nella dinamica.

Se questo termine viene impostato con un valore relativamente basso il quadricottero risponderà ai comandi molto lentamente.

Il coefficiente derivativo serve per smorzare le oscillazioni presenti nella dinamica.

Tuttavia, se è impostato ad un valore troppo elevato i motori si surriscalderebbero eccessivamente perché il termine derivativo introduce molto rumore.

Il coefficiente integrativo ha memoria dell'errore passato.

Se è impostato ad un valore elevato introduce delle oscillazioni lente nella dinamica.

Se è impostato ad un valore basso il veicolo non manterrà l'assetto impostatogli.

Un buon settaggio di questi parametri implicherà la qualità di volo del nostro quadricottero. Esso risponderà bene ai vari input dati dal radiocomando avendo un comportamento quanto più vicino a quello di riferimento.

SIMULAZIONI DI VOLO

Dopo aver creato il modello del quadricottero e dopo aver proceduto alla taratura dei PID si può passare alla simulazione di volo del veicolo.

Le simulazioni effettuate all'interno del progetto sono:

1. Simulazione in modalità MANUAL
2. Simulazione in modalità ALTITUDE
3. Simulazione in modalità POSITION

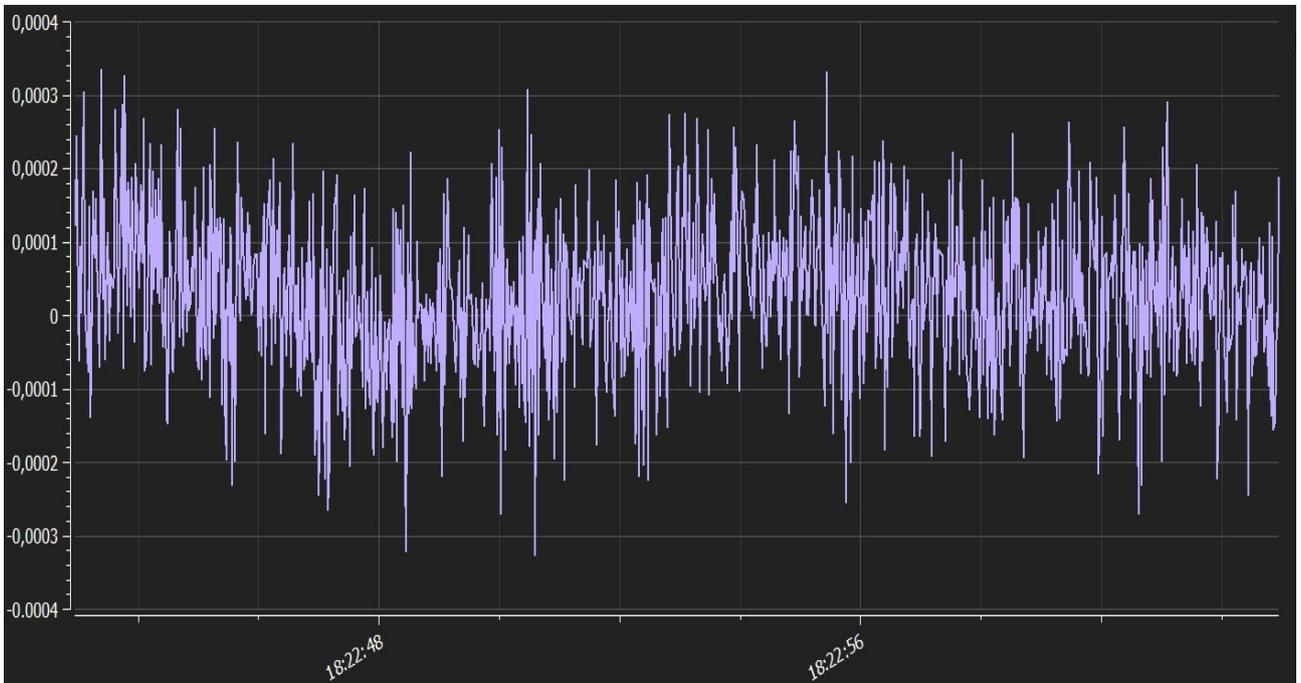
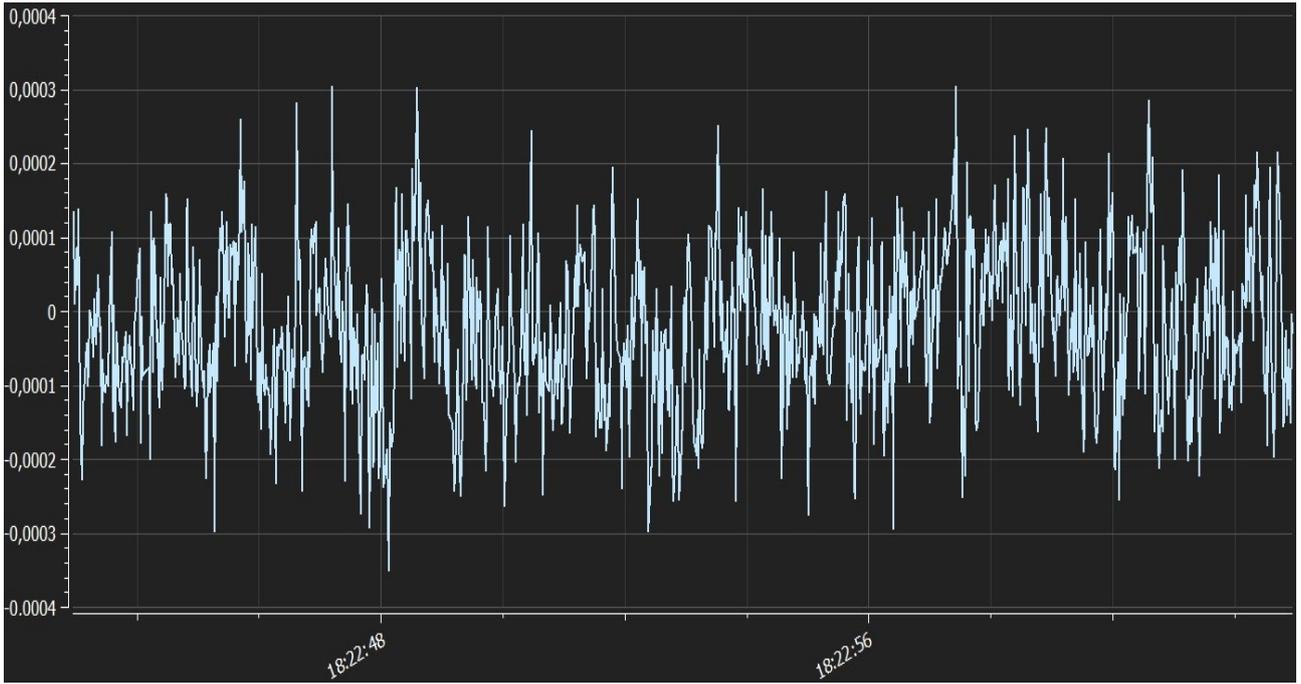
Ognuna di queste modalità di volo gestisce in maniera differente sia gli input forniti dal radiocomando e sia il controllo dei movimenti.

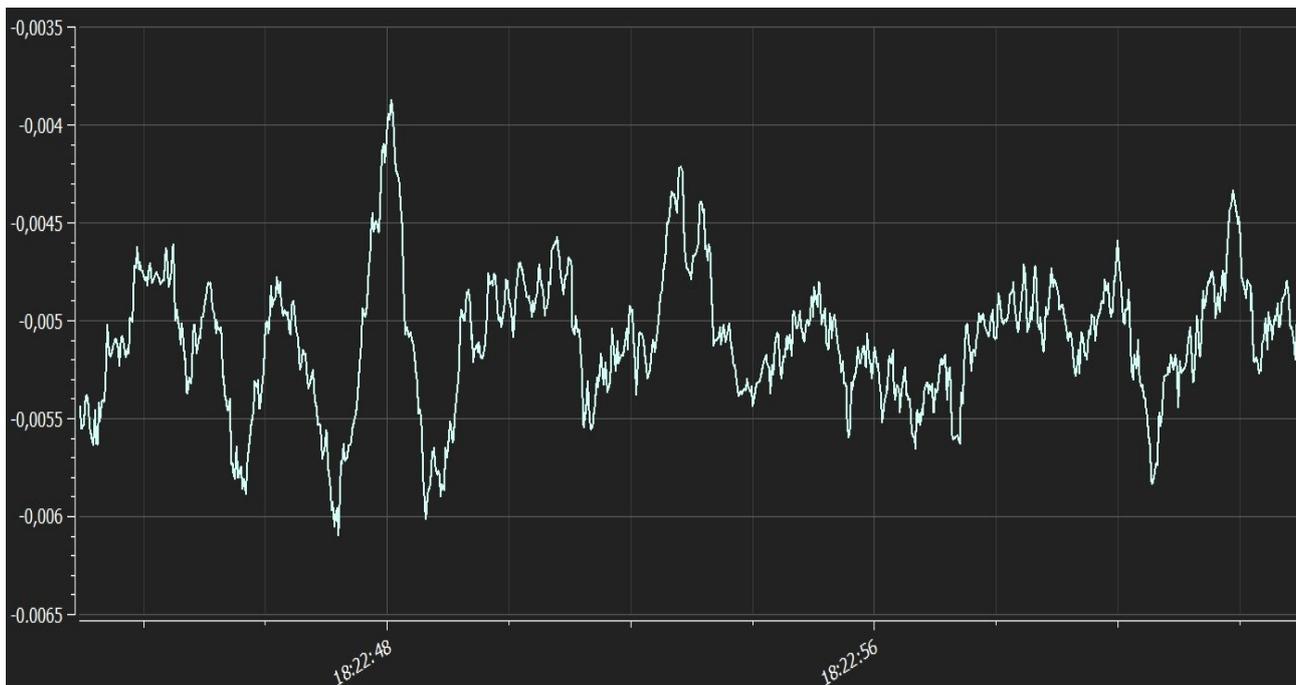
SIMULAZIONE MANUAL MODE

In questa modalità di volo il firmware controlla l'assetto del drone.

La simulazione è avvenuta in periodo temporale di 20 secondi lasciando il drone sospeso in aria senza fornirgli nessun comando.

Il controllore di assetto in questo caso dovrà mantenere a zero gli angoli di roll e pitch, e mantenere costante l'angolo di yaw.

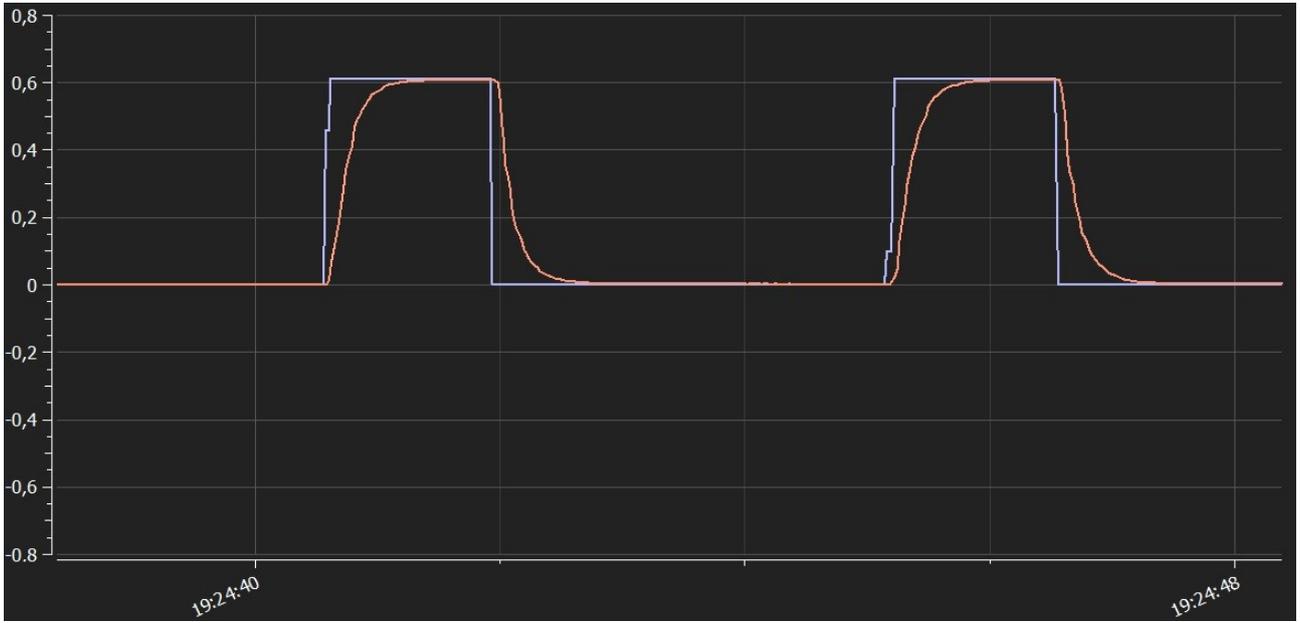
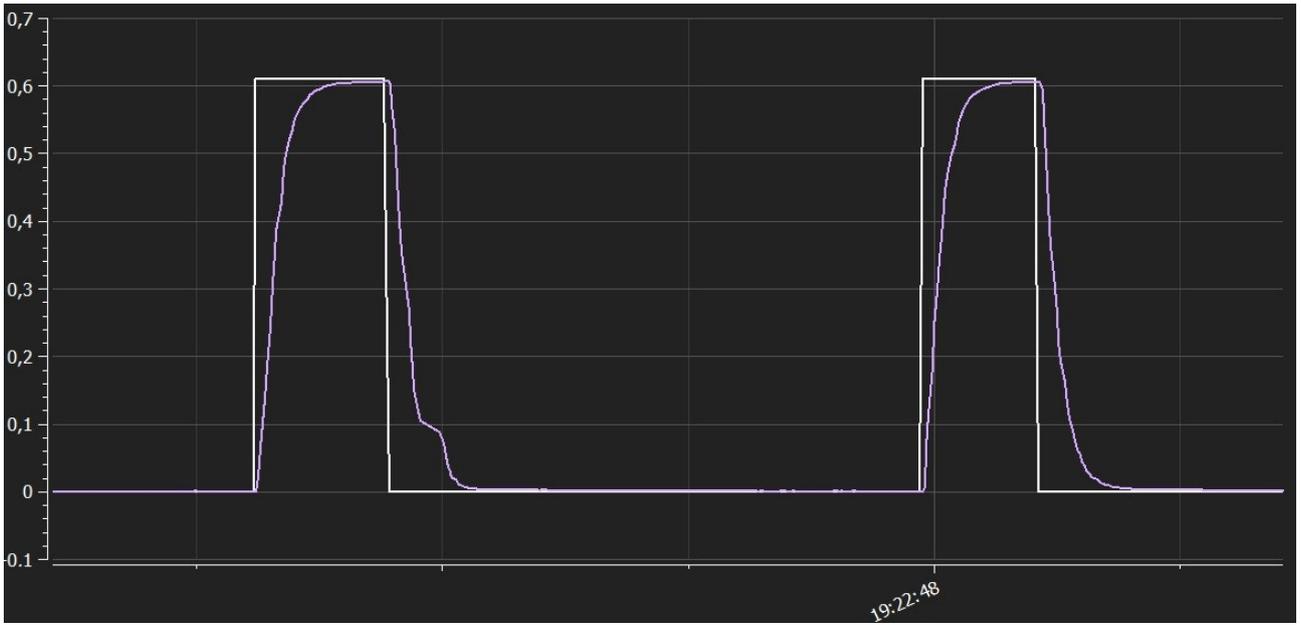


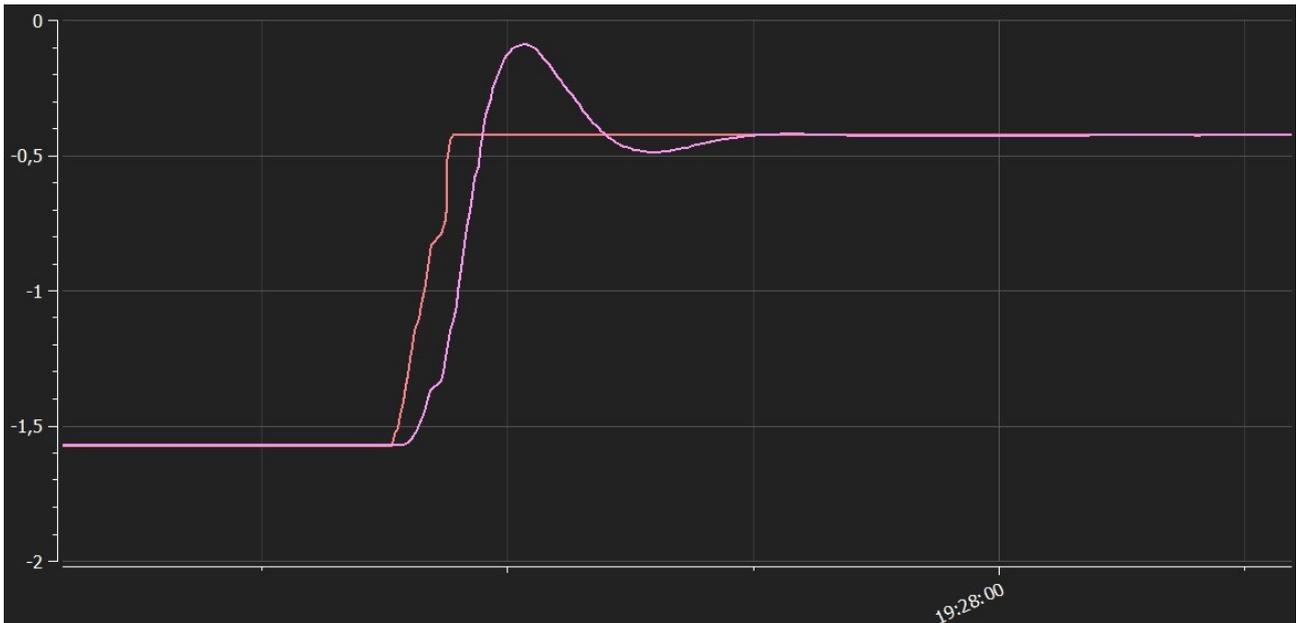


Dai grafici sopra riportati sia l'angolo di roll che il pitch si assestano a zero con un errore di circa $\pm 4 \cdot 10^{-4}$.

L'angolo di yaw si mantiene costante al valore di 0.005 con un errore di circa $\pm 10^{-3}$

Fornendo uno scalino come input si può verificare il corretto funzionamento del controllore. Il valore stimato è prossimo al valore di setpoint come riportato nei grafici seguenti:

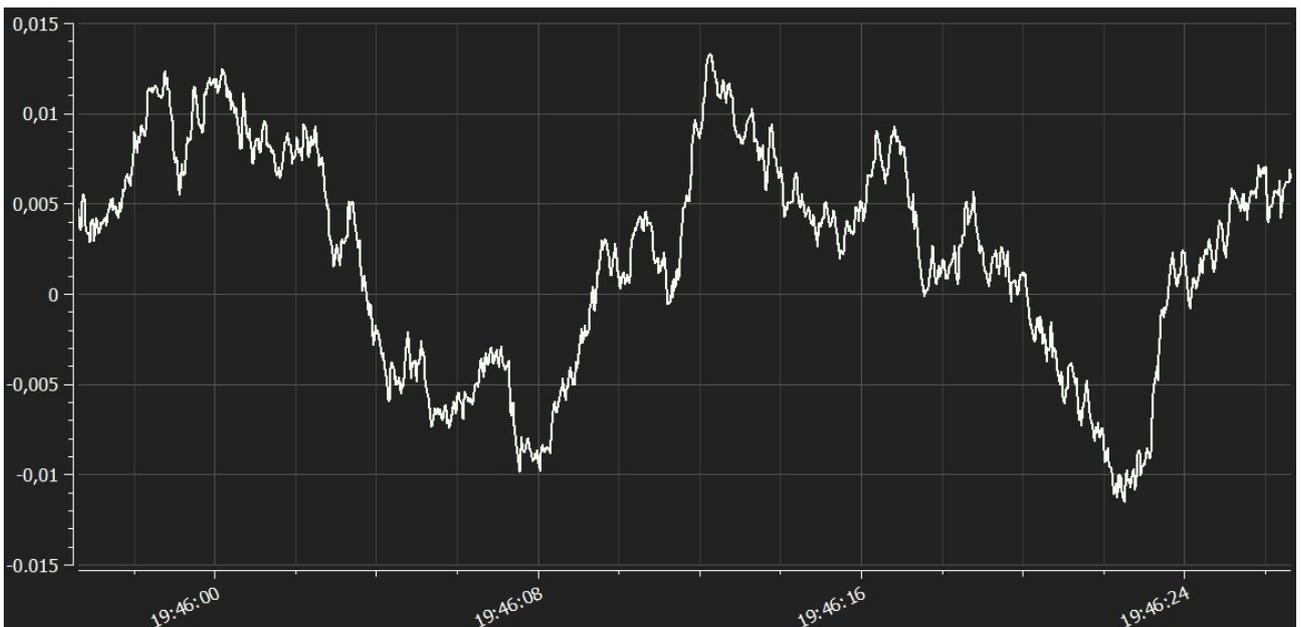
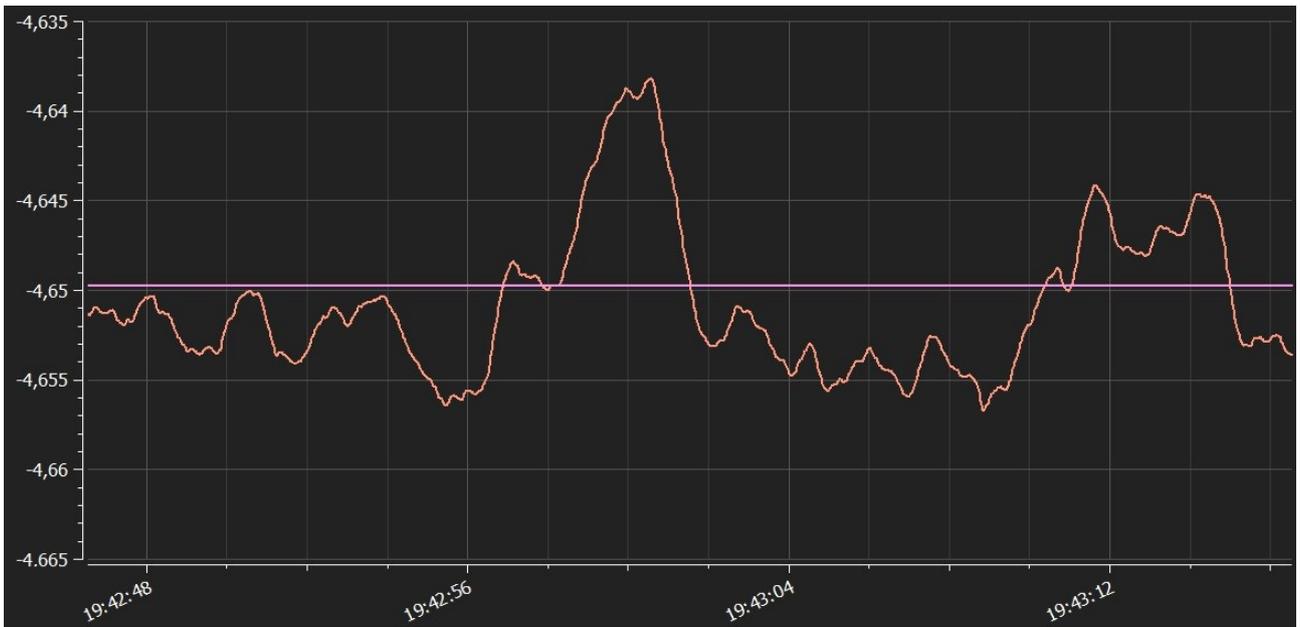




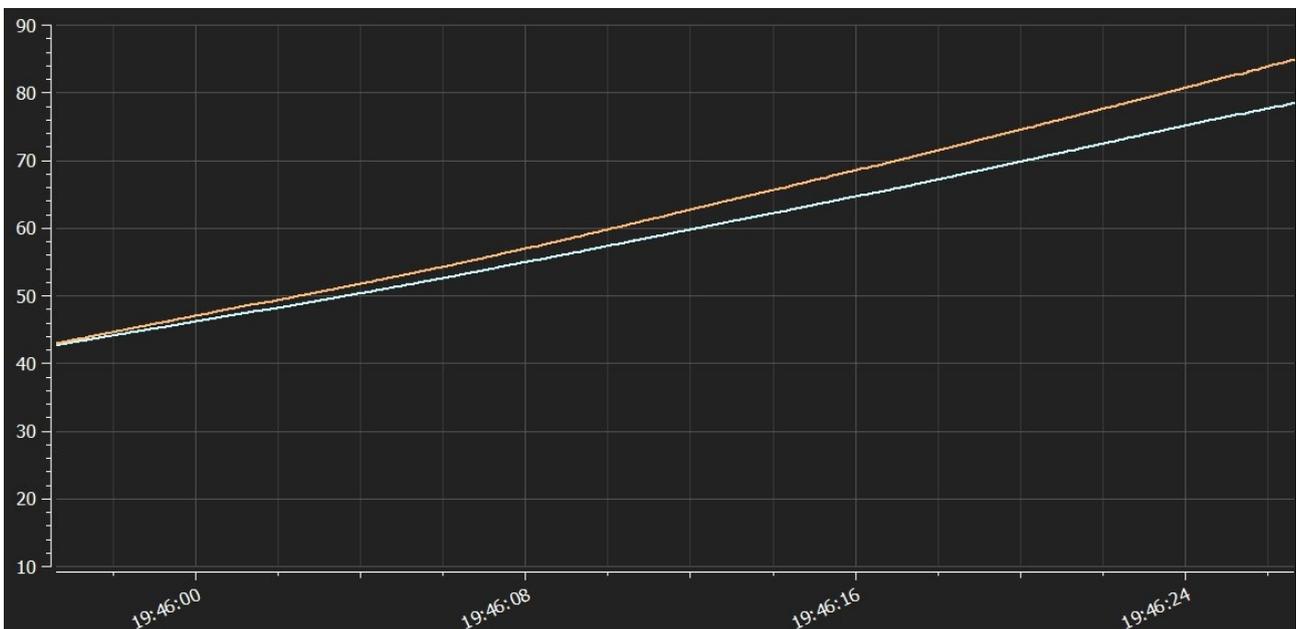
SIMULAZIONE ALTITUDE MODE

In questa modalità il controllore ha il compito di mantenere la quota imposta e di conseguenza la velocità relativa in quell'asse nulla.

La simulazione è avvenuta portando il drone ad una quota di circa 4,6 metri per 30 secondi.



Dai grafici si deduce che la quota è stata mantenuta con un errore di circa $\pm 1.5 \cdot 10^{-3}$.
La velocità verticale si è tenuta intorno lo zero con un errore di circa $\pm 1.3 \cdot 10^{-3}$.
In questa modalità il controllore non si occupa di gestire la posizione orizzontale.
Infatti, siccome nell'ambiente di simulazione è riprodotto il vento, il drone mantiene la quota ma si sposta nella direzione del vento come mostra la figura seguente:



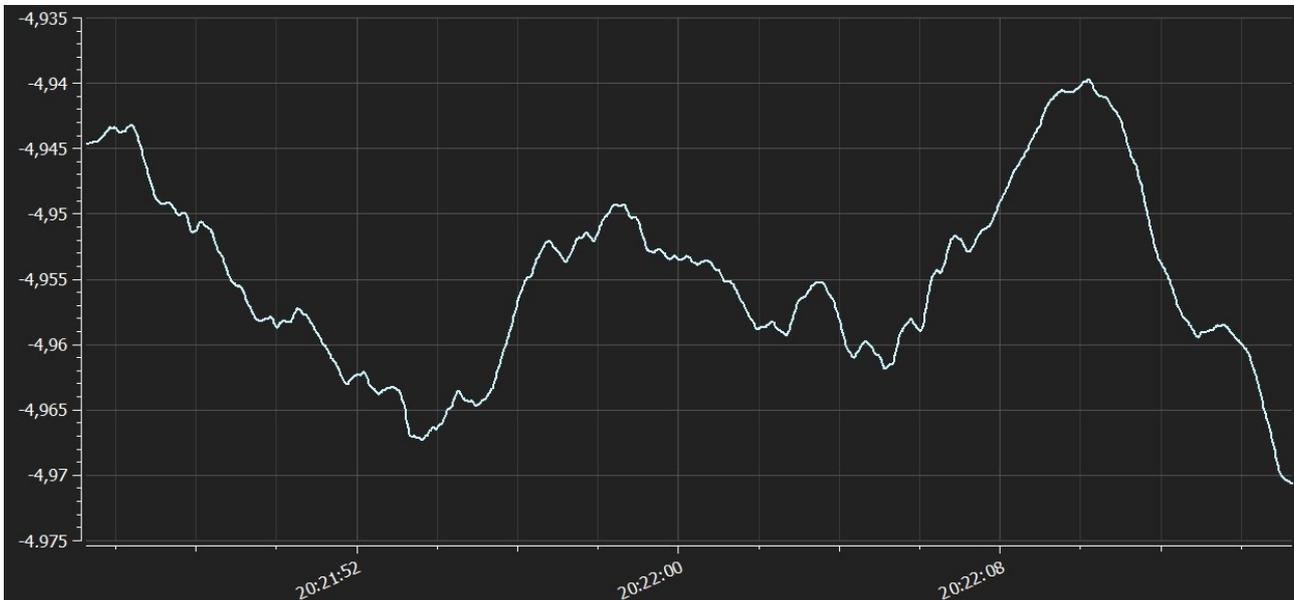
SIMULAZIONE POSITION MODE

In questa modalità il controllore gestisce sia l'altitudine che la posizione orizzontale.
Il drone manterrà la quota e anche la posizione sugli assi x e y.

La simulazione è avvenuta in un istante temporale di 30. Il drone è stato portato ad una quota di circa 5 metri.



La posizione sugli assi x (arancione) e y (viola) è stata mantenuta con un errore di circa ± 0.13



La quota è stata mantenuta sui 5 metri con un errore di ± 0.025

CONCLUSIONE E SVILUPPI FUTURI

Possiamo dedurre dai dati raccolti dalle varie simulazioni che il controllore si comporta in maniera ottimale. L'assetto, l'altitudine e la posizione sono stati controllati, in ambiente simulato, con errori relativamente bassi. Questo è stato possibile grazie ad una buona taratura dei guadagni proporzionali, integrativi e derivativi. Una buona taratura permette al drone di essere stabile in volo rispondendo positivamente anche in presenza di disturbi, purtroppo i suddetti parametri non sono universali e dunque è necessario

configurarli di volta in volta a seconda delle caratteristiche fisiche. Un pessimo settaggio porterebbe il drone ad avere comportamenti instabili con seri rischi di schiantarsi. Per questo la possibilità di poter testare il drone in un ambiente simulato è un aspetto molto importante che merita di essere approfondito.

Grazie alla simulazione è stato possibile testare il volo del quadricottero in maniera sicura senza nessun rischio. Una volta osservato la risposta in volo è possibile passare alla fase successiva caricando il firmware modificato sulla scheda di controllo reale e procedere con il volo vero e proprio.

Un'idea per proseguire questo lavoro potrebbe essere quello di approfondire l'ambiente di simulazione in maniera tale da progettare il più realistico possibile.

Inoltre, il software PX4 vanta altre funzioni non citate in questa tesi come:

- Geofence - grazie al quale è possibile creare delle regioni virtuali dove il drone può volare all'interno senza uscire al di fuori del perimetro;
- Obstacle Avoidance - che permette al drone di eseguire un percorso programmato evitando degli ostacoli;
- Collision Prevention - che può essere usato per impartire al drone il comportamento da attuare in presenza di un ostacolo non programmato.

Bibliografia

- [1] M. E. Penati e G. Bertoni, Sistemi di controllo: modellistica e tecnologie, Bologna: Zanichelli, 1989.
- [2] M. L. Corradini e G. Orlando, Fondamenti di Automatica, Bologna: Pitagora, 2002.
- [3] E. Bonello, A flight control scheme for a quadcopter: implementation and validation, Malta, 2018.
- [4] C. A. Amadi, Design and implementation of model predictive control on pixhawk flight controller, Stellenbosch, 2018.
- [5] F. Baiocchi, V. B. Mei e D. Sciarroni, Ardupilot: panoramica sul controllo, Ancona, 2015.
- [6] S. Longo, F. Merli e V. Saverino, Analisi del software di controllo della piattaforma Ardupilot, Ancona.
- [7] S. Bouadallah, Design and control of quadrotors with application to autonomous flying, Tlemcen, 2007.
- [8] D. Brescianini, M. Hehn e R. D'Andrea, Nonlinear quadcopter attitude control technical report, Zurigo, 2013.
- [9] M. Tamburini, Framework per lo sviluppo e la simulazione di algoritmi di controllo per velivoli autonomi, Pisa, 2014.
- [10] G. P. Carratelli e M. D. Duca, Controllo di un quadcopter, Napoli, 2012.
- [11] G. Testolin, Controllori pid e tecniche anti wind-up.
- [12] M. Gigli, Studio e sviluppo di modelli per simulatori di volo di quadricotteri, Ancona, 2018.
- [13] M. L. Corradini e G. Orlando, Controllo digitale di sistemi dinamici, Milano: Franco Agneti, 2005.
- [14] A. Isidori, Sistemi di controllo, Roma: Siderea, 1993.
- [15] PX4, «User guide,» 2019. [Online]. Available: <https://docs.px4.io/>.
- [16] Qgroundcontrol, «User guide,» 2019. [Online]. Available: <https://docs.qgroundcontrol.com/>.