

ALMA MATER STUDIORUM
UNIVERSITA' DI BOLOGNA

SCUOLA DI INGEGNERIA
Sede di Forlì

Corso di Laurea in
INGEGNERIA AEROSPAZIALE
Classe L-9

ELABORATO FINALE DI LAUREA
In
MECCANICA DEL VOLO

**Guida, Navigazione e Controllo di rover
terrestri attraverso sistemi GPS**

Candidato:
Chiara Paceschi

Relatore:
Emanuele Luigi de Angelis

Anno Accademico 2021/2022
Seduta di laurea: Maggio 2023

Ringraziamenti

Desidero ringraziare il Professore Emanuele Luigi de Angelis per avermi permesso di svolgere questa attività di tesi. Oltre ad avermi trasmesso profonde conoscenze durante tutto il percorso di stesura dell'elaborato, mi ha fornito preziosi ed indispensabili consigli durante lo sviluppo del modello. Desidero inoltre ringraziare l'Ing. Giulia Bertolani per avermi aiutato durante parte del percorso.

Vorrei continuare ringraziando dal profondo del mio cuore i miei genitori, mio fratello e i miei parenti per avermi sostenuto in tutto questo percorso e i miei amici per essermi stati accanto durante i miei periodi di difficoltà e di gioia.

Sommario

Il lavoro descritto nel presente elaborato è fondato sulla realizzazione di un sistema di guida autonoma di un rover terrestre attraverso dispositivi GPS. Per eseguire tale studio vengono riportate le basi della guida autonoma di velivoli e le tecnologie utilizzate. Successivamente vengono descritti gli script Matlab e il programma per l'implementazione del modello del rover e dei suoi sistemi di controllo. Le basi del modello presentato partono da un algoritmo di tipo Beam Rider per la navigazione a waypoint e una legge di controllo proporzionale di sterzo. Per passare ad un'ambiente reale si utilizza un modello di rover terrestre munito di PX4. L'implementazione del programma "rover" (in Simulink) ha permesso, tramite l'utilizzo delle coordinate fornite dal GPS, di controllare il rover tramite 3 modalità: Manual, Steering, Guided. I risultati numerici vengono validati dalla campagna sperimentale.

INDICE

1.	INTRODUZIONE	6
2.	MODELLAZIONE MATEMATICA	8
2.1	Modellazione del rover	8
2.2	Navigazione, guida e controllo	13
3	RISULTATI SPERIMENTALI.....	20
3.1	Ambiente implementativo	20
3.2	Prove sperimentali.....	31
4	CONCLUSIONI E SVILUPPI FUTURI	34

Indice delle figure

Figura 1.1:	cube flight controller, conosciuto anche come pixhawk 2.1	7
Figura 1.2:	H-RTK F9P Helical	7
Figura 2.1:	sistema di riferimento veicolo assi fissi	9
Figura 2.2:	blocco Vehicle body 3DOF.....	9
Figura 2.3:	disposizione forze per tipologia di blocco Dual.....	12
Figura 2.4:	rover con GPS RTK e pixhawk cube black	12
Figura 2.5:	rover del laboratorio C.I.R.I Aeronautica	13
Figura 2.6:	blocco navigazione, guida e controllo.....	14
Figura 2.7:	sistema di guida 'beam rider' nel sistema NED	15
Figura 2.8:	sottosistema navigazione.....	16
Figura 2.9:	iterazione che permette di assegnare il successivo WPf	16
Figura 2.10:	beam rider e sistema di controllo	18
Figura 2.11:	grafico traiettoria del rover per χ_s e χ_{br}	20
Figura 3.1:	schermata principale di Qground	21
Figura 3.2:	blocco Vehicle Attitude	22
Figura 3.3:	blocco GPS.....	22
Figura 3.4:	FrSky ACCST TANARIS X9D digital telemetry radio system	23
Figura 3.5:	blocco Radio Control Transmitter.....	23
Figura 3.6:	sottosistema per prima interpolazione lineare per velocità	25
Figura 3.7:	sottosistema per prima interpolazione lineare per sterzo	25
Figura 3.8:	sottosistema per prima interpolazione lineare per modalità.....	26
Figura 3.9:	sottosistema per seconda interpolazione lineare	26
Figura 3.10:	conversione per sterzo e manetta	27
Figura 3.11:	sezione per comunicare con sterzo e motori	28
Figura 3.12:	sottosistema buzzer	29
Figura 3.13:	sottosistema led.....	30
Figura 3.14:	stateflow per assegnare la modalità, led e buzzer	31
Figura 3.15:	programma implementato nel PX4	31
Figura 3.16:	foto di google earth con waypoints rover.....	32
Figura 3.17:	rover nel percorso desiderato	33
Figura 3.18:	traiettoria sperimentale del rover assegnati i waypoints in esempio	33

Indice delle tabelle

Tabella 1: Buzzer tune id..... 29
Tabella 2: LED color values..... 30
Tabella 3: LED blinking mode 30

1. INTRODUZIONE

I veicoli a navigazione autonoma utilizzano tecnologie che permettono la guida senza la necessità di un intervento umano, con sistemi di sicurezza e di controllo adatti. Per poter raggiungere un certo livello di autonomia, il sistema di comando sfrutta la capacità di rilevare l'ambiente circostante attraverso tecniche come radar, lidar, GPS, telecamere, sensori o anche Intelligenza Artificiale (AI). Dunque, l'interazione tra tali componenti ed i sistemi di controllo avanzato a bordo permettono all'autovettura di prendere decisioni circa i percorsi da seguire e di evitare eventuali ostacoli. Gli obiettivi di un sistema di guida autonoma sono percepire l'ambiente in cui è posto, pianificare le manovre successive per poter adempiere ad un obiettivo dato ed infine controllare il mezzo.

Molti rover sono stati progettati come veicoli da trasporto per i membri di missioni spaziali, mentre altri sono parzialmente o totalmente autonomi. Il primo rover telecomandato fu quello utilizzato nell'esplorazione lunare da parte dei sovietici, chiamato rover Lunokhod 1. Atterrò sulla Luna nel novembre 1970 e riuscì ad inviare immagini ravvicinate del suolo lunare e viste panoramiche, analizzando anche il suolo. Numerosi sono i rover telecomandati, autonomi e con pilota, tutti essenziali per l'esplorazione spaziale.

Tipicamente questi sono costruiti per essere utilizzati nella perlustrazione di altri pianeti, ma con il passare degli anni c'è stato uno sviluppo della loro applicazione, con tecnologie più semplici, anche a terra, diffusi in ambito commerciale e domestico, come gli aspirapolveri della Roomba.

Un rover con guida autonoma è indispensabile per vari ambiti. Pensando al trasporto merci con droni è possibile utilizzare un rover che, in piena autonomia, possa trasportare un oggetto per poi consegnarlo al drone, oppure ritirarlo da esso e consegnarlo a destinazione.

È proprio in questo contesto in cui è necessaria una meccanizzazione autonoma e precisa, che nasce il presente lavoro di tesi; il quale ha come obiettivo la guida autonoma tramite beam rider di un rover per il trasporto merci. Il lavoro nasce in seguito all'attività di tirocinio svolta presso il laboratorio C.I.R.I Aerospaziale, presso il Tecnopolo di Forlì; una struttura che promuove lo sviluppo di conoscenze e servizi per le aziende e gli enti di ricerca operanti nei settori dell'aeronautica, spazio, sistemi energetici, materiali avanzati e sistemi meccanici, sensoristica, ingegneria navale e trasporti.

La guida è stata pensata per essere adatta a diversi percorsi e modalità di guida; per tale motivo è stato utilizzato un radiocomando che consente di attuare le modalità Manual, Steering e Guided. Il

veicolo è progettato con un sistema sterzante che permette di cambiare l'angolazione delle ruote anteriori e da un sistema meccanico che consente anche di eseguire la retromarcia.

La massa del veicolo, considerando lo chassis e le quattro ruote, è pari a 4737,5 g. Lo chassis misura 41 x 16 x 8,7 cm. Il Rover, per lo studio effettuato, è munito di

- un componente principale: l'hex Cubepilot Cube Black, precedentemente noto come Pixhawk 2.1. Questo è un autopilota flessibile destinato principalmente ai produttori di sistemi commerciali. All'interno del PX4 sono presenti diversi sensori come il giroscopio e l'accelerometro;



Figura 1.1: cube flight controller, conosciuto anche come pixhawk 2.1

- uno strumento di posizionamento: il GNSS Holybro H-RTK F9P helical. Tale strumento è compatibile con il controllore di volo della serie Pixhawk open source. H-RTK F9P è l'ultima serie di sistemi di posizionamento GNSS differenziali di alta precisione di Holybro, con modulo UBLOX F9P, una bussola IST8310, un interruttore di sicurezza integrato e un indicatore LED tricolore. Questo sistema fornisce RTK multibanda con tempi di convergenza rapidi e prestazioni affidabili, ricezione simultanea di GPS e velocità di aggiornamento rapida per applicazioni altamente dinamiche e con precisione centimetrica;



Figura 1.2: H-RTK F9P Helical

- un doppio motore TITAN da 16,8 V a coppia elevata;
- un doppio servosterzo;

- un buzzer;
- un regolatore elettronico exc evx-2 Traxxas;
- un ricevente FrSky X8R, da 2.4GHz ACCST, 8CH PWM, 16CH SBUS;
- due batterie rocket pack 2 da 7,2 V, che alimentano i motori. La posizione delle batterie è bassa ed inclinata verso il centro del rover per aumentare la stabilità;
- una batteria Flight Discharge Li-Po da 35-11,1 V/ 2200 mAh, usata per l'avionica;
- sospensioni a sfera;
- diversi cavi per il collegamento tra i vari dispositivi.

L'elaborato segue due fasi principali:

- Modellazione matematica: in cui vengono poste le basi matematiche per navigazione, guida e controllo del rover. In tale sezione viene introdotto l'utilizzo del beam rider;
- Validazione numerica e sperimentale: in cui ci si interfaccia con il rover fisico e viene sviluppato il modello Simulink di guida autonoma.

2. MODELLAZIONE MATEMATICA

La modellazione matematica è fondamentale per la navigazione autonoma del rover, il cui studio può essere diviso in due sezioni principali: una prima parte che consiste nella modellazione del rover in un ambiente 3D ed una seconda parte che si compone del sistema di guida e controllo, secondo filosofia model-based design.

2.1 Modellazione del rover

Per la navigazione in autonomia del rover è stato realizzato un primo modello Simulink in cui viene rappresentato il rover in un ambiente 3D tramite l'utilizzo di Simscape e modelli 3D. Questo ha permesso di individuare il comportamento del rover con controllo manuale tramite Joystick. Per poter studiare la dinamica del veicolo è possibile inserire il blocco Vehicle body 3DOF Dual Track. Questo implementa un modello rigido di carrozzeria a due assi per poter calcolare il movimento longitudinale, laterale e di imbardata del veicolo. Con tale blocco si tiene conto della massa e della resistenza aerodinamica dovuta all'accelerazione e allo sterzo. Per determinare il movimento del veicolo, il blocco implementa le equazioni della dinamica, delle forze esterne e degli pneumatici.

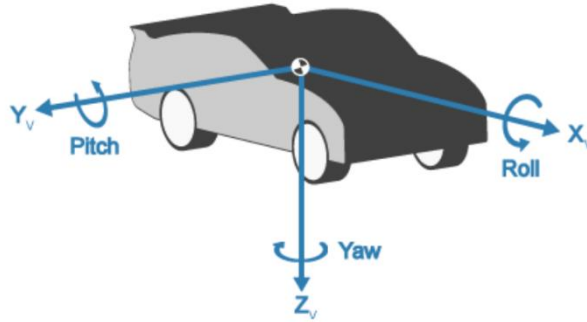


Figura 2.2: sistema di riferimento veicolo assi fissi

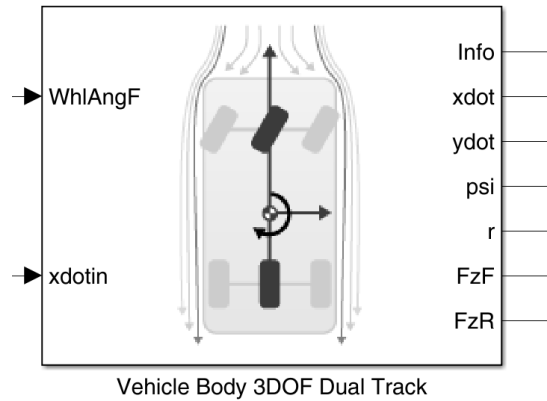


Figura 2.1: blocco Vehicle body 3DOF

Per calcolare la dinamica planare del corpo rigido vengono utilizzate le seguenti formule:

$$\ddot{x} = \dot{y}r + \frac{F_{xfl} + F_{xfr} + F_{xrl} + F_{xrr} + F_{xext}}{m}; \quad (1)$$

$$\ddot{y} = -\dot{x}r + \frac{F_{yfl} + F_{yfr} + F_{yrl} + F_{yrr} + F_{yext}}{m}; \quad (2)$$

$$\dot{r} = \frac{a(F_{yfl} + F_{yfr}) - b(F_{yrl} + F_{yrr}) + \frac{w_f(F_{xfl} - F_{xfr})}{2} + \frac{w_r(F_{xrl} - F_{xrr})}{2} + M_{zext}}{I_{zz}}; \quad (3)$$

$$r = \dot{\psi}; \quad (4)$$

le forze esterne riguardano sia la resistenza che le forze esterne in input. Quest'ultime vengono prese come nulle.

$$F_{x,y,z \text{ ext}} = F_{d \ x,y,z} + F_{x,y,z \text{ input}} ; \quad (5)$$

$$M_{x,y,z \text{ ext}} = M_{d \ x,y,z} + M_{x,y,z \text{ input}} ; \quad (6)$$

Il blocco è settato in “External longitudinal velocity” quindi le forze esterne sono le seguenti

$$F_{xflt} = 0 ; \quad (7)$$

$$F_{yflt} = -C_{yfl} \alpha_{fl} \mu_{fl} \frac{F_{zfl}}{2F_{znom}} ; \quad (8)$$

$$F_{xfrt} = 0 ; \quad (9)$$

$$F_{yfrt} = -C_{yfr} \alpha_{fr} \mu_{fr} \frac{F_{zfr}}{2F_{znom}} ; \quad (10)$$

$$F_{xrlt} = 0 ; \quad (11)$$

$$F_{yrlt} = -C_{yrl} \alpha_{rl} \mu_{rl} \frac{F_{zrl}}{2F_{znom}} ; \quad (12)$$

$$F_{xrtr} = 0 ; \quad (13)$$

$$F_{yrtr} = -C_{yrr} \alpha_{rr} \mu_{rr} \frac{F_{zrr}}{2F_{znom}} ; \quad (14)$$

Le equazioni utilizzano le seguenti variabili:

- x, \dot{x}, \ddot{x} posizione, velocità e accelerazione CG del veicolo lungo l’asse x nel sistema vehicle-fixed;
- y, \dot{y}, \ddot{y} posizione, velocità e accelerazione CG del veicolo lungo l’asse y nel sistema vehicle-fixed;
- ψ rotazione del telaio fisso del veicolo sull’asse z (imbardata);
- $\dot{\psi}$ velocità angolare del veicolo rispetto all’asse z;
- $F_{x,y,z \text{ ext}}$ forze esterne applicate al CG del veicolo lungo gli assi x, y e z fissati nel veicolo;

- $F_{d\ x,y,z}$ forze di trascinamento applicate al CG del veicolo, lungo gli assi fissi;
- $F_{x,y,z\ input}$ forze di trascinamento applicate al CG del veicolo, lungo gli assi x , y e z fissati dal veicolo;
- $M_{x,y,z\ ext}$ momento esterno sulla CG del veicolo, sugli assi fissi;
- $M_{d\ x,y,z}$ momento di trascinamento sul CG del veicolo, sugli assi x , y e z fissi dal veicolo;
- $M_{x,y,z\ input}$ momento di input sulla CG del veicolo, sugli assi x , y e z ;
- I_{zz} momento d'inerzia della carrozzeria del veicolo Informazioni sull'asse Z fissato dal veicolo;
- F_{xft}, F_{xrt} forza longitudinale dello pneumatico applicata alle ruote anteriori e posteriori, lungo l'asse X fissato dal veicolo;
- F_{yft}, F_{yrt} forza laterale dello pneumatico applicata alle ruote anteriori e posteriori, lungo Asse Y fisso sul veicolo;
- F_{xfl}, F_{xfr} forza longitudinale applicata al fronte sinistro e al davanti a destra ruote, lungo l'asse X fissato dal veicolo;
- F_{yfl}, F_{yfr} forza laterale applicata alla parte anteriore sinistra e anteriore destra ruote, lungo l'asse Y fissato dal veicolo;
- F_{xrl}, F_{xrr} forza longitudinale applicata alla parte posteriore sinistra e posteriore destra ruote, lungo l'asse X fissato dal veicolo;
- F_{xflt}, F_{xfrt} forza longitudinale dello pneumatico applicata alla parte anteriore sinistra e anteriore ruote destre, lungo l'asse X fissato dal veicolo;
- F_{yflt}, F_{yfrt} pneumatico a forza laterale applicato all'anteriore sinistro e all'anteriore destro ruote, lungo l'asse Y fissato dal veicolo;
- F_{xrlt}, F_{xrrt} forza longitudinale dello pneumatico applicata alla parte posteriore sinistra e posteriore ruote destre, lungo l'asse X fissato dal veicolo;
- F_{zfl}, F_{zfr} forza normale applicata alle ruote anteriori sinistra e destra, lungo Asse Z fisso sul veicolo;
- F_{zrl}, F_{zrr} forza normale applicata alle ruote posteriori sinistra e destra, lungo Asse Z fisso;
- m massa della carrozzeria del veicolo;
- w_f, w_r grandezza della ruota anteriore e posteriore;
- a, b distanza delle ruote anteriori e posteriori, rispettivamente, dal punto di proiezione normale del veicolo CG sull'asse comune aereo;
- C_{yf}, C_{yr} rigidità in curva della ruota anteriore e posteriore;
- α_{fl}, α_{fr} angoli di slittamento delle ruote anteriori sinistra e destra;

- α_{rl}, α_{rr} angoli di slittamento della ruota posteriore sinistra e destra;
- μ_{fl}, μ_{fr} coefficiente di frizione della ruota anteriore sinistra e destra;
- μ_{rl}, μ_{rr} coefficiente di frizione della ruota posteriore sinistra e destra;
- F_{znom} forza normale nominale applicata agli assi, lungo l'asse z fissato.

Nella tipologia Dual le forze verticali, longitudinali e laterali agiscono ai quattro angoli del veicolo o hard points.

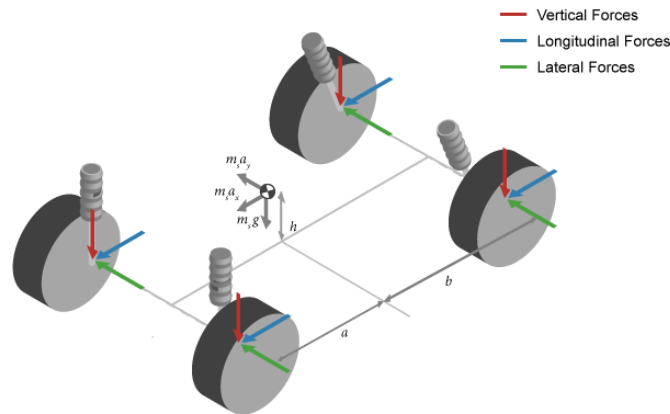


Figura 2.3: disposizione forze per tipologia di blocco Dual

Nel modello Simulink per la modellazione 3D è stato utilizzato il tool Simscape Multibody, che permette di modellare e simulare sistemi meccanici multibody. Nello specifico vengono usati i blocchi Simulation 3D Actor Transform Get, Simulation 3D Scene Configuration e Simulation 3D Vehicle with Ground Following.

In seguito, per la guida autonoma del rover fisico, è stata eliminata la parte di simulazione ed inserita la parte di ricezione dei sensori tramite PX4 Autopilots Support from UAV Toolbox. Nel modello è rimasto inalterato il sottosistema di navigazione, guida e controllo. In questa parte viene utilizzato il rover terrestre del Laboratorio C.I.R.I Aeronautica dell'UNIBO con sede a Forlì.

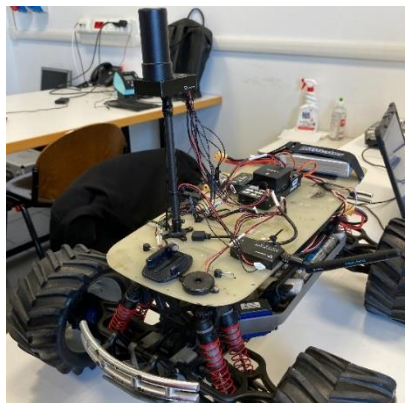


Figura 2.4: rover con GPS RTK e pixhawk cube black



Figura 2.5: rover del laboratorio C.I.R.I Aeronautica

2.2 Navigazione, guida e controllo

Il sistema navigazione, guida e controllo ha lo scopo di assegnare un angolo di rotta desiderato. Per poter fare ciò è necessario conoscere:

- l'angolo di rotta del rover χ ;
- la velocità angolare dell'asse di imbardata r ;
- la latitudine, longitudine ed altitudine del rover nel sistema di riferimento assi terra, indicati rispettivamente con μ , l e h ;
- la matrice dei waypoint, WP ;
- la latitudine, longitudine ed altitudine di riferimento indicati con lat_{ref} , lon_{ref} e h_{ref} .

Per tracciare la posizione del rover in latitudine e longitudine utilizziamo il GPS, i cui dati vengono poi trasformati nel sistema di riferimento assi terra. La posizione del rover in assi terra viene inserita nelle variabili X_e ed R . Per quanto riguarda i WP anche questi devono essere trasformati da LLA a flat Earth per il corretto funzionamento del Beam Rider.

Il sistema di navigazione consente al rover di compiere un percorso iterativo. Partendo dalla matrice dei WP, di dimensioni $n \times m$, e dall'indice della matrice pari a 0 si procede con l'assegnazione del WP di partenza o iniziale, WP target e della velocità desiderata V_d tramite la seguente scrittura:

$$WP_i = WP(\text{index}, 1:2)'; \quad (15)$$

$$WP_t = WP(\text{index}+1, 1:2)'; \quad (16)$$

$$V_d = WP(\text{index}+1, 3); \quad (17)$$

In seguito, viene calcolata la distanza tra WP_t e la posizione del rover e confrontata con un raggio di tolleranza preassegnato indicata con r_t . Se questa è minore o uguale al raggio di tolleranza viene sommato uno all'indice in memoria. Quindi partendo dall'indice pari a 0 si passerà ad indice pari a 1 e così via. Per la memorizzazione dell'indice è possibile utilizzare il blocco Memory di Simulink. Per far compiere al rover lo stesso percorso in continuo è stato aggiunto il tool Wrap to Zero, che permette di reinizializzare la matrice; infatti, questo imposta l'indice su zero quando l'input supera il valore soglia, in questo caso pari a $n-2$, mentre si pone l'indice uguale all'input quando l'input è minore o uguale alla soglia.

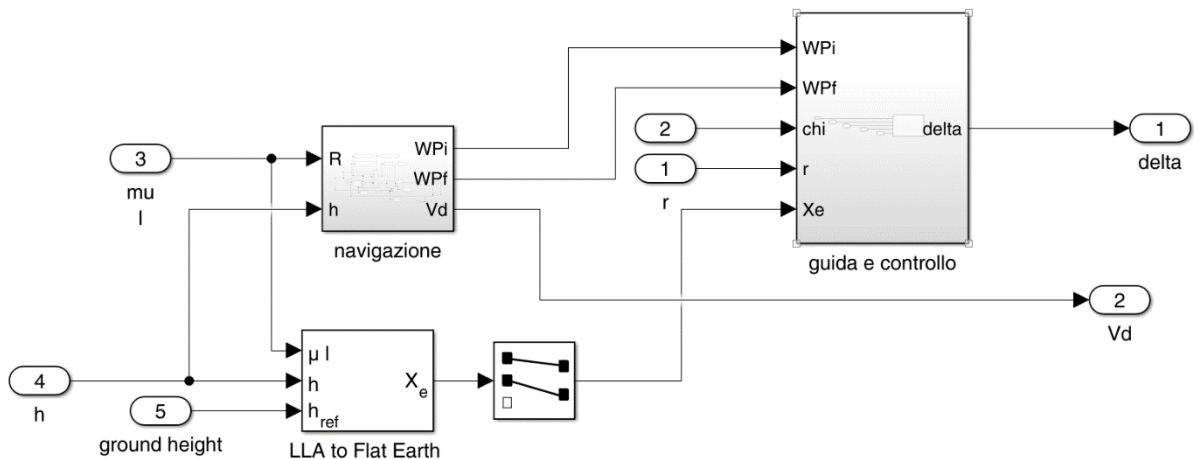


Figura 2.6: blocco navigazione, guida e controllo

Il sistema di guida e di controllo serve a manovrare il rover con opportune leggi. Queste si suddividono in due: la legge di guida e la legge di controllo.

La guida da remoto è una tecnologia che invia informazioni di guida al rover tramite la stazione di guida e lo dirige al target. Questa tecnica viene solitamente utilizzata nei missili per intercettare un obiettivo in movimento. La guida da remoto può essere divisa in “remote command guidance and beam-riding guidance”¹, quest’ultima può poi essere suddivisa in radio (radar) e ottica (laser). La guida utilizzata in questa tesi è la radar beam-riding guidance. Il principio base del beam rider è il seguente: la stazione di guida emette il raggio guida detto beam e dirige la linea centrale del beam al target. Il rover, conoscendo la propria posizione, produce l’istruzione di controllo per indirizzarsi lungo la linea centrale del beam e raggiungere il target.

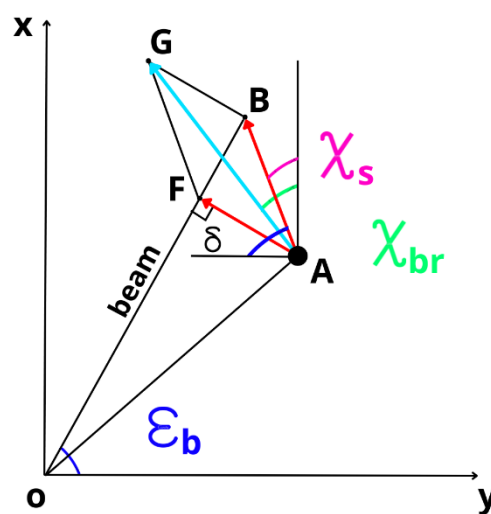


Figura 2.7: sistema di guida 'beam rider' nel sistema NED

Lo scopo del beam rider è quello di minimizzare $|AF|$ (vedi Figura 2.7). Infatti, in caso di deviazione, il rover si immette in una traiettoria che lo riporta sul raggio del beam. La strumentazione necessaria per questo consiste nel GPS.

Nella Figura 2.7 i punti $O(x_O, y_O)$ e $B(x_B, y_B)$ rappresentano due waypoint consecutivi. $|AB|$ è la distanza tra il rover ed il waypoint obiettivo, mentre $|AF|$ è la distanza tra velivolo ed il beam. L’informazione necessaria per far seguire al rover la traiettoria è la conoscenza degli angoli di rotta desiderati χ_s e χ_{br} .

¹ Qi Guodong, Yu Jianqiao, Ai Xiaolin and Jiang Jun, “remote command guidance and beam-riding guidance”, p2, Multi-missile Coordination High Precision Guidance and Control Method for Beam-riding Guidance

Nella modellazione è possibile scegliere quale dei due angoli di rotta si vuole utilizzare:

- χ_s heading straight angle [rad], il quale tiene conto solo del target finale (punto B in Figura 2.7);
- χ_{br} heading reference angle [rad], calcolato con beam rider, in cui si considera per il calcolo un nuovo punto indicato con G in Figura 2.7. Metodologia molto più accurata rispetto alla precedente.

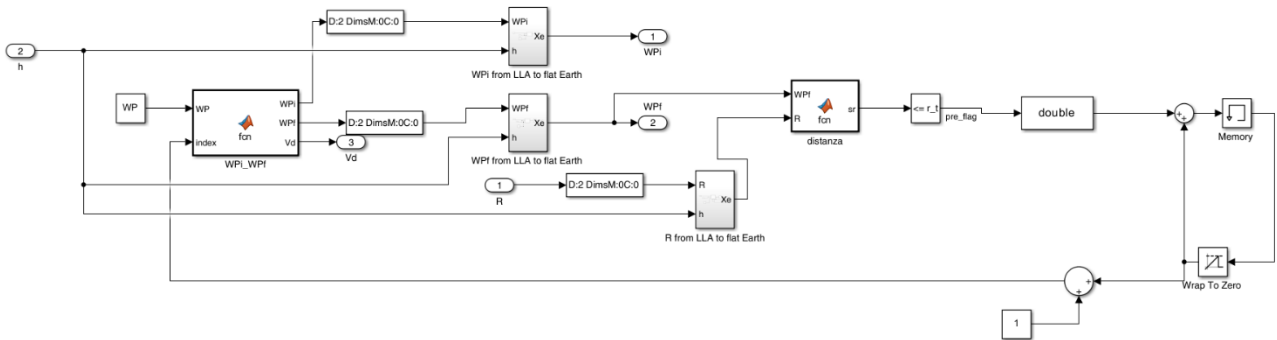


Figura 2.8: sottosistema navigazione

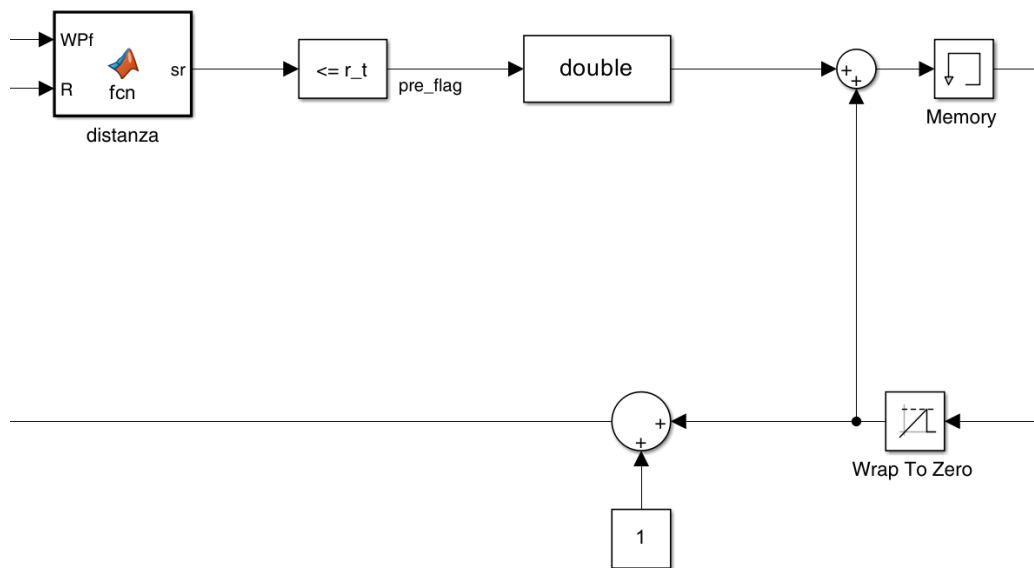


Figura 2.9: iterazione che permette di assegnare il successivo WPf

Come input al modello si hanno le posizioni dei waypoints, mentre come output si ottiene la distanza che il rover deve compiere che corrisponde ad AB, se si sceglie come angolo desiderato χ_s , mentre AF se si sceglie χ_{br} .

Le formule utilizzate all'interno del Beam rider sono le seguenti:

$$\vec{AB} = \vec{B} - \vec{A} ; \quad (18)$$

$$OB = \sqrt{(x_B - x_O)^2 + (y_B - y_O)^2} ; \quad (19)$$

$$\varepsilon_b = \sin^{-1}\left(\frac{x_B - x_O}{OB}\right) ; \quad (20)$$

$$\theta = \varepsilon_b - \text{atan2}((x_A - x_O), (y_A - y_O)) ; \quad (21)$$

$$OA = \sqrt{(x_A - x_O)^2 + (y_A - y_O)^2} ; \quad (22)$$

$$\vec{AF} = OA * \sin(\theta) ; \quad (23)$$

$$y_{AF} = -AF * \sin(\varepsilon_b) ; \quad (24)$$

$$x_{AF} = AF * \cos(\varepsilon_b) ; \quad (25)$$

$$\vec{AF} = [x_{AF}, y_{AF}] ; \quad (26)$$

$$\chi_s = \tan^{-1}\left(\frac{y_{AB}}{x_{AB}}\right) ; \quad (27)$$

$$y_{AG} = y_{AF} + y_{AB} ; \quad (28)$$

$$x_{AG} = x_{AF} + x_{AB} ; \quad (29)$$

$$\vec{AG} = [x_{AG}, y_{AG}] ; \quad (30)$$

$$\chi_{br} = \tan^{-1}\left(\frac{y_{AG}}{x_{AG}}\right) ; \quad (31)$$

La seconda legge è una legge proporzionale derivativa per il controllo del rover (anello di controllo), che permette di avere una guida proporzionale dipendente da ε , che corrisponde all'errore di disallineamento, dipendente dal target. La formula varia a seconda del valore dell'errore.

$$\varepsilon = \chi_d - \chi ; \tag{32}$$

se $|\varepsilon| > \pi$ la relazione da usare è la seguente

$$\varepsilon = \varepsilon - 2\pi * \text{sign}(\varepsilon); \tag{33}$$

altrimenti si utilizza la formula (32)

$$\delta = k_\chi * \varepsilon + k_r * r ; \tag{34}$$

dove:

- k_χ e k_r sono i guadagni del controllore;
- δ l'angolo di sterzo;
- χ rappresenta l'angolo di rotta assegnato;
- r è la velocità angolare di imbardata;
- χ_d angolo di rotta desiderato.

Nella Figura di seguito l'implementazione delle leggi di guida e controllo in ambiente Simulink.

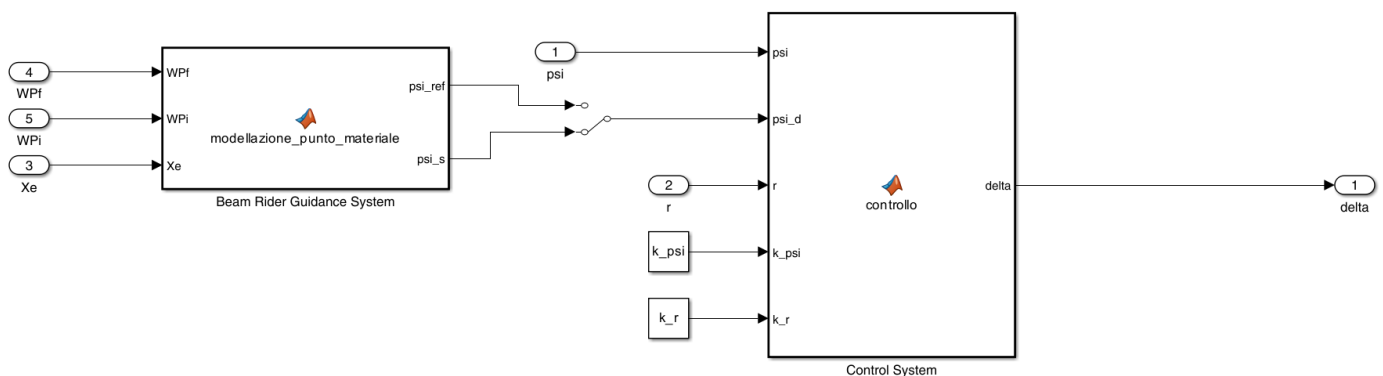


Figura 2.10: beam rider e sistema di controllo

Dal blocco guida e controllo si ottiene il valore dell'angolo di sterzo del veicolo indicato con delta.

È possibile validare il funzionamento del modello di guida nel simulatore andando a considerare i seguenti dati:

$$\bullet \quad WP = \begin{bmatrix} 44.20063923320479 & 12.066689496109902 & 0.1 \\ 44.20045848190607 & 12.066533927984638 & 0.1 \\ 44.20036089520192 & 12.066784043979917 & 0.1 \\ 44.2005334745312 & 12.066935588792951 & 0.1 \\ 44.20063923320479 & 12.066689496109902 & 0.1 \end{bmatrix} \quad \text{per far compiere al rover}$$

un circuito continuo è necessario assegnare all'ultima riga della matrice WP il primo waypoint assegnato.

Per quanto riguarda lo sterzo i dati presi in esame sono i seguenti:

- $\text{maxSteeringDisplacement} = 0.4125 \text{ rad};$
- $\text{minSteeringDisplacement} = -0.4125 \text{ rad};$
- $k_r = 0, \text{ r gain};$
- $k_\chi = 0.25, \text{ psi gain}.$

Come raggio di tolleranza, entro il quale il rover passerà al WP successivo, viene preso $r = 0,5 \text{ [m]}$.

I parametri di simulazione utilizzati sono i seguenti:

- $ST = 0.001 \text{ Sample time di simulazione};$
- $TS = 0.2 \text{ Sample time hardware};$

Le condizioni ambientali sono:

- $p_{amb} = 101325 \text{ Pa}$ pressione assoluta;
- $T_{amb} = 273 \text{ K}$ temperatura dell'aria;
- $g = 9.81 \frac{\text{m}}{\text{s}^2}$ accelerazione gravitazionale.

Infine, le coordinate di riferimento sono:

- $lat_{ref} = 44.2005741 \text{ deg}$, latitudine di riferimento per l'origine del sistema di coordinate in assi terra;
- $lon_{ref} = 12.0669805 \text{ deg}$, longitudine di riferimento per l'origine del sistema di coordinate in assi terra;
- $h_{ref} = 24 \text{ m}$, altezza di riferimento della superficie terrestre trasformata nel sistema assi terra.

Implementando il modello, è possibile ottenere i risultati in Figura, in cui viene tracciato il percorso compiuto dal rover nel caso in cui venga utilizzata χ_s oppure a χ_{br} .

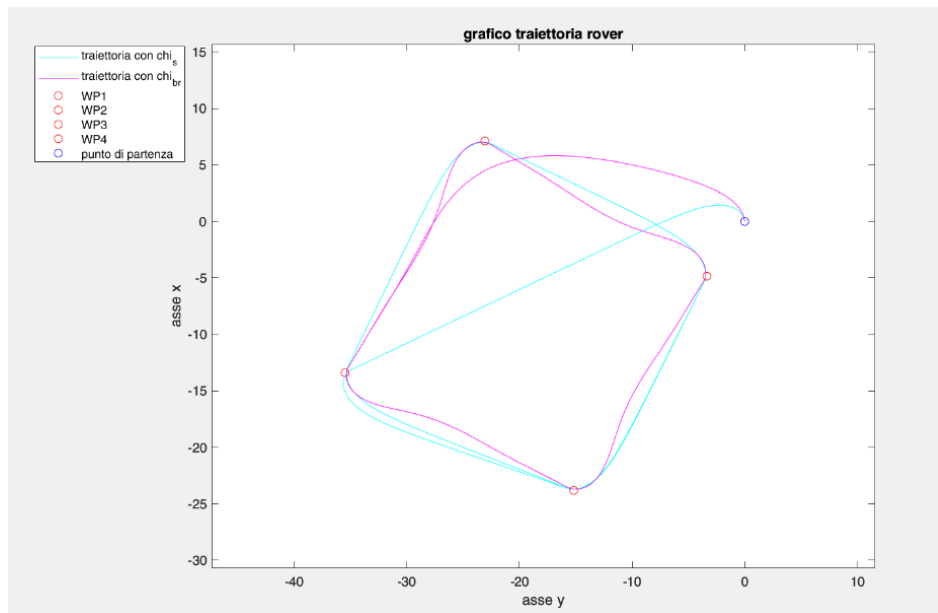


Figura 2.11: grafico traiettoria del rover per χ_s e χ_{br}

È possibile osservare un diverso andamento del rover e dei grafici a seconda della scelta della $\chi_{desiderata}$, se questa è pari a χ_{br} oppure a χ_s . Come visibile in Figura 2.11 se viene scelto χ_s il rover prosegue dritto verso il WP target, mentre se si utilizza χ_{br} il rover si avvicina al target minimizzando la distanza dal “beam”.

3 RISULTATI SPERIMENTALI

È stato possibile validare il funzionamento del modello di guida attraverso prove pratiche con l'utilizzo di un rover terrestre, controllato tramite il radiocomando. Tale veicolo è stato munito di GPS RTK e PX4.

3.1 Ambiente implementativo

L'ambiente di lavoro consiste nel PX4, un autopilota open source. PX4 fa parte della piattaforma Dronecode, una piattaforma end-to-end completa per lo sviluppo di droni, fornita con una licenza open source comune adatta al settore. Include, tra le altre cose, lo stack di volo PX4, la stazione di controllo a terra QGroundControl, l'SDK Dronecode e il Dronecode Camera Manager. Lo studio

che segue è basato su un software realizzato ad hoc, in cui non si utilizza una guida di base del velivolo fornita da PX4.

È necessario anche l'utilizzo di QGroundControl, la stazione di controllo a terra Dronecode. Si tratta di un programma che permette di caricare PX4 sull'hardware di controllo del veicolo, di configurare il veicolo ed i sensori principali (bussola, GPS, giroscopio, ecc.), modificarne i parametri ed ottenere informazioni in tempo reale.

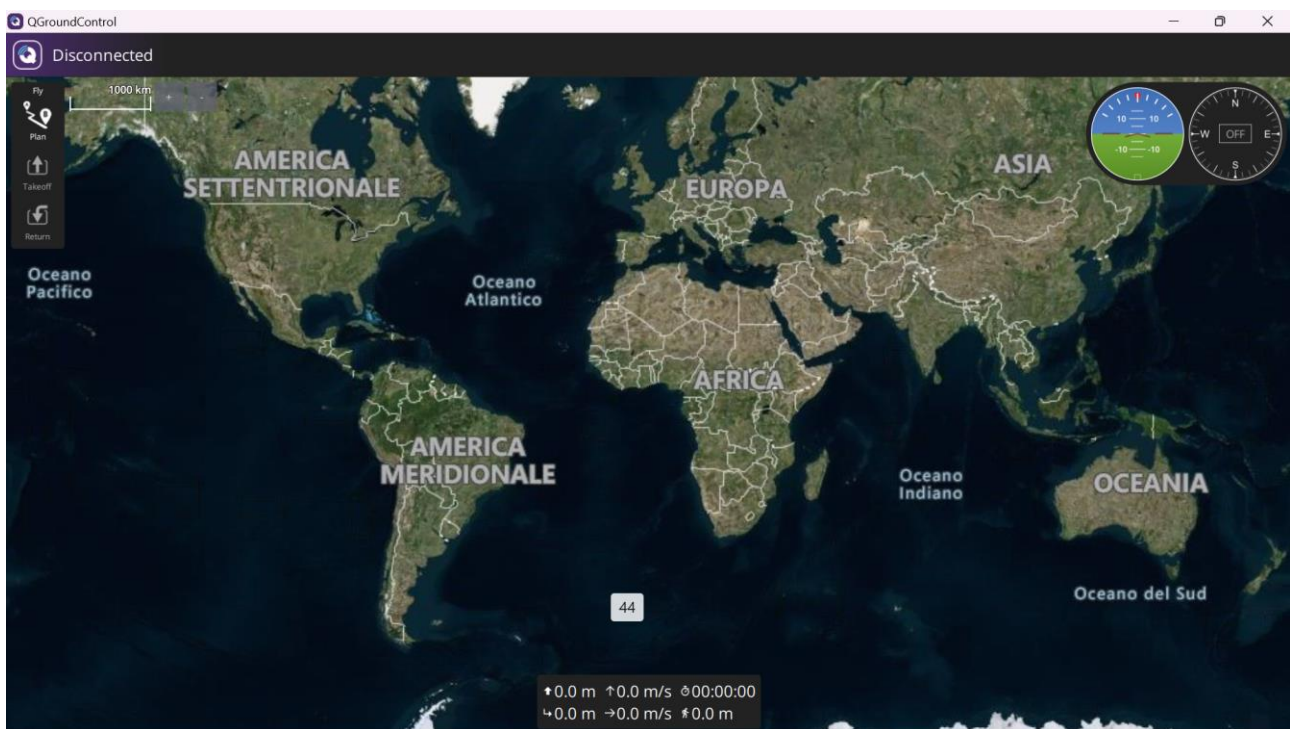


Figura 3.1: schermata principale di Qground

Il rover è munito di un GPS RTK. Attraverso una rete dedicata di satelliti artificiali in orbita il GPS fornisce informazioni sulle sue coordinate geografiche ovunque sulla Terra. La localizzazione avviene tramite la trasmissione di un segnale radio da parte di ciascun satellite e l'elaborazione dei segnali ricevuti da parte del ricevitore.

Per poter comunicare con il PX4 è necessario l'utilizzo del pacchetto PX4 Autopilots Support from UAV Toolbox. Nel programma sono stati utilizzati i seguenti blocchi:

- PX4 Vehicle Attitude, il quale legge l'argomento uORB vehicle_odometry ed emette le misurazioni dell'assetto dall'hardware Pixhawk. Nel programma l'output di tale blocco consiste nel Yaw rate indicato in seguito con r [rad/s].

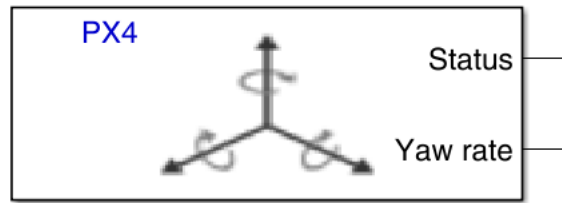


Figura 3.2: blocco Vehicle Attitude

- PX4 GPS, il quale legge il vehicle_gps_position uORB topic. Le informazioni che conferisce il blocco sono le coordinate del GPS, quindi la latitudine, longitudine e altitudine del rover che in seguito vengono indicati con μ [degrees], l [degrees] ed h [m], indicate con X_e . Le informazioni di latitudine e longitudine vengono in seguito inglobate nella variabile R . Come ulteriore output è necessario ottenere l'angolo di rotta (Course) indicato in seguito come χ [rad], course $\in [-\pi; \pi]$. Siccome la latitudine e la longitudine vengono forniti dal blocco in 1×10^{-7} degrees è necessario trasformarli in [degrees].

Mentre l'altitudine viene data in [mm] e deve essere trasformata in [m].

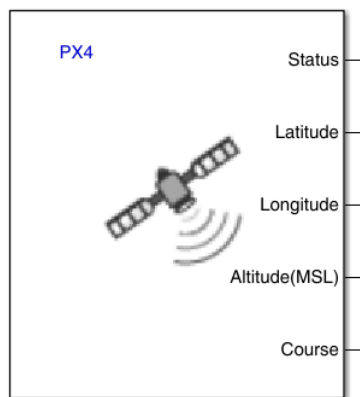


Figura 3.3: blocco GPS

- PX4 Radio Control Trasmittente che legge l'input_rc uORB topic per ottenere il segnale mandato dal radiocomando al canale specificato. È pertanto necessario controllare a quali canali corrispondono i comandi coi quali si vuole guidare il rover. I channels presi in esame sono Ch 1 per il controllo della manetta, Ch 4 per lo sterzo, Ch 5 per il tipo di modalità ed infine Ch 6 per la modalità failsafe. L'output Status indica se i dati sono stati ricevuti durante un passaggio temporale precedente o no. Un valore compreso tra 1 e 7 indica che nessun valore è stato letto dall'argomento uORB durante il passaggio temporale corrente oppure la

perdita di comunicazione con il trasmettitore radiocomandato o che il trasmettitore di controllo radio è passato alla modalità failsafe. Un valore pari a 0 indica che l'informazione al canale viene aggiornata.

Le modalità con cui è possibile guidare il rover sono 3:

- 1) Manual, che consiste nel controllo completamente manuale del rover, con la possibilità di controllare sia sterzo che manetta.
- 2) Steering, nella quale è possibile solo controllare la manetta del rover attraverso il radiocomando e la parte di sterzo viene guidata dalla sezione Beam rider.
- 3) Guided, la guida in piena autonomia del rover, il quale varia manetta e sterzo in modo autonomo a seconda dei WP assegnati.

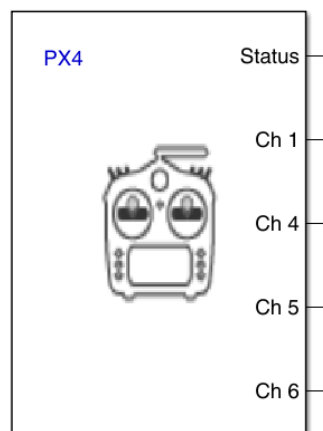


Figura 3.5: blocco Radio Control Transmitter



Figura 3.4: FrSky ACCST TARANIS X9D digital telemetry radio system

Per poter attuare tali modalità è necessario armare il rover, cambiando la modalità failsafe.

Dall'immagine 3.5 è possibile vedere quale leva viene utilizzata per modificare le modalità. Con la leva alzata si attiva la modalità Manual, con leva in mezzo si attiva lo Steering, infine con la leva abbassata si ha la Guided. Mentre per attivare il failsafe è necessario alzare la leva adiacente alla precedente. Tale modalità può essere disattivata posizionando la leva verso il basso.

La scelta delle modalità viene gestita attraverso uno schema Stateflow, che fornisce un linguaggio grafico per la modellazione della logica decisionale e combinatoria. In tale percorso la modalità Manual corrisponde ad un output pari a 1, lo steering ad un valore pari a 2 ed infine Guided a un output uguale a 3. Per poter passare da una modalità all'altra sono stati indicati dei requisiti a seconda del tipo di input in ingresso allo Stateflow.

Come radiocomando viene utilizzato FrSky 2.4GHz ACCST TANARIS X9D digital telemetry radio system.

- PX4 Parameter Read, legge il parametro del PX4 specificato e ne fornisce l'output. Nel programma tale blocco viene utilizzato in primo luogo per ricevere i valori massimi e minimi dei diversi canali del radiocomando. In seguito, si utilizza tale blocco per ricevere informazioni sui led e buzzer.
- PX4 PWM Output, utilizzato per configurare il PWM output del PX4, che nel nostro caso permette di regolare i motori e lo sterzo.
- PX4 uORB Write, il quale accetta un bus non virtuale che corrisponde allo specifico uORB topic e pubblica il messaggio al network uORB. Utilizzato per comunicare il colore e il suono rispettivamente del led e del buzzer.

Dal radiocomando è possibile, nella modalità Manual, governare la manetta e lo sterzo. Il valore dovrà essere convertito in tipologia single. Successivamente è necessario trasformare il valore del radiocomando in uno compatibile con la velocità, con lo yaw e con la modalità attraverso un'interpolazione lineare, seguendo la seguente formula, inserita all'interno di una Matlab function

$$OUT = OUTMIN + (IN - INMIN) \times \frac{(OUTMAX - OUTMIN)}{(INMAX - INMIN)} \quad (35)$$

OUTMIN e OUTMAX rappresentano i valori minimi e massimi che si hanno in output; INMIN e INMAX rappresentano i valori minimi e massimi che si hanno in input dal comando del radiocomando ed IN è il valore che si ha in input dal comando del radiocomando scelto. È possibile avere i valori INMAX e INMIN utilizzando il blocco PX4 Parameter Read. Per quanto riguarda PWM_MAIN vengono assegnati come valore massimo 1900, mentre come valore minimo 1100. Il valore massimo e minimo in output per la modalità è rispettivamente pari a 1 e -1.

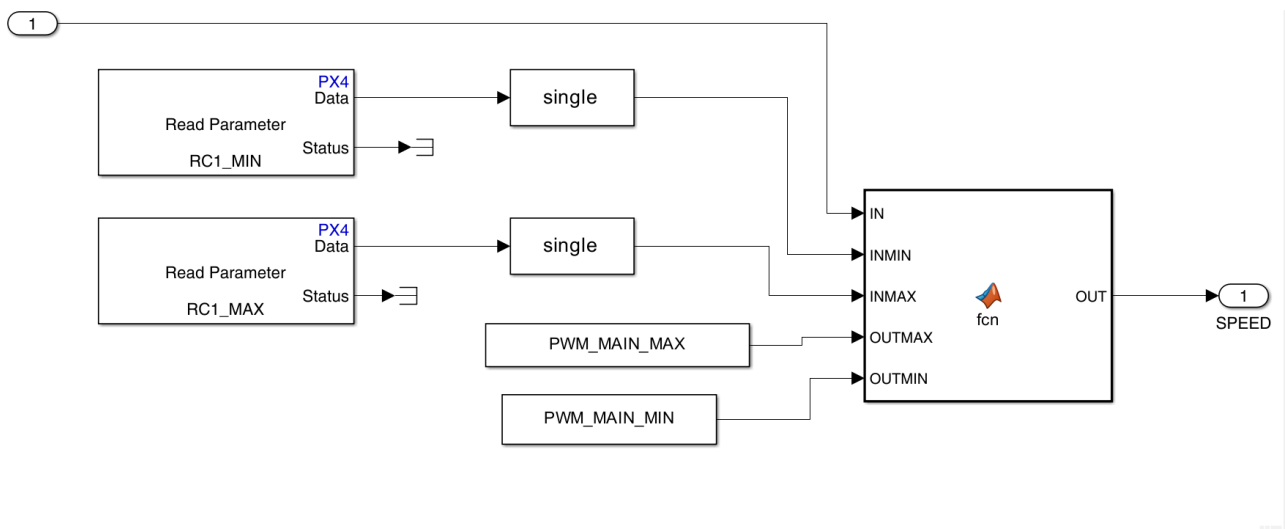


Figura 3.6: sottosistema per prima interpolazione lineare per velocità

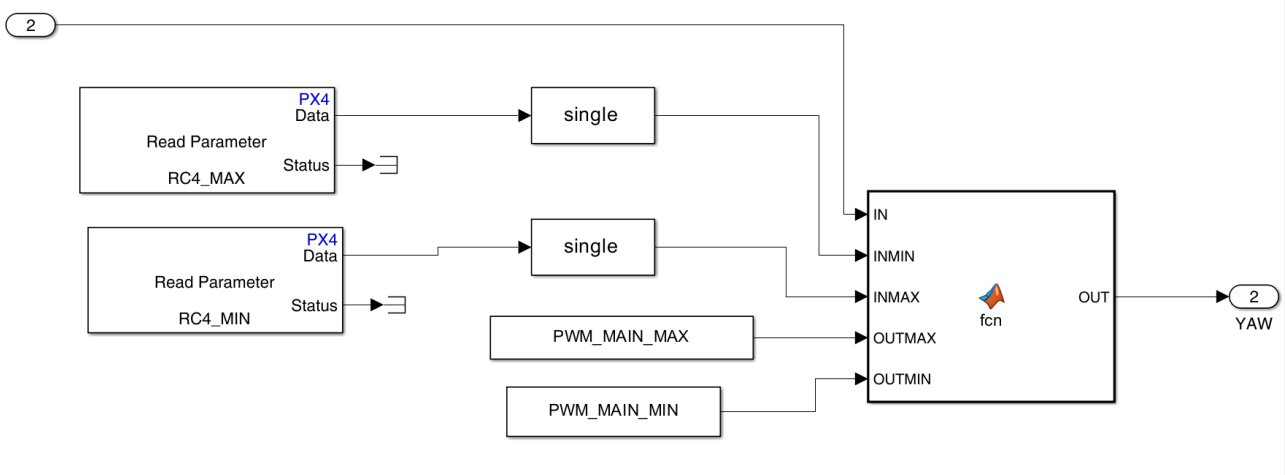


Figura 3.7: sottosistema per prima interpolazione lineare per sterzo

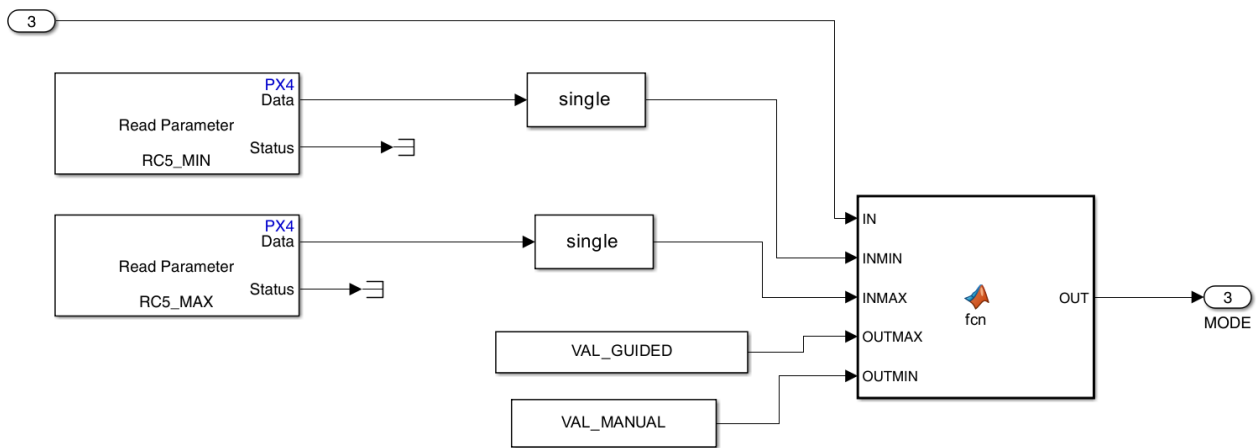


Figura 3.8: sottosistema per prima interpolazione lineare per modalità

In seguito, il valore della velocità e dello yaw dovrà essere ulteriormente convertito, utilizzando la stessa function, in un valore leggibile ai motori e allo sterzo. Tale procedimento viene eseguito all'interno del blocco "to servo conversion".

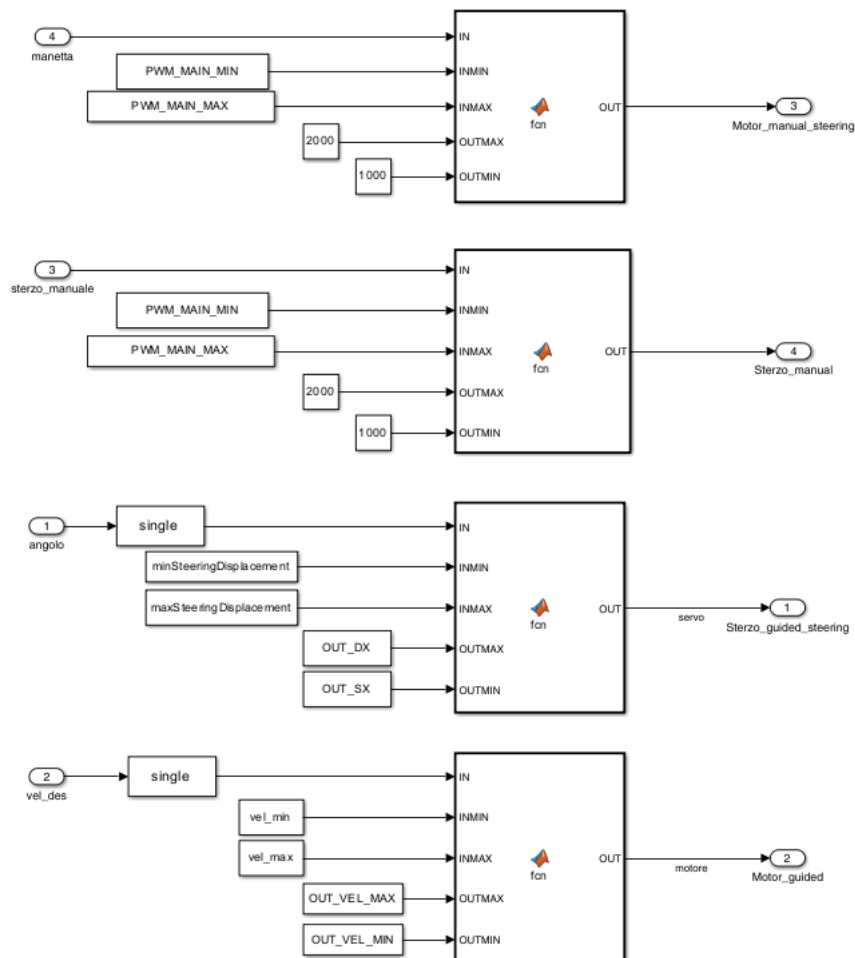


Figura 3.9: sottosistema per seconda interpolazione lineare

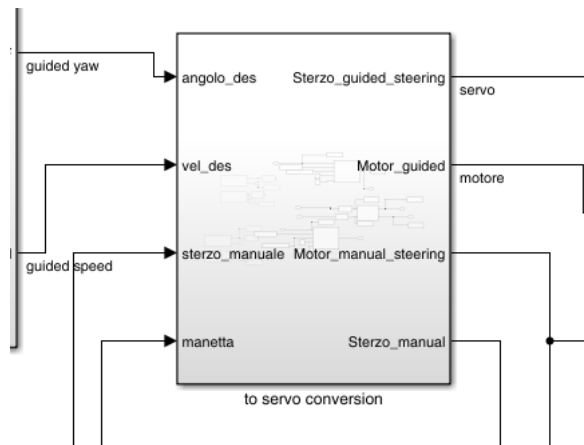


Figura 3.10: conversione per sterzo e manetta

Si ottengono 4 diversi output che corrispondono allo yaw calcolato con il beam rider, alla velocità assegnata tramite matrice dei WP, allo yaw e alla velocità manuale. Vengono assegnate le seguenti costanti:

per quanto riguarda lo sterzo

- OUT_DX = 1000 [°];
- OUT_SX = 2000 [°];
- maxSteeringDisplacement;
- minSteeringDisplacement;

per quanto riguarda la manetta

- OUT_VEL_MIN = 1000 [°];
- OUT_VEL_MAX = 2000 [°];
- vel_min = $-1 \frac{m}{s}$;
- vel_max = $1 \frac{m}{s}$.

A seconda delle modalità si sceglie quale valore di velocità e sterzo utilizzare:

- Manual utilizza la velocità e lo yaw assegnate dal radiocomando, indicate in Figura come Motor_manual_steering e sterzo_manual;
- Steering si avvale della velocità assegnata dal radiocomando e dalla yaw del beam rider, indicate in Figura come Motor_manual_steering e Sterzo_guided_steering;
- Guided assegna la velocità e lo yaw implementate dal beam river e dal WP, indicate in Figura come Motor_guided e Sterzo_guided_steering.

Tale ragionamento viene implementato in una Matlab function avente come input un vettore a 6 componenti e mode, mentre come output un solo valore di sterzo e manetta. All'interno della funzione vengono assegnati i valori a seconda della modalità scelta.

È necessario inserire un limite alle possibili grandezze di sterzo e manetta utilizzando il blocco Saturation, in cui si inserisce come upper limit 2000 [°] e lower limit 1000 [°]. Infine, la velocità e yaw vengono trasmessi ai motori e allo sterzo inserendo il blocco PX4 PWM Output, con una conversione in uint16. Il blocco PX4 PWM Output presenta diversi canali, nello studio sono stati scelti Ch1 e Ch3 che corrispondono ai canali motori e sterzo nel rover fisico.

Il blocco PX4 PWM Output accetta anche un segnale per abilitare le condizioni di failsafe. È quindi possibile configurare un comando di sicurezza nel radiocomando così da poter fermare il rover quando necessario. Nello studio in esame è stato preso il canale Ch6 come comando in sicurezza. Il Set failsafe viene attivato quando al blocco viene dato con input 0, mentre disattivato con un input pari a 1, tali valori devono essere di tipologia boolean. Con failsafe disattivato è possibile la mobilità del rover. Il blocco ha anche l'immissione Arm, con valore pari a 1 questo permette di abilitare il segnale per l'inserimento del controllore di volo PX4.

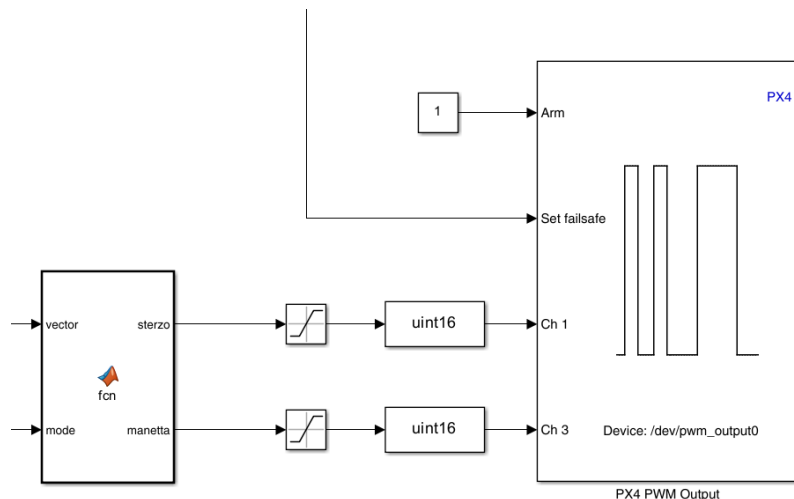


Figura 3.11: sezione per comunicare con sterzo e motori

Per maggiore praticità sono stati configurati i sottosistemi per comunicare con un led e buzzer. Questi variano a seconda della modalità scelta. In entrambi i casi è necessario assegnare ad ogni modalità un valore diverso di led e buzzer ed inserire un blocco rate transition. Per poter leggere e modificare il colore e il suono si utilizzano i blocchi uORB Message e uORB Write. Per quanto riguarda il led il

topic è *led_control*, mentre per il buzzer è *tune_control*. Per quanto riguarda l'id del tune è possibile scegliere tra i valori in Tabella.

Tabella 1: Buzzer tune id

Buzzer Tune	ID
CUSTOM	0
STARTUP	1
ERROR_TUNE	2
NOTIFY_POSITIVE	3
NOTIFY_NEUTRAL	4
NOTIFY_NEGATIVE	5
ARMING_WARNING	6
BATTERY_WARNING_SLOW	7
BATTERY_WARNING_FAST	8
GPS_WARNING	9
ARMING_FAILURE	10
PARACHUTE_RELEASE	11
EKF_WARNING	12
BARO_WARNING	13
SINGLE_BEEP	14
HOME_SET	15

Di seguito è presente il sottosistema per il buzzer.

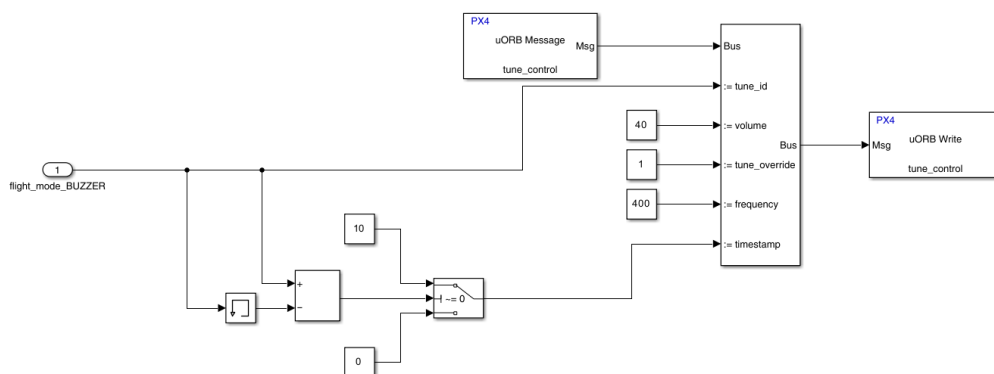


Figura 3.12: sottosistema buzzer

Per quanto riguarda il colore del led e la modalità con cui si illumina i valori sono quelli in Tabella.

Tabella 2: LED color values

LED Color	Values
No Color	0
RED	1
GREEN	2
BLUE	3
YELLOW	4
PURPLE	5
AMBER	6
CYAN	7
WHITE	8

Tabella 3: LED blinking mode

LED Blinking Mode	values
LED OFF	0
LED ON	1
DISABLED	2
BLINK SLOW	3
BLINK NORMAL	4
BLINK FAST	5
BREATHE	6
FLASH	7

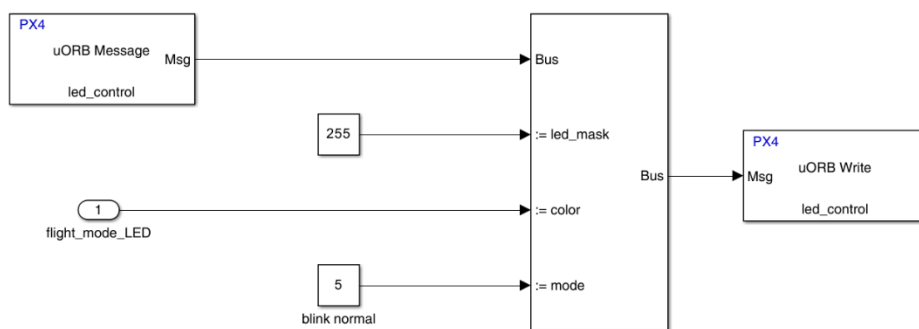


Figura 3.13: sottosistema led

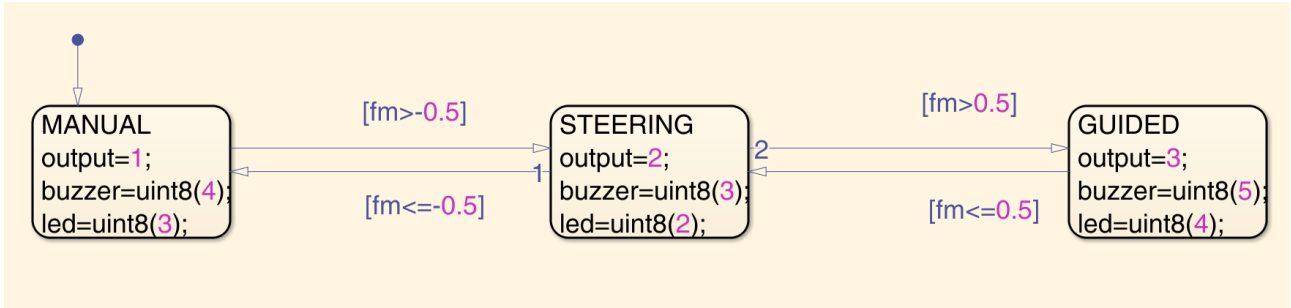


Figura 3.14: stateflow per assegnare la modalità, led e buzzer

Il programma finale è il seguente

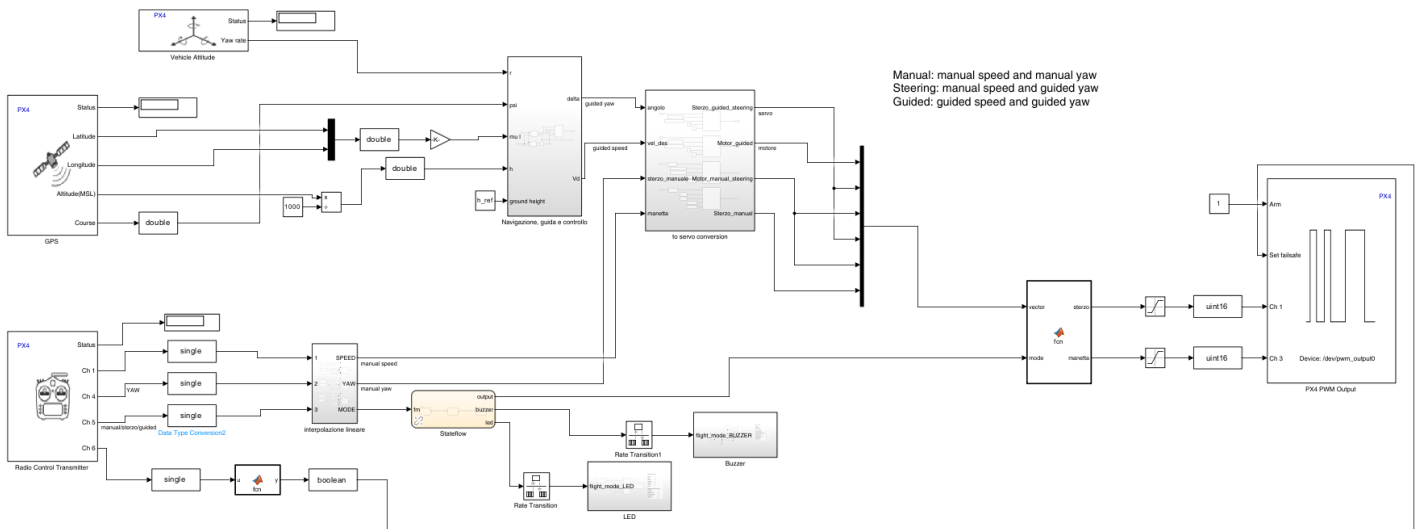


Figura 3.15: programma implementato nel PX4

3.2 Prove sperimentali

Per verificare il corretto funzionamento della configurazione software-hardware, è necessario procedere con prove sperimentali. A tal proposito, risulta necessario configurare il PX4 e il GPS con Qground.

Come visto nel capitolo “Navigazione, guida e controllo” è necessario assegnare un certo numero di waypoints, variando il numero di righe n della matrice WP. Si ottiene così una matrice nx3 in cui la prima colonna corrisponde alla latitudine [rad], la seconda alla longitudine [rad] ed infine la terza alla velocità.

Considerando una matrice 3x3 si ha

$$WP = \begin{bmatrix} WP1 \\ WP2 \\ WP3 \end{bmatrix} = \begin{bmatrix} lat1 & lon1 & vel1 \\ lat2 & lon2 & vel2 \\ lat3 & lon3 & vel3 \end{bmatrix} \quad (36)$$

Come esempio vengono inseriti i seguenti dati:

$$\bullet \quad WP = \begin{bmatrix} 44.20063923320479 & 12.066689496109902 & 0.1 \\ 44.20045848190607 & 12.066533927984638 & 0.1 \\ 44.20036089520192 & 12.066784043979917 & 0.1 \\ 44.2005334745312 & 12.066935588792951 & 0.1 \\ 44.20063923320479 & 12.066689496109902 & 0.1 \end{bmatrix} \quad \text{per far compiere al rover}$$

un circuito continuo è necessario assegnare all'ultima riga della matrice WP il primo waypoint assegnato. Le coordinate assegnate corrispondono ai WP in figura 3.16.



Figura 3.16: foto di google earth con waypoints rover

Per quanto riguarda lo steering i dati presi in esame sono i seguenti:

- $maxSteeringDisplacement = 0.4125 \text{ rad}$;
- $minSteeringDisplacement = -0.4125 \text{ rad}$;
- $k_r = 0$, $r \text{ gain}$;
- $k_\chi = 0.25$, $psi \text{ gain}$.

Come raggio di tolleranza, entro il quale il rover passerà al WP successivo, viene preso $r_t = 0,5 \text{ [m]}$.

I riferimenti locali e di altezza:

- $lat_{ref} = 44.2005741$ deg, latitudine di riferimento per l'origine del sistema di coordinate in assi terra;
- $lon_{ref} = 12.0669805$ deg, longitudine di riferimento per l'origine del sistema di coordinate in assi terra;
- $h_{ref} = 24$ m, altezza di riferimento della superficie terrestre trasformata nel sistema assi terra.

Come riferimento è stato preso il C.I.R.I Aeronautica via Baldassare Carnaccini, 12 a Forlì.

Il parametri di simulazione utilizzato è $TS = 0.2$ Sample time hardware.

Infine, l'environment scelto è:

- $p_{amb} = 101325$ Pa pressione assoluta;
- $T_{amb} = 273$ K temperatura dell'aria;
- $g = 9.81 \frac{m}{s^2}$ accelerazione gravitazionale.

Inseriti tali dati nel programma è possibile fare l'upload del sistema all'interno del PX4 tramite l'opzione "Build, Deploy & Start" nella sezione Hardware presente in Simulink. Si procede con l'accensione del rover e del radiocomando. Per armare il rover è necessario disattivare il failsafe ed è così possibile scegliere quale modalità attuare.

Con la modalità guidata il rover compirà il percorso in totale autonomia. Le prove hanno portato allo stesso percorso che si otteneva in simulazione, con maggiore precisione se viene scelto χ_{br} .



Figura 3.17: rover nel percorso desiderato



Figura 3.18: traiettoria sperimentale del rover assegnati i waypoints in esempio

4 CONCLUSIONI E SVILUPPI FUTURI

Lo studio effettuato ha permesso di trarre conclusioni in merito alla guida di un rover autonomo. Si è passati da un ambiente simulato ad uno sperimentale. I risultati precedentemente ottenuti nel Simscape Multibody e nel programma di guida sono stati verificati con il reale funzionamento del rover. È stato fondamentale l'utilizzo di un GPS RTK e PX4-Cube Black per il corretto rilevamento dell'ambiente esterno e per la realizzazione del sistema di navigazione. In particolare, si è osservato come il rover sia in grado di seguire una certa traiettoria per raggiungere una lista di waypoint assegnata. La guida autonoma è fondamentale ed indispensabile in molte applicazioni. Infatti, questo è pensato per evitare la presenza dell'uomo, la quale diventa superflua per un veicolo del genere.

Il programma redatto è finalizzato a minimizzare i problemi legati alla navigazione senza pilota di un rover tramite l'utilizzo di leggi di controllo proporzionale, di modellazione punto materiale e beam rider. La legge di controllo ed il beam rider permettono di attenuare gli errori che si possono generare per consentire al rover di riallinearsi con la corretta traiettoria.

Nel caso preso in esame la manetta viene implementata come costante, in un ciclo aperto. Per uno sviluppo futuro della navigazione è possibile modificare tale ciclo trasformandolo in uno chiuso. Infatti, è possibile inserire un anello di controllo in cui la variabile da stabilizzare è la velocità lungo la traiettoria. Tale implementazione può essere utile se si pensa ad un controllo di velocità dato un WP di partenza e un WP target in cui la velocità viene regolata ad esempio alterandola in base alla distanza dal beam rider oppure dal target. Un altro utilizzo potrebbe essere quello di modulare la velocità a seconda dello sterzo, così da non avere un capovolgimento del rover. Un'ulteriore aggiunta potrebbe essere l'utilizzo di sensori ottici.

Riferimenti Bibliografici

- (1) Garret D. Asper, Benjamin M. Simmons, Rapid Flight Control Law Deployment and Testing Framework for Subscale VTOL Aircraft, NASA, Settembre 2022.
- (2) A. Cina, "Dal GPS al GNSS (Global Navigation Satellite System). Per la geomatica," Torino, Celid, 2014.
- (3) Martin-Alvarez, S. Hayati, R. Volpe, R. Petras, Advanced Design and Implementation of a Control Architecture for Long Range Autonomous Planetary Rovers, s.l., s.d.
- (4) S. Miller, T. Soares, Y. Van Weddingen, J. Wendlandt, Modeling Flexible Bodies with Simscape Multibody Software, MathWorks.
- (5) M. Pavan, J. Jaidev Vyas, E. Balamurugan, "Modelling and simulation of aircraft nose wheel steering system", 39° National Systems Conference, Noida, India, 2015.
- (6) A. Shabana "Dynamics of multibody system", Cambridge University press, 1998.
- (7) Qi Guodong, Yu Jianqiao, Ai Xiaolin and Jiang Jun, Multi-missile Coordination High Precision Guidance and Control Method for Beam-riding Guidance, Dicembre 2018, Volume 1215.
- (8) Zarchan, Tactical and Strategic Missile Guidance: An Introduction, Timothy C. Lieuwen, 1 Luglio, 2019, Seventh Edition, Volume 1.
- (9) UAV Toolbox Support Package for PX4 Autopilots, User's Guide, MathWorks, 2023.
- (10) UAV Toolbox Support Package for PX4 Autopilots, Reference, MathWorks, 2023.
- (11) FrSky Telemetry, PX4 v1.9.0 User Guide, Dronecode, disponibile su: https://docs.px4.io/v1.9.0/en/peripherals/frsky_telemetry.html (visitato il 2 Maggio 2023).
- (12) Sensors, QGroundControl User Guide, Dronecode, disponibile su: <https://docs.qgroundcontrol.com/master/en/SetupView/Sensors.html> (visitato il 3 Maggio 2023).

(13)PX4 Autopilot User Guide, PX4 autopilot, disponibile su: <https://docs.px4.io/v1.9.0/en/>
(visitato il 2 Maggio 2023).