

# **Progettazione e realizzazione di un pilota automatico per aeromodelli basato su architettura Arduino**

*Relatrice:*

Prof.ssa Reggiani Monica

*Laureando:*

Tassinari Paolo

## **SOMMARIO**

Dal 2010, in tutto il mondo è cresciuto moltissimo l'interesse, sia di singoli cittadini che di aziende, verso i droni, specialmente verso piccoli velivoli; il mercato offre una vasta scelta in questo senso, soprattutto per quanto riguarda mezzi radiocomandati, mentre è molto carente nell'ambito dei piloti automatici.

In questa tesi viene sviluppato un software di questo tipo per aeromodelli ad ala fissa, prestando particolare attenzione alle possibilità di personalizzazione, così che possa essere utilizzato con successo a bordo del maggior numero di velivoli possibile.

# INDICE

<b>1</b>	<b>Definizione del problema.....</b>	<b>3</b>
1.1	Rotta	4
1.2	Posizione	6
1.3	Orientamento	6
<b>2</b>	<b>Il piano di volo.....</b>	<b>7</b>
2.1	Waypoints	7
2.1	ModeType	7
2.1	Esempio di piano di volo	8
<b>3</b>	<b>Software.....</b>	<b>9</b>
3.1	Torre di controllo	9
3.2	Relay	12
3.3	Autopilota	13
3.3.1	Setup	14
3.3.2	Loop	15
<b>4</b>	<b>Hardware.....</b>	<b>19</b>
4.1	Specifiche del velivolo	19
4.2	Controllori	22
4.3	Sensori	22
4.4	Altri componenti	23
<b>5</b>	<b>Sviluppo dell'aereo.....</b>	<b>19</b>
5.1	Layout	23
5.2	Costruzione	23
5.3	Esecuzione di un volo	24
<b>6</b>	<b>Risultati sperimentali.....</b>	<b>24</b>
<b>7</b>	<b>Conclusioni.....</b>	<b>27</b>

# Capitolo 1

## Definizione del problema

L'obiettivo è realizzare un pilota automatico in grado di guidare l'aereo su cui è installato lungo una rotta prestabilita e di gestirne decollo e atterraggio, il tutto in completa autonomia. Inoltre, i dati rilevanti del volo devono poter essere trasmessi in tempo reale ad una stazione a terra, che li renda disponibili all'utilizzatore sotto forma di tabelle, grafici o indicatori.

Visto che deve poter essere utilizzato da una vasta gamma di modelli, ognuno con esigenze anche molto diverse (es.: un aliante e un aereo acrobatico), il software sarà open-source e le parti rilevanti saranno messe in evidenza e scritte il più chiaramente possibile, in modo che possano essere modificate con semplicità e senza rischiare di commettere errori gravi.

In questo capitolo viene introdotta la terminologia utilizzata, in particolare in riferimento al movimento dell'aereo.

## 1.1 Rotta

La rotta (fig. 1) è formata da due componenti: rotta perpendicolare e rotta parallela, in riferimento al campo gravitazionale terrestre nel punto.

- *Asse longitudinale corrente*: è il versore che indica la direzione che l'aereo sta avendo;
- *Asse longitudinale richiesto*: è il versore che indica la direzione che l'aereo dovrebbe avere;
- *Rotta perpendicolare*: è l'angolo tra le proiezioni su un piano perpendicolare al campo gravitazionale terrestre dell'asse magnetico terrestre e di quello longitudinale (corrente o richiesto) dell'aereo; anche chiamata direzione;
- *Rotta parallela*: è l'angolo tra l'asse longitudinale (corrente o richiesto)(1) dell'aereo e un versore (2) perpendicolare al vettore campo gravitazionale terrestre (3) calcolato sulla posizione dell'aereo, tali che (1), (2) e (3) giacciono sullo stesso piano; anche chiamata inclinazione.

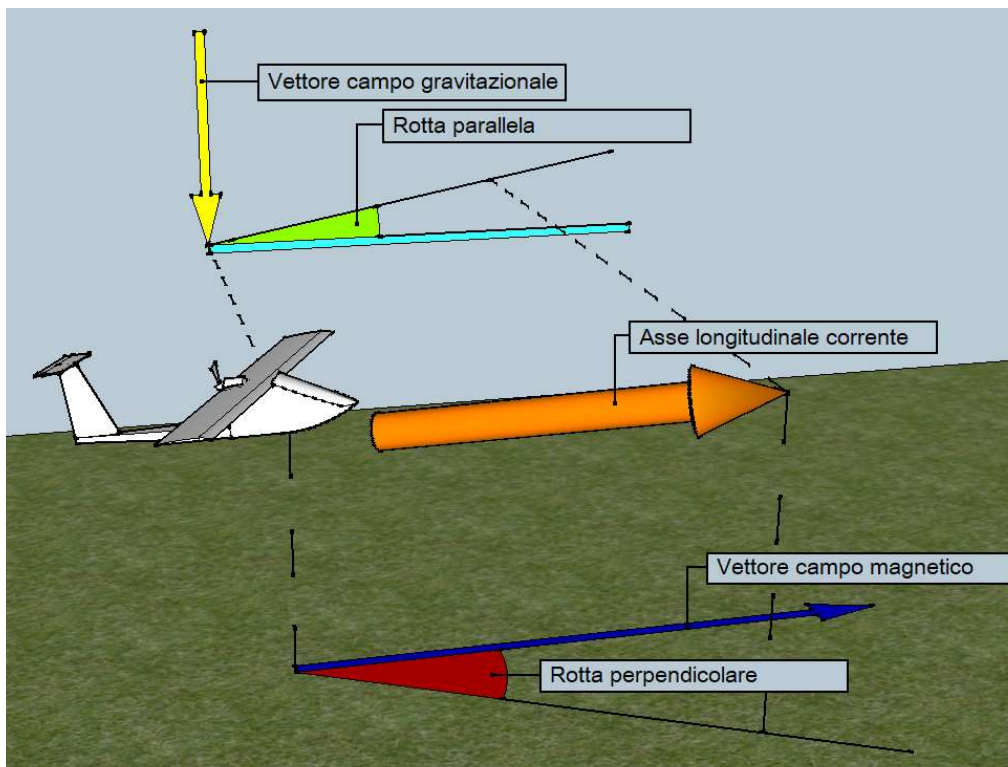


Figura 1: Rappresentazione schematica di rotta perpendicolare (o direzione) e rotta parallela o (inclinazione).

In questo documento, si fa riferimento a più tipi di rotta:

- *Rotta*: se non è meglio specificato, è la rotta che l'aereo cerca di seguire; direzione e inclinazione sono calcolati partendo dall'asse longitudinale richiesto;
- *Rotta corrente*: è la rotta che l'aereo sta seguendo nell'istante corrente; direzione e inclinazione sono calcolati partendo dall'asse longitudinale corrente;

A sua volta, la rotta corrente è detta *rotta magnetica* se calcolata utilizzando un sensore di campo magnetico e un accelerometro, o *rotta GPS* se ricavata a partire da due rilevamenti GPS effettuati in due istanti differenti.

Questi due metodi di calcolo presentano alcuni vantaggi e svantaggi che li rendono più o meno preferibili in base alle esigenze dell'aereo in quel momento:

Rotta magnetica: è molto imprecisa [1], principalmente per tre ragioni:

- il nord magnetico non coincide con il nord geografico, ed è in costante movimento;
- il campo magnetico locale differisce da quello di riferimento a causa di elementi come masse ferrose, cavi elettrici, centrali elettriche, ecc.;
- il campo magnetico calcolato dal sensore differisce da quello locale a causa della presenza nell'aereo di elementi metallici strutturali, batterie, motori, servomotori e altri moduli (sensori, ricetrasmittente, ecc.).

Il vantaggio risiede nell'istantaneità della misurazione, per cui viene utilizzata durante le manovre per stimarne la durata (v.di "Esecuzione delle manovre").

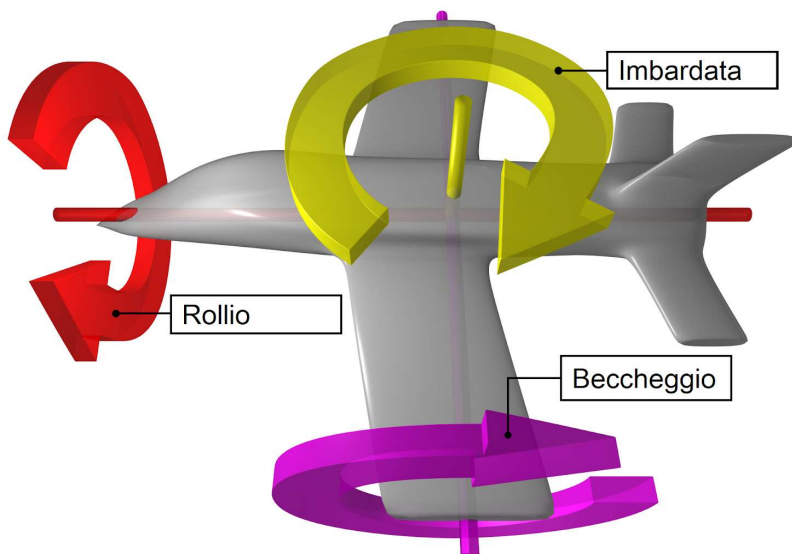
Rotta GPS: è tanto più attendibile quanto più tempo trascorre tra le misurazioni: la precisione attesa dei rilevamenti GPS non è inferiore a 2 m, per cui, supponendo una velocità orizzontale minima di 36 km/h (10 m/s), è necessario che trascorrono almeno 0,2 s tra le due misurazioni. Di conseguenza, la rotta GPS viene calcolata soltanto al termine di una manovra, così da verificare l'orientamento dell'aereo e, se necessario, impostare una nuova rotta da seguire.

## 1.2 Posizione

Per determinare la posizione di un punto nello spazio, sono necessarie tre coordinate: nel caso di un aereo, quelle più facilmente ottenibili sono espresse come latitudine, longitudine, altitudine; queste vengono ricavate tramite GPS e barometro (purtroppo il ricevitore GPS non si è rivelato affidabile nel calcolo dell'altitudine, variandola di diverse decine di metri ad aereo fermo e condizioni meteo favorevoli). Da notare che, dato che la pressione atmosferica in un dato luogo può variare considerevolmente anche nel giro di poche ore, l'altitudine viene misurata dal punto di partenza e la pressione in quel punto è presa come riferimento, anziché quella al livello del mare; in questo modo si evitano errori nell'estimo dell'altitudine da raggiungere, essenziale in fasi delicate come decollo e atterraggio.

## 1.3 Orientamento

Per stabilirne l'orientamento, è utile fissare un sistema di riferimento al velivolo stesso (fig. 2): siano l'asse x quello che va dall'estremità dell'ala sinistra verso quella dell'ala destra, l'asse y dalla poppa verso la prua, l'asse z dal basso verso l'alto (presumendo l'aereo a terra). Allora, le coordinate che esprimono la rotazione intorno a questi assi sono rispettivamente beccheggio, rollio e imbardata [1]. Ma per la maggior parte delle applicazioni che un aereo può avere, conviene utilizzare la direzione al posto dell'imbardata: in questo modo stabilire l'orientamento dell'aereo è più intuitivo, e lo sono anche i calcoli che il controllore deve effettuare.



*Figura 2: rotazioni utilizzate per identificare l'orientamento dell'aereo. La direzione delle frecce corrisponde ad un angolo positivo.*

## Capitolo 2

### Il piano di volo

Per ottenere un programma in grado di pilotare aerei con caratteristiche e impieghi diversi, si è deciso di utilizzare per la definizione del piano di volo un sistema basato su waypoints da raggiungere in linea retta dopo aver eseguito una serie di istruzioni (che possono rappresentare acrobazie, manovre di saluto, cambi di velocità del motore, ecc...).

Questa struttura permette alla fine di ogni waypoint di ricalcolare la rotta GPS, limitando gli errori derivanti dall'uso prolungato della rotta magnetica, ma anche di eseguire manovre complesse come decollo e atterraggio in autonomia e sicurezza (infatti le istruzioni permettono di studiare percorsi molto dettagliati, impossibili da ottenere con i soli waypoints).

Questa serie di comandi è codificata in un file (detto appunto *piano di volo*) (fig. 3), assieme ad alcune istruzioni per i controllori che gestiscono le superfici di controllo.

Il piano di volo è diviso in 3 sezioni: Waypoints, modeType e pid:

- Waypoints elenca in ordine i waypoints;
- modeType elenca tutte le istruzioni necessarie al volo.
- pid contiene le tre costanti (proporzionale, integrale e derivata) necessarie ai 3 controllori, uno per il beccheggio, uno per il rollio e l'ultimo per l'imbardata (quindi 9 valori in tutto) e la frequenza di campionamento per l'integrazione;

```
Waypoints
04571534100 01219753000 0015 01
04571550600 01219735200 0015 02
04571534100 01219753000 0015 01
modeType
01 00 0000 0000 0000 07 0080
01 01 0000 0045 0000 04 0045
01 02 0000 -045 0000 04 -045
01 03 0000 0000 0000 02 0002
02 00 0000 0000 0000 07 0050
02 01 0000 0000 0000 03 0010
pid
0150 1000 0000 0150 1000 0000 0200 0050 0000 0100
end
```

*Figura 3: Esempio del file contenente il piano di volo*

#### 2.1 Waypoints

`04571534100 01219753000 0015 01` *Figura 4: Primo waypoint del file preso in esempio*

Le 4 diverse stringhe numeriche che si notano (fig. 4) indicano, da sinistra a destra: latitudine, longitudine, altitudine (rispetto al punto di partenza) e modo del waypoint.

L'obiettivo dell'aereo è raggiungere (entro un margine di errore di alcuni metri) le coordinate indicate seguendo le direttive imposte dai *modi* (in questo caso, modo numero 1).



## 2.2 ModeType

```
01 00 0000 0000 0000 07 0080
01 01 0000 0045 0000 04 0045
01 02 0000 -045 0000 04 -045
01 03 0000 0000 0000 02 0002
```

Figura 5: Modo numero 1 del file preso in esempio  
(v.di fig. 3)

I modi (fig. 5) sono costituiti da un elenco contenente una serie di *istruzioni*, e possono essere modificati prima del decollo dall'utilizzatore. Ad ogni modo è associato un numero variabile di istruzioni, ognuna formata da 7 valori (da sinistra a destra):

- 1) Modo a cui l'istruzione appartiene;
- 2) Numero dell'istruzione: in un modo formato da più istruzioni, indica in che ordine vanno eseguite;
- 3) Inclinazione richiesta;
- 4) Rollio richiesto;
- 5) Deviazione richiesta (rispetto alla rotta corrente); detta anche  $\Delta$  direzione;
- 6) Tipo di condizione da soddisfare per passare all'istruzione successiva;
- 7) Valore associato a (6).

Se l'istruzione indica di raggiungere direttamente il waypoint, i valori (3) e (5) non vengono presi in considerazione.

Le condizioni per ora esistenti sono:

- 01) Raggiungere un'altezza con l'angolo di beccheggio indicato nell'istruzione;
- 02) Raggiungere il waypoint entro la distanza in metri indicata nell'istruzione;
- 03) Come la 02, ma senza tenere conto della differenza di altezza. Quindi l'angolo di beccheggio è quello indicato nell'istruzione;
- 04) Raggiungere l'angolo di rollio indicato nell'istruzione;
- 05) Raggiungere l'angolo di beccheggio indicato nell'istruzione;
- 06) Variare la direzione di un angolo pari a quello indicato nell'istruzione;
- 07) Regolare la potenza del motore (il valore è in %);
- 08) Mantenere gli angoli indicati per un certo tempo;
- 09) Mantenere le velocità angolari indicate per un certo tempo. Nel caso di valori troppo alti, l'aereo raggiungerà velocità angolari inferiori, ma per lo stesso tempo indicato.

Una volta completate le istruzioni del modo indicato dal waypoint, si passa a quello successivo, per cui in teoria è possibile compiere un volo soltanto con istruzioni che non tengano conto della posizione del waypoint (ovvero senza le condizioni 02 e 03), ma è sconsigliabile, perché in questo modo è molto difficile calcolare la rotta GPS e il pilota automatico si affiderà a quella magnetica, la cui imprecisione rischia di essere eccessiva se accumulata da manovre precedenti.

## 2.3 Esempio di piano di volo

```
Waypoints
04571534100 01219753000 0015 01
04571550600 01219735200 0015 02
04571534100 01219753000 0015 01
modeType
01 00 0000 0000 0000 07 0080
01 01 0000 0045 0000 04 0045
01 02 0000 -045 0000 04 -045
01 03 0000 0000 0000 02 0002
02 00 0000 0000 0000 07 0050
02 01 0000 0000 0000 03 0010
pid
0150 1000 0000 0150 1000 0000 0200 0050 0000 0100
end
```

*Figura 6: Semplice piano di volo, illustrato anche in figura 2*

Il modo 01 impone lo svolgimento di 4 istruzioni: nell'ordine, portare la potenza del motore all'80%, rollare fino ad un angolo di rollio di 45° verso destra, rollare fino ad un angolo di 45° verso sinistra, raggiungere una distanza di 2 metri dal waypoint.

Il modo 02 invece è composto di sole 2 istruzioni: portare la potenza del motore al 50 % e raggiungere una distanza di 10 m dal waypoint, ignorandone l'altezza e con un angolo di beccheggio pari a 0.

Il primo ed il terzo waypoint hanno modo 1, mentre il secondo ha modo 2, quindi il volo si svolgerà pressappoco così:

- potenza 80%;
- rollio a 45°;
- rollio a -45°;
- virata verso il primo waypoint; raggiungimento del waypoint;
- potenza 50%;
- virata verso il secondo waypoint; raggiungimento del waypoint;
- potenza 80%;
- rollio a 45°;
- rollio a -45°;
- virata verso il terzo waypoint; raggiungimento del waypoint;
- potenza 0% (avviene automaticamente alla fine del volo).

## Capitolo 3

### Software

Come spiegato nel capitolo 1, oltre ad un sistema per controllare l'aereo sarà presente anche un'interfaccia utente installata su un computer a terra, e per mettere in comunicazione i due programmi sarà necessario un sistema di trasmissione dati. Quindi servono 3 programmi separati (fig. 7): *Torre di controllo*, installato sul computer a terra, *Relay*, installato su un Arduino Nano collegata al PC via USB e dotata di ricetrasmittitore, e *Autopilota*, installato a bordo dell'aereo su un Arduino MEGA anch'esso dotato di ricetrasmittitore.

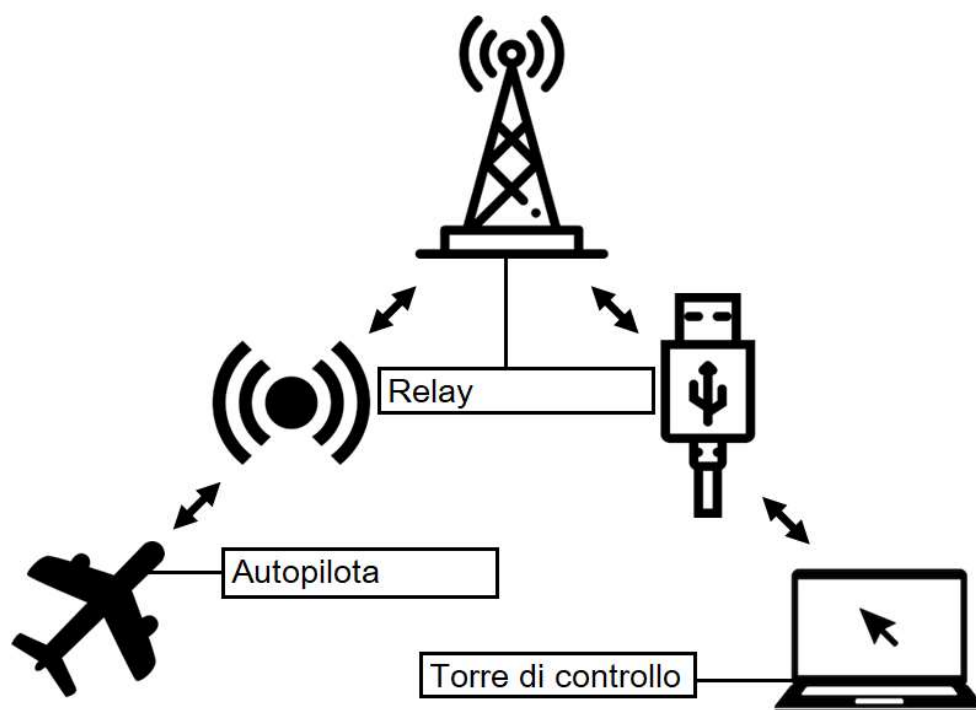


Figura 7: comunicazione tra i tre programmi.

Inoltre, essendo open-source, il progetto è disponibile su GitHub<sup>1</sup>, completo di librerie, programmi, esempi di piani di volo e manuale.

---

<sup>1</sup> <https://github.com/Paolotassi/Planes-/blob/master/README.md>

### 3.1 Torre di controllo

È una macchina a stati che permette all'utente di seguire passo passo la procedura di avviamento dell'aereo, in modo da consentire l'individuazione di eventuali errori, e di seguire e analizzare il volo attraverso i dati ricevuti e opportunamente illustrati.

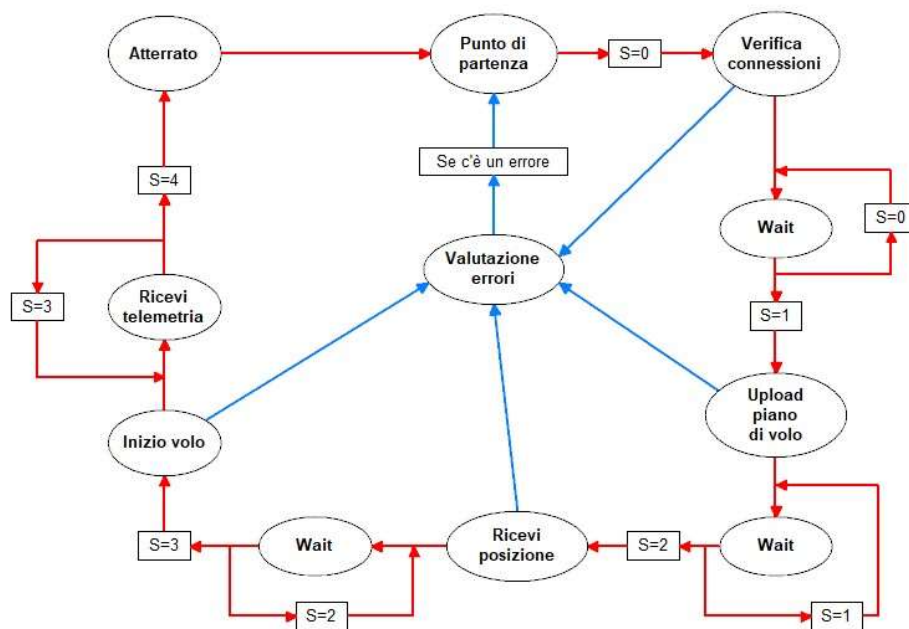


Figura 8: Schema del funzionamento di Torre di controllo. S indica lo stato.

Gli stati di Torre di controllo sono incrementati utilizzando gli interruttori della sezione *Checklist* del primo pannello (fig. 9), mentre i segnali di errore sono segnalati nella sezione *Fatal error description*. Questi possono essere generati in due modi: dal confronto tra i messaggi ricevuti da Autopilota e quelli attesi, oppure dall'utente se la Checklist non viene seguita correttamente. In entrambe i casi, la procedura deve essere eseguita nuovamente. In figura 8 si nota che 5 blocchi non inviano segnali di errore: i 3 *Wait* sono semplicemente dei cicli in cui si attende il comando di procedere, *Atterrato* indica la conclusione del volo e quindi non avrebbe senso segnalare errori, mentre *Ricevi telemetria* è il ciclo in cui resta Torre di controllo durante tutto il volo, ed è la funzione che riceve e rappresenta la telemetria. Quindi è fondamentale che non sia necessario un reset in questa fase, per cui gli errori che si possono riscontrare qui (cioè stringhe di dati corrotte o incomplete, o periodi inattesi di silenzio radio) risultano semplicemente in dei "vuoti" nei grafici (fig. 11).

Per la sua realizzazione è stato scelto LabView, un ambiente di sviluppo particolarmente indicato per la realizzazione di macchine a stati. Inoltre, consente di creare e modificare facilmente interfacce uomo-macchina, ed in particolare ha una scelta molto ampia di grafici e indicatori.

La schermata è divisa in 3 pannelli: il primo (fig. 9) contiene tutti i pulsanti, le spie e gli indicatori necessari perché le procedure possano essere eseguite, ed eventuali errori diagnosticati; il secondo (fig. 10) è formato da una tabella ed alcuni display che rappresentano i dati di volo in cifre; il terzo (fig. 11) è formato da grafici ed indicatori che rappresentano i dati di volo in maniera più intuitiva e di immediata comprensione.

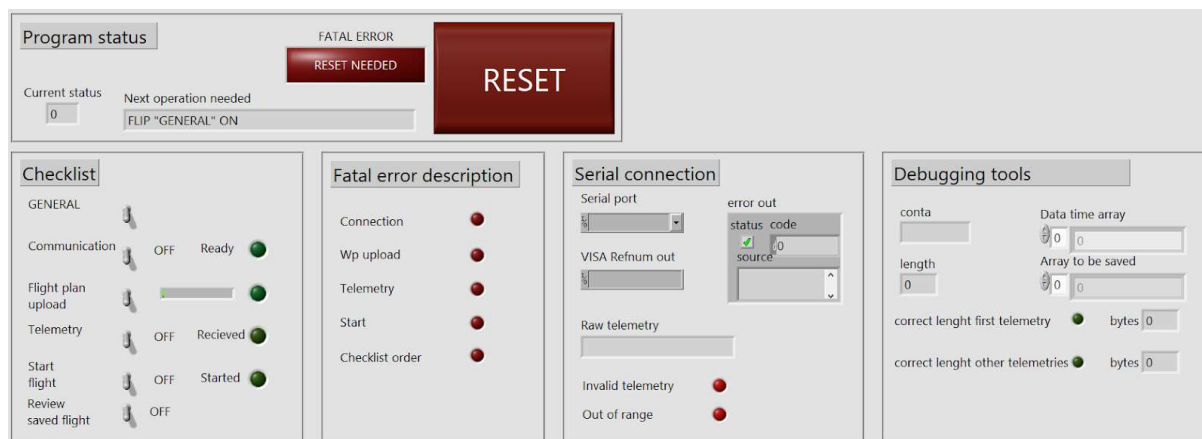


Figura 9: Pannello per controlli pre-volo, invio piano di volo e ricerca errori: Program status e checklist permettono l'interazione con l'aereo e con i files necessari; Serial connection consente la verifica dei dati ricevuti, e indica eventuali errori nelle stringhe ricevute; Fatal error description contiene diverse spie, che si accendono in base al tipo di errori riscontrati nel setup; infine, debugging tools permette di controllare lo svolgimento di alcuni processi critici (utile più che altro in fase di test di eventuali modifiche al programma).

Number	Time (s)	Waypoint	Latitude (deg)	Longitude (deg)	Altitude (m)	Engine power (%)	Speed (%)	Pitch (deg)	Elevator posizion (deg)	Roll (deg)	Aileron position (deg)
12	4127	0	-1	2	0	0	0	2	-1	42	-170
13	4129	0	-1	2	0	0	0	0	0	-22	-88
14	4130	0	-1	2	0	0	0	0	0	-31	-32
15	4131	0	-1	2	0	0	0	0	0	-33	-24
16	4133	0	-1	2	0	0	0	0	0	-29	-34
17	4134	0	-1	2	0	0	0	0	-1	0	-92
18	4136	0	-1	2	0	0	0	0	-1	-11	-92
19	4137	0	-1	2	0	0	0	0	0	-28	-38
20	4139	0	-1	2	0	0	0	0	0	-31	-30
21	4140	0	-1	2	0	0	0	0	0	-32	-30
22	4142	0	-1	2	0	0	0	0	0	-36	-24
23	4143	0	-1	2	0	0	0	0	0	-40	-16

Starting LAT	Starting LNG	Saved Data points	Saved LAT	Saved Ailerons	Saved Stabilizer	Wp number
-1	2	29	0	0	0	0
Current LAT	Current LNG	Discarded Data points	Saved LNG	Saved Roll	Saved Pitch	Flight record array
-1	2	3	0	-3	0	0

Figura 10: Pannello contenente i dati in forma scritta: anche se non strettamente necessario per seguire il volo, soprattutto durante i test si è rivelato uno strumento prezioso per la comprensione dei problemi riscontrati.

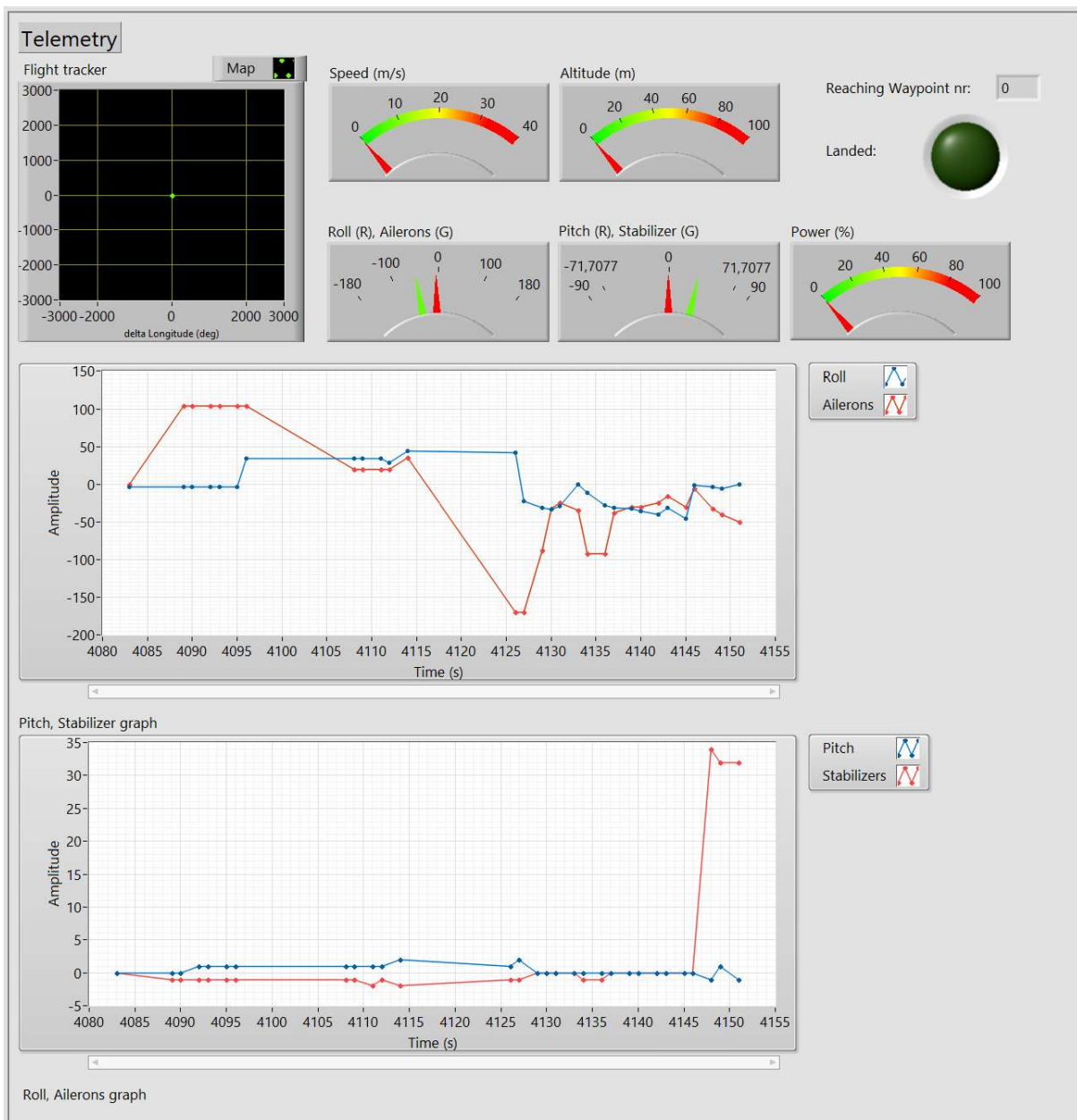


Figura 11: Pannello per la visualizzazione della telemetria: gli indicatori e la mappa (Flight tracker) permettono di seguire il volo in tempo reale, mentre i grafici consentono una comprensione accurata delle azioni dei controllori nel tempo; soprattutto, la sovrapposizione dei grafici di rollio e beccheggio e delle azioni dei rispettivi controllori rendono molto intuitivo il confronto tra i due. In due punti (attorno a  $t = 4100$  s e  $t = 4120$  s) si notano dei “vuoti”, probabilmente dovuti a stringhe di telemetria corrotte.

Dato che la connessione con l'aereo può essere disturbata o interrotta per varie cause, da interferenze sulla stessa frequenza a distanze troppo elevate, l'autopilota è progettato per poter volare in totale autonomia; comunque, per evitare distorsioni nei dati salvati, il computer controlla 3 elementi delle stringhe ricevute dall'aereo: lunghezza, carattere iniziale, carattere finale. Se anche solo uno di questi non corrisponde con quelli attesi, il computer indica un errore non fatale (quindi non interrompe le sue operazioni), e scarta la stringa ricevuta. Per segnalarlo all'utilizzatore, viene incrementato il valore *Discarded Data points* (fig. 10) (nei dati dell'esempio, si noti che sono stati scartati 3 punti); viceversa, per ogni stringa di dati corretta viene incrementato *Saved Data points*. Ogni eventuale interruzione viene indicata nei grafici tramite spazi vuoti, mentre gli indicatori continuano a visualizzare gli ultimi valori ricevuti fino alla successiva ricezione corretta.

Comunque durante il volo i dati possono anche essere salvati in una microSD: così anche in caso di interruzione delle trasmissioni sarà possibile rivedere i dati completi (ovviamente non in tempo reale, ma in questo caso non è poi così importante, dato che in ogni caso l'utilizzatore non ha controllo sull'aereo mentre è in volo).

## 3.2 Relay

È un programma molto semplice, il cui funzionamento si può dividere in due parti:

- Prima del volo, invia ad Autopilota le comunicazioni di Torre di controllo, poi attende una risposta che viene inoltrata a quest'ultimo;
- Durante il volo, riceve in continuazione dati da Autopilota, e li trasmette a Torre di controllo soltanto quando questo ne fa richiesta.

Questa divisione permette di ottimizzare le risorse a disposizione di Autopilota: infatti, nella prima fase ha tutto il tempo necessario per elaborare le informazioni ricevute e la risposta adeguata, mentre la seconda è strutturata in modo da consentirgli l'invio dei dati in ogni momento, così da non dover interrompere operazioni più importanti per il volo.

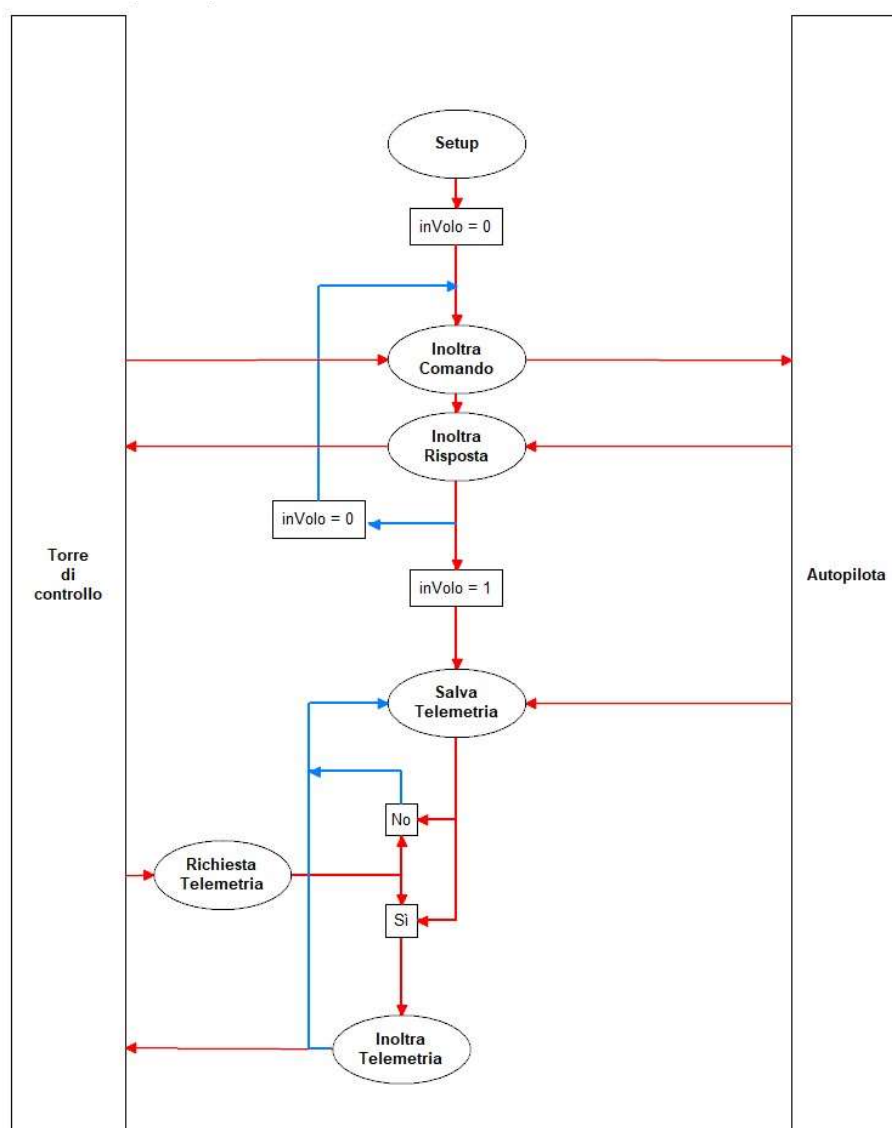


Figura 12: Schema di funzionamento di Relay: la variabile *inVolo* indica il cambio di stato del programma.

Inoltre, dato che il valore di *inVolo* viene trasmesso da Torre di controllo a Relay in vari momenti, l'utilizzatore può evitare di resettarlo manualmente.



### 3.3 Autopilota

Questo è il programma che controlla direttamente l'aereo: una volta ricevute tutte le informazioni necessarie e avviato il volo, è completamente autonomo.

Anche Autopilota è diviso in due sezioni principali, *Setup* e *Loop*:

#### 3.3.1 Setup

È la prima fase di autopilota, dove viene verificato il corretto funzionamento dell'aereo e dei suoi sistemi e vengono ricevute tutte le informazioni necessarie dal computer, tra cui il piano di volo e i parametri per l'utilizzo delle superfici di controllo.

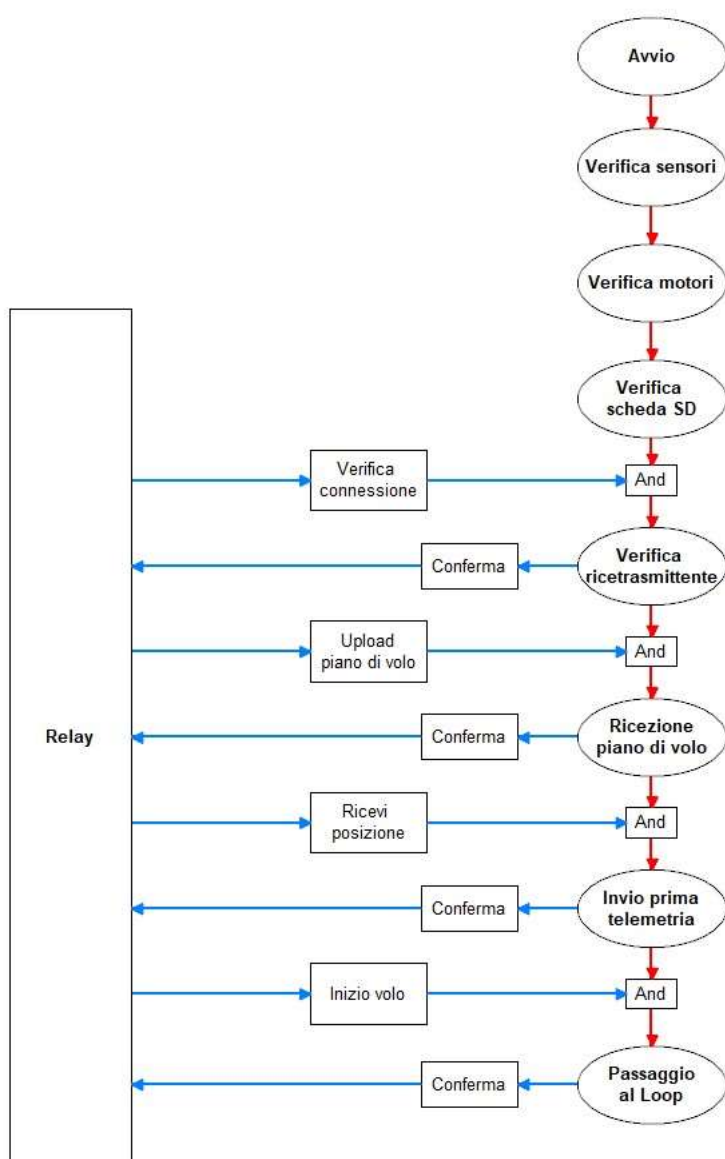


Figura 13: Schema di funzionamento di Setup.

All'avvio del programma, prima di stabilire una connessione con Relay, vengono testati il motore, i servo e i sensori: dato che i primi sono collegati a parti mobili (rispettivamente, elica e superfici di controllo), il modo scelto per verificarne il funzionamento è il controllo visivo di una sequenza di movimenti che avviene dopo quella dei sensori (in particolare, una connessione stabile per il GPS può richiedere anche qualche minuto); in questo modo, i motori indicano il successo o il fallimento di queste operazioni, in base alla sequenza dei loro movimenti.

Poi si passa alla verifica della microSD (e alla creazione di un file dedicato a quel volo), e quindi della ricetrasmittente: prima attraverso una funzione che ne controlla i collegamenti, poi attraverso un tentativo di connessione con Relay; da questo momento i controlli pre-volo sono terminati, e, se effettuati con successo, si può passare alla fase successiva.

Qui l'autopilota riceve dal computer tutte le informazioni necessarie al volo (queste sono salvate nel file creato precedentemente). In ordine:

- waypoints;
- modi;
- costanti per il PID.

Se il piano di volo è stato ricevuto correttamente, Autopilota invia al computer una stringa di dati sulla telemetria, e attende il permesso di cominciare il volo (fig. 14).

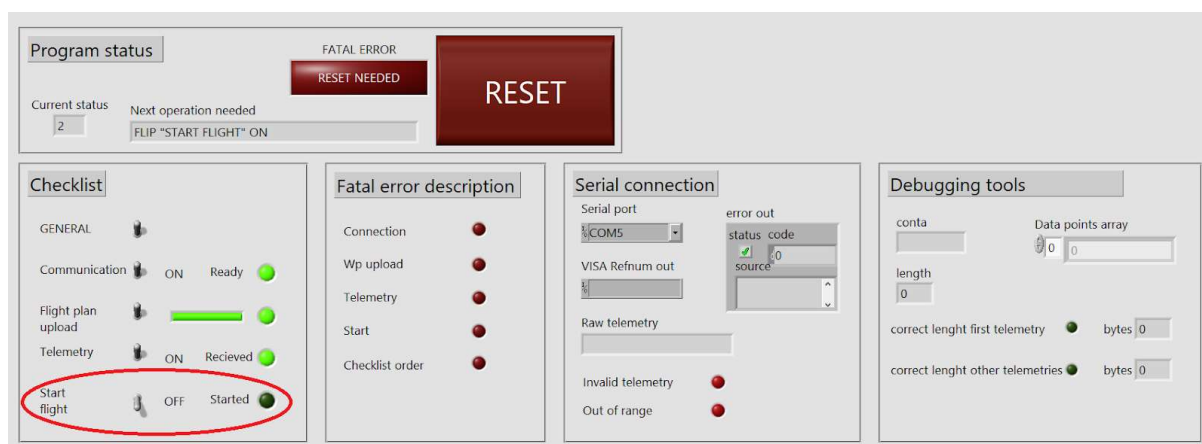


Figura 14: Se il Setup viene completato con successo, la relativa schermata di Torre di controllo si presenta così; per iniziare il volo, e abbandonare quindi la fase di setup, basta premere l'interruttore Start flight, ultimo passaggio della checklist da eseguire prima del decollo.

### 3.3.2 Loop

È la sezione di Autopilota che controlla l'aereo durante il volo, dal decollo all'atterraggio; come si può intuire dal nome, è un ciclo che viene eseguito fino al raggiungimento dell'ultimo waypoint. Svolge principalmente due funzioni: calcola la rotta da seguire (ed esegue di conseguenza le manovre necessarie) e trasmette al computer i dati sul volo.

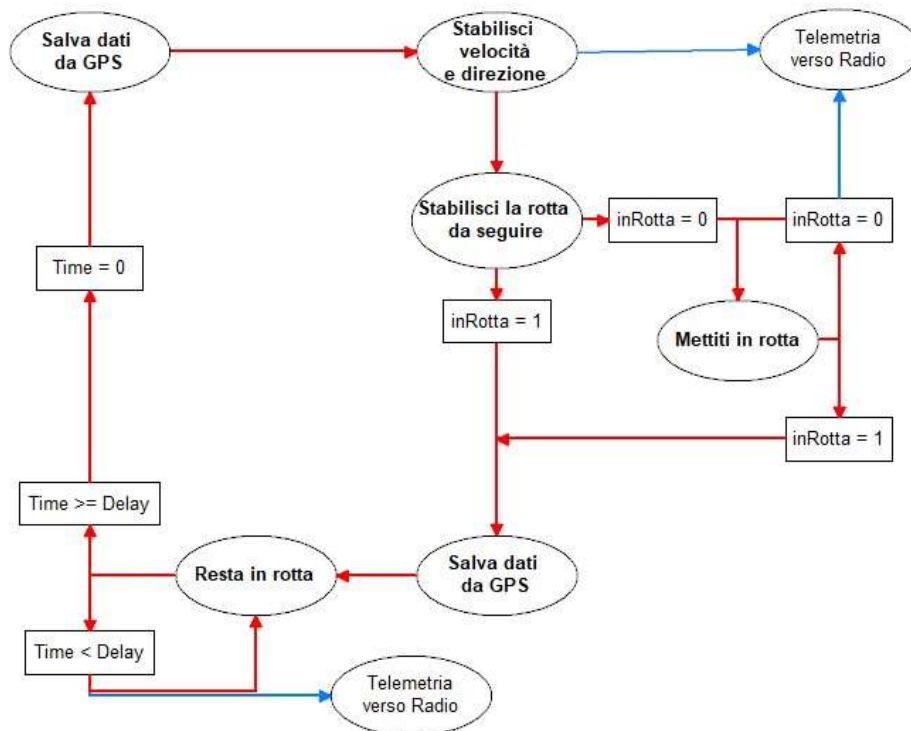


Figura 15:  
Schema  
generale di  
funzionamento  
di Loop.

La prima azione eseguita in Loop è la raccolta dei dati dal GPS, che genera il secondo punto necessario a determinare la rotta GPS. Per il primo ciclo *Stabilisci velocità e direzione* fornirà valori completamente sbagliati, ma questo non è un problema in quanto l'aereo deve ancora decollare. Il primo punto necessario per la prossima iterazione di *Stabilisci velocità e direzione* è ricavato successivamente, prima di *Resta in rotta*. Questa decisione è stata presa perché *Resta in rotta* è la fase in cui l'aereo mantiene l'assetto per un certo tempo (detto *Delay*), ed è quindi il momento più indicato per la raccolta di punti per il calcolo della rotta GPS. *Stabilisci la rotta da seguire* è la fase in cui viene identificata dal piano di volo l'istruzione da seguire e viene calcolata la nuova rotta, mentre *Mettiti in rotta* è la funzione che controlla le superfici mobili dell'aereo per raggiungerla. Infine *inRotta* è un valore booleano che indica se l'aereo sta seguendo la rotta corretta.

Durante ogni ciclo la telemetria viene trasmessa in almeno 3 momenti (*Mettiti in rotta* viene eseguita più volte, e per ogni iterazione viene inviata la telemetria, per cui in base alla manovra può essere che questo avvenga anche decine di volte), il che consente di seguire da Torre di controllo tutte le fasi di ogni manovra.

### **Esecuzione di una manovra:**

Durante il volo, l'aereo esegue il loop finché non viene spento o resettato manualmente. Di seguito sono indicate soltanto le istruzioni presenti nel loop (fig. 15) che riguardano la navigazione (ovvero *Stabilisci la rotta da seguire*, *Mettiti in rotta* e *Resta in rotta*), senza approfondire quelle riguardanti i sensori e la trasmissione dati:

- *Stabilisci la rotta da seguire*: fornisce  $\Delta$  direzione, inclinazione e angolo di rollio. Prima di tutto effettua un rilevamento della rotta magnetica, e utilizza la direzione rilevata come '0'. Dopodiché, aggiorna  $\Delta$  direzione, inclinazione e angolo di rollio, secondo le indicazioni fornite dall'istruzione corrente del modo del waypoint: o utilizza quelli indicati dall'istruzione, o punta il waypoint; in questo secondo caso, considera la direzione corrente quella ricavata dalla rotta GPS (dal primo al secondo rilevamento GPS), e  $\Delta$  direzione e inclinazione sono calcolati come differenza tra la rotta che va dal secondo rilevamento GPS al waypoint e la rotta GPS, mentre il rollio resta quello indicato dall'istruzione corrente;
- *Mettiti in rotta* (fig. 16.1 - 16.6): è un ciclo contenente 3 controllori PID, i cui valori saranno chiamati PID\_beccheggio, PID\_rollio e PID\_imbardata. Il loro compito è verificare le correzioni da apportare per raggiungere, rispettivamente, inclinazione, angolo di rollio e  $\Delta$  direzione: per effettuarle il più precisamente possibile, all'inizio del ciclo vengono effettuati un rilevamento dall'accelerometro/giroscopio e uno della rotta magnetica. Dato però che non c'è una superficie di controllo dedicata all'imbardata, PID\_imbardata viene sommato a PID\_rollio: l'aereo tende a rollare verso la rotta (fig. 16.2), e "scivola" lungo un piano ideale parallelo all'ala (fig. 16.3); la presenza dell'impennaggio verticale fa sì che la coda scivoli più lentamente della prua, e in questo modo l'inclinazione corrente cresce verso il basso; infine, PID\_beccheggio aumenta, correggendo l'inclinazione dell'aereo (fig. 16.4), che ora si trova parallelo alla rotta;
- *Resta in rotta*: agisce esattamente come *mettiti in rotta*, ma  $\Delta$  direzione, inclinazione e angolo di rollio sono tutti inizialmente pari a 0 (essendo l'aereo già posizionato sulla rotta corretta). La differenza consiste nella condizione di uscita dalla funzione: *Mettiti in rotta* termina non appena raggiunta la rotta, mentre *Resta in rotta* agisce per un tempo non inferiore a *Delay* (il tempo viene verificato al termine di ogni ciclo, in modo da non rischiare di interrompere una manovra in corso).

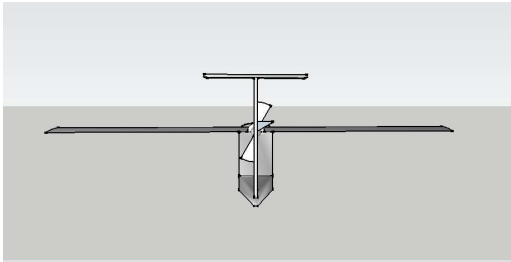


Figura 16.1: Posizione iniziale, prima di una virata di 30° a sinistra.

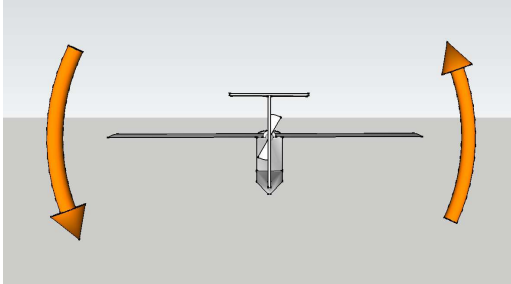


Figura 16.2: Rollio verso sinistra: i controllori di rollio e imbardata si sommano, così da variare il rollio anche in base a  $\Delta$  direzione.

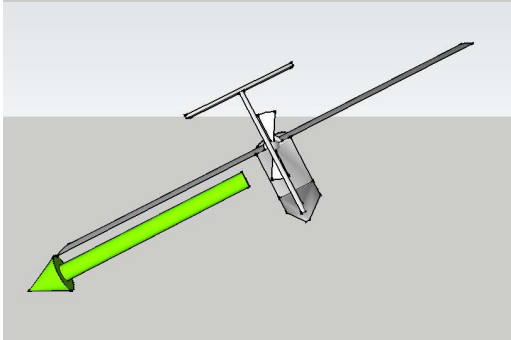


Figura 16.3: L'aereo scivola lungo un piano circa parallelo all'ala; l'impennaggio verticale oppone resistenza, scivolando più lentamente della prua (che è più vicina al baricentro dell'aereo, cioè appena sotto l'ala).

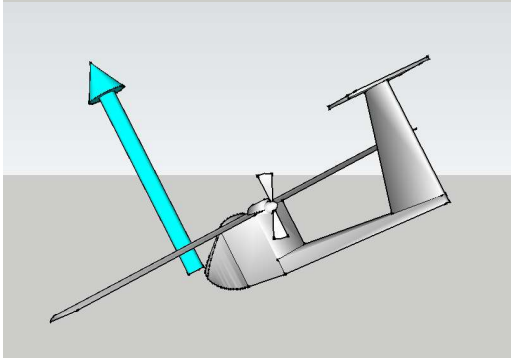


Figura 16.4: L'aereo sta picchiando, quindi il PID\_beccheggio cresce per riportarlo all'inclinazione corretta. così facendo, inoltre, contribuisce alla virata: infatti più il rollio aumenta, più aumenta la componente del beccheggio parallela al terreno.

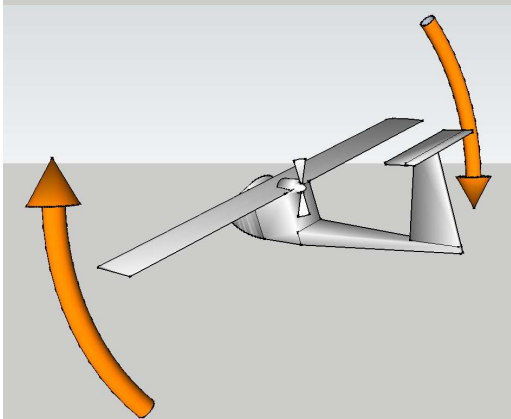


Figura 16.5: Ora che è stata raggiunta la rotta corretta, per evitare deviazioni indesiderate è necessario azzerare il rollio.

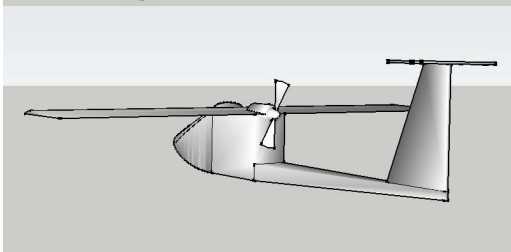
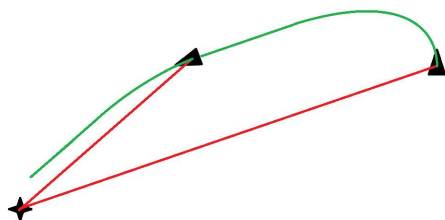


Figura 16.6: La virata è completa e l'aereo è di nuovo in volo lungo la rotta richiesta.

Come descritto in [1], anche se le figure 16.1-16.6 sono utili a comprendere le operazioni che compongono una virata, è importante notare che le fasi da 2 a 5 avvengono in contemporanea: l'angolo di rollio deve aumentare o diminuire in base alla deviazione attuale dalla rotta, mentre quello di beccheggio deve rimanere costante durante tutta la manovra; di conseguenza la posizione istantanea delle superfici di controllo viene ricalcolata più volte al secondo, in base alla frequenza indicata nel piano di volo (10 Hz è quella impostata di default, ma è possibile raggiungere un massimo di 140 Hz).

Ovviamente, dato che l'aereo non vira sul posto, la rotta che raggiungerà non sarà sovrapposta a quella richiesta, ma soltanto parallela (fig. 17). Di conseguenza, prima di passare al waypoint successivo saranno richieste più iterazioni del ciclo con la stessa istruzione; comunque, ad ogni iterazione la distanza tra rotta e rotta corrente viene significativamente ridotta, per cui in pochi secondi si arriva lo stesso ad una rotta che porta abbastanza vicino al waypoint da rientrare nei margini di errore.

Sarebbe possibile calcolare il raggio di una virata, in modo da richiedere una sola iterazione, ma dato che è un parametro strettamente correlato alle prestazioni e dimensioni dell'aereo stesso (e che richiederebbe molti voli di prova per essere determinato in maniera affidabile), questo metodo contrasta con l'idea di un dispositivo universale e facile da usare, per cui il primo metodo si adatta meglio alle specifiche scelte per questo software.



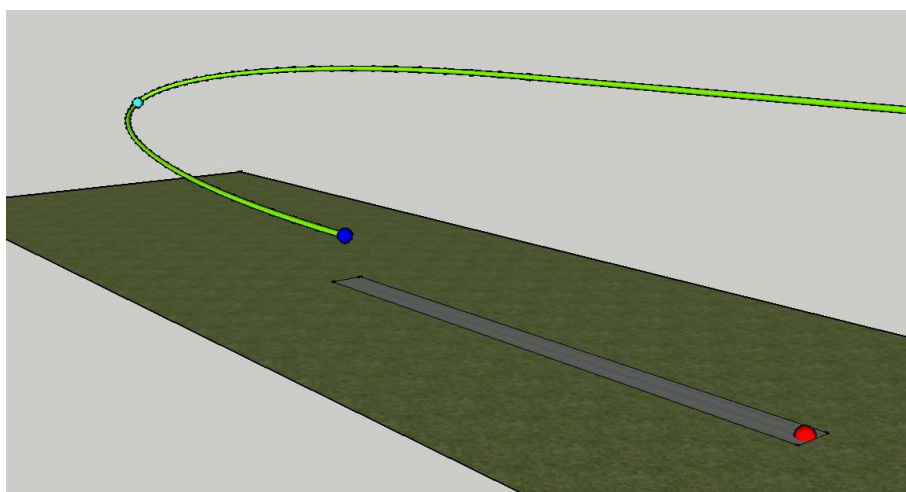
*Figura 17: L'aereo non raggiunge la rotta (rosso), ma una rotta parallela (verde). La distanza tra l'aereo e la rotta sarà tanto maggiore tanto più ampia la virata che deve eseguire, ma già dopo la seconda iterazione si può notare come tale distanza sia significativamente ridotta.*

Inoltre, causa dell'imprecisione della rotta magnetica, è probabile che siano necessari più cicli perché l'aereo punti effettivamente verso il waypoint: infatti più  $\Delta$  direzione è elevato, più lo è anche l'errore tra rotta e rotta magnetica alla fine della manovra (idealmente, sarebbe nullo). In alternativa, se  $\Delta$  direzione viene definito dall'istruzione del modo, la direzione corrente è calcolata solo attraverso il sensore magnetico, per cui è possibile che rotta e rotta corrente differiscano sostanzialmente. È quindi consigliabile inserire tra i modi alcune impostazioni che richiedano il raggiungimento del waypoint, così da poter utilizzare i rilevamenti GPS (e mitigare quindi la differenza tra la rotta effettivamente percorsa dall'aereo e quella desiderata dall'utilizzatore).

### Atterraggio e fine del volo:

Dato che per memorizzare i waypoints si utilizza un array di dimensione predeterminata, quelli che non vengono utilizzati sono contrassegnati da una variabile; in questo modo, il loop può determinare se l'ultimo waypoint è stato raggiunto con successo, che significherebbe la fine del volo.

Alternativamente (fig. 18), il motore può essere spento durante l'avvicinamento all'ultimo waypoint (posizionato alla fine della pista d'atterraggio): l'aereo quindi planerà lungo la pista, picchiando leggermente, finché non toccherà il suolo, presumibilmente prima del waypoint. Anche se così facendo l'aereo potrebbe non segnalare la fine del volo al computer, nei primi test questo metodo è consigliabile perché il motore sarà spento prima dell'atterraggio, prevenendo possibili danni all'elica e ribaltamenti (il centro di spinta nel modello scelto è relativamente alto, il che potrebbe essere dannoso se non viene seguita questa procedura o se il controllore non è ancora calibrato correttamente).



*Figura 18: Esempio di traiettoria consigliata per l'atterraggio: 2 waypoints (azzurro e blu) servono ad allineare l'aereo con la pista; il terzo waypoint (rosso) contiene soltanto due istruzioni: la prima, che viene eseguita appena superato il waypoint blu, è di spegnere il motore, mentre la seconda è di raggiungere il waypoint rosso. così l'aereo planerà lungo la pista, fino a toccarla e quindi fermarsi (prima di aver raggiunto il waypoint rosso).*

# Capitolo 4

## Hardware

### 4.1 Specifiche del velivolo

Nonostante il pilota automatico sia pensato per operare sulla gamma più vasta possibile di aeromodelli, per creare un velivolo adatto ai test può essere stati seguiti alcuni accorgimenti:

È stato costruito utilizzando il più possibile il foam, perché è un materiale poco costoso, leggero e facile da tagliare con precisione. Quindi, in caso di incidenti, è un aereo più facile e veloce da riparare rispetto ad uno di balsa e tela, stampato in 3D, o realizzato in vetroresina.

Come per gli aerei con pilota, uno dei momenti più pericolosi nel volo di un aeromodello è l'atterraggio [2]; nel caso di un modello economico, senza carrello o pattini, questo avviene di solito su un prato, e visto che l'intera superficie inferiore dell'aereo, e in base all'altezza dell'erba anche parte di quella frontale, agisce da freno grazie all'attrito con il tessuto erboso, è relativamente brusco. Di conseguenza, se l'assetto dell'aereo non è adeguatamente controllato, è molto probabile che questo venga danneggiato durante l'atterraggio; date queste condizioni, qui sono riportati i due incidenti più comuni:

- ribaltamento in avanti (fig. 19): se l'aereo sta picchiando troppo, toccherà terra a partire dal muso; le conseguenze più gravi normalmente sono la rottura dell'elica e il danneggiamento dell'impennaggio verticale;
- rotazione intorno ad un'ala (fig. 20): se mentre atterra l'aereo sta rollando oltre un certo angolo, toccherà terra a partire dall'estremità di un'ala; le conseguenze più gravi normalmente sono la rottura dell'elica e il danneggiamento dell'ala che ha toccato terra per prima.



Le seguenti immagini<sup>2</sup> ritraggono incidenti realmente avvenuti:



*Figura 19.1: l'aereo atterra con una picchiata eccessiva: si può notare come l'elica sia pericolosamente vicina al terreno;*

*Figura 19.2: l'elica impatta con il terreno, causando il ribaltamento del velivolo;*

*Figura 19.3: il violento impatto probabilmente ha danneggiato l'elica e l'impennaggio verticale. Inoltre, l'ala non è stata deformata soltanto grazie alla presenza dei due rinforzi diagonali.*



*Figura 20.1: l'aereo atterra con un eccessivo rollio, e l'ala destra tocca terra assieme al carrello;*

*Figura 20.2: essendo la forza frenante sbilanciata pesantemente verso la destra del velivolo, questo ruota velocemente attorno al punto di contatto tra ala e terreno;*

*Figura 20.3: i danni più evidenti sono la deformazione dell'elica e il distaccamento del carrello, e anche in questo caso la presenza di rinforzi ha prevenuto la deformazione dell'ala.*

---

<sup>2</sup> [https://www.youtube.com/watch?v=mkOyyG\\_rXSg](https://www.youtube.com/watch?v=mkOyyG_rXSg)

Le scelte progettuali volte a limitare al massimo i danni in caso di atterraggio scorretto sono allora l'utilizzo di un'elica spingente e posizionata il più in alto possibile ed un'ala alta con rinforzi diagonali [3]: l'altezza dell'ala consente un angolo di rollio maggiore rispetto all'ala bassa, mentre l'elica spingente è più al riparo da impatti diretti rispetto ad una trainante; inoltre, l'altezza dell'impennaggio verticale la protegge almeno parzialmente da impatti dovuti ai ribaltamenti.

Perciò il modello scelto per testare il pilota automatico è la versione in scala 1:5,7 dello Sky Arrow (prodotto da 3I):



*Figura 21: Foto del modello costruito dal progettista originale<sup>3</sup>*

L'elevata apertura alare (170 cm) rende possibile l'installazione del sistema di pilotaggio senza che il suo peso influisca eccessivamente sulle capacità del velivolo, e rispetta tutti i requisiti di cui sopra: ala alta rinforzata, elica spingente, grande impennaggio verticale.

Nonostante il progetto originale sia già adatto allo scopo, alcune piccole modifiche ne possono ulteriormente migliorare le caratteristiche.

Per limitare i pesi a bordo, è stato deciso di ridurre il numero di servomotori necessari a gestire le superfici di controllo da 3 a 2, eliminando il timone di direzione (ma mantenendo l'impennaggio); questo è possibile perché i piccoli aggiustamenti alla rotta che dovrebbe operare possono essere eseguiti anche attraverso le altre superfici di controllo: su aerei con persone a bordo si preferisce evitare questa soluzione soprattutto perché il continuo rollio causa nausea molto più facilmente, ma su un aeromodello questo non è un problema.

Un vantaggio secondario di questa decisione è che, essendo ora l'impennaggio verticale una superficie unica, è più resistente e più facile da riparare. Inoltre, per lo stesso scopo è possibile eliminare il carrello: questa scelta elimina la possibilità di atterrare su una pista, ma in compenso consente di semplificare molto la procedura di atterraggio e di alleggerire ulteriormente, per quanto poco, il velivolo.

---

<sup>3</sup> <http://rc-plans.com/catalog/item303.html>

## 4.2 Controllori

L'autopilota è installato su un Arduino MEGA<sup>4</sup>: la scelta di un Arduino è stata dettata dal prezzo accessibile e dall'enorme quantità di librerie per sensori, motori e ricetrasmittitori già esistenti e disponibili gratuitamente. La versione MEGA è necessaria per la maggiore memoria disponibile e il numero di pin dedicati alla comunicazione e alla PWM (indispensabili visto il numero di sensori e motori necessari, senza considerare quelli che l'utilizzatore può decidere di aggiungere).

Un secondo Arduino viene utilizzato come ricetrasmittitore dal PC a terra, per permettere la comunicazione via radio tra quest'ultimo e l'aereo. Dato che in questo ruolo memoria e numero di pin sono significativamente meno importanti, la scelta migliore è stata l'Arduino NANO<sup>5</sup>, soprattutto per il minor prezzo.

## 4.3 Sensori

Il compito principale della suite di sensori è l'identificazione delle 6 coordinate:

- *latitudine* e *longitudine* si possono ricavare utilizzando un ricevitore GPS;
- l'*altitudine* viene calcolata con un barometro;
- *beccheggio* e *rollio* con un accelerometro, e con un giroscopio la velocità angolare di queste due.

Anche l'imbardata può essere calcolata integrando nel tempo le misurazioni di un giroscopio, ma dato l'inevitabile errore che il processo di integrazione genera sulla misurazione, è più sicuro utilizzare al suo posto la *rotta corrente*; il suo valore si ricava elaborando le letture dell'accelerometro e di un sensore magnetico a 3 assi, oppure attraverso due rilevamenti GPS distanziati nel tempo.

Per la realizzazione del prototipo i sensori scelti sono:

- ricevitore GPS: NEO-6M<sup>6</sup>;
- accelerometro/giroscopio: GY-521<sup>7</sup>;
- sensore magnetico: GY-271<sup>8</sup>;
- barometro: GY-BMP 280<sup>9</sup>.

Gli unici fattori che hanno influenzato questa scelta sono meramente economici e di esistenza di librerie dedicate ad Arduino, per cui, se le condizioni di utilizzo dell'aereo richiedono una suite differente, l'utilizzatore può scegliere i componenti che più ritiene opportuni (per esempio, l'altitudine può essere stimata anche con un telemetro, o con il GPS stesso, anche se quest'ultimo sistema non è stato ritenuto abbastanza affidabile per quest'applicazione).

---

<sup>4</sup> <https://store.arduino.cc/arduino-mega-2560-rev3>

<sup>5</sup> <https://store.arduino.cc/arduino-nano>

<sup>6</sup> <https://www.electroschematics.com/neo-6m-gps-module/>

<sup>7</sup> [http://win.adrirobot.it/sensori/MPU-6050/sensore\\_MPU-6050.htm](http://win.adrirobot.it/sensori/MPU-6050/sensore_MPU-6050.htm)

<sup>8</sup> <https://www.meccanismocomplesso.org/arduino-magnetic-magnetic-magnetometer-hmc58831/>

<sup>9</sup> <https://startingelectronics.org/pinout/GY-BMP280-pressure-sensor-module/>

Da notare che per quanto riguarda latitudine e longitudine, c'è una notevole discrepanza tra la precisione dei valori forniti dal GPS (stimata attorno a 2,5 m in condizioni ottimali, a seguito di opportune prove) e l'incertezza delle variabili salvate ( $10^{-7}$  gradi sia di latitudine che di longitudine, ovvero distanze inferiori ad 1 cm). Questo è dovuto al numero minimo di bits necessari a garantire un'incertezza inferiore a 2,5 m (corrispondenti a  $9 \times 10^{-6}$  gradi), ovvero 26, che in un'architettura ad 8 bits come quella utilizzata qui si traduce in una variabile di 4 bytes (32 bits).

Un vantaggio secondario di questa differenza è che sarà possibile integrare sistemi più precisi per il rilevamento della posizione senza dover apportare modifiche al software.

Invece tutte le altre distanze angolari utilizzate hanno un'incertezza di 1 grado, ritenuta sufficiente per lo scopo di questo progetto.

Per facilitare l'utilizzo di un'altra suite di sensori, le funzioni per la lettura dei dati si trovano in un modulo a parte; vale la stessa cosa per il ricetrasmittitore, a cui ne è dedicato un altro. In questo modo, se si dovessero integrare dei sensori differenti, per renderli funzionanti basterà modificare un modulo relativamente breve anziché dover cercare all'interno del programma vero e proprio (più lungo e complesso) le funzioni interessate.

## 4.4 Altri componenti

Vista la necessità di una linea di comunicazione tra aereo e computer, il velivolo richiede una ricetrasmittente, e dato che le distanze potrebbero, potenzialmente, superare il chilometro, l'unico metodo di trasmissione consigliabile è la radio; infatti, i moduli bluetooth e wi-fi non sono affidabili oltre qualche decina di metri, mentre gli infrarossi sono da escludersi per l'interferenza causata dal Sole. Per questo è stato scelto il modulo radio SI4432<sup>10</sup>, anche qui in funzione soltanto del costo e della disponibilità di una libreria per Arduino.

Inoltre, per assicurare la possibilità di studiare il volo, i dati di questo vengono salvati su una microSD: in questo modo, anche in caso di problemi di ricezione, sarà possibile studiare i dati integrali, e non solo quelli la cui trasmissione è riuscita. Per cui è stato necessario acquistare un lettore per queste schede, e vista l'enorme scelta, ma anche l'omogeneità delle caratteristiche, che il mercato offre per questi moduli, la scelta è ricaduta ancora su quello più economico, prodotto dalla HER.

Arduino MEGA inoltre offre la possibilità di collegare altri componenti, dato il numero di PIN disponibili: sia sensori, come un tubo di pitot, che servomotori; in particolare, questi ultimi possono risultare indispensabili per l'utilizzo in velivoli più grandi o con prestazioni migliori: l'utilizzo di flap, di un timone di direzione o di un carrello retrattile può migliorare notevolmente le capacità dell'aereo, da una migliore aerodinamicità, e quindi dell'autonomia, ad una portanza variabile, che estende sia il range di velocità che di pressioni atmosferiche (e quindi di altitudini) in cui può operare.

---

<sup>10</sup> <https://www.alldatasheet.net/datasheet-pdf/pdf/538051/SILABS/SI4432.html>

# Capitolo 5

## Sviluppo dell'aereo

### 5.1 Layout

Per collegare efficacemente i vari elementi che compongono l'hardware dell'aereo, è stato realizzato un circuito stampato (fig. 22) sotto al quale è posizionato l'Arduino MEGA, mentre al di sopra sono stati duplicati tutti i suoi pin (ad eccezione di 6, posizionati al centro e inutili ai fini di questa piattaforma), di cui alcuni vengono collegati attraverso le piste ai rispettivi componenti. Inoltre, sono stati predisposti i circuiti per due regolatori di tensione, i cui output sono 5 V e 3,3 V (ovvero le tensioni di alimentazione dell'Arduino MEGA e di tutti i componenti utilizzati ad eccezione del motore collegato all'elica, che viene alimentato separatamente), nel caso in cui la potenza erogabile dall'Arduino sia insufficiente. In base alle prove eseguite, con i componenti correnti è sufficiente il regolatore per i 5 V, mentre i 3,3 V vengono erogati dall'Arduino stesso.

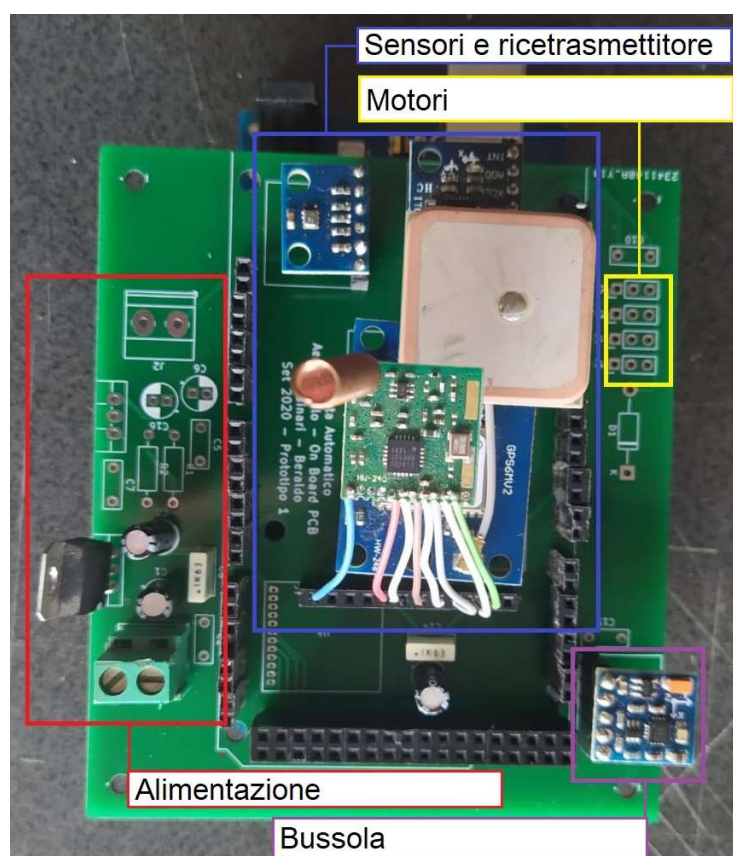


Figura 22: vista dall'alto del circuito stampato; sono evidenziate le 4 sezioni in cui è stato diviso. La bussola è stata posizionata vicino a pin inutilizzati e il più possibile distante da quelli dei motori e di alimentazione per cercare di limitare le interferenze nella lettura del campo magnetico terrestre.

Questo circuito ha due buone caratteristiche: è abbastanza compatto (10 x 8,7 cm), quindi può essere inserito anche in aeromodelli relativamente piccoli, e permette un facile accesso a tutti i pin dell'Arduino, così da permettere di cambiare o aggiungere componenti molto rapidamente e senza dover stampare un nuovo circuito. Questo è avvenuto per esempio con il lettore SD, il cui inserimento è stato deciso in un secondo momento, senza che questo abbia richiesto un nuovo circuito stampato, dato che è stato possibile collegarlo direttamente ai pin dell'Arduino.

## 5.2 Costruzione

Anche se il progetto originale (v.di *Specifiche del velivolo*) è stato seguito quasi alla perfezione, a seguito dei primi tre voli di prova (di cui i primi due conclusi con uno schianto al suolo) sono state apportate varie modifiche (figura 23.2 - 23.3).

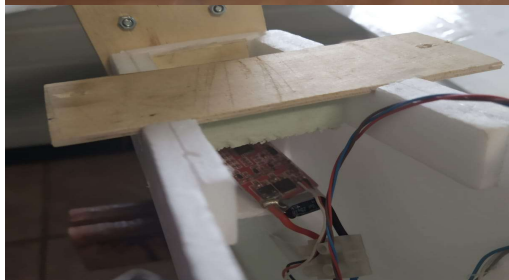
Nella versione iniziale, la differenza più importante tra l'aereo costruito e le indicazioni del progetto è il materiale utilizzato per i supporti diagonali dell'ala: anziché una sottile striscia di alluminio, è stato utilizzato del filo di nylon; inoltre i pezzi di collegamento tra ala e fusoliera non sono inseriti in quest'ultima, ma semplicemente incollati al di sopra.



*Figura 23.1: Prima versione dell'aereo: la fusoliera non è stata chiusa davanti per mostrare la posizione della scheda del pilota automatico; mancano ancora il supporto del motore e quelli diagonali dell'ala.*



*Figura 23.2: Immagine tratta dalle istruzioni originali per la costruzione dell'aereo: si noti come i supporti in legno siano stati "annegati" nella fusoliera.*



*Figura 23.3: Fotografia di uno dei supporti (quello posteriore) dell'ala realizzati per questo progetto: hanno una sezione a T e sono realizzati in foam (verde) e legno e sono incollati alla fusoliera.*

### **Primo volo:**

Lo schianto del primo volo è avvenuto a partire dal contatto tra l'estremità sinistra dell'ala e il terreno, e si è svolto esattamente come descritto in precedenza (v.di cap. 4.1, *Specifiche del velivolo*). Le modifiche apportate sono:

- Eliminati i supporti diagonali dell'ala: il danno arrecato a questa dalla tensione del filo, nonostante i rinforzi sul punto di collegamento tra filo ed ala, sono stati più gravi di quelli causati all'ala dalla caduta stessa;
- Rinforzata internamente la fusoliera con strisce di foam longitudinali: il violento contatto con il terreno ha causato il distacco della coda e la deformazione del muso, per cui si è dimostrata necessaria l'aggiunta di rinforzi lungo la base dell'aereo (in pratica una *chiglia*) e attorno al punto in cui viene inserita la cabina.

### **Secondo volo:**

Durante il secondo incidente l'aereo ha colpito il ramo di un albero con l'estremità sinistra dell'ala e si è violentemente avvitato attorno al suo asse verticale (condizione detta *vite piatta*) ed è caduto quasi perfettamente sulla base della fusoliera, con il muso inclinato leggermente (circa 30 gradi verso il basso). Lo schianto ha causato il distacco dell'ala dalla fusoliera (i supporti sono rimasti attaccati all'ala), dimostrando vincente la scelta di eliminare i fili di nylon: infatti l'energia dell'impatto è stata assorbita per lo più dallo strato di colla (molto semplice da rimpiazzare) tra supporti e fusoliera, lasciando questi praticamente intatti. Le modifiche apportate sono:

- Cambio del tipo di colla tra supporti dell'ala e fusoliera, da colla caldo a colla acrilica: la colla a caldo quando viene applicata causa la contrazione del foam, per cui utilizzarla più volte sullo stesso punto potrebbe compromettere la struttura;
- Riposizionamento dello stabilizzatore orizzontale (fig. 24): nell'impatto col terreno si è staccato dal timone di coda, probabilmente a causa della zona molto sottile su cui era stato incollato (circa 70 mm in lunghezza e 8 mm in larghezza); per cui è stato posizionato alla base del timone, potendo così essere incollato sia sopra che sotto, e sotto su una superficie più ampia (larga circa 25 mm);
- Cambio dell'elica: l'elica scelta inizialmente non sfruttava appieno la potenza del motore, per cui ne è stata installata una con passo e superficie delle pale maggiori.



*Figura 24: Nuova posizione dello stabilizzatore orizzontale: in questa configurazione, gli sforzi che il timone deve sopportare sia durante il volo sia in caso di impatto col terreno sono significativamente ridotti.*

### Terzo volo:

Questo volo si è concluso con un atterraggio abbastanza morbido: grazie ai voli precedenti, è stato possibile tarare meglio le superfici di controllo, permettendo la riuscita di questo. È stato però deciso di reintrodurre i fili di nylon come supporti diagonali per le ali, con ulteriori rinforzi nei punti di collegamento con ala e fusoliera: infatti un controllo più preciso dell'aereo, unito alla maggior potenza ottenibile con la nuova elica (circa il 20% in più), ha significativamente ridotto il rischio di impatti violenti con il suolo, rendendo superflua, se non deleteria, la funzione sacrificale dei supporti in legno dell'ala (l'ala si è separata dalla fusoliera nonostante non avesse ricevuto colpi eccessivi). È stato comunque modificato il sistema con cui i fili sono collegati alla fusoliera: ora la loro tensione può essere regolata con precisione prima di ogni volo attraverso una vite, anziché il metodo precedente, che consisteva nell'annodarli a mano.

Inoltre sono stati inseriti alcuni spessori tra l'ala e la fusoliera, e un timone di coda fisso leggermente inclinato verso sinistra: così sono stati modificati gli angoli relativi tra le varie superfici dell'aereo, rendendolo molto più agile.

## 5.3 Esecuzione di un volo

Il quarto (ed ultimo) volo radiocomandato è stato un successo: l'aereo si è dimostrato abbastanza maneggevole e reattivo ai comandi.

L'ampia superficie alare ha consentito basse velocità di decollo e atterraggio, permettendo di effettuare correzioni molto velocemente.



*Figura 25: Aereo durante il quarto volo: è stato scelto di volare ad una quota relativamente bassa per sfruttare la protezione dal vento fornita dagli alberi.*

Purtroppo, prima a causa delle avverse condizioni meteo (soprattutto il forte vento) e poi dell'aggravarsi della pandemia, non è stato possibile accedere nuovamente al campo di volo per volare con il pilota automatico attivo, per cui non è stato ancora eseguito un volo in autonomia, ma sicuramente questo avverrà non appena sarà possibile.



# Capitolo 6

## Risultati sperimentali

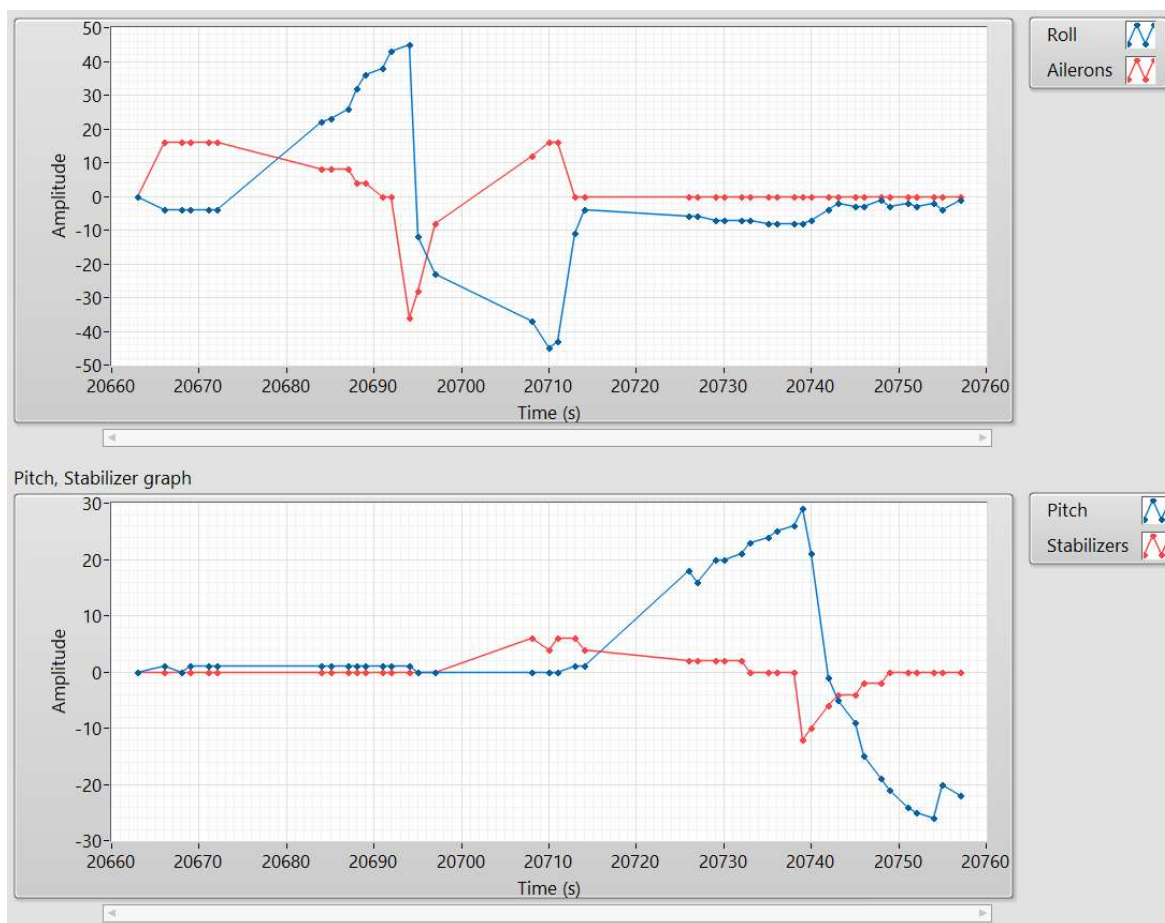
Una volta completati sia i programmi che il circuito di Autopilota, sono stati eseguiti vari test per scoprire e correggere eventuali errori nel sistema, in modo da assicurare la sua capacità di gestire autonomamente un volo.

Di seguito, i test più significativi:

**Test dei controllori:** progettato per verificare il corretto svolgimento del loop e l'azione dei servo collegati alle superfici di controllo. L'aereo è stato mosso a mano, e il piano di volo prevede il rollio prima a  $45^\circ$  e poi a  $-45^\circ$ , seguito da beccheggio prima di  $30^\circ$  e poi di  $-30^\circ$ , mentre la posizione rispetto al waypoint non viene presa in considerazione.

```
Waypoints
04571550600 01219735200 0015 01
modeType
01 00 0000 0045 0000 04 0045
01 01 0000 -045 0000 04 -045
01 02 0030 0000 0000 05 0030
01 03 -030 0000 0000 05 -030
pid
0010 0001 0000 0010 0001 0000 0000 0000 0100
end
```

*Figura 26.1: Piano di volo; il test è stato ripetuto più volte, con diversi valori per i PID e per gli angoli richiesti. Questa configurazione è quella corrispondente ai risultati riportati in seguito.*



*Figura 26.2*

Il test è stato svolto correttamente più volte, con diverse configurazioni per i controllori: la loro risposta ha avuto verso e modulo corrispondente alle aspettative in tutte le configurazioni, e questo, assieme alle prestazioni dell'aereo valutate nei voli radiocomandati, ha permesso di identificare le impostazioni più promettenti per il primo volo autonomo, che nel piano di volo sono rappresentate dalla stringa "0200 0100 0000 0200 0100 0000 0300 0050 0000 0100" (v.di cap. 2, *Il piano di volo*).

Inoltre, è possibile osservare tre momenti in cui non sono stati ricevuti dati corretti: tra i secondi 20674 e 20684, 20700 e 20705 e infine 20715 e 20725; la causa ipotizzata inizialmente è la bassa potenza del trasmettitore, ma a seguito di altri test (v.di *Test per l'ottimizzazione*) si è potuto osservare come in realtà il problema fosse una combinazione di bugs di Relay e Torre di controllo che fermava la ricezione per tempi multipli di 5 secondi in caso di errori nell'ultimo pacchetto di dati ricevuto. Risolti questi, i grafici sono risultati molto più uniformi.

**Test della virata:** progettato per verificare che i controllori rispondano correttamente ad una richiesta di una virata di 30° a sinistra. Questa non è richiesta attraverso il volo tra waypoints, perchè il test è effettuato da fermi, e sfrutta invece l'apposita condizione.

```
Waypoints
04571550600 01219735200 0015 01
04571550600 01219735200 0015 01
modeType
01 00 0000 0000 -030 06 -030
pid
0010 0010 0000 0010 0010 0000 0100 0100 0000 0100
end
```

*Figura 27: Piano di volo; sono presenti due waypoints identici per verificare l'esecuzione di due virate di seguito.*

La prima volta questo test ha portato alla luce due problemi: il dato sulla rotta perpendicolare fornito da sensore magnetico e accelerometro aveva segno invertito, e la deviazione per effettuare una virata veniva aggiunta alla direzione corrente ad ogni ciclo dei PID anziché soltanto la prima volta (praticamente, ad ogni iterazione veniva richiesto di virare di altri 30°, invece di un valore progressivamente più vicino a 0).

La seconda volta ha funzionato come atteso, ovvero i controllori di rollio e imbardata (che vengono sommati) inizialmente hanno fatto rollare l'aereo, per poi riportarlo gradualmente in piano mentre la virata proseguiva. Contemporaneamente, quello del beccheggio ha cercato di mantenere un'inclinazione costante di 0°.

**Test di controllori e motore:** progettato per assicurarsi che il motore funzioni correttamente in simbiosi con gli altri sistemi, intervalla alle istruzioni del primo test (test dei controllori) istruzioni per variare la velocità dell'elica.

```
Waypoints
04571550600 01219735200 0015 01
modeType
01 00 0000 0045 0000 04 0045
01 01 0000 0000 0000 07 0020
01 02 0000 -045 0000 04 -045
01 03 0000 0000 0000 07 0080
01 04 0030 0000 0000 05 0030
01 05 0000 0000 0000 07 0010
01 06 -030 0000 0000 05 -030
01 07 0000 0000 0000 07 0000
pid
0010 0001 0000 0010 0001 0000 0000 0000 0100
end
```

*Figura 28: Piano di volo; si noti il layout delle istruzioni: il raggiungimento del corretto angolo di rollio o beccheggio porta ad una variazione di velocità (in ordine: 20%, 80%, 10% e 0%)*

Durante tutto il programma, compresa la fase di setup, si è rilevata un'interferenza nel segnale che controlla il motore. L'ipotesi è che sia causata dal collegamento seriale che, in questa fase, deve avvenire tra autopilota e computer (il pilota automatico non è ancora dotato di un'alimentazione indipendente; inoltre, il monitor seriale consente di verificare alcune informazioni utili per controllare i punti più critici del programma, come le funzioni che stanno venendo eseguite). La conseguenza è un picco nella velocità del motore dalla durata inferiore al secondo, per cui non è così grave da compromettere un volo; verrà comunque studiata una volta assicurato un sistema di alimentazione separato per il pilota automatico. Altre ipotesi, escluse a seguito di opportune verifiche, sono uno scorretto collegamento a massa dell'ESC (Electronic Speed Controller, necessario per controllare la velocità del motore) e una dichiarazione non voluta del valore dello stesso pin PWM da parte delle librerie dei sensori.

**Test per l'ottimizzazione:** progettato per studiare i tempi richiesti per eseguire singoli processi, come l'acquisizione di dati dai sensori, o intere parti del programma, come il calcolo e raggiungimento di una nuova rotta, utilizza come piano di volo lo stesso del primo test; è stato eseguito più volte, per valutare di volta in volta l'impatto delle modifiche ad Autopilota. Di seguito i risultati delle varie iterazioni:

1. Evidenziate le criticità maggiori: la trasmissione della telemetria richiede 230 ms, mentre il salvataggio su scheda SD ne richiede tra 490 e 510, mentre ogni altro processo non dura più di 10 ms. Di conseguenza, ogni iterazione del ciclo del PID dura circa 740 ms, quindi la frequenza di campionamento per la componente integrale (uguale a quella di aggiornamento della posizione delle superfici di controllo) è inferiore a 1,35 Hz.

Inoltre, ogni secondo circa (in questo caso, ogni 2 cicli PID, quindi ogni 1,5 s) la rotta viene ricalcolata, processo che richiede altri 735 ms.

In conclusione, ogni minuto vengono apportate circa 27 correzioni alla posizione delle superfici di controllo, chiaramente insufficienti.

2. Aumentato il tempo di ricalcolo della rotta da 1 a 3 secondi, eliminate le funzioni di trasmissione e salvataggio dal ciclo PID: ora ogni iterazione di questo dura tra 6 e 7 ms, rendendo possibile una frequenza di campionamento e aggiornamento di circa 140 Hz.

Però ogni 3 secondi c'è un'attesa di 735 ms, come per l'iterazione precedente; quindi, anche se le correzioni al minuto sono ora più di 6000 (massimo 6875), non sono distribuite uniformemente.

3. Eliminata la funzione di trasmissione: il salvataggio dei dati nella scheda SD richiede così tanto tempo perché c'è un conflitto con la ricetrasmittente, tale per cui è indispensabile togliere l'alimentazione alla prima per utilizzare la seconda; questo richiede che siano utilizzate ogni volta le funzioni di riavvio, molto dispendiose in termini di tempo. Considerando poi che la ricetrasmittente utilizzata si è rivelata inaffidabile e che seguire il volo in tempo reale è meno importante di avere dei dati completi, è stato deciso di non utilizzare la ricetrasmittente durante il volo. Così facendo, il salvataggio della telemetria sulla scheda SD richiede soltanto 16 ms, per cui mantenendo un tempo di ricalcolo di 3 secondi è teoricamente possibile effettuare 8495 correzioni al minuto (in media 141,6 al secondo), con un intervallo massimo tra due consecutive di appena 26 ms.

4. Reintrodotta la trasmissione, eliminato il salvataggio su scheda SD: ottimizzando i valori di bit rate e potenza della ricetrasmittente, è stato possibile abbassare il tempo di trasmissione da 230 ms a meno di 6 ms, perciò è possibile inviare pacchetti di dati anche dall'interno del ciclo PID, portandone la durata media a 13 ms e massima a 14 ms; così il numero di correzioni resta comunque molto elevato (4255 al minuto, o una media di 71 al secondo), ma la telemetria viene inviata ad ogni ciclo (quindi ogni 14 ms) anziché ad ogni ricalcolo (cioè ogni 3 secondi) come invece accade per i salvataggi nell'iterazione precedente. Questo permette inoltre di sopperire all'inaffidabilità dei trasmettitori utilizzati, dato che solitamente i problemi si sono verificati con singoli pacchetti di dati.

# Capitolo 7

## Conclusioni

Nonostante non sia ancora stato possibile completare un volo in autonomia, il progetto può essere considerato un successo. Lo sviluppo del software e la costruzione (anziché l'acquisto) dell'aereo hanno consentito di acquisire esperienza e dati importanti per la costruzione di un secondo velivolo, dotato di un sistema di pilotaggio derivato da quello qui esposto, facendo tesoro degli errori fatti e delle soluzioni trovate.

In particolare:

- La divisione in 3 macro sezioni (*Torre di controllo, Relay e Autopilota*) è risultata la più efficace tra le varianti considerate, per cui verrà mantenuta;
- La mancanza del timone di coda non è risultata problematica, per cui il nuovo velivolo manterrà questa configurazione. Però l'utilizzatore potrà selezionare quali superfici sono presenti nel suo aereo, per cui verrà introdotta questa opzione;
- L'interfaccia utente realizzata con LabView non è particolarmente adatta allo scopo, soprattutto per l'impossibilità di renderla gratuita e open source, per cui quella nuova sarà realizzata con Visual Studio;
- Nonostante la grammatica per la scrittura dei file dei piani di volo non sia complicata, la presenza di un editor capace di prevenire eventuali errori renderebbe il processo molto più veloce, per cui verrà introdotto nel nuovo *Torre di controllo*;
- La ricetrasmittente ha un raggio molto breve, per cui ne verrà utilizzata una più potente;
- Arduino è risultato un'ottima piattaforma per sviluppare un prototipo, soprattutto grazie alle librerie già pronte per quasi tutti i sensori utilizzati, ma è stato portato ai suoi limiti, quindi per il nuovo aereo è stato scelto l'stm32.

Comunque, le imperfezioni rimaste in questo primo prototipo non sono molto più che estetiche, e l'aereo una volta collaudato si è dimostrato un ottimo velivolo per sperimentare, per cui continuerà ad essere utilizzato come banco di prova per lo sviluppo del prossimo.

## **Bibliografia**

<sup>[1]</sup>Visani G., *Il nuovo pilota moderno*, Milano, Mursia, 1987

<sup>[2]</sup>Lora-Lamia S., Sport pilot!, in “*Volare Sport*”, a. XIX, n. 206, Febbraio 2000

<sup>[3]</sup>San Bovi F., Cosa compro con meno di 50 milioni?, in “*Volare Sport*”, a. XIX, n. 207, Marzo 2000