

ALMA MATER STUDIORUM  
UNIVERSITÀ 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

**Stima della posizione e assetto di un multirottore indoor  
tramite l'utilizzo di sensori laser**

CANDIDATO:  
Edoardo Baldacci

RELATORE:  
Emanuele Luigi De Angelis

Anno accademico 2019-2020

# Ringraziamenti

Ritengo necessario e doveroso dedicare questo spazio della mia tesi a tutte le persone che hanno reso possibile, grazie a consigli e supporto, il raggiungimento di questo traguardo.

Ringrazio il mio relatore il Prof. De Angelis per la disponibilità e la fiducia dimostrata nei miei confronti.

Ringrazio la mia famiglia, che mi ha dato la possibilità di arrivare fin qui.

Ringrazio la mia fidanzata Giulia per avermi incentivato alla scelta di questo percorso formativo.

# Indice

<b>Abstract.....</b>	<b>5</b>
<b>1. Introduzione .....</b>	<b>6</b>
<b>2. Analisi del problema .....</b>	<b>9</b>
2.1 VL53L1X.....	9
2.2 Definizione dei parametri.....	10
2.3 Posizionamento dei sensori.....	12
2.4 Determinazione angoli e posizione .....	15
2.5 Emulazione sensori .....	20
2.6 Filtraggio dei dati .....	23
<b>3. Realizzazione pratica .....</b>	<b>26</b>
3.1 Hardware.....	26
3.2 Realizzazione del Case.....	26
3.3 Cablaggio .....	30
3.4 Software di controllo.....	34
3.5 Filtraggio delle misure .....	37
3.6 Caratterizzazione dei sensori .....	38
<b>4. Validazione risultati.....</b>	<b>41</b>
4.1 Simulazione numerica.....	41
4.2 Test sperimentali preliminari .....	45
4.3 Stima dei costi.....	48
<b>5. Conclusioni .....</b>	<b>49</b>
<b>Bibliografia .....</b>	<b>50</b>
<b>Appendici .....</b>	<b>52</b>

# Indice delle figure

Figura 1: Schema a blocchi del sistema .....	7
Figura 2: Principio di funzionamento sensore Tof.....	9
Figura 3: Sensore V15311x. ....	9
Figura 4: Sistemi di riferimento. ....	11
Figura 5: Vettore posizione $\bar{R}^s$ .....	11
Figura 6: Rotazioni positive degli angoli. ....	12
Figura 7: Dimostrazione ridondanza delle misure. ....	13
Figura 8: Posizionamento dei sensori. ....	14
Figura 9: Rappresentazione dei vettori dei sensori 1 e 2. ....	15
Figura 10: Determinazione dell'angolo di $\varphi$ . ....	16
Figura 11: Rotazione di un angolo $-\varphi$ . ....	16
Figura 12: Determinazione angolo $\theta$ .....	17
Figura 13: Rotazione SDR di un angolo $-\varphi$ e $-\theta$ . ....	18
Figura 14: Determinazione angolo $\psi$ . ....	19
Figura 15: Schema emulatore dei sensori. ....	20
Figura 16: Scomposizione del vettore del sensore.....	21
Figura 17: Schema a blocchi del sistema. ....	23
Figura 18: Confronto tra misure del sensore 1.....	24
Figura 19: Confronto tra misure del sensore 4.....	25
Figura 20: Case completo. ....	27
Figura 21: Supporto sensore. ....	27
Figura 22: Configurazione dei sensori con modifiche. ....	28
Figura 23: Confronto prima e dopo la modifica. ....	29
Figura 24: Interferenze tra i due sensori. ....	29
Figura 25: Vista dei layer di stampa. ....	30
Figura 26: Case completo. ....	30
Figura 27: Sequenza del protocollo. ....	31
Figura 28: Immagini cablaggio del prototipo. ....	32
Figura 29: Schema di collegamento.....	33
Figura 30: Schema a blocchi dell'elaborazione delle misure. ....	39
Figura 31: Curva di caratterizzazione del sensore. ....	40

Figura 32: Simulazione al calcolatore della manovra di hovering.....	42
Figura 33: Innalzamento quota. ....	43
Figura 34: Spostamento della coordinata $X_G$ .....	44
Figura 35: Stanza costruita per le simulazioni. ....	45
Figura 36: Vista del prototipo durante le misurazioni. ....	45
Figura 37: rotazione di un angolo $\theta$ di $-16^\circ$ .....	46
Figura 38: Rotazione di un angolo $\psi$ di $22^\circ$ .....	46
Figura 39: Rotazione di un angolo di $\varphi$ di $-8^\circ$ .....	46

# Abstract

Oggi giorno, la navigazione indoor si basa su sistemi complessi e costosi, che prevedono l'utilizzo di varie tecnologie. È scopo di questo progetto fornire un'alternativa a questi sistemi di stima e assetto, basandosi su misure fornite da sensori laser.

Il primo punto affrontato per la realizzazione del sistema è stata un'analisi approfondita della natura del problema. In particolare, è stato necessario stimare il numero minimo di sensori da utilizzare ed il loro collocamento sul multirotores, il quale ha permesso di costruire il modello matematico adatto alle soluzioni ricercate.

Il passo successivo ha previsto l'implementazione nell'ambiente di calcolo del modello matematico, ricavato nella precedente fase.

Dopo di che, si è passato a verificare, tramite simulazioni numeriche, il modello ottenuto.

Tramite i risultati raggiunti è stato progettato e sviluppato il prototipo per la stima di posizione e assetto.

In ultima fase si è proceduto alla verifica, per mezzo di test sperimentali in laboratorio, dei dati acquisiti dal prototipo.

I risultati emersi dal lavoro svolto, hanno evidenziato che è possibile stimare la posizione e assetto, grazie all'utilizzo di sensori laser, seppur con una risoluzione angolare. A seguito dei test svolti in laboratorio, sono stati sottolineati anche i limiti tecnologici dei sensori utilizzati.

Il prototipo presenta ancora alcuni aspetti da migliorare. Tramite un'analisi più approfondita degli argomenti trattati in questo elaborato sarà possibile rendere quest'ultimo più funzionale.

# 1. Introduzione

Lo scopo di questo progetto è la realizzazione di un sistema di stima e assetto di un multirotores indoor, con l'utilizzo di sensori laser. A partire dalle misure fornite dai sensori si vogliono ricavare gli angoli di Roll, Pitch, Yaw, che caratterizzano l'assetto del multirotores e la posizione di esso all'interno di una generica stanza.

Il sistema ottenuto è un dispositivo compatto, basato sull'emissione di un numero limitato di raggi laser con tecnologia "Time of Flight". Ogni raggio laser colpisce le pareti di una stanza di forma elementare (parallelepipedo), rimbalza su di esse e ritorna verso il sensore, provvisto di un apposito ricevitore in grado di rilevare il segnale di ritorno. Calcolando il tempo di "volo" del fascio laser e conoscendo la sua velocità, ogni sensore è in grado di fornire la distanza tra esso e la parete.

Puntando i fasci dei sensori in determinate posizioni e installandone un determinato numero, è possibile, utilizzando delle specifiche equazioni, ricavare la posizione e l'assetto del velivolo.

Il principale motivo per la realizzazione di questo progetto è cercare di fornire un sistema più economico e meno complesso rispetto ai pochi sistemi, presenti oggi sul mercato, che si occupano di determinare assetto e posizione indoor.

Uno di questi è il sistema VPS (Vision Positioning System), che è in grado di individuare oggetti e ambienti circostanti, tramite una telecamera installata sotto il multirotores. Questo sistema tipicamente unisce la tecnologia di riconoscimento delle immagini, (grazie all'intelligenza artificiale), alla tecnologia di realtà aumentata.

Un'altra soluzione è l'utilizzo della tecnologia LIDAR (Laser Imaging Detection and Ranging) che si basa sulla scansione dell'ambiente circostante in modo molto simile ad un comune radar. A differenza di quest'ultimo però, vengono impiegate lunghezze d'onda ultraviolette in grado di rilevare anche piccoli oggetti.

Un altro possibile sistema si basa sull'utilizzo della Motion Capture, che prevede l'installazione di un certo numero di telecamere fisse all'interno dell'ambiente. Sull'oggetto, di cui si vuole seguire il movimento, vengono applicati dei marker, visibili dalle telecamere. Dopo di che, per mezzo di software appositi le telecamere identificano e seguono il movimento dei marker. Grazie a questo tracciamento è possibile ricostruire la posizione dell'oggetto nello spazio.

Si potrebbe pensare, come possibile soluzione di posizione, all'utilizzo di un ricevitore GPS (Global Position System), utilizzato in cooperazione con una bussola magnetica. Tuttavia, questo sistema presenta grossi inconvenienti: in primo luogo, all'interno di strutture la ricezione del segnale GPS non è ottimale, dunque si potrebbe avere una notevole incertezza sulla posizione. Un ulteriore

problema è l'errore di deviazione della bussola magnetica, causato dalle possibili masse ferromagnetiche nella struttura.

Si è così giunti alla soluzione del problema utilizzando sensori laser che, uniti ad un apposito software e hardware, sono in grado di determinare la posizione e l'assetto di un multirobot indoor.

Nel secondo capitolo dell'elaborato saranno esposti i risultati ottenuti durante l'esperienza di tirocinio, svolto presso il Laboratorio di Meccanica del Volo (Dipartimento di Ingegneria Industriale), sotto il tutorato del professor Emanuele Luigi De Angelis, durante il quale si è costruito un simulatore completo, in Matlab e Simulink, del sistema. Quest'ultimo è composto da 3 blocchi principali:

- Il simulatore del multirobot, che è un tool di simulazione sviluppato per il corso di Laboratorio di Meccanica del volo. La sua utilità consiste nell'ottenere i valori reali di posizione e angoli, che saranno sfruttati nei successivi blocchi.
- L'emulatore dei sensori, il cui scopo è simulare in ambiente di calcolo il reale funzionamento dei sensori.
- L'algoritmo di stima della posizione e assetto, in grado di risolvere il problema della posizione e dell'assetto prendendo in input le distanze misurate dai sensori laser.

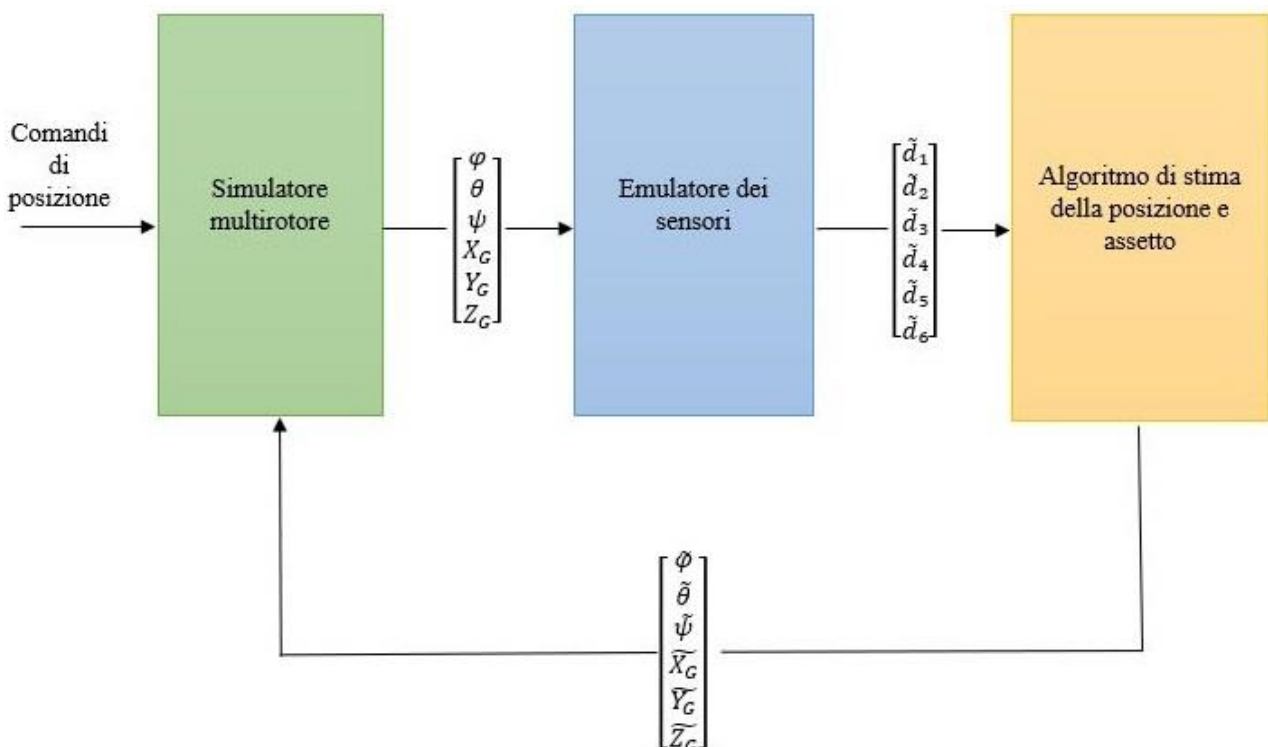


Figura 1: Schema a blocchi del sistema

I capitoli 3 e 4 tratteranno della realizzazione del prototipo finale, argomento che è stato approfondito come attività di tesi. In particolare, si affronteranno:



- I requisiti che stanno alla base della progettazione del case e la sua realizzazione tramite stampa 3D.
- Scelta dell'hardware, collegamento dei sensori e scrittura del software di controllo per ottenere posizione e assetto del sistema.
- Filtraggio delle misure e caratterizzazione del sensore.
- Test sperimentali e considerazioni derivate da questa esperienza.

## 2. Analisi del problema

### 2.1 VL53L1X

I sensori usati per la realizzazione del progetto sono i VL53L1X, che basano il loro funzionamento sulla tecnologia ToF (Time of Flight).

Il Time of Flight indica la misura del tempo impiegato da un'onda o una particella per percorrere una certa distanza. Conoscendo a priori la velocità con cui viaggia l'onda, è possibile risalire alla distanza percorsa.

Nel nostro caso sul sensore sono installati rispettivamente un emettitore laser e un ricevitore: il primo emette un fascio laser che, sbattendo su una superficie solida, viene riflesso e successivamente captato dal ricevitore. Grazie a quest'ultimo, il dispositivo riesce a misurare l'intervallo di tempo (tempo di volo) che è stato necessario al fascio per colpire l'oggetto e tornare indietro. Dopo di che, utilizzando questa misura del tempo e la velocità del laser, il sensore calcola la distanza percorsa.

Di seguito sono riportate le caratteristiche del sensore, fornite dalla casa costruttrice:

- Tensione nominale: 3.3V/5V
- Dimensioni: 20x24mm
- Dimensioni fori di montaggio: 3 mm
- Range di misura: 40 a 4000 mm
- Precisione di misura:  $\pm 5\%$
- Tempo di misurazione: 20 ms (short distance mode), 33 ms (Medium distance mode), 140 ms (Long distance mode).
- Campo visivo:  $27^\circ$
- Lunghezza d'onda laser: 940 nm
- Temperatura di utilizzo:  $-20^\circ$  a  $80^\circ$

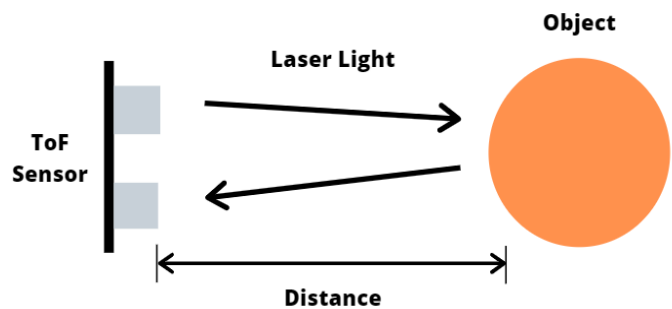


Figura 2: Principio di funzionamento sensore ToF. Fonte: <https://www.seeedstudio.com/blog/2020/01/08/what-is-a-time-of-flight-sensor-and-how-does-a-tof-sensor-work/>



Figura 3: Sensore VL5311x. Fonte: <https://www.amazon.com/Raspberry-Distance-Accuracy-Frequency-Detection/dp/B07F3TV3G4>

Bisogna tenere presente che le prestazioni dei sensori possono dipendere dall'ambiente in cui si svolge la misura. I due parametri che influiscono maggiormente sono la luminosità della stanza e la qualità delle superfici di riflessione.

Il sensore possiede diverse modalità di misura in base alla portata desiderata:

- Short; questa è la modalità con la portata più ridotta (circa 1360 mm) e necessita di un minimum timing budget di 20 ms. Inoltre, offre il vantaggio che le misure ottenute non risentono particolarmente della luminosità della stanza.
- Medium; la portata è di 2900 mm, con un minimum timing budget di 33 ms. Qui la dipendenza dalla luminosità è molto più accentuata: in un ambiente con un'intensa luminosità la portata scende a 760 mm.
- Long; questa funzione permette di sfruttare tutto il potenziale del sensore con una portata di 3600 mm, occorre però impostare un minimum timing budget di 140 ms. Anche in questo caso abbiamo una forte dipendenza dalla luminosità come nella modalità precedente.

Il minimum timing budget consiste nel tempo minimo che il sensore impiega per la misurazione. Più questo tempo è elevato, più la portata del dispositivo cresce, migliorando anche l'errore di ripetibilità. È possibile impostare un timing budget in base alle necessità, rimanendo comunque all'interno di un range da 20-1000 ms.

Nel progetto la modalità impostata è quella intermedia con un minimum timing budget di 40 ms.

## **2.2 Definizione dei parametri**

Un corpo rigido è un oggetto materiale che è soggetto al vincolo di rigidità, ovvero sia che il corpo sia fermo o in movimento, esso non si deforma mai. Grazie a questa definizione è possibile affermare che un multirobot è un corpo rigido, di conseguenza il suo movimento nello spazio può essere ricondotto allo studio della cinematica di un corpo rigido.

Per poter individuare posizione e assetto di un corpo rigido si ha necessità di introdurre il concetto di gradi di libertà, ovvero il numero minimo di parametri indipendenti tra loro che sono necessari ad individuare una generica configurazione nello spazio.

Sfruttando i risultati della fisica e della meccanica razionale, si può affermare che i gradi di libertà per un corpo rigido sono 6, in particolare un vettore posizionale rispetto ad un SDR inerziale (3 componenti), ed una terna di angoli che identificano le rotazioni intorno a particolari assi.

Come accennato nelle righe precedenti, i sei parametri necessitano di un sistema di riferimento per essere definiti, occorre perciò introdurre quali saranno usati nella trattazione:

- **SISTEMA DI ASSI COORDINATI:** È un sistema di assi coordinati e perpendicolari tra loro, solidale con la stanza (in figura, in nero). Grazie a questa terna di assi coordinati è possibile definire il vettore posizione  $\bar{\mathbf{R}}$ .
- **ASSI VERTICALI LOCALI:** Un sistema di riferimento che è locale, con origine fissata nel baricentro del nostro corpo rigido e con gli assi che rimangono paralleli, ma di verso opposto, al Sistema di Assi Coordinati (in figura, in blu).
- **ASSI BODY:** Un sistema che ha assi solidali con il corpo, che quindi oltre a muoversi come il baricentro, assume anche l'orientamento del corpo rigido (in figura, in rosso).

Nella trattazione si ha necessità di dover scrivere i vettori in diversi sistemi di riferimento. Per una

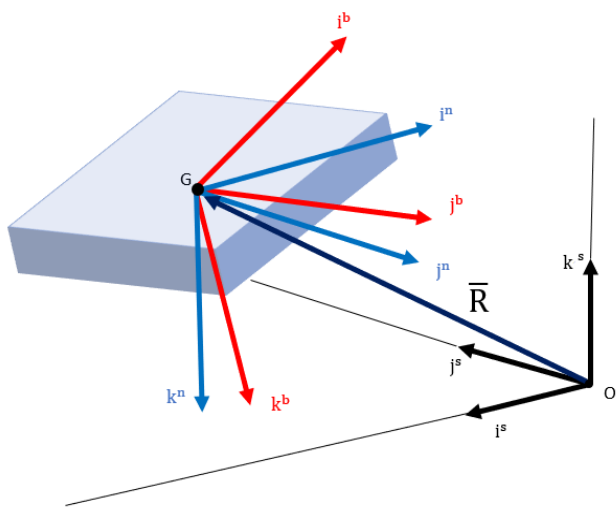


Figura 4: Sistemi di riferimento.

maggiore chiarezza definiamo degli apici per identificare i vari sistemi riferimento. L'apice "s" verrà utilizzato per identificare i vettori scritti nel sistema di Assi Coordinati, mentre l'apice "n" sarà usato per il SDR Assi Verticali Locali e in ultimo l'apice "b" per il sistema di riferimento Body.

Definiti i sistemi di riferimento, è più semplice visualizzare i 6 parametri indipendenti, che caratterizzano la posizione e l'orientamento del drone.

Come prima cosa si può definire il vettore posizionale  $\bar{\mathbf{R}}^s = (X_G, Y_G, Z_G)$ , che congiunge il baricentro del nostro sistema con l'origine del sistema Assi Coordinati.

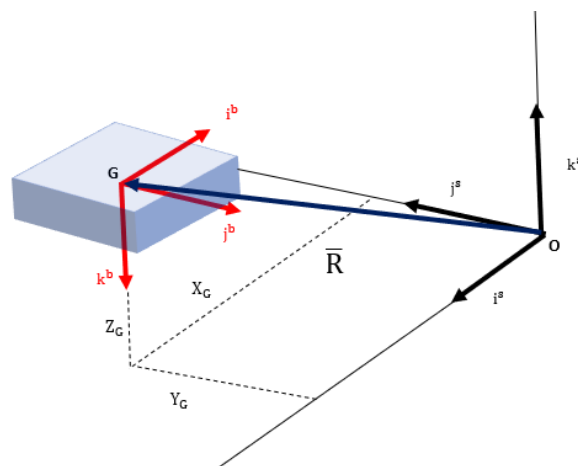


Figura 5: Vettore posizione  $\bar{\mathbf{R}}$ .

La seconda terna da definire sono gli angoli, che identificano le rotazioni del SDR Body, rispetto all'SDR Assi Verticali Locali. Questi angoli vengono comunemente chiamati angolo di Yaw  $\psi$ , angolo di Pitch  $\theta$  e angolo di Roll  $\varphi$ .

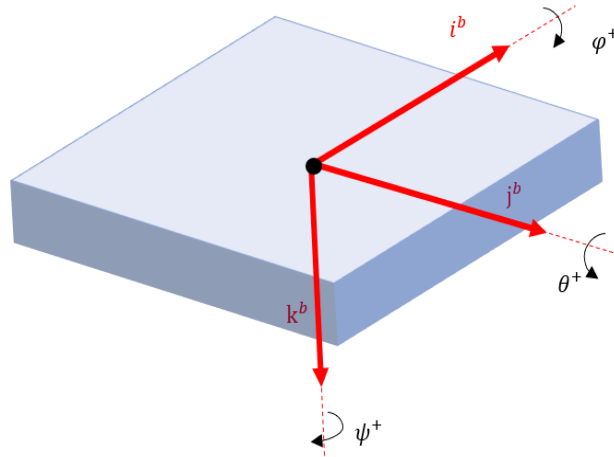


Figura 6: Rotazioni positive degli angoli.

Eslicitata la terna di angoli utilizzata è possibile ricavare una matrice di rotazione che, applicata a un vettore in assi di riferimento Body, lo riscrive nel sistema di riferimento Assi Verticali.

Per applicazioni legate alla meccanica del volo si utilizza una sequenza 3-2-1, ovvero la prima rotazione viene fatta intorno al terzo asse  $\mathbf{k}^b$ , poi una rotazione intorno al secondo asse  $\mathbf{j}^b$  ed infine una rotazione intorno al primo asse  $\mathbf{i}^b$ .

$$R_b^n = \begin{pmatrix} \cos(\theta)\cos(\psi) & \sin(\varphi)\sin(\theta)\cos(\psi) - \cos(\varphi)\sin(\psi) & \cos(\varphi)\sin(\theta)\cos(\psi) + \sin(\varphi)\sin(\psi) \\ \cos(\theta)\sin(\psi) & \sin(\varphi)\sin(\theta)\sin(\psi) + \cos(\varphi)\cos(\psi) & \cos(\varphi)\sin(\theta)\sin(\psi) - \sin(\varphi)\cos(\psi) \\ -\sin(\theta) & \sin(\varphi)\cos(\theta) & \cos(\varphi)\cos(\theta) \end{pmatrix} \quad (1)$$

## 2.3 Posizionamento dei sensori

Nel paragrafo precedente è stato esposto il numero minimo di parametri necessari ad individuare l'assetto e la posizione di un corpo rigido nello spazio, ovvero 6.

Il passo successivo, dunque, sarà trovare una determinata configurazione utilizzando sei sensori che rappresenteranno i sei parametri indipendenti citati sopra.

Perciò in questo paragrafo verrà affrontato il problema del posizionamento dei sensori, ovvero dove e come posizionarli per avere la migliore stima di assetto e posizione.

È importante sottolineare le basi su cui si è costruito il modello dei sensori: un primo vincolo consiste nell'aver almeno un sensore in ogni direzione principale  $i^b, j^b, k^b$ .

Un secondo requisito da soddisfare consiste nell'utilizzare il minor numero di sensori per ricavare la posizione e assetto.

Questo concetto lo si può comprendere meglio con un esempio: consideriamo solo il multirobot fisso, in una posizione generica nella stanza, con angoli di Pitch e Roll nulli ma con un angolo Yaw arbitrario e posizioniamo per ora solo tre sensori sulla superficie frontale normale all'asse ad una distanza arbitraria B.

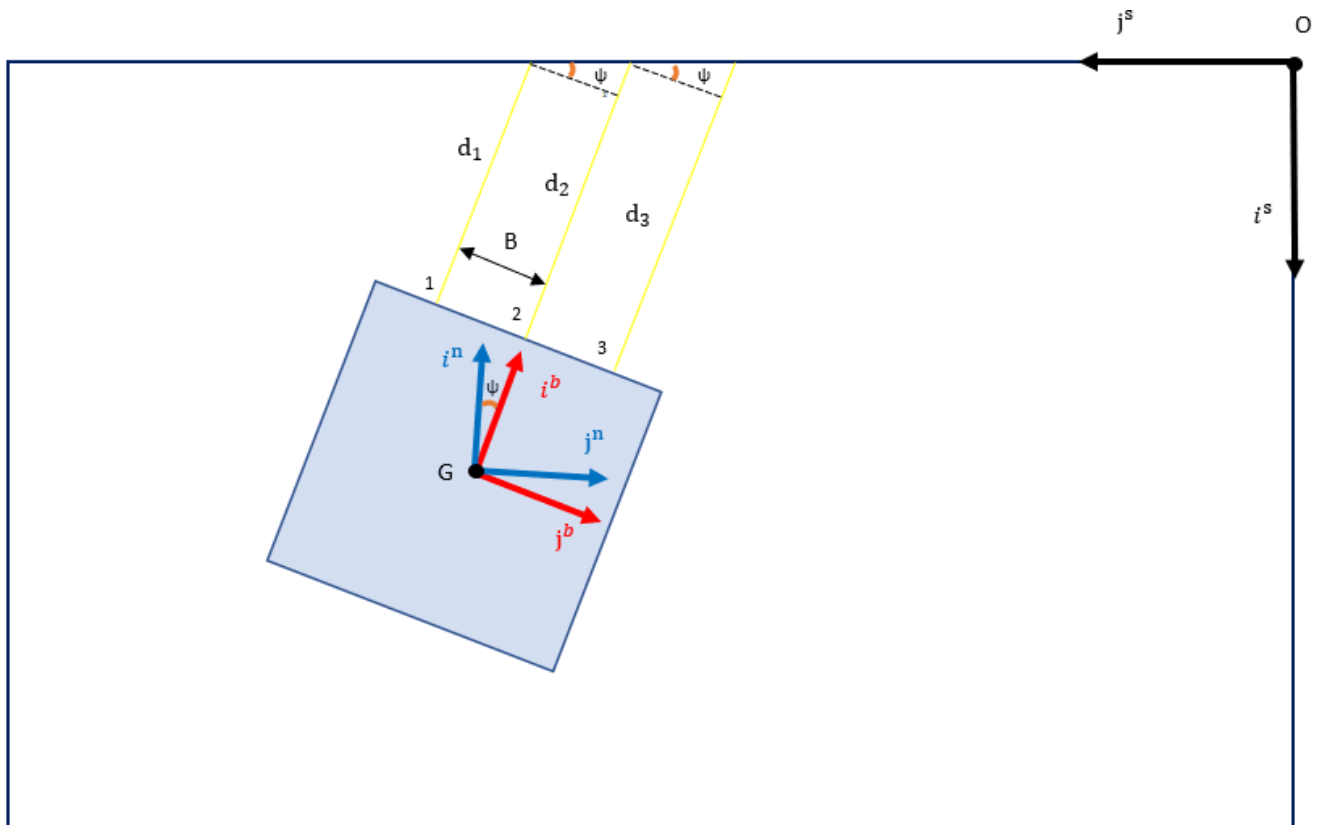


Figura 7: Dimostrazione ridondanza delle misure.

$$\psi = \text{atan}\left(\frac{d_2 - d_1}{B}\right) \quad (2)$$

$$\psi = \text{atan}\left(\frac{d_3 - d_2}{B}\right) \quad (3)$$

Come si può vedere tramite l'equazione 2 e l'equazione 3 è possibile risalire all'angolo di Yaw. Si può notare inoltre che il sensore numero 3 è superfluo, questo perché a meno degli errori di misura, i rapporti nell'equazioni 2 e 3 sono identici, dunque si otterrebbe lo stesso angolo in tutti e due i casi. È possibile, perciò, eliminare uno dei tre sensori.

Tenendo conto di tutti i vincoli considerati per il collocamento dei sensori si è giunti a una configurazione sperimentale:

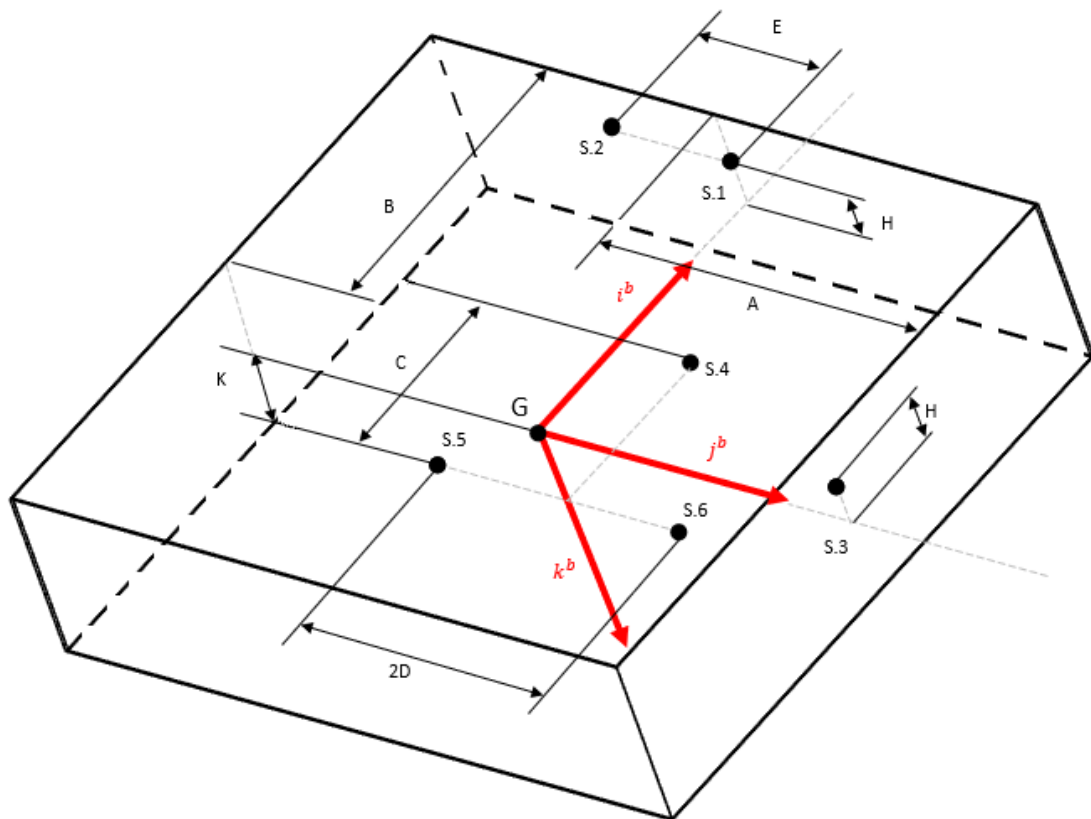


Figura 8: Posizionamento dei sensori.

I parametri A, B, C, D, E, H, K sono le distanze dei sensori dal baricentro del multirottore scritte rispetto all'SDR Body.

## 2.4 Determinazione angoli e posizione

Per poter semplificare le operazioni nei prossimi passaggi definiamo ogni distanza misurata come un vettore scritto in assi Body, con origine nel baricentro del corpo rigido.

Visto che ogni sensore è in posizione diversa rispetto a quella del baricentro, si ha necessità di aggiungere delle componenti, ovvero la distanza tra la posizione del sensore e il baricentro, tutto scritto nei tre assi principali  $\mathbf{i}^b, \mathbf{j}^b, \mathbf{k}^b$ .

$$\bar{\mathbf{r}}_1^b = \begin{pmatrix} d_1 + B \\ 0 \\ -H \end{pmatrix} \quad \bar{\mathbf{r}}_2^b = \begin{pmatrix} d_2 + B \\ -E \\ -H \end{pmatrix} \quad \bar{\mathbf{r}}_3^b = \begin{pmatrix} 0 \\ d_3 + A \\ -H \end{pmatrix} \quad (4)$$

$$\bar{\mathbf{r}}_4^b = \begin{pmatrix} C \\ 0 \\ d_4 + K \end{pmatrix} \quad \bar{\mathbf{r}}_5^b = \begin{pmatrix} 0 \\ -D \\ d_5 + K \end{pmatrix} \quad \bar{\mathbf{r}}_6^b = \begin{pmatrix} 0 \\ D \\ d_6 + K \end{pmatrix}$$

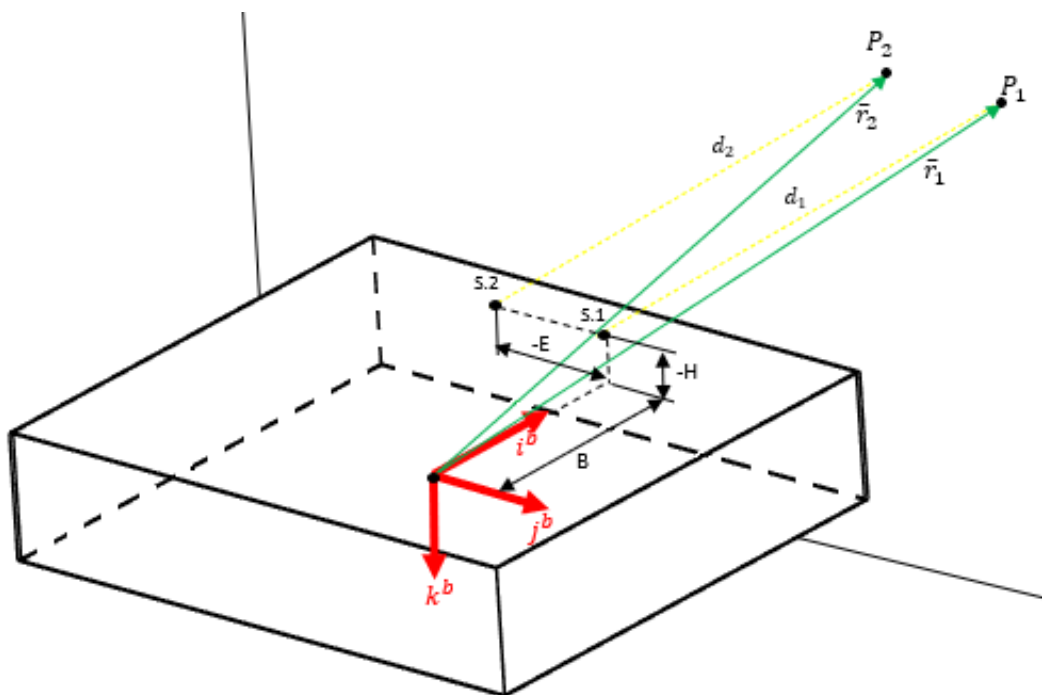
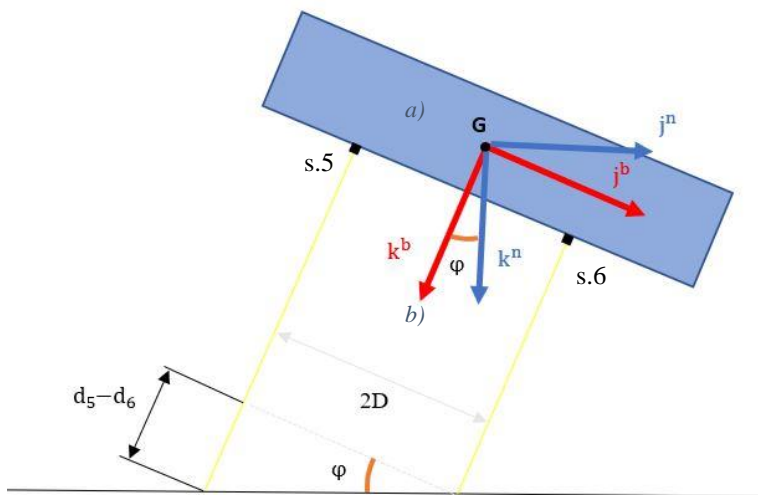


Figura 9: Rappresentazione dei vettori dei sensori 1 e 2.

Questo passaggio è fondamentale, perché più avanti si sfrutteranno le proprietà di rotazione dei vettori per trovare gli angoli di assetto.



Il primo angolo che si può trovare è l'angolo  $\varphi$ , infatti avendo scelto una matrice di rotazione 3-2-1, la rotazione dell'angolo di Rollio è l'ultima in sequenza, quindi non è influenzata dalle altre rotazioni. Con delle semplici considerazioni geometriche possiamo trovare l'angolo  $\varphi$  a partire dalle distanze  $d_5$  e  $d_6$ .



$$\tan(\varphi) = \frac{d_5 - d_6}{2D} \quad (5)$$

$$\varphi = \text{atan} \left( \frac{d_5 - d_6}{2D} \right) \quad (6)$$

Figura 10: Determinazione dell'angolo di  $\varphi$ .

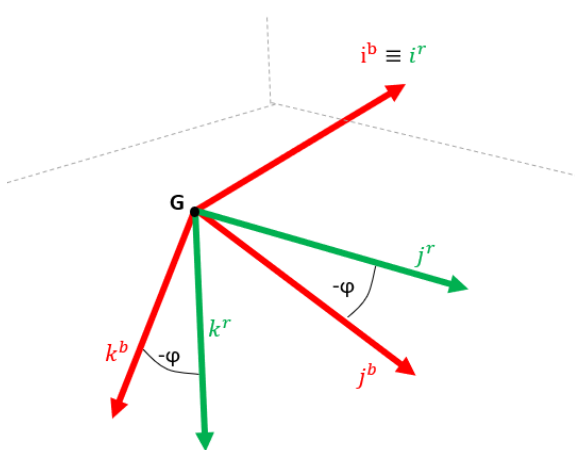


Figura 11: Rotazione di un angolo  $-\varphi$ .

Trovato l'angolo  $\varphi$ , è possibile costruire una matrice di rotazione di un angolo  $-\varphi$ , da applicare ai vettori  $\bar{\mathbf{r}}_4^b$ ,  $\bar{\mathbf{r}}_5^b$ ,  $\bar{\mathbf{r}}_6^b$ . Così facendo otteniamo i vettori espressi in un nuovo sistema di riferimento "r". Questo sistema possiede il versore  $\mathbf{i}^r$  coincidente con  $\mathbf{i}^b$  e gli altri versori ruotati di  $-\varphi$ .

Indichiamo con  $R_b^r$  la matrice di rotazione dal sistema di riferimento Body, al nuovo sistema di riferimento ruotato r:

$$R_b^r = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & \text{sen}(\varphi) \\ 0 & -\text{sen}(\varphi) & \cos(\varphi) \end{pmatrix} \quad (7)$$

Possiamo ora riscrivere i vettori nel nuovo sistema di riferimento “r”:

$$\bar{\mathbf{r}}_4^r = \begin{pmatrix} C \\ (d_4 + K)\text{sen}(\varphi) \\ (d_4 + K)\cos(\varphi) \end{pmatrix} \quad (8)$$

$$\bar{\mathbf{r}}_5^r = \begin{pmatrix} 0 \\ -D\cos(\varphi) + (d_5 + K)\text{sen}(\varphi) \\ D\text{sen}(\varphi) + (d_5 + K)\cos(\varphi) \end{pmatrix} \quad (9)$$

$$\bar{\mathbf{r}}_6^r = \begin{pmatrix} 0 \\ D\cos(\varphi) + (d_6 + K)\text{sen}(\varphi) \\ -D\text{sen}(\varphi) + (d_6 + K)\cos(\varphi) \end{pmatrix} \quad (10)$$

Avendo ora a disposizione i nuovi vettori, si può trovare l’angolo di Pitch considerando solo la componente in direzione di  $\mathbf{k}^r$ :

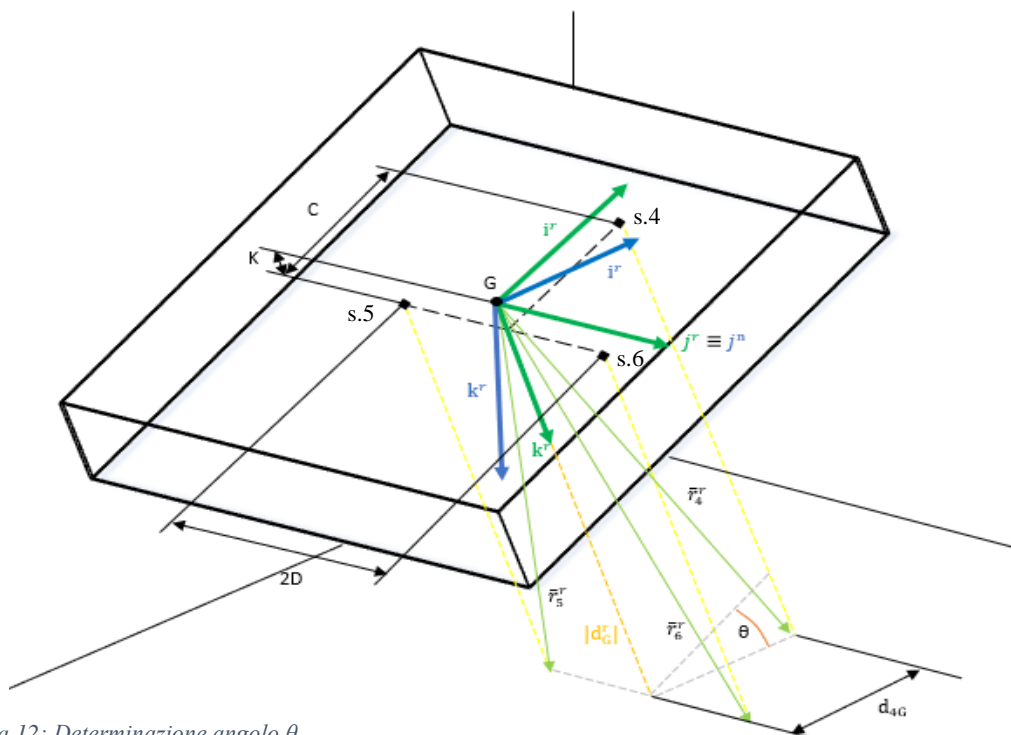


Figura 12: Determinazione angolo  $\theta$ .

$$|\mathbf{d}_G^r| = \frac{(d_6 + d_5 + 2K)\cos(\varphi)}{2} \quad (11)$$

$$\tan(\theta) = \frac{((d_4 + K)\cos(\varphi)) - |\mathbf{d}_G^r|}{C} \quad (12)$$

$$\theta = \text{atan}\left(\frac{((d_4 + K)\cos(\varphi)) - |\mathbf{d}_G^r|}{C}\right) \quad (13)$$

Una volta ricavato  $\theta$ , è possibile definire un nuovo sistema di riferimento “p”, ruotato di un angolo  $-\theta$  e  $-\varphi$ .

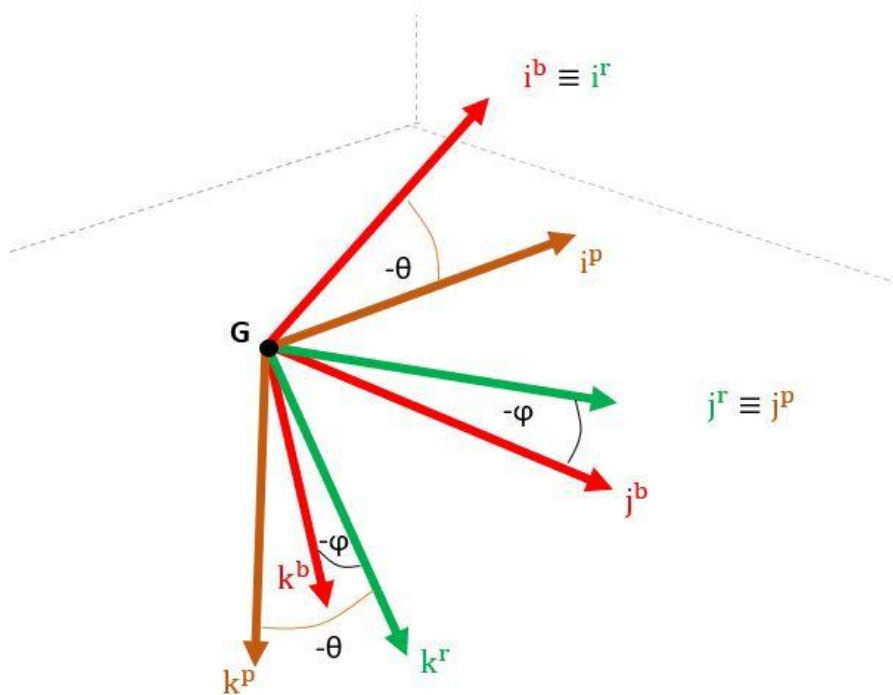


Figura 13: Rotazione SDR di un angolo  $-\varphi$  e  $-\theta$ .

A partire da una costruzione geometrica è possibile ricavare l'angolo  $\psi$ .

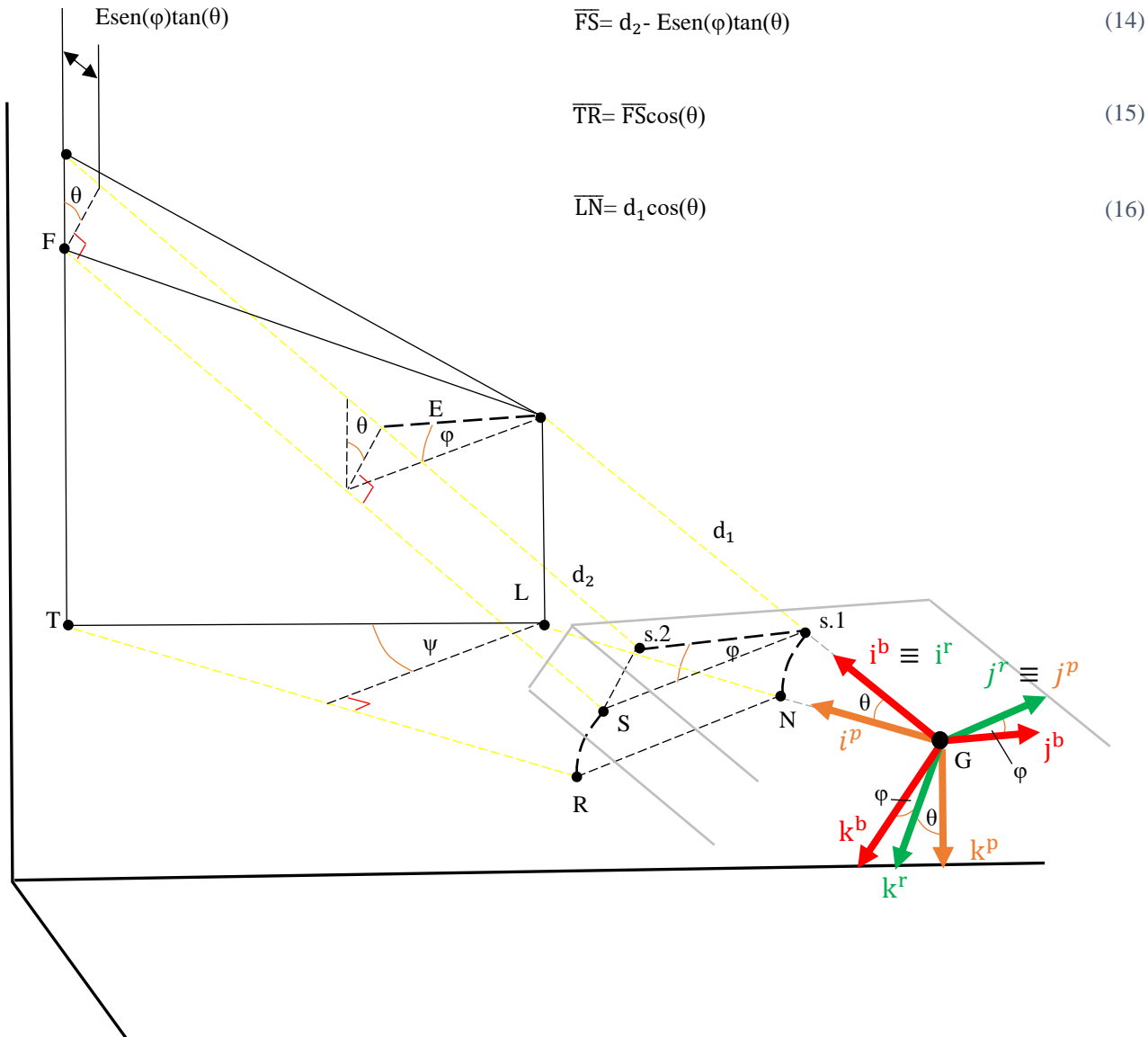


Figura 14: Determinazione angolo  $\psi$ , per semplificazione si suppone solo in questo caso  $H=0$ .

$$\overline{LN} = d_1 \cos(\vartheta) \quad (17)$$

$$\overline{TR} = \left( d_2 - (\text{Esen}(\varphi)\tan(\vartheta)) \right) \cos(\vartheta) \quad (18)$$

$$\psi = \text{atan} \left( \frac{(\overline{LN} - \overline{TR})}{(E|\cos(\varphi)|)} \right) \quad (19)$$

Una volta determinati i tre angoli, possiamo trovare il vettore posizione utilizzando la distanza del sensore 1, del sensore 3 e la distanza  $\mathbf{d}_G^r$  (Equazione 11).

$$X_G = (B + d_1)\cos(\theta)\cos(\psi) - H(\cos(\varphi)\sin(\theta)\cos(\psi) + \sin(\varphi)\sin(\psi)) \quad (20)$$

$$Y_G = (A + d_3)(\sin(\varphi)\sin(\theta)\sin(\psi) + \cos(\varphi)\cos(\psi)) - H(\cos(\varphi)\sin(\theta)\sin(\psi) - \sin(\varphi)\cos(\psi)) \quad (21)$$

$$Z_G = |\mathbf{d}_G^r|\cos(\theta) \quad (22)$$

## 2.5 Emulazione sensori

In questo paragrafo viene descritto il modello matematico utilizzato per emulare i sensori e i relativi errori di misura.

Il modello utilizza gli angoli e il vettore posizionale, che gli sono forniti in ingresso, per determinare le distanze misurate dai sensori durante il loro funzionamento.

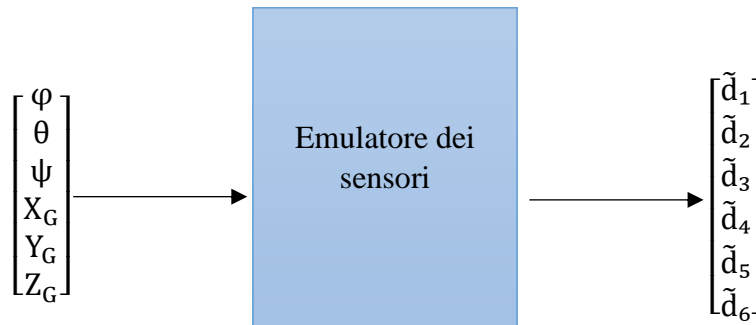


Figura 15: Schema emulatore dei sensori.

L'emulazione viene fatta utilizzando i vettori generati da ogni sensore (Equazione 4) e la matrice di rotazione (Equazione 1).

Per prima cosa, ad ogni vettore si applica la matrice di rotazione in modo da ottenere le proiezioni dei vettori nel sistema di riferimento Assi verticali locali.

Per come abbiamo posizionato ogni sensore, otteniamo che una precisa componente del vettore sensore equivale a una componente del vettore posizionale  $\bar{\mathbf{R}}^s$ .

Prendendo in esame il sensore numero 1 e facendo il prodotto riga per colonna otteniamo;

$$\bar{\mathbf{r}}_1^b = \begin{bmatrix} d_1 + B \\ 0 \\ -H \end{bmatrix} \quad (23)$$

$$\bar{\mathbf{r}}_1^n = \begin{bmatrix} (d_1 + B)\cos(\vartheta)\cos(\psi) - H(\cos(\varphi)\sin(\vartheta)\cos(\psi) + \sin(\varphi)\sin(\psi)) \\ 0 \\ -(d_1 + B)\sin(\vartheta) - H\cos(\varphi)\cos(\vartheta) \end{bmatrix} \quad (24)$$

Osservando la figura 16 si può notare che la prima componente di  $\bar{\mathbf{r}}_1^n$  equivale alla prima componente di  $\bar{\mathbf{R}}^s$ , ovvero  $X_G$ .

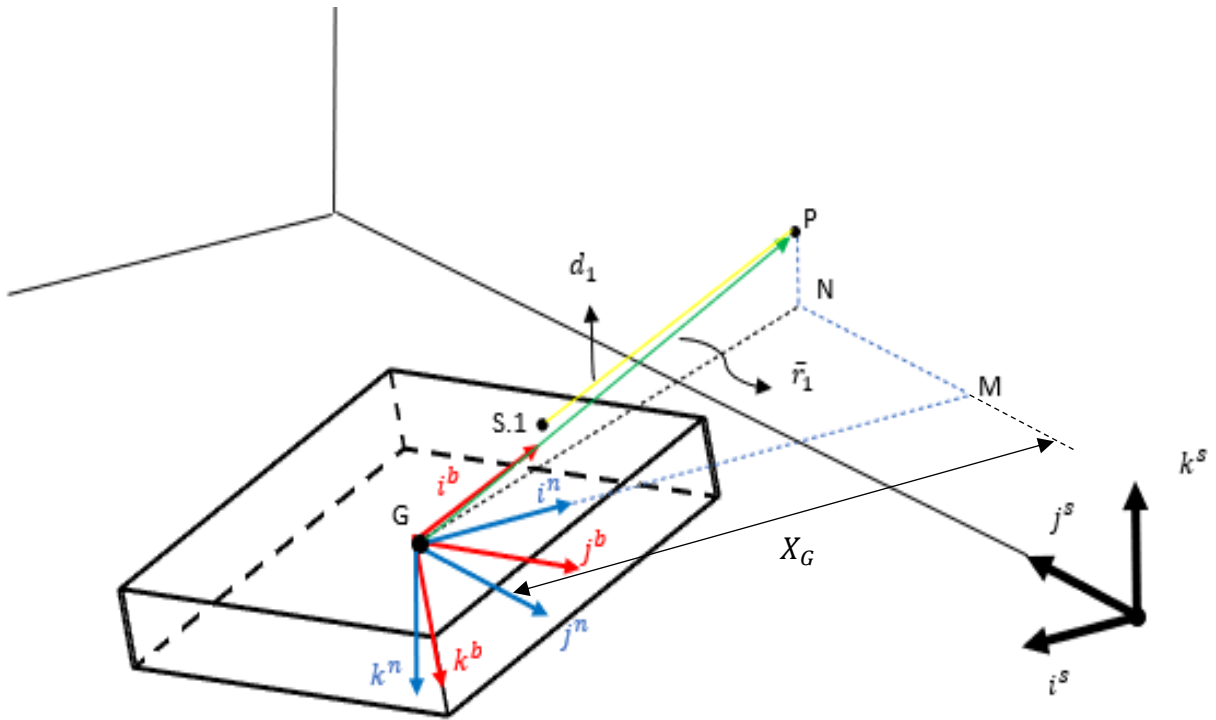


Figura 16: Scomposizione del vettore del sensore.

Eguagliando le due componenti si ottiene:

$$(d_1 + B)\cos(\vartheta)\cos(\psi) - H(\cos(\varphi)\sin(\vartheta)\cos(\psi) + \sin(\varphi)\sin(\psi)) = X_G \quad (25)$$

Ora, avendo a disposizione angoli e componente  $X_G$  si ottiene un'espressione in una sola incognita

$d_1$ , invertibile e facilmente risolvibile.

$$d_1 = \frac{X_G + H(\cos(\varphi)\sin(\vartheta)\cos(\psi) + \sin(\varphi)\sin(\psi))}{\cos(\vartheta)\cos(\psi)} - B \quad (26)$$

Lo stesso procedimento è applicato agli altri vettori, tenendo conto che per i vettori  $\bar{\mathbf{r}}_4^n, \bar{\mathbf{r}}_5^n, \bar{\mathbf{r}}_6^n$  bisogna eguagliare la loro terza componente con  $Z_G$ , mentre per il vettore  $\bar{\mathbf{r}}_3^n$  è necessario imporre la seconda componente di esso uguale a  $Y_G$ .

Otteniamo, infine, 6 equazioni nelle 6 incognite desiderate, qui sotto l'equazione finale per il sensore 1.

$$d_1 = \frac{X_G + H(\cos(\varphi)\sin(\vartheta)\cos(\psi) + \sin(\varphi)\sin(\psi))}{\cos(\vartheta)\cos(\psi)} - B \quad (27)$$

Trovate le equazioni che governano il modello matematico dei sensori e che ci forniscono le distanze desiderate, il passo successivo è emulare gli errori di misura. In accordo con specifiche del sensore preso in esame, si dovrà introdurre un'incertezza di  $\pm 1$  mm sulle misure.

Questo errore è possibile introdurlo nei dati semplicemente aggiungendo in modo del tutto casuale +1,0,-1, ai 6 valori trovati con le equazioni.

L'aggiunta di questi valori si può effettuare con il comando `randi([])` in matlab, questo comando genera un valore intero compreso in un intervallo a nostra scelta, dopo di che andremo ad aggiungere questo valore alle nostre distanze calcolate con le equazioni, ottenendo:

$$\tilde{d}_1 = d_1 + \text{randi}([-1 \ 1], 1)$$

$$\tilde{d}_2 = d_2 + \text{randi}([-1 \ 1], 1)$$

$$\tilde{d}_3 = d_3 + \text{randi}([-1 \ 1], 1)$$

$$\tilde{d}_4 = d_4 + \text{randi}([-1 \ 1], 1)$$

$$\tilde{d}_5 = d_5 + \text{randi}([-1 \ 1], 1)$$

$$\tilde{d}_6 = d_6 + \text{randi}([-1 \ 1], 1)$$

Una volta simulati i disturbi, si è passato ad unire i 3 blocchi principali:

- Simulatore del multirottore implementato nel Laboratorio di meccanica del volo, esso si basa sulla scrittura delle equazioni che governano la dinamica e cinematica di un quadricottero. Come input questo blocco riceve angoli e posizione stimati dal nostro algoritmo, in output fornisce angoli e posizione vera.
- Emulatore dei sensori, questo blocco simula i 6 sensori. Presi in input angoli e posizione, calcola i fasci laser che avremmo con i vari sensori.
- Algoritmo di stima dell'assetto e posizione, quest'ultimo blocco è il sistema vero e proprio che, prese le distanze calcolate dai vari sensori (nel nostro caso sono fornite dall'Emulatore dei sensori e disturbi), ci fornisce angoli e posizione stimata, che possono essere usate per il controllo del velivolo.

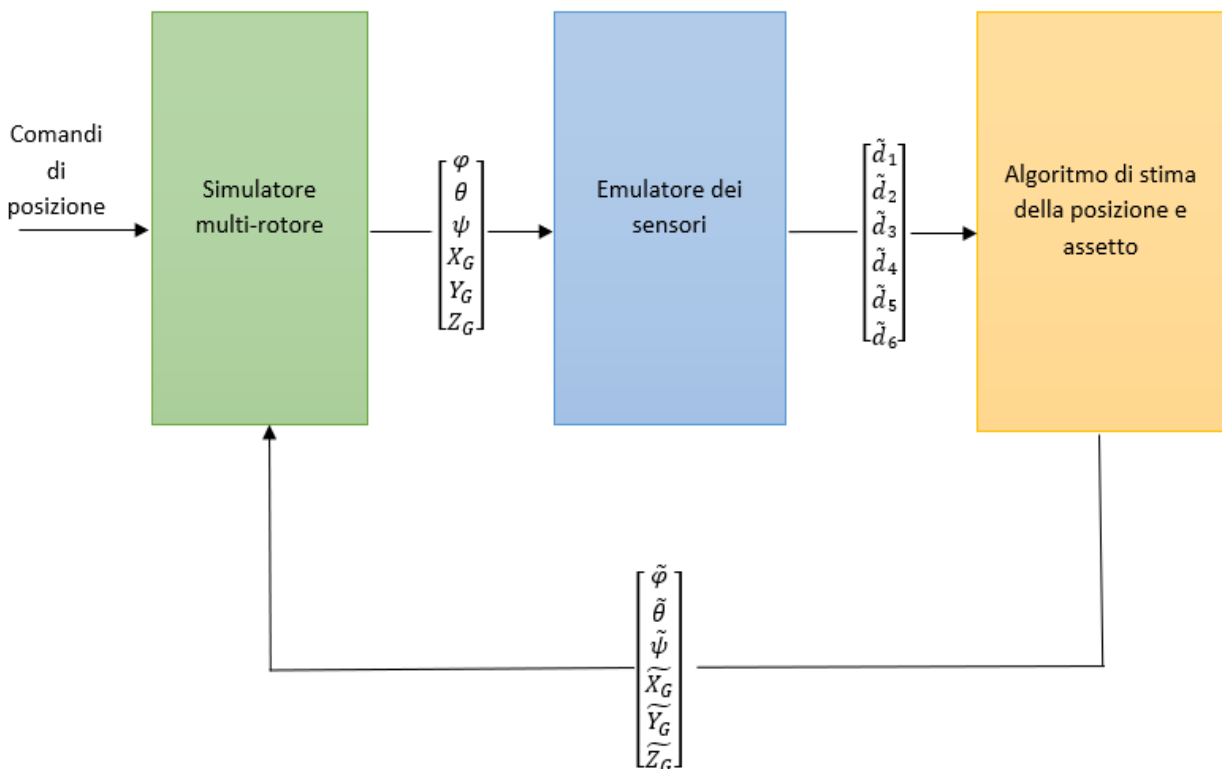


Figura 17: Schema a blocchi del sistema.

## 2.6 Filtraggio dei dati

In ingresso, l'algoritmo di stima e assetto necessita delle 6 distanze fornite dai sensori. Tuttavia, prima di elaborare i dati forniti dall'emulatore dei sensori, si è pensato di inserire un filtro a media mobile.



Questo filtro prende in ingresso gli ultimi N campioni e restituisce, come dato di output, la media di essi. Così facendo il nuovo valore della misura non dipende solo dal valore istantaneo (affetto da disturbi e quindi impreciso), ma dagli ultimi N elementi.

Questa operazione permette di avere dei dati che variano con più continuità senza avere bruschi cambiamenti improvvisi. Di conseguenza le misure fornite dall' algoritmo risultano più stabili.

Un problema legato all'aggiunta di questo filtro è rappresentato da un ritardo nella lettura dei dati. Questo è dovuto al fatto che nelle prime iterazioni risultano dei valori nulli, infatti, non essendoci valori della grandezza prima dell'istante  $t_0$  (istante di inizio simulazione), il filtro di default assegna il valore nullo. Dunque, all'inizio ci potrebbero essere delle criticità nell'ambiente di simulazione, cosa non vera nella realtà fisica perché il multirobotore partendo da fermo è in grado di stabilizzare le misure prima di effettuare il volo.

In Matlab è già implementato un blocco di questo genere chiamato Moving Average, come range di media utilizzeremo gli ultimi cinque valori.

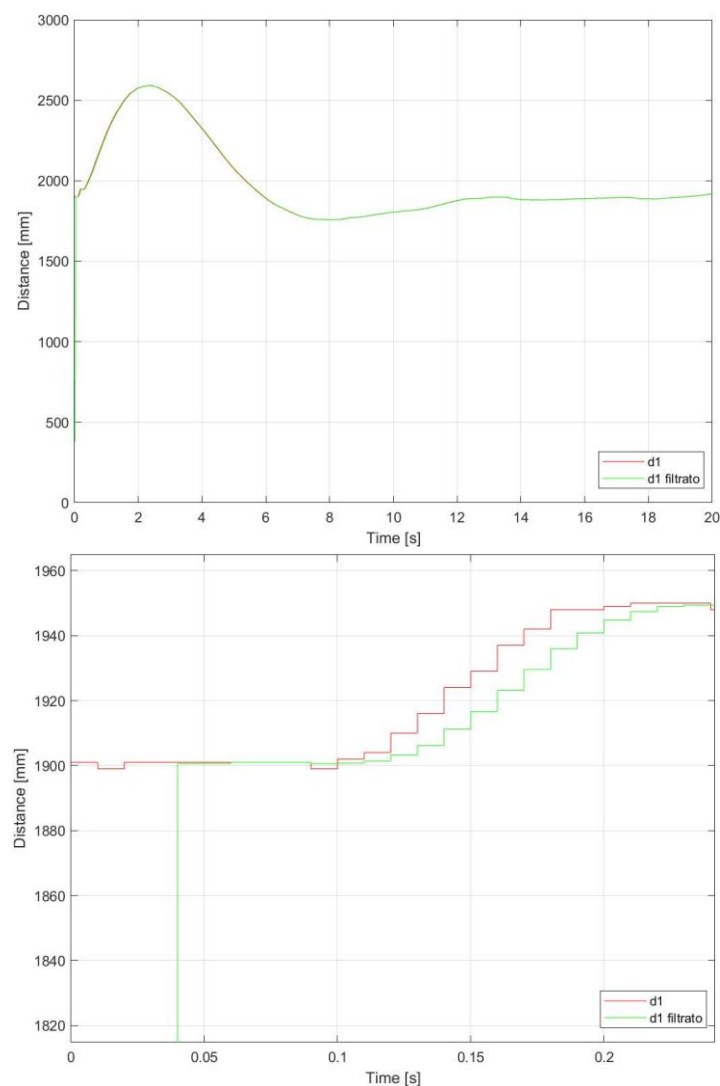


Figura 18: Confronto tra misure del sensore 1: in rosso le misure "grezze" mentre in verde le misure filtrate.

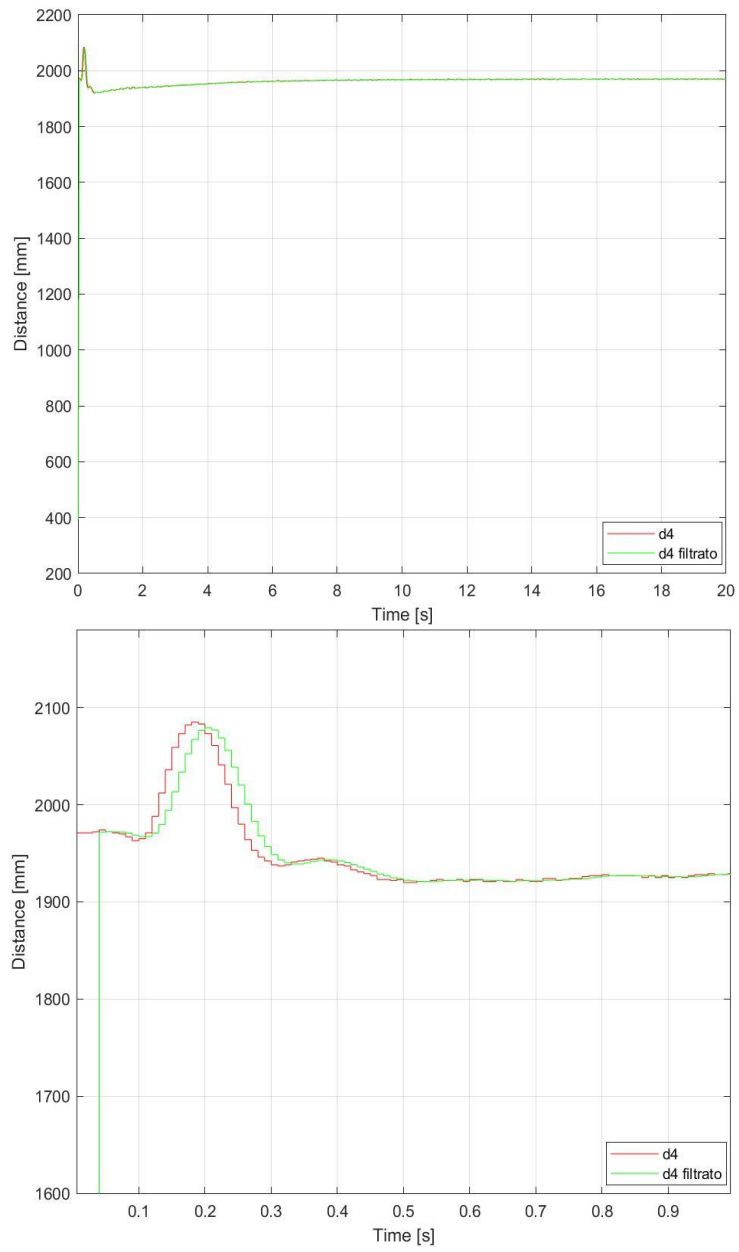


Figura 19: Confronto tra misure del sensore 4: in rosso le misure “grezze” mentre in verde le misure filtrate.

## 3. Realizzazione pratica

### 3.1 Hardware

Un primo passo per la realizzazione del prototipo è la scelta dell'hardware. I sensori scelti permettono di interfacciarsi con una vasta gamma di controllori, dai più complessi, come Raspberry Pi o le Nucleo Board, a sistemi più semplici ed economici come Arduino.

Avendo già familiarità con l'ambiente di lavoro di Arduino e avendo a disposizione una vasta gamma di contenuti open source, la scelta è ricaduta su quest'ultimo.

Per motivi legati all'ingombro che il sistema dovrà avere, la scheda selezionata è un Arduino Nano: compatto e con sufficiente memoria di calcolo per elaborare le equazioni di risoluzione.

Come conseguenza, le equazioni saranno implementate direttamente nell'IDE di Arduino, in un linguaggio simile al C++.

### 3.2 Realizzazione del Case

La configurazione descritta nel capitolo precedente ha prodotto buoni risultati durante le simulazioni al calcolatore. Per tale motivo sarà utilizzata come base di partenza per sviluppare il prototipo del sistema.

A tale configurazione dovranno essere apportate alcune modifiche: questo perché, nell'ambiente di simulazione ciascun sensore è stato modellizzato come un punto matematico, ovvero tralasciando il suo ingombro reale. Occorrerà dunque, tenere in considerazione le reali dimensioni delle shield.

Un altro aspetto da considerare è la progettazione e realizzazione dei supporti per gli stessi sensori. Essi dovranno garantire un preciso orientamento di quest'ultimi nelle varie direzioni, in accordo a quanto detto nel capitolo precedente.

Sempre sfruttando i risultati ottenuti durante il periodo di tirocinio, si è notato come all'aumentare della distanza tra due sensori, ad esempio il sensore 5 e il sensore 6, la risoluzione angolare aumenta, portando ad avere misurazioni più precise. Questo apparente beneficio si scontra però con le reali

dimensioni che il prototipo dovrà avere: per permettere una futura installazione del sistema su un multirottore e allo stesso tempo permettere di svolgere ulteriori compiti a quest'ultimo, il Case non dovrà avere dimensioni ingombranti.

Per riassumere, le nuove posizioni dei sensori sono state determinate tenendo in considerazione diversi fattori:

- Un compromesso tra risoluzione angolare e dimensioni del prototipo.
- Ingombri reali delle shield dei sensori.
- Progettazione dei supporti.

Oltre ai vincoli progettuali che sono forniti dalla natura stessa del progetto, un ulteriore vincolo è relativo alla strumentazione usata, ovvero alla Stampante 3D.

Questo perché, per permettere una stampa più semplice, e adatta alle dimensioni dell'area di stampa, si è valutato di dividere il prototipo in due parti più piccole che andranno poi unite con l'utilizzo di bulloni.

Tenendo presente queste considerazioni, il progetto che è stato realizzato tramite software di disegno lo si può vedere nelle figure successive:

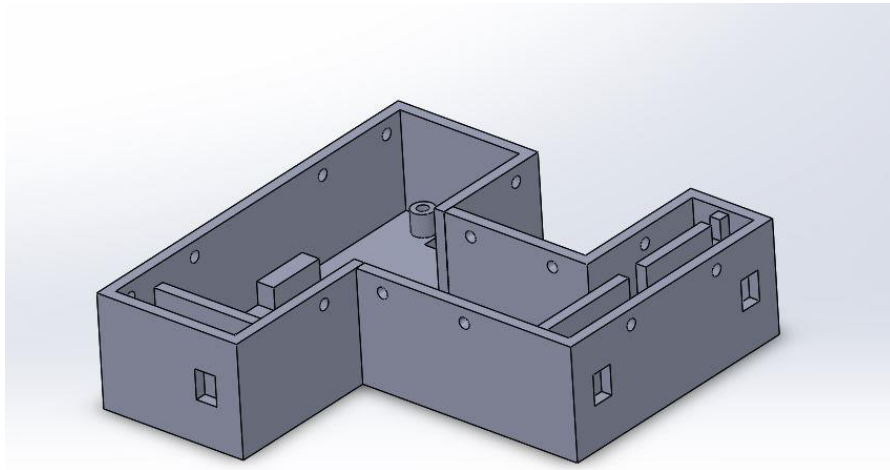


Figura 20: Case completo.

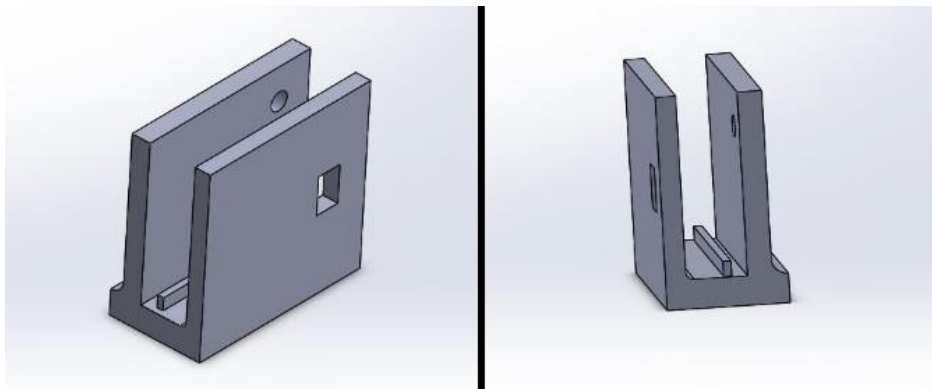


Figura 21: Supporto sensore. Questo è stato realizzato in un primo momento per testare il fissaggio del sensore. Verificata la sua funzionalità, la parte (così definito un corpo solido in SOLIDWORKS) è stata unita al corpo principale del case, per realizzare il disegno finale.

Nelle figure viene mostrata la geometria del Case e il supporto del sensore. Rispetto alla configurazione di partenza (figura 8), le modifiche apportate consistono nello spostamento del sensore 3 in direzione  $\mathbf{i}^b$ , inoltre, anche il sensore 1 risulta spostato in direzione del versore  $\mathbf{j}^b$ . Di conseguenza è possibile riscrivere i vettori dei sensori (equazione 4), che andremo a utilizzare nelle equazioni:

$$\begin{aligned} \bar{\mathbf{r}}_1^b &= \begin{pmatrix} d_1 + B \\ +E \\ -H \end{pmatrix} & \bar{\mathbf{r}}_2^b &= \begin{pmatrix} d_2 + B \\ -S \\ -H \end{pmatrix} & \bar{\mathbf{r}}_3^b &= \begin{pmatrix} T \\ d_3 + A \\ -H \end{pmatrix} \\ \bar{\mathbf{r}}_4^b &= \begin{pmatrix} C \\ 0 \\ d_4 \end{pmatrix} & \bar{\mathbf{r}}_5^b &= \begin{pmatrix} 0 \\ -D \\ d_5 \end{pmatrix} & \bar{\mathbf{r}}_6^b &= \begin{pmatrix} 0 \\ D \\ d_6 \end{pmatrix} \end{aligned} \quad (28)$$

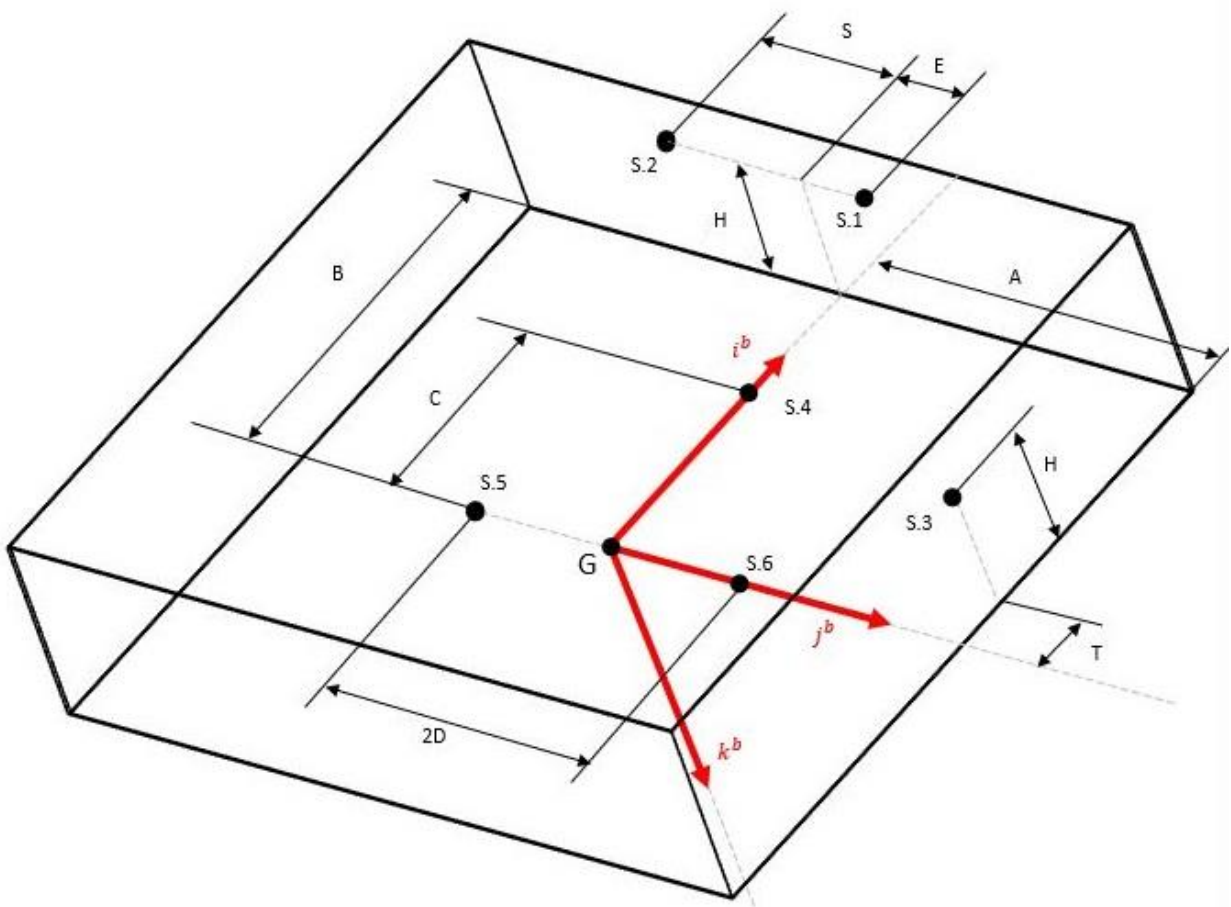


Figura 22: Configurazione dei sensori con modifiche.

Il prototipo mostrato sopra oltre a essere figlio dei requisiti considerati nel capitolo 2 è anche il risultato di test sperimentali.

Infatti, in origine il sensore 1 era ruotato di  $180^\circ$  intorno all'asse  $i^b$ : durante i primi test sperimentali, ci si è accorti che questo posizionamento portava a una interferenza tra il sensore 1 e il sensore 2, disturbando così sia le letture del primo che quelle del secondo.

Questa interferenza era causata dal fatto che il fascio laser di ogni sensore non è un raggio rettilineo, ma un cono, con un'apertura di  $27^\circ$ . A una certa distanza questi coni venivano a sovrapporsi come in figura, alterando le misurazioni.

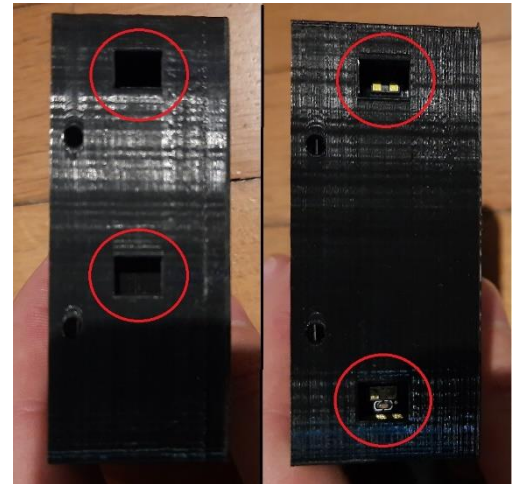


Figura 23: Confronto prima e dopo la modifica.

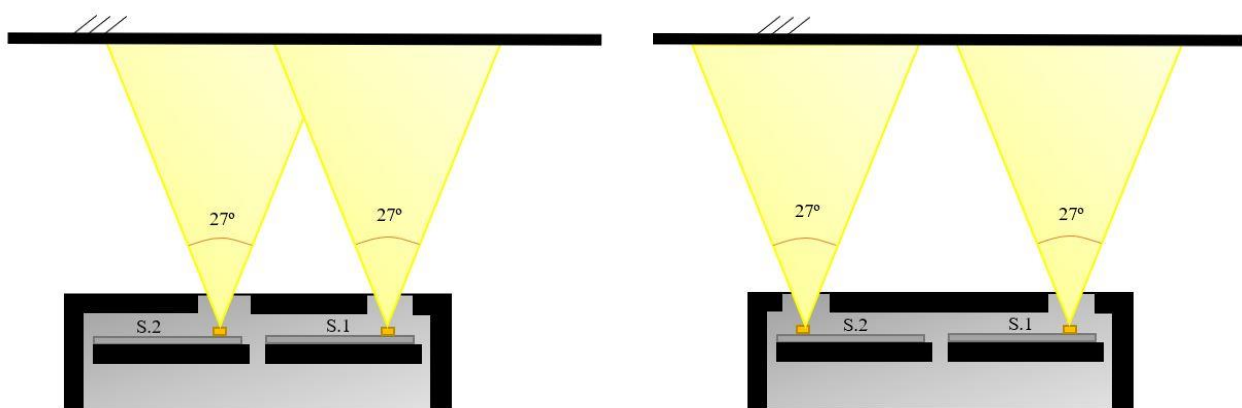


Figura 24: Interferenze tra i due sensori.

Per evitare questo problema si è deciso di allontanare il più possibile i due sensori mantenendo la struttura del Case originale, dunque si è ruotato il sensore 1 di  $180^\circ$ .

Esportando il disegno in ambiente virtuale, con l'estensione STL, lo si può elaborare con un software per la stampa 3D dove gli verranno attribuite le caratteristiche meccaniche (peso, resistenza, finitura) andando a scegliere il materiale e i parametri di stampa.

Il materiale impiegato è il PLA (acido polilattico) e ogni layer ha un'altezza di 0,15 mm, con un ugello dal diametro di 0,4.



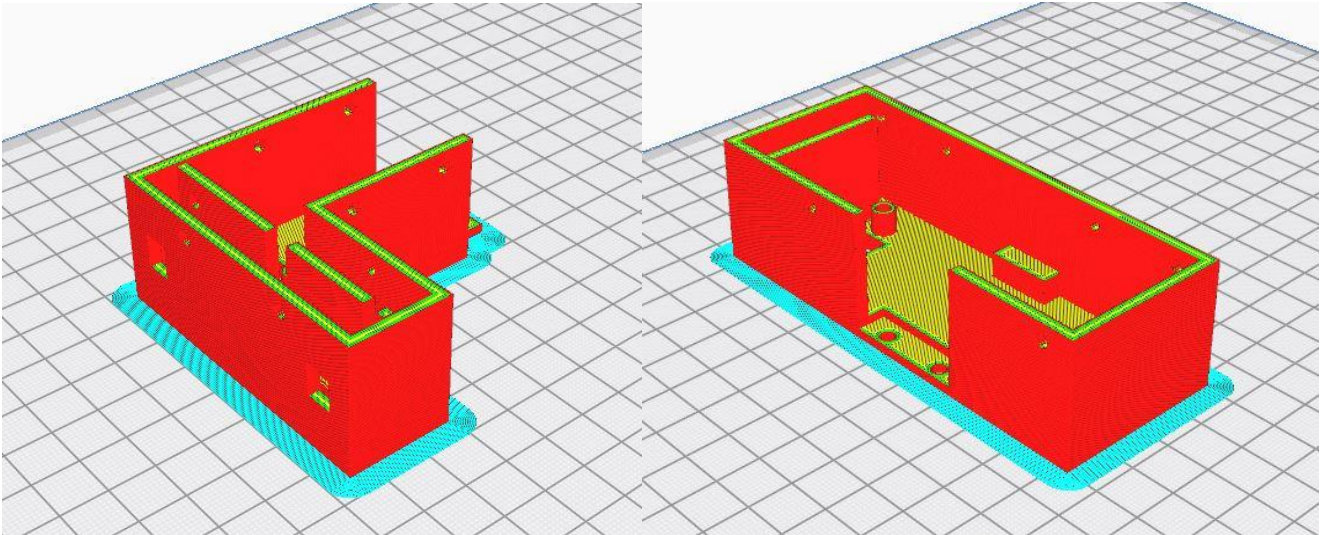


Figura 25: Vista dei layer di stampa.

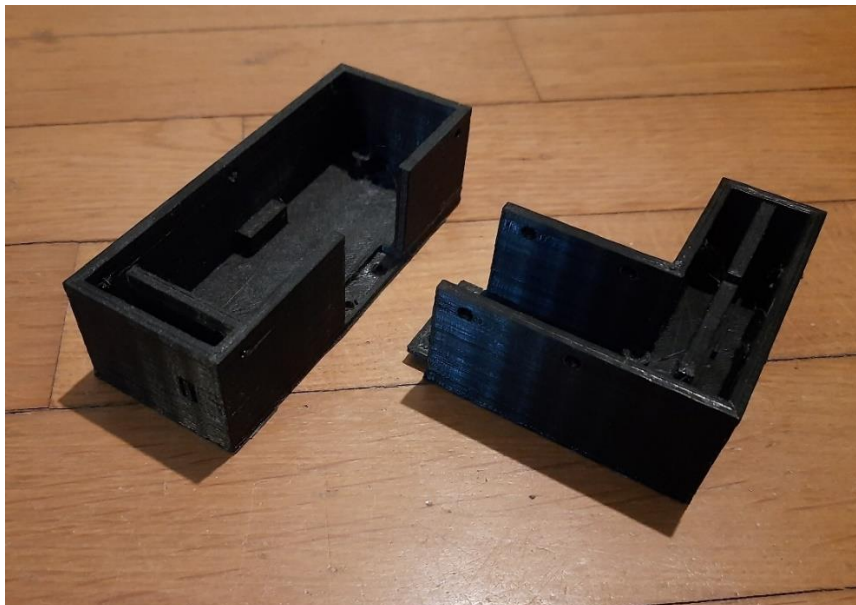


Figura 26: Case completo.

### 3.3 Cablaggio

Ogni sensore ha come ingresso 6 pin, per il nostro utilizzo sarà sufficiente utilizzarne solo 5:

- VCC è il pin che porta l'alimentazione del sensore ovvero +5V
- GND è il pin di massa.
- SDA (Serial Data) è il pin da cui vengono trasmessi o ricevuti i dati

- SCL (Serial Clock) è il pin che fornisce il clock di tutto il sistema
- SHUT è il pin necessario a spegnere il sensore

La gestione dei sei sensori viene effettuata tramite tecnologia I<sup>2</sup>C, ovvero una comunicazione seriale paragonabile a un bus.

Il protocollo I<sup>2</sup>C è una soluzione che prevede dei ruoli precisi, ovvero quello di un Master che controlla, e di “n” Slave che eseguono i comandi impartiti dal Master.

Lo scopo del Master, dunque, è quello di gestire in modo ordinato e senza conflitti tutti i dispositivi Slave collegati alla linea bus. È importante sottolineare che all’interno di un circuito può esserci un solo Master.

Il modo per selezionare un dispositivo Slave è quello di inviare il suo indirizzo sul bus di comunicazione attraverso una sequenza particolare, dunque ogni Slave dovrà avere un suo specifico indirizzo, per permettere una comunicazione chiara. L’indirizzo, di solito, viene fornito durante la fabbricazione del dispositivo ed è il medesimo per dispositivi uguali. Tuttavia, se si vogliono utilizzare più sensori dello stesso tipo, è possibile riassegnarlo tramite opportune procedure.

La comunicazione avviene rispettando precise fasi. Nella fase iniziale, denominata anche sequenza di start, il Master verifica che le linee di SDA e SCL siano a livello alto, questo passaggio serve ad assicurarsi che nessun altro dispositivo sia attivo sulla linea.

Verificata la condizione precedente, il master porta a livello basso la linea di SDA. Questa variazione fa sì che tutti i dispositivi collegati interpretino che la linea bus sia occupata.

A questo punto viene inviato un segnale di clock sulla linea SCL ed esattamente al primo fronte di discesa del segnale di clock, la linea SDA torna alta e viene predisposto l’invio dell’indirizzo.

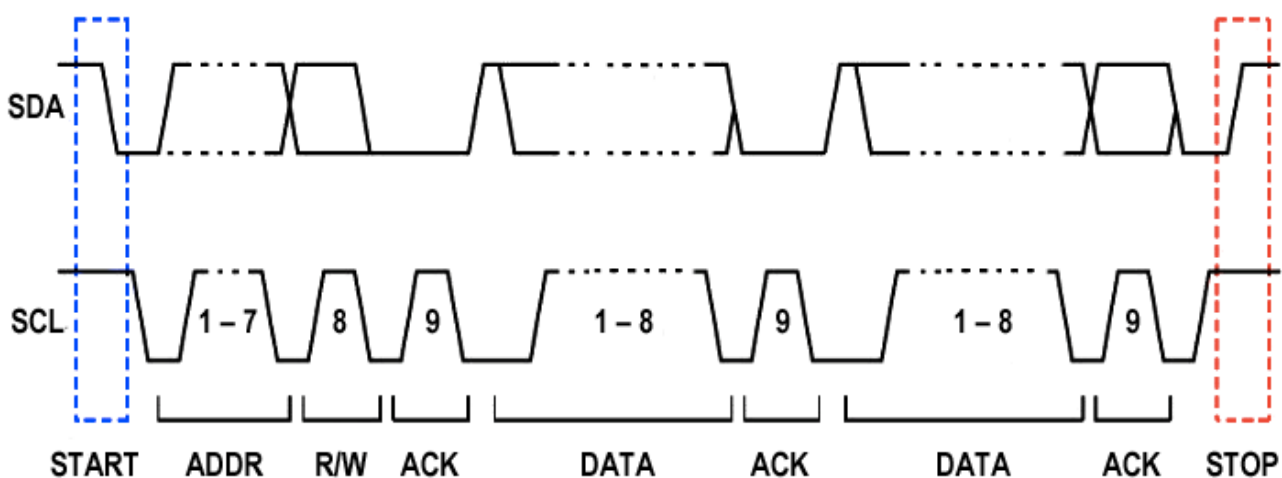


Figura 27: Sequenza del protocollo. Fonte <https://developer.electricimp.com/resources/i2cerrors>



Terminata la sequenza di Start, il master invia l'indirizzo del dispositivo con cui si vuole dialogare, sulla linea SDA. L'indirizzo viene inviato con una sequenza di 7 o 10 bit in base alle specifiche del dispositivo. Alla fine della sequenza però viene inserito un ulteriore bit che specifica se il dispositivo deve inviare (bit 1), o ricevere (bit 0), dati dal Master.

Se tutta la procedura è andata a buon fine, la periferica collegata risponde indicando che ha correttamente ricevuto le informazioni.

A questo punto parte il dialogo tra Master e Slave sotto forma di sequenza da otto bit in base alla richiesta avanzata nella sequenza precedente.

Una regola fondamentale di questa comunicazione è che la linea SDA assume un valore valido solo se la linea SCL è a livello basso. Il livello alto delle linee viene garantito dalle resistenze inserite nel circuito, chiamate resistenze di pullup, ma nel momento in cui un dispositivo viene collegato la linea SCL viene portata a basso livello e quindi è possibile la lettura di bit validi in accordo con il segnale di clock. La velocità di lettura dei dati, dunque, è scandita da quest'ultimo, preimpostato nell'inizializzazione del programma.

Alla fine della comunicazione con un dato dispositivo verrà eseguito un ciclo di stop per determinare la fine della comunicazione.

Questo particolare protocollo permette di semplificare la comunicazione con più dispositivi e di lavorare con il minor numero possibile di cavi, nel nostro caso 5.

Tutti i sensori dovranno condividere la stessa linea VCC e GND, le quali forniscono l'alimentazione vera e propria al sensore. Stessa cosa vale per le linee SDA e SCL che devono essere le medesime per tutti i sensori.

Rimane ora solo da specificare il collegamento dell'ultima uscita utilizzata SHUT: questa, a differenza delle precedenti, deve essere specifica per ogni sensore, dedicando un pin di Arduino a ciascuno di essi.

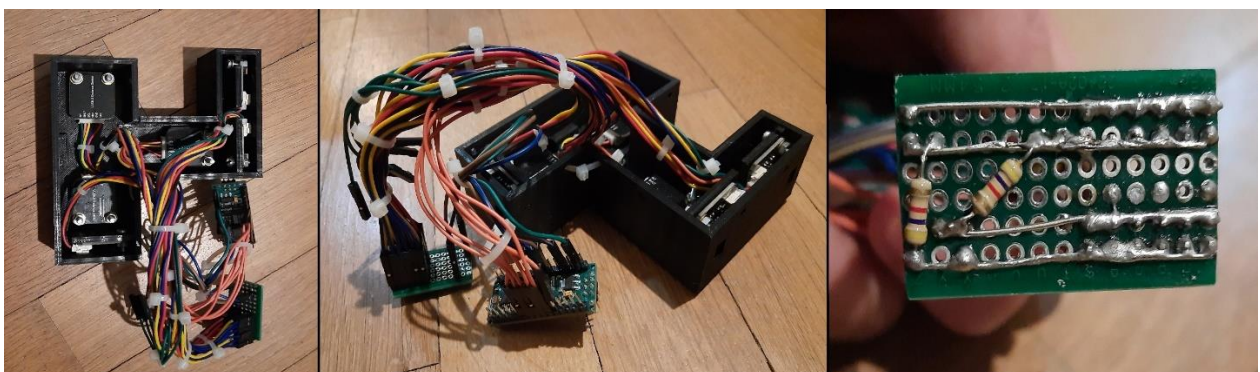


Figura 28: Immagini cablaggio del prototipo.

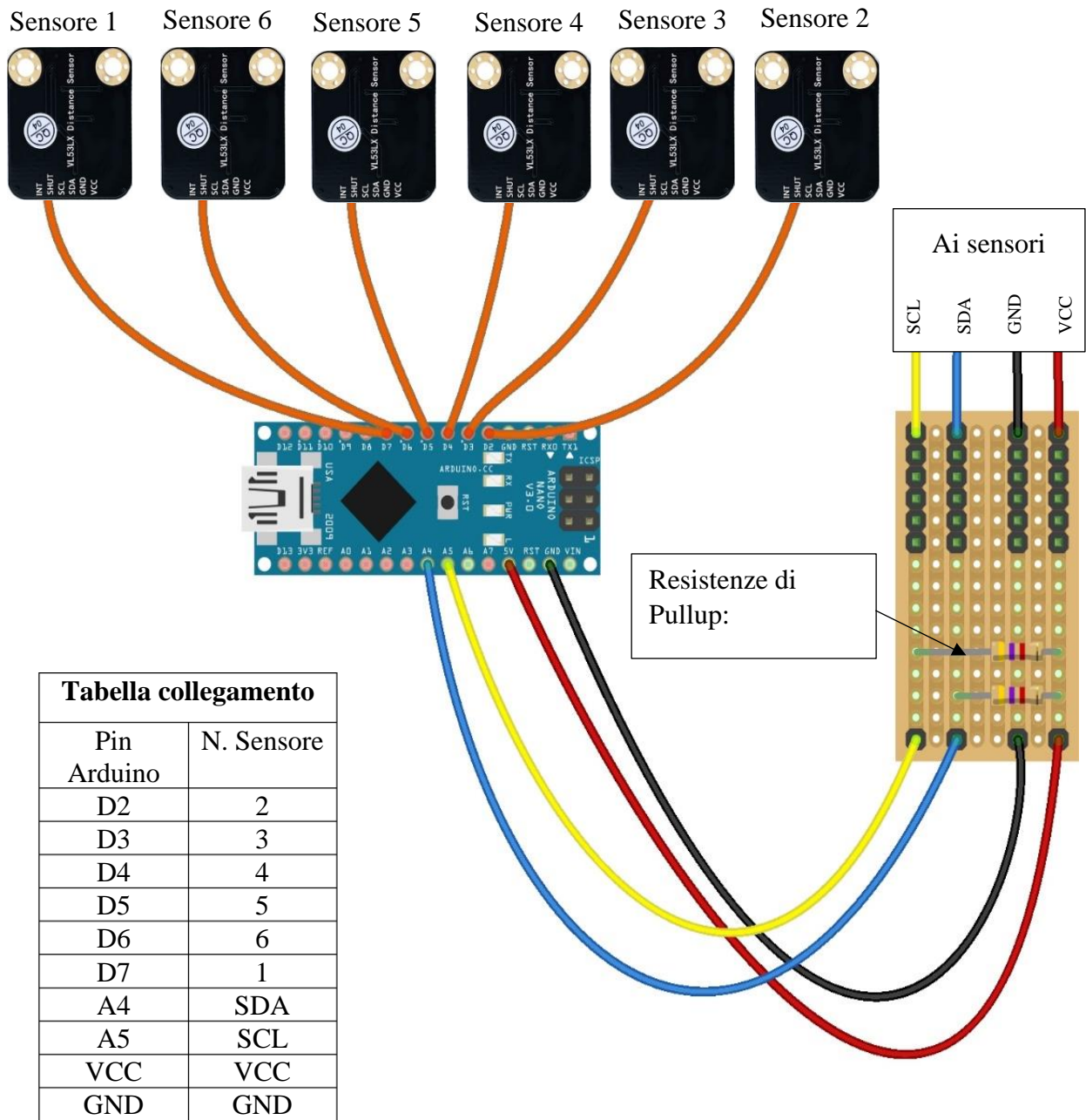


Figura 29: Schema di collegamento.

## 3.4 Software di controllo

I protocolli per la comunicazione presentati nel paragrafo precedente, si possono condensare in poche righe di codice, utilizzando una libreria specifica per il sensore VL53L1X. Questa libreria è disponibile nella finestra “gestione librerie” di Arduino e viene denominata VL53L1X, by Pololu.

Il software si compone di tre parti principali:

1. Assegnazione variabili

In questa sezione vengono inizializzate le variabili e assegnati i pin che si useranno all’interno del programma. Nello specifico viene attribuito a ogni sensore il pin di SHUT, che servirà a riassegnare l’indirizzo dello stesso.

Altri parametri fondamentali sono quelli geometrici (A, B, C, D, E, S, H) che identificano la posizione, in assi Body, di ogni sensore rispetto al punto G.

Le ulteriori variabili sono usate per il filtro a media mobile implementato nel programma.

2. Inizializzazione

La parte di inizializzazione consiste nell’impostare i parametri di misurazione a ogni sensore e a modificare ogni loro indirizzo.

Nello stesso momento vengono anche designati i pin di input, output e la frequenza di Clock del sistema ovvero 400000 Hz.

3. Elaborazione programma

In questa sezione viene eseguito in loop l’algoritmo che ricava gli angoli e la posizione. Per non appesantire la stesura del programma ogni compito è stato diviso in funzioni più piccole, che vengono richiamate ciclicamente.

La prima funzione chiamata rappresenta il filtro a media mobile. Inserendo nelle parentesi i parametri richiesti, la funzione restituisce il valore della distanza filtrato per un dato sensore.

Per ottenere le 6 distanze la funzione viene ripetuta per ogni sensore.

```
unsigned long cal_media(int N, VL53L1X sensor, unsigned long media[], int offset)
```

A questo punto il programma tiene in memoria le distanze calcolate che saranno usate nelle prossime tre funzioni:

```
float find_phi (int D, unsigned long d_5, unsigned long d_6)
float find_theta (int C, unsigned long d_5, unsigned long d_6, unsigned long d_4, float phi)
float find_psi (int E, int S, unsigned long d_1, unsigned long d_2, float phi, float theta)
```

Le tre funzioni eseguite in sequenza calcolano gli angoli di assetto espressi in radianti.

In fine le ultime tre funzioni eseguite sono quelle per la determinazione della posizione, questo perché il calcolo del vettore posizione richiede che si sia già a conoscenza degli angoli di assetto.

```
unsigned long find_XG(unsigned long d_2,float phi, float theta,float psi, int B, int E, int H)
unsigned long find_YG(unsigned long d_3,float phi, float theta,float psi, int T, int A, int H)
unsigned long find_ZG(unsigned long d_4,float phi, float theta,float psi, int C )
```

Per rendere il programma più comprensibile, qui sotto sono riportate alcune funzioni principali della libreria, il tutto viene specificato per il sensore 1 (useremo la denominazione sensore\_1).

```
#include <VL53L1X.h>
```

Questo comando permette di includere librerie esterne ad Arduino. Nel nostro caso andremo a includere la libreria specifica per il sensore che utilizzeremo.

```
VL53L1X sensore_1;
```

In questo modo specifichiamo che la variabile sensore\_1 è un sensore VL53L1X, perciò d'ora in poi sarà trattato come un oggetto sensore.

```
sensore_1.setTimeout(500);
```

Viene impostato il tempo massimo di attesa da parte del master durante la fase di inizializzazione.

```
sensore_1.setAddress(42);
sensore_1.setDistanceMode(VL53L1X::Medium);
sensore_1.setMeasurementTimingBudget(40000);
sensore_1.startContinuous(40);
```

Grazie a questi comandi viene riassegnato l'indirizzo del sensore (intervallo utilizzabile da 42-54), successivamente impostata una configurazione Medium di misura e infine settato il tempo di attesa minimo per ogni sensore.

```
media[0]=sensore.read();
```

Infine, abbiamo il comando utilizzato per leggere la misura del sensore.

All'interno del programma sono state implementate le equazioni per la stima della posizione e assetto, apportando alcune modifiche rispetto a quelle ricavate nel capitolo 2.

Per trovare l'angolo  $\varphi$  si utilizza sempre l'equazione 5, mentre per gli altri due angoli,  $\theta$  e  $\psi$ , alcuni termini delle equazioni 13 e 19 si possono elidere.

$$\varphi = \text{atan} \left( \frac{d_5 - d_6}{D} \right) \quad (29)$$

$$\theta = \text{atan} \left( \frac{((d_4)\cos(\varphi)) - |(d_6 + d_5)\cos(\varphi)/2|}{C} \right) \quad (30)$$

$$\psi = \text{atan} \left( \frac{d_1 \cos(\vartheta) - (d_2 - ((E + S)\text{sen}(\varphi)\tan(\vartheta))) \cos(\vartheta)}{((E + S)|\cos(\varphi)|)} \right) \quad (31)$$

A causa del riposizionamento dei sensori è possibile rimuovere dalle prime due equazioni il termine  $K$  e aggiungere il termine  $S$  nell'ultima. Nel primo caso, osservando l'equazione 28 si può notare che nei vettori  $\bar{\mathbf{r}}_4^b$ ,  $\bar{\mathbf{r}}_5^b$ ,  $\bar{\mathbf{r}}_6^b$  la componente  $K$  in direzione del versore  $\mathbf{k}^b$  non è più presente rispetto alla configurazione mostrata in figura 8. Nel secondo caso, invece, rispetto all'equazione 4, nel vettore  $\bar{\mathbf{r}}_1^b$  è stata aggiunta una componente in direzione  $\mathbf{j}^b$ .

Discorso analogo lo si può fare per le equazioni per ricavare il vettore di stato  $\bar{\mathbf{R}}^s$ . partendo dalle equazioni 20, 21, 22 si può scrivere:

$$X_G = (B + d_2)\cos(\theta)\cos(\psi) - S(\text{sen}(\varphi)\text{sen}(\theta)\cos(\psi) - \cos(\varphi)\text{sen}(\psi)) - H(\cos(\varphi)\text{sen}(\theta)\cos(\psi) + \text{sen}(\varphi)\text{sen}(\psi)) \quad (32)$$

$$Y_G = T(\cos(\theta)\text{sen}(\psi)) + (A + d_3)(\text{sen}(\varphi)\text{sen}(\theta)\text{sen}(\psi) + \cos(\varphi)\cos(\psi)) - H(\cos(\varphi)\text{sen}(\theta)\text{sen}(\psi) - \text{sen}(\varphi)\cos(\psi)) \quad (33)$$

$$Z_G = -C\text{sen}(\theta) + d_4(\cos(\varphi)\cos(\theta)) \quad (34)$$

Per come sono state scritte le equazioni i vari parametri geometrici devono essere inseriti in valore assoluto. In caso si volesse spostare un sensore e uno dei suoi parametri cambiasse direzione, basterà inserire il nuovo parametro con un segno meno.

A dimostrazione di ciò, consideriamo per ora solo il vettore  $\bar{\mathbf{r}}_1^b$ :

$$\bar{\mathbf{r}}_1^b = \begin{pmatrix} d_1 + B \\ +E \\ -H \end{pmatrix} \quad (35)$$

In accordo con quello detto sopra i parametri E ed H devono essere inseriti nelle equazioni con segno positivo.

Supponiamo ora di modificare la posizione di  $\bar{\mathbf{r}}_1^b$ , in particolare invertendo la sua componente in direzione  $\mathbf{j}^b$ :

$$\bar{\mathbf{r}}_1^b = \begin{pmatrix} d_1 + B \\ -E \\ -H \end{pmatrix} \quad (36)$$

In questo caso occorrerà sempre inserire un parametro H positivo, mentre il parametro E, dovrà avere un segno meno per garantire la validità delle equazioni implementate.

### 3.5 Filtraggio delle misure

Testando i sensori singolarmente si è potuto notare come le misure fornite da ciascuno presentino un andamento oscillatorio casuale, anche se il sensore è mantenuto fermo.

Questo può in parte essere dovuto a disturbi esterni quali la variazione significativa della luce dell'ambiente, o una cattiva riflessione della superficie, che fa sì che arrivino segnali in ritardo o anticipo.

Inoltre, il problema può anche essere dovuto al tipo di hardware utilizzato e al relativo software, infatti, è possibile che utilizzando un hardware migliore, o un'altra libreria di controllo questi problemi non sussistano. Tuttavia, all'inizio del progetto è stata fatta una scelta ben precisa dell'hardware da utilizzare.

Quindi, la soluzione adottata per annullare la variazione è inserire un filtro a media mobile, replicando così il lavoro svolto durante il tirocinio, dove si era inserito un filtro di questo genere in entrata al blocco dell'algoritmo.

La differenza, rispetto al filtro usato nell'ambiente virtuale, sta nel numero di campioni N che vengono utilizzati. Infatti, non esiste un numero preciso da utilizzare ma il numero varia a seconda del tipo di sensore utilizzato.

Dopo varie prove effettuate per diversi valori di N si è trovato un range, da 10 a 15, nel quale le misure si stabilizzano, variando solo di 1/2 mm.

Al di sopra di questi valori non si nota alcun miglioramento significativo, viceversa al di sotto, si ha ancora una forte variabilità dei dati.

```
unsigned long cal_media(int N, VL53L1X sensor ,unsigned long media[], int offset){  
  
for(int i=(N-1); i>=1;i--){  
    media[i]=media[i-1];  
}  
  
unsigned long totale=0;  
media[0]=sensor.read();  
  
for( int i=0; i<N; i++){  
    totale=totale+media[i];  
}  
unsigned long Val=totale/(N);  
unsigned long valore_c=((0.00000000555*(Val^3))-(0.00001921595*(Val^2))+(1.029318598154028*Val)+14.55552824191158)+offset;  
delay(1);  
return valore_c;  
  
}
```

### 3.6 Caratterizzazione dei sensori

Le misure ottenute dai sensori non sono ancora utilizzabili all'interno del programma, questo perché ci sono errori dovuti alla non linearità della curva caratteristica del sensore.

La soluzione adottata consiste nell'effettuare una serie di misure a delle distanze note, in questo modo si avranno due serie di dati: una con le distanze vere e prestabilite, l'altra con le corrispondenti misure effettuate dai sensori.

Ottenuti questi dati si cerca di trovare, tramite metodi di approssimazione, una nuova curva caratteristica di ordine N che meglio approssima, ai minimi quadrati, le misure sperimentali.



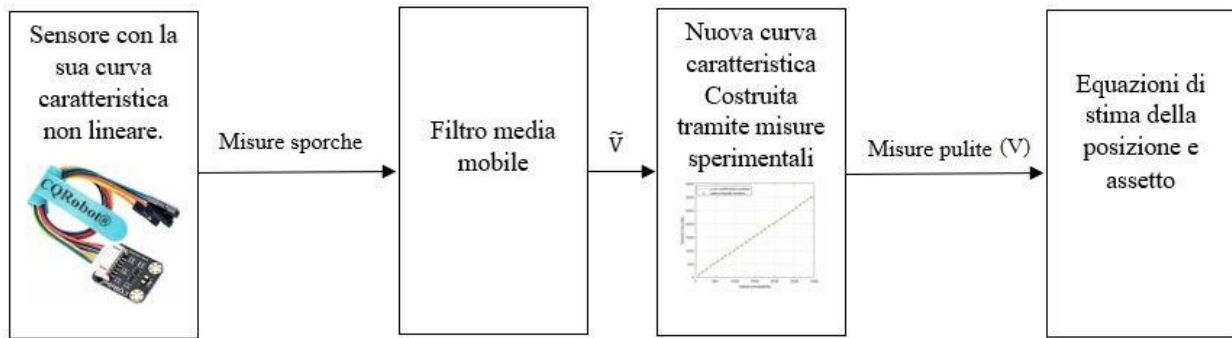


Figura 30: Schema a blocchi dell'elaborazione delle misure.

Ripetendo varie volte il procedimento con differenti gradi del polinomio di approssimazione, si è potuto notare che i coefficienti oltre il grado 3, sono dell'ordine di  $10^{-15}$ , dunque per i nostri scopi sono del tutto trascurabili.

Qui sotto vengono riportati i dati con i quali si è ricavata l'equazione caratteristica:

<b>Tabella misurazioni sperimentali</b>										
Riferimento (mm)	100	200	300	400	500	600	700	800	900	1000
Misure (mm)	86	180	274	386	471	571	671	766	869	074
Riferimento (mm)	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
Misure (mm)	1071	1171	1272	1372	1465	1569	1663	1760	1858	1958
Riferimento (mm)	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
Misure (mm)	2056	2152	2252	2352	2449	2546	1246	1735	1830	1922

Tabella 1: Misurazioni sperimentali.

Attraverso, dunque, metodi numerici si è trovata una curva di terzo grado:

$$V = 14.555 + 1.029\tilde{V} + (1.921 \times 10^{-5})\tilde{V}^2 + (5.551 \times 10^{-9})\tilde{V}^3 \quad (37)$$

Con  $\tilde{V}$  vengono intese le misure che arrivano in uscita dal filtro a media mobile, mentre  $V$  rappresenta il valore pulito. In realtà, occorrerebbe eseguire questa procedura per ogni sensore, infatti, ciascuno di essi potrebbe avere problemi di non linearità differenti. Tuttavia, eseguendo



alcune misurazioni è risultato che tutti i sensori hanno caratteristiche sugli errori di non linearità abbastanza simili. Quindi, l'equazione sarà unica e applicata a ognuno di essi.

Ciò non toglie la possibilità di poter caratterizzare in futuro ciascun sensore con la propria curva sperimentale, attraverso uno studio più approfondito dell'argomento.

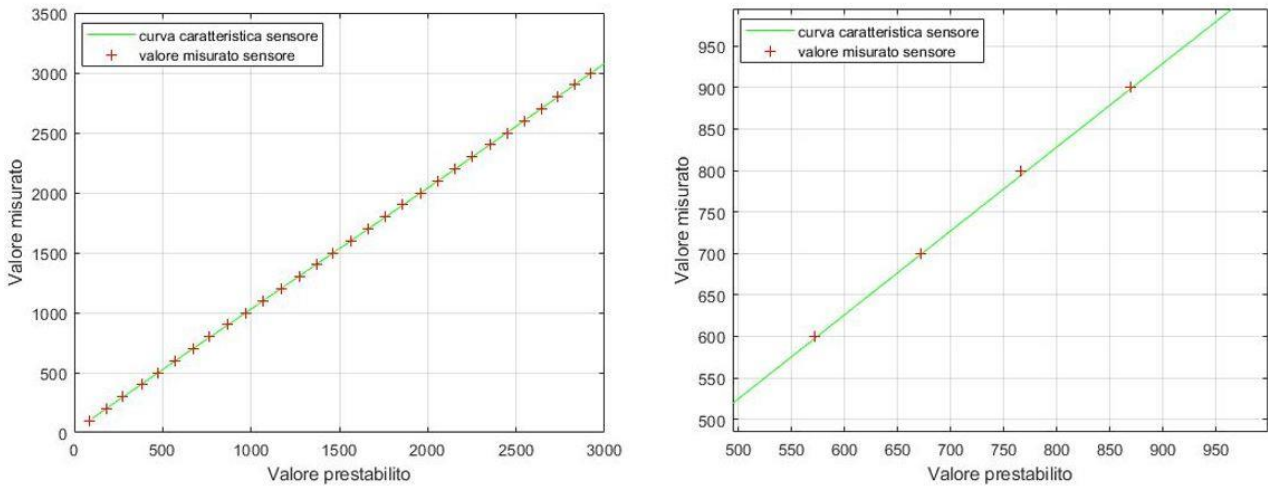


Figura 31: A sinistra la curva di caratterizzazione del sensore a destra un suo zoom.

Occorre però fare un'ultima precisazione. Nei precedenti paragrafi si è parlato della forte dipendenza dei sensori dall'ambiente esterno, è bene quindi prevedere un offset specifico per ogni sensore da aggiungere al risultato dell'equazione sopra.

$$V = 14.555 + 1.029\tilde{V} + (1.921 \times 10^{-5})\tilde{V}^2 + (5.551 \times 10^{-9})\tilde{V}^3 + \text{offset} \quad (38)$$

Ogni qual volta si effettueranno voli in ambienti diversi occorrerà posizionare il sistema a delle distanze note e controllare se i valori forniti dai sensori corrispondono ad esse, se così non fosse, basterà semplicemente assegnare alla variabile offset la differenza tra esse.

## 4. Validazione risultati

### 4.1 Simulazione numerica

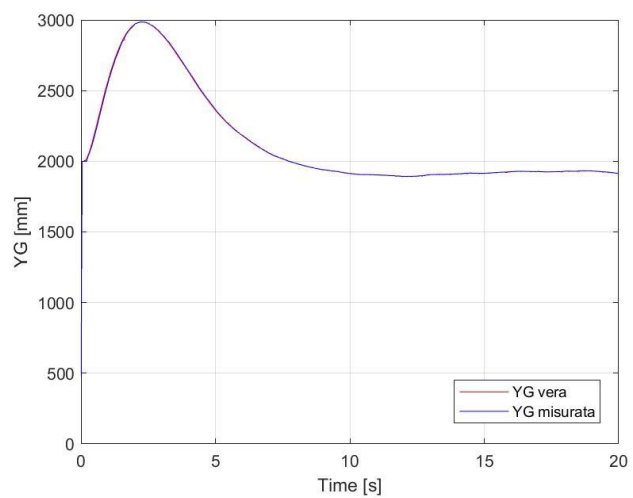
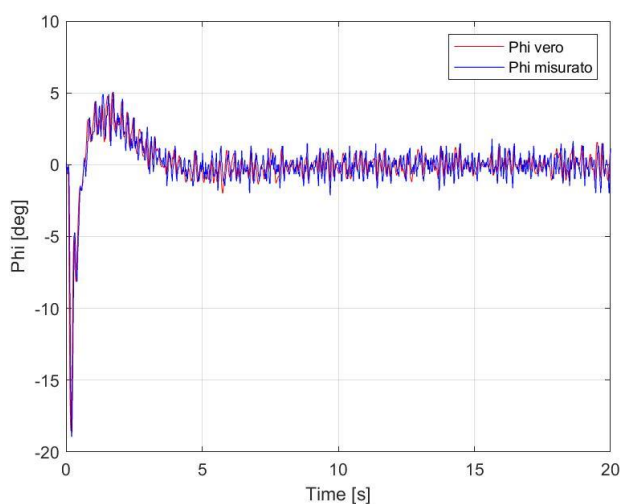
In questo paragrafo vengono riportate alcune simulazioni del sistema implementato al calcolatore con lo scopo sia di verificare il modello matematico implementato, sia di fornire delle basi di partenza per il suo futuro sviluppo.

La simulazione consiste nel far compiere al multirottore alcune manovre desiderate, in modo da confrontare le grandezze fornite dal simulatore con le grandezze stimate dal sistema implementato.

Una prima manovra simulata rappresenta una stabilizzazione in condizioni di hovering.

Questa manovra consiste nel mantenere il multirottore fisso in una posizione dello spazio. In particolare, la posizione iniziale del baricentro del velivolo, rispetto agli Assi Coordinati, sarà di 2 metri in ogni direzione.

Grazie alle misure fornite dall'algorithm di stima e assetto ai controllori, il sistema sarà in grado di mantenere la posizione.



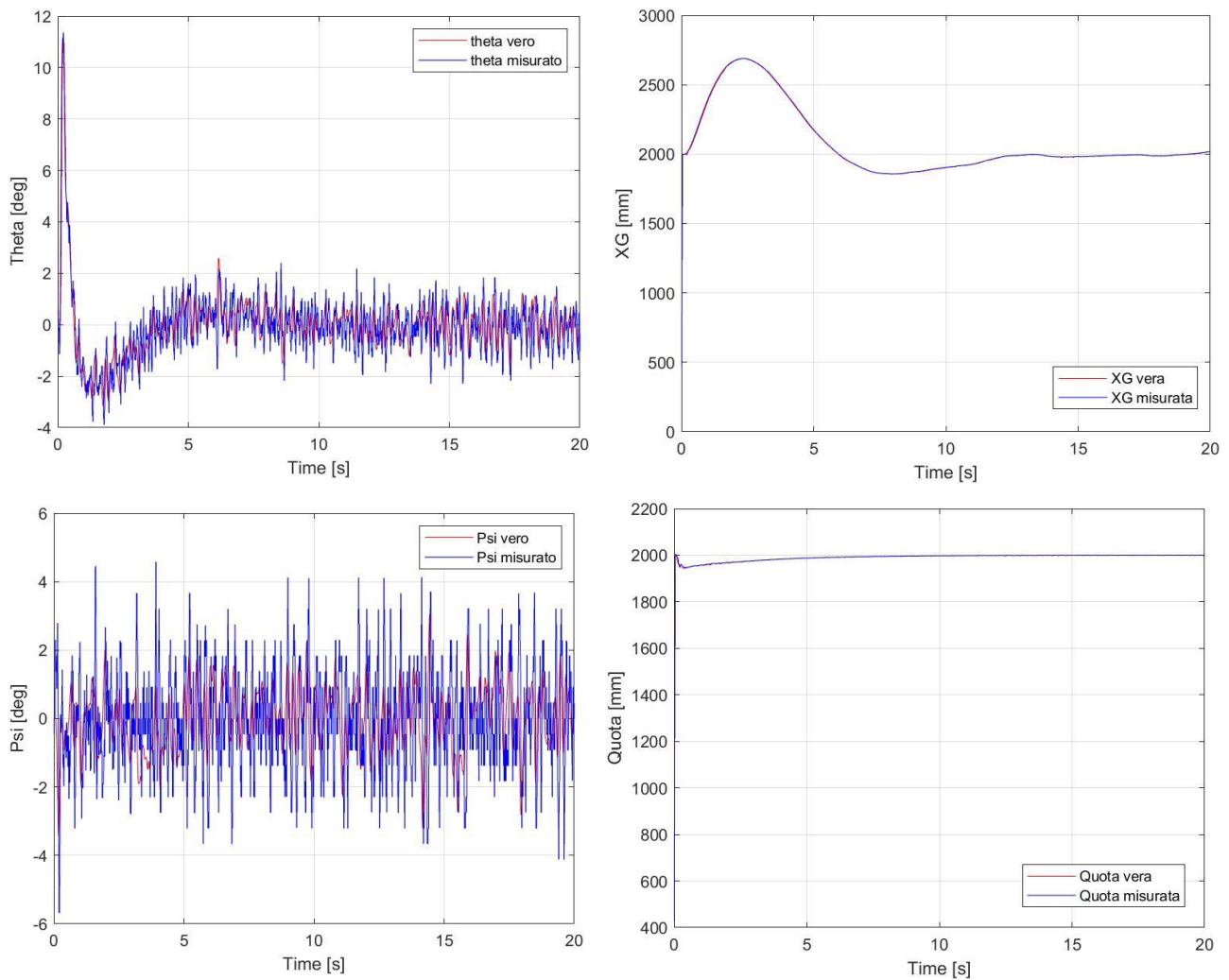


Figura 32: Simulazione al calcolatore della manovra di hovering.

All'inizio abbiamo delle forti oscillazioni con un periodo di esse relativamente grande: questo è dovuto al fatto che nei primi istanti della simulazione le letture fornite dall'algoritmo ai controllori sono errate a causa del filtro a media mobile. Di conseguenza, si genererà una correzione da parte dei controllori che in realtà non sarebbe necessaria.

In aggiunta a questo errore, un altro aspetto da tenere in considerazione sono i parametri impostati all'interno dei controllori. Questi ultimi infatti potrebbero generare una lenta risposta ai disturbi che porterebbe il sistema a stabilizzarsi solo dopo transitori abbastanza lunghi.

Nonostante ciò, la stabilità del sistema è garantita dal fatto che nel lungo periodo il velivolo si stabilizza nella posizione desiderata.

Il risultato principale da osservare è l'andamento delle misure reali, fornite dal simulatore del multiroto, confrontate con i dati forniti dall'algoritmo implementato. Questi tendono a coincidere, validando così le equazioni implementate per ricavare posizione e assetto.

I prossimi grafici mostrano un innalzamento di quota di 1 m.

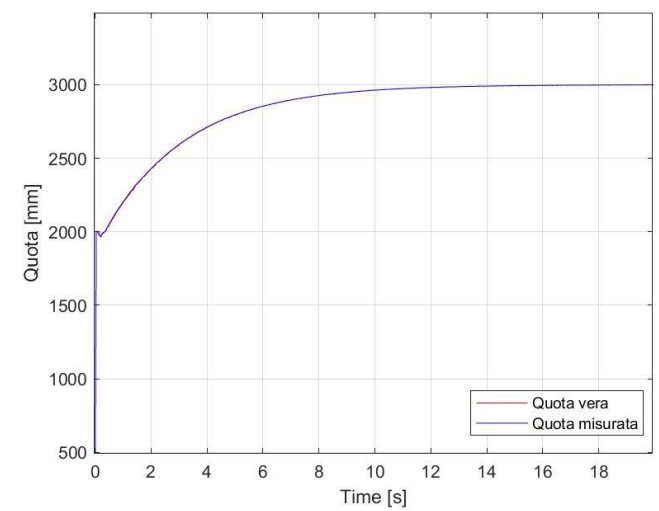
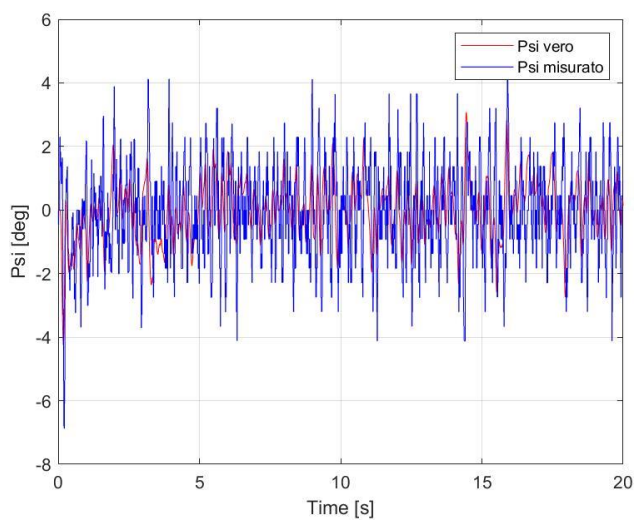
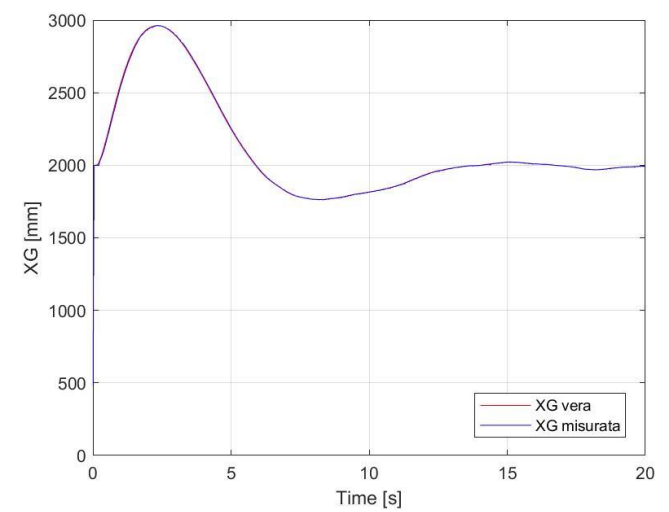
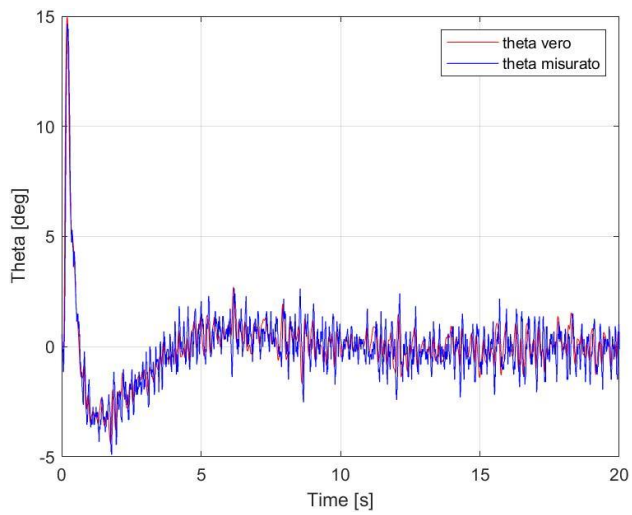
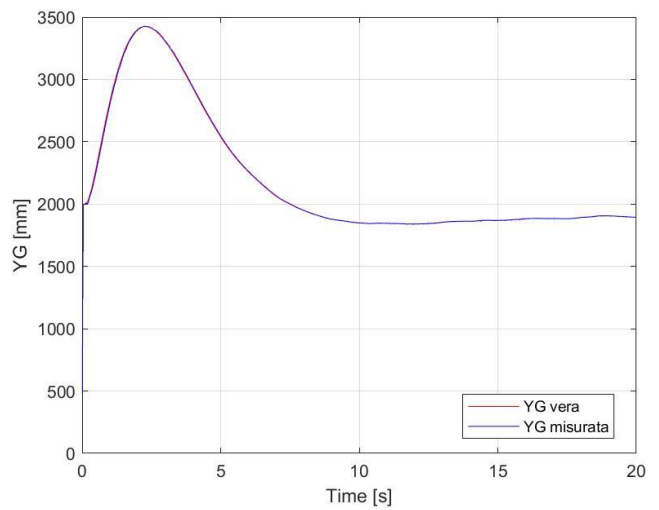
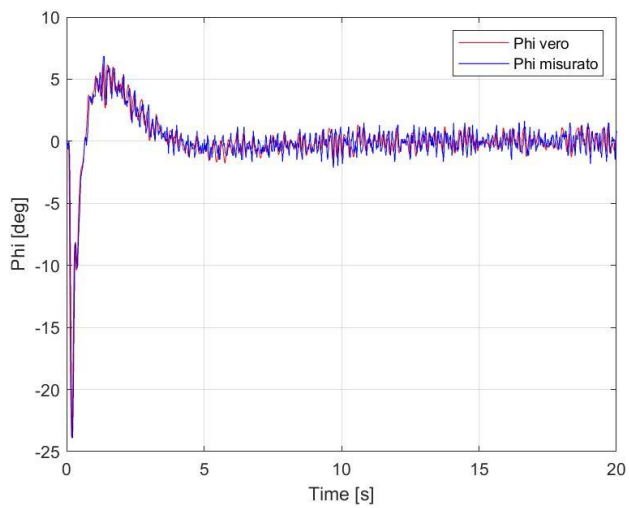


Figura 33: Innalzamento quota.

Anche in questo caso notiamo sempre delle oscillazioni che nel lungo periodo vanno ad attenuarsi.

Un'ulteriore manovra eseguita in simulazione è l'aumento della coordinata  $X_G$ .

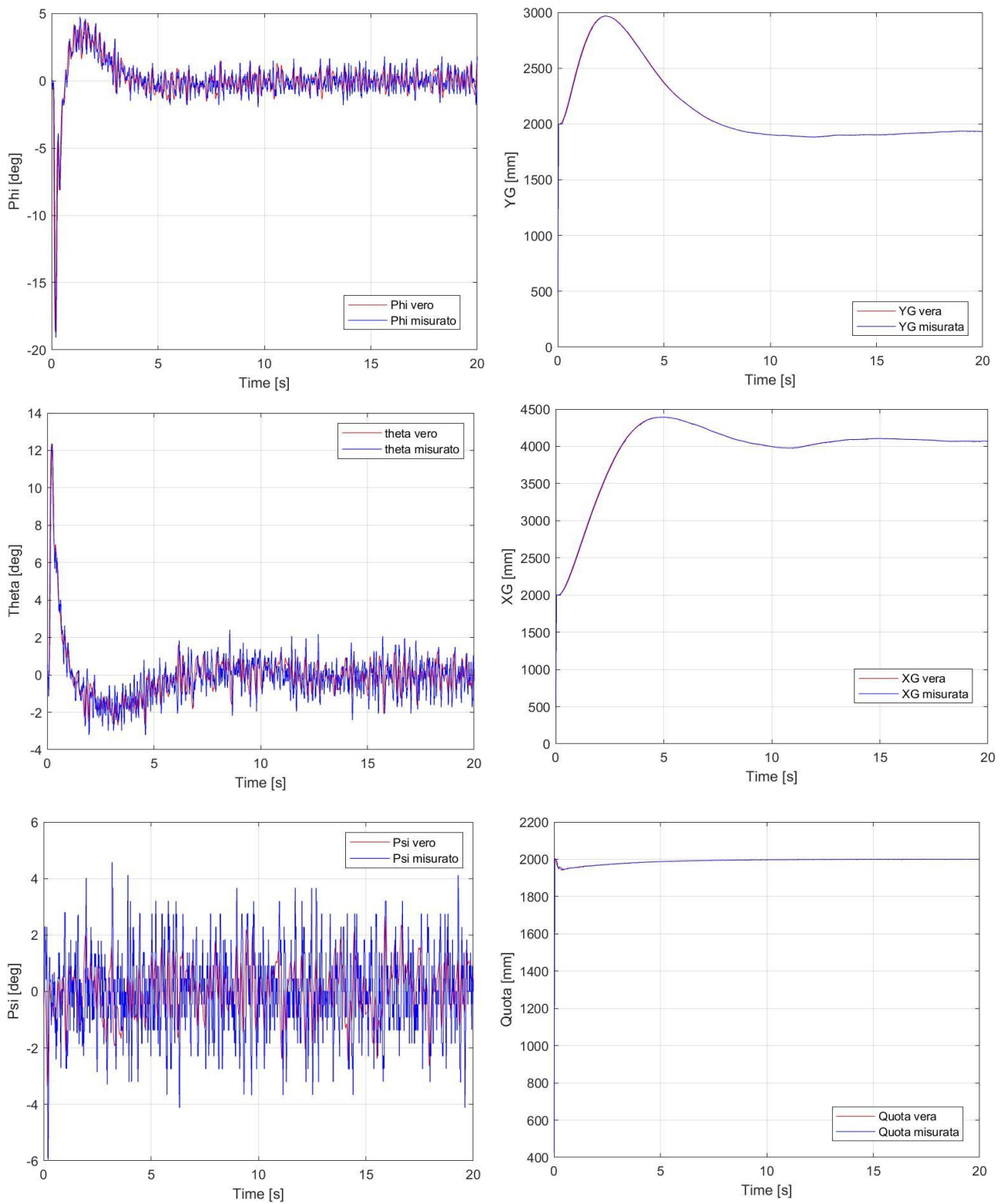


Figura 34: Spostamento della coordinata  $X_G$



## 4.2 Test sperimentali preliminari

Allo scopo di verificare la funzionalità del prototipo, si sono svolti dei test sperimentali. Per prima cosa è stato costruito un ambiente dove svolgere questi test: in particolare è stata realizzata una stanza in legno di dimensioni note e del tutto arbitrarie, dopo di che per fare assumere le diverse posizioni al prototipo, è stato necessario realizzare un supporto.



Figura 35: Stanza costruita per le simulazioni.

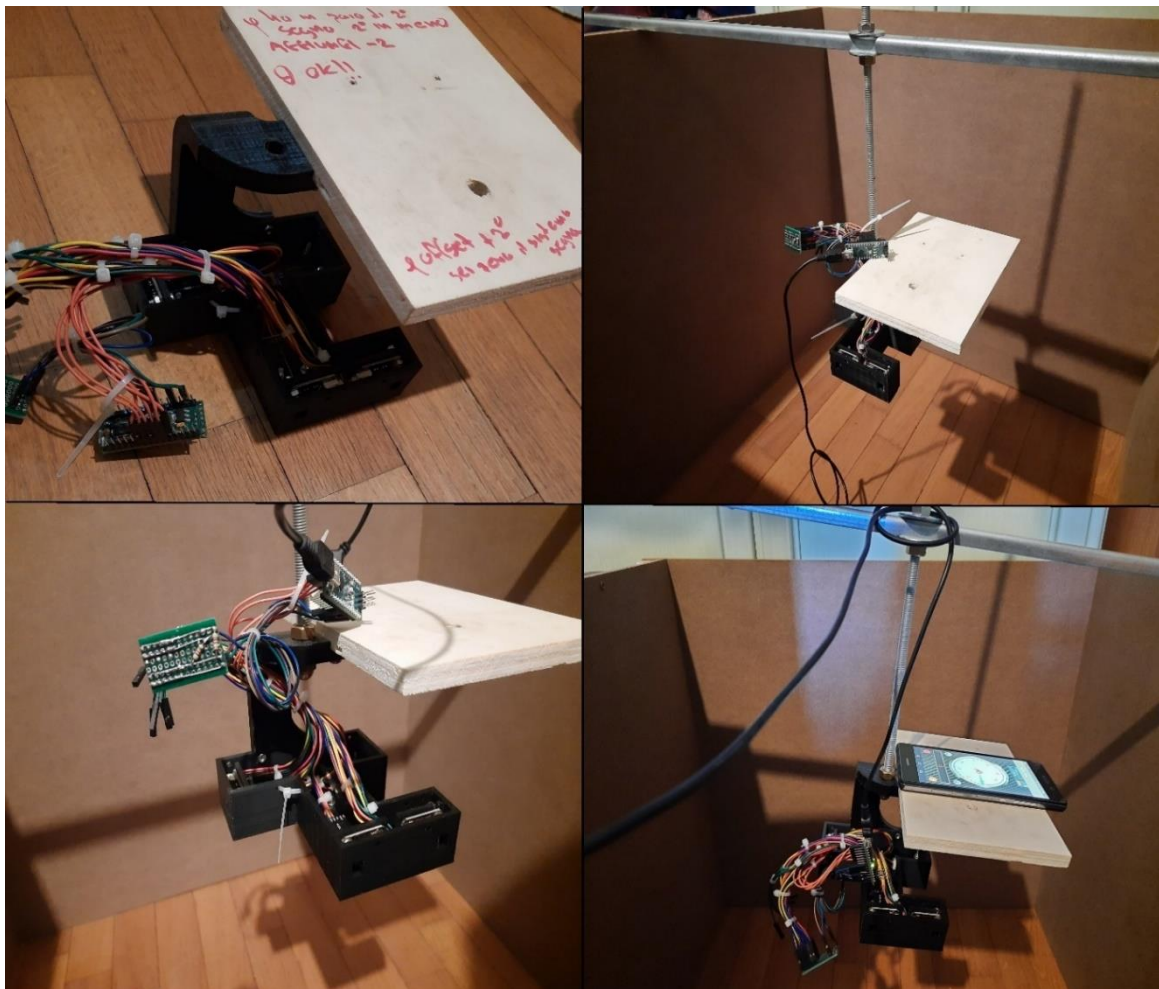


Figura 36: Vista del prototipo durante le misurazioni.

Come mostrato in figura il supporto permette di sostenere il prototipo all'interno della stanza durante le misurazioni.

Per verificare i dati calcolati dal prototipo, essi verranno messi a confronto con le misure fornite da uno smartphone che si muove insieme al sistema.

Qui sotto si presentano varie misurazioni effettuate durante i test.

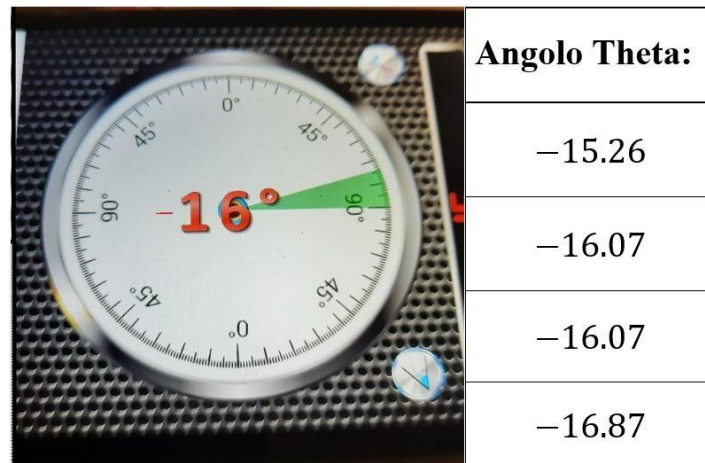


Figura 37: rotazione di un angolo  $\theta$  di  $-16^\circ$ .



Figura 38: Rotazione di un angolo  $\psi$  di  $22^\circ$ . Nella figura a sinistra abbiamo il colore blu che corrisponde all'andamento della Yaw, il verde per il Pitch e infine il giallo per l'angolo di Rollio.



Figura 39: Rotazione di un angolo di  $\phi$  di  $8^\circ$ .

I risultati dei test, oltre a verificare la validità delle misure, permettono anche di individuare eventuali criticità nella progettazione.

Infatti, nei primi test sul campo, si è individuato un errore nel posizionamento dei sensori 1-2 che portava a una forte oscillazione dell'angolo  $\psi$ , come già spiegato nel capitolo 3.

In un primo momento si era pensato a un difetto di uno dei due sensori, ma isolandoli, le distanze calcolate erano corrette.

Per ovviare al problema, dunque, si è deciso di riposizionare il sensore 1, modificando il case come spiegato precedentemente.

Un ulteriore difetto evidenziato con test aggiuntivi è il limite tecnologico dato dai sensori.

Precisamente, come già spiegato nei capitoli precedenti, seppur si è costruita una curva di caratterizzazione del sensore, ognuno di essi necessita di un'un'ulteriore correzione del suo offset di misurazione ogni qual volta si modificano significativamente le condizioni ambientali.

Tale problema può essere risolto semplicemente posizionando il sensore a una distanza nota e impostando come offset per ogni sensore la differenza tra la distanza nota e la distanza misurata dal sensore.

Questa procedura dovrà essere eseguita tutte le volte che il sistema cambierà il luogo di lavoro.

In ultima analisi, un ulteriore limite dei sensori è rappresentato dalla qualità delle misure.

Infatti, le misure fornite dai sensori possiedono un'incertezza di  $\pm 1\text{mm}$ , che porta inevitabilmente a un'incertezza angolare.

La propagazione di questo errore e di conseguenza l'incertezza angolare, dipende dalle varie distanze tra i sensori. Qualora si volesse migliorare questo aspetto, una possibile soluzione potrebbe essere aumentare le dimensioni caratteristiche del progetto (A, B, C, D, E, S, H).

Ovviamente le dimensioni massime che il sistema potrà assumere vengono stabilite in base al velivolo su cui andrà installato. È possibile aumentare le dimensioni del case adeguandole in accordo al tipo di multirottore in possesso, aumentando così la precisione dei dati.



### 4.3 Stima dei costi

N.	Prodotto	Costo unitario	TOT
6	Sensore VL53L1X Time-of-Flight (ToF) <sup>1</sup>	20.99 €	125.94 €
1	Arduino Nano <sup>2</sup>	14.40 €	14.40 €
1	Telaio di supporto	15 €	15€
1	Componentistica varia (bulloni, cavi, basette)	15 €	15€
<b>Totale progetto</b>			<b>170.34 €</b>

Tabella 2: 1 [https://www.amazon.it/CQRobot-Distance-Raspberry-Interface-Detection/dp/B07F3TV3G4/ref=pd\\_sbs\\_147\\_2/259-9143159-0286711?\\_encoding=UTF8&pd\\_rd\\_i=B07F3TV3G4&pd\\_rd\\_r=d2598170-23f1-4b55-915d-3fc1ef94078b&pd\\_rd\\_w=mbZEX&pd\\_rd\\_wg=RbF0o&pf\\_rd\\_p=85d7fa16-9140-48a8-92c7-6a88c9b2bb49&pf\\_rd\\_r=KKZZ2BGFZNWMGHM02WRM&pvc=1&refRID=KKZZ2BGFZNWMGHM02WRM](https://www.amazon.it/CQRobot-Distance-Raspberry-Interface-Detection/dp/B07F3TV3G4/ref=pd_sbs_147_2/259-9143159-0286711?_encoding=UTF8&pd_rd_i=B07F3TV3G4&pd_rd_r=d2598170-23f1-4b55-915d-3fc1ef94078b&pd_rd_w=mbZEX&pd_rd_wg=RbF0o&pf_rd_p=85d7fa16-9140-48a8-92c7-6a88c9b2bb49&pf_rd_r=KKZZ2BGFZNWMGHM02WRM&pvc=1&refRID=KKZZ2BGFZNWMGHM02WRM). 2 [https://www.amazon.it/Arduino-Nano-Every-Single-Board/dp/B07VX7MX27/ref=sr\\_1\\_8?adgrpid=58322028131&dchild=1&gclid=EAAlaQobChMI\\_reHzKii6wIVDbLVCh1DhAo6E AAYASAAEgJS2\\_D\\_BwE&hvadid=255177846693&hvdev=c&hvlocphy=1008745&hvnetw=g&hvqmt=e&hvrnd=1047161729886835624&hvtargid=kwd-296545676784&hydadcr=28874\\_1803253&keywords=arduino+nano&qid=1597670016&sr=8-8&tag=slhyin-21](https://www.amazon.it/Arduino-Nano-Every-Single-Board/dp/B07VX7MX27/ref=sr_1_8?adgrpid=58322028131&dchild=1&gclid=EAAlaQobChMI_reHzKii6wIVDbLVCh1DhAo6E AAYASAAEgJS2_D_BwE&hvadid=255177846693&hvdev=c&hvlocphy=1008745&hvnetw=g&hvqmt=e&hvrnd=1047161729886835624&hvtargid=kwd-296545676784&hydadcr=28874_1803253&keywords=arduino+nano&qid=1597670016&sr=8-8&>tag=slhyin-21)

All'interno della tabella è riportato il costo complessivo del prototipo, senza considerare il costo di realizzazione della stanza e del supporto.

Osservando il totale, ci si rende conto che la spesa è relativamente contenuta rispetto ai sistemi citati nell'introduzione: ad esempio, il costo di un sistema di tipo Lidar può arrivare anche a qualche migliaio di euro.

Per quanto riguarda la tecnologia Motion Capture, il suo costo è difficilmente stimabile a causa della quantità e dal tipo di telecamere utilizzate, inoltre bisogna tenere conto anche dalla potenza di calcolo del Pc. Tuttavia, si può affermare che il suo costo in termini economici è superiore, senza considerare che risulta essere un sistema laborioso e troppo complicato per i possibili scopi di un multirobot indoor.

## 5. Conclusioni

Le ultime considerazioni affrontano un breve confronto tra le varie tecnologie e il prototipo realizzato fornendo anche qualche spunto per il suo miglioramento.

Un punto di forza di questo sistema, in confronto alla tecnologia Lidar/VPS, è la compattezza del sistema e il consumo energetico: risparmiando carica si può risparmiare sulle dimensioni della batteria e dunque diminuire il peso.

Un ulteriore vantaggio è la poca potenza di calcolo necessaria rispetto, ad esempio, alla tecnologia Motion Capture, la quale ha bisogno di un elevato costo computazionale per ricavare la posizione, unito a un hardware costoso, quali telecamere e Pc per la gestione delle immagini.

Ovviamente le tecnologie citate sopra offrono vantaggi rispetto al nostro sistema, come ad esempio la possibilità di scannerizzare l'ambiente circostante in modo da creare un ambiente virtuale. Di fatto il funzionamento del sistema proposto si limita ad ambienti con forme semplici, come parallelepipedi di dimensioni note.

Un'altra differenza risulta essere la portata dei sensori: la tecnologia Lidar viene impiegata anche per scansioni di paesaggi, o ambienti urbani, tuttavia la possibilità di svolgere scansioni su larga scala, comporta un costo economico complessivo maggiore.

Nonostante tutto, sono presenti ancora aspetti da migliorare: uno su tutti, la realizzazione di una sequenza automatica di calibrazione da effettuare all'inizio di ogni utilizzo. Questa sequenza dovrà essere affidabile e ripetibile, grazie anche all'utilizzo di supporti che si potranno rimuovere una volta terminato il programma di calibrazione.

Un ulteriore miglioramento potrebbe consistere nella caratterizzazione di ogni sensore. In questo modo ognuno di essi avrebbe la propria curva sperimentale, che terrebbe conto dei singoli errori di non linearità.

Inoltre, sono possibili nuovi sviluppi di questo progetto: uno fra questi è connettere il sistema al computer di bordo di un velivolo, o addirittura gestire i sensori direttamente con esso eliminando la presenza di Arduino Nano.

Infine, un secondo potenziamento potrebbe prevedere la stima di ulteriori misure, come le velocità angolari e lineari, utilizzando un filtro osservatore dello stato.

# Bibliografia

- [1] Carlo Casarosa – Meccanica del volo – Pisa university press – Pisa – 2013
  
- [2] P. Biscari T. Ruggeri G. Saccomandi M. Vianello – Meccanica razionale per l'ingegneria – Monduzzi Editore – Bologna – 2007
  
- [3] Daniele Catellani – Una rassegna di tecnologie per posizionamento indoor – Scuola di scienze, laurea specialistica in informatica – Università di Bologna – 2016/2017 – Relatore: Prof. Luciano Bonomi
  
- [4] Gianluca Rossetti - Development of a flight control architecture for rotary-wing UAVs with model-based design approach – Dottorato di ricerca in Meccanica e Scienze Avanzate dell'Ingegneria (Ciclo 30), Settore Concorsuale: 09/A1 Ingegneria aeronautica, aerospaziale e navale, Settore Scientifico Disciplinare: ING-IND/03 Meccanica del volo – Università di Bologna – 2018 – Coordinatore Dottorato: Prof. Marco Carricato, supervisore: Prof. Fabrizio Giulietti
  
- [5] Simone Majocchi – Arduino Uno, Programmazione avanzata e Librerie di sistema – Vispa Edizioni – Gallarate (VA) – 2012
  
- [6] Michael Margolis – Arduino progetti e soluzioni 2<sup>a</sup> edizione – Tecniche Nuove – Milano – 2013
  
- [7] R. Bevilacqua, D. Bini, M. Capovani, O. Menchi - Metodi Numerici – Zanichelli – Bologna – 1992
  
- [8] Mahony, R., Kumar, V., and Corke, P. – "Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor" – IEEE Robotics & Automation Magazine – Vol. 19 No. 3 – 2012
  
- [9] Kidus, Guye – Development of an Indoor Multirotor Testbed for Experimentation on Autonomous Guidance Strategies – Electronic Theses and Dissertations - 2972 – <https://openprairie.sdstate.edu/etd/2972> – 2018

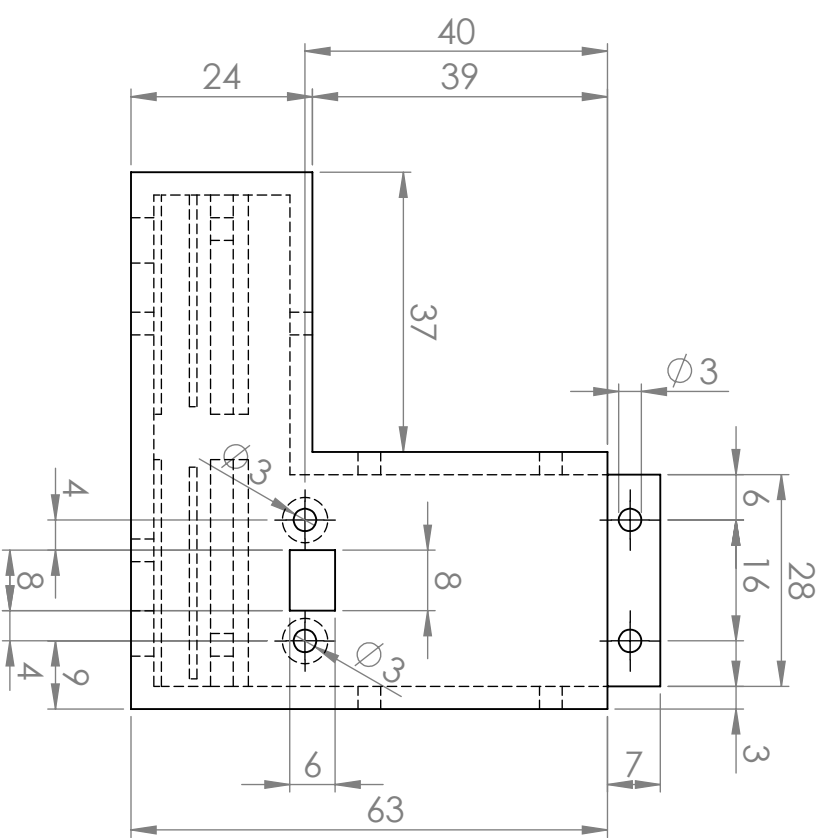
- [10] Yuntian Lia, Matteo Scanavino, Elisa Capello, Fabrizio Dabbene, Giorgio Guglieri, Andrea Vilardi – A novel distributed architecture for UAV indoor navigation – Published by Elsevier B.V. – International Conference on Air Transport – INAIR 2018
- [11] Moshe Ben Moshe, Shvalb Nir, Baadani Jonathan, Levy Harel, Nagar Itay – Indoor Positioning and Navigation for micro UAV – Work in progress – Convention of Electrical and Electronics Engineers in Israel – 2012



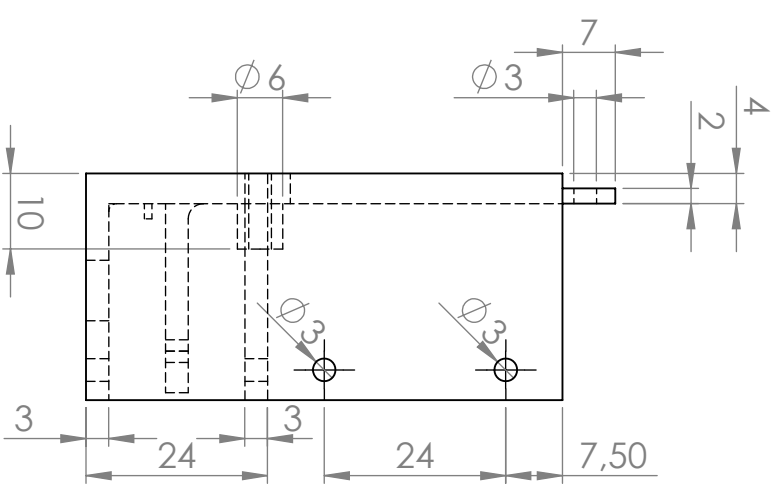




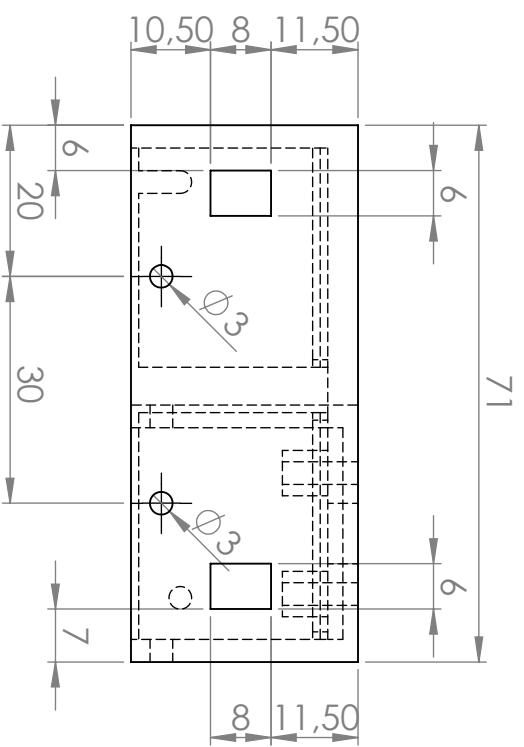
**Appendice 2: Parte anteriore del Case**



Vista Inferiore



Vista laterale



Vista frontale

Alma Mater Studiorum - Università di Bologna, Dipartimento di  
Ingegneria industriale, Meccanica del Volo

Appendice elaborato finale di Tesi

Titolo elaborato: Stimola della posizione e assetto di un multirattore indoor tramite l'utilizzo di sensori laser.

Candidato: Baldacci Edoardo

Relatore: Emanuele Luigi de Angelis

Matricola: 0000839212

Anno accademico: 2019/2020

Scala: 1:1

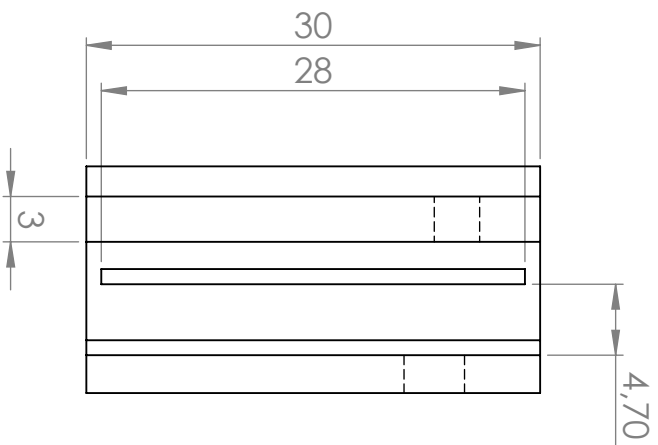
Unita di misura: millimetri

Numero foglio: 1/3

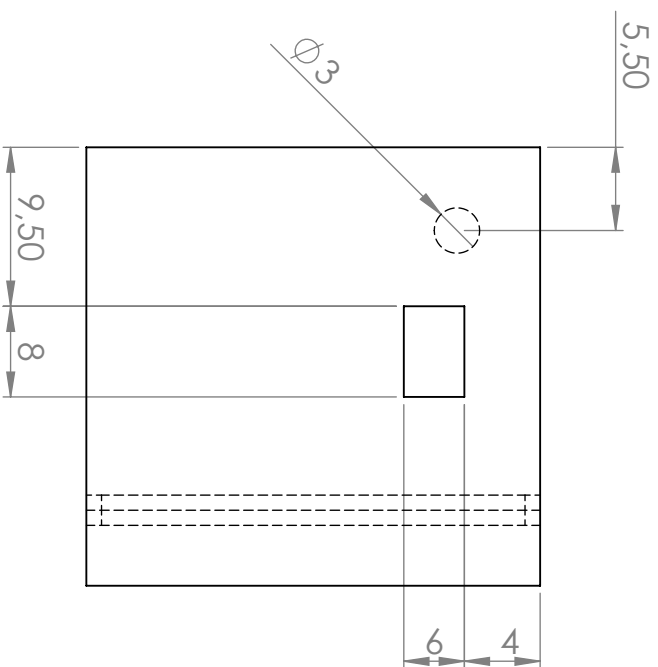




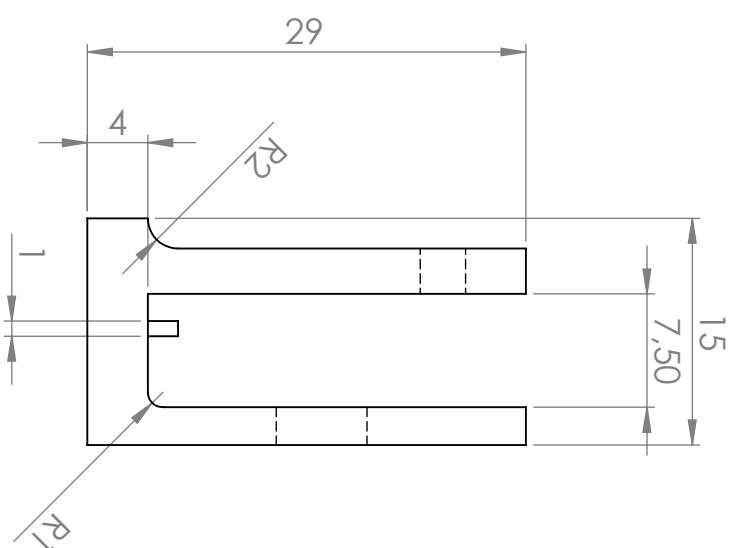
# Allegato 2: Supporto sensore



Vista superiore



Vista laterale



Vista frontale

R1 = 3  
R2 = 5

Alma Mater Studiorum - Università di Bologna, Dipartimento di  
Ingegneria industriale, Meccanica del Volo

Appendice elaborato finale di Tesi

Titolo elaborato: Stimazione della posizione e assetto di un multistrutture indoor tramite l'utilizzo di sensori laser.

Candidato: Baldacci Edoardo

Relatore: Emanuele Luigi de Angelis

Matricola: 0000839212

Anno accademico: 2019/2020

Scala: 1:1

Unità di misura: millimetri

Numero foglio: 3/3