

Corso di Laurea Magistrale in Ingegneria aerospaziale - Spazio

Tesi di Laurea Magistrale

# Ottimizzazione del simulatore di volo di un pallone stratosferico

#### Relatori

Prof.ssa Manuela BATTIPEDE Ing. Victor MIHEREA Dott. Gabriele SARTOR

> **Candidato** Alessio La Mesa

Anno Accademico 2022-2023

# Ringraziamenti

Ringraziamenti di vero cuore vanno a chi ha reso possibile questo enorme traguardo. A partire da mia mamma Claudia, mio papà Alessandro e mio fratello Matteo, sempre presenti e colonne portanti di questo successo; senza i loro sforzi non sarei mai arrivato qui.

A mio nonno Ugo, che mi ha reso l'uomo che sono oggi, e a tutti i parenti vicini e lontani.

A Francesca, la persona speciale che mi supporta e mi sta accanto, una "collega" diventata compagna di vita a suon di progetti e materie da dare, di cui ho imparato a fidarmi e con cui spero di condividere più di una Laurea. Che questo sia solo l'inizio del nostro futuro.

Ai "con-legiali", per forza, va un grazie immenso per essere stati parte di un pezzo della mia vita, senza di loro gran parte del percorso non sarebbe stato possibile. Elencarli tutti impossibile, essere loro grato un dovere.

Agli amici che, nonostante la lontananza, sono sempre pronti ad ascoltarmi e a supportarmi.

Ringrazio infine la Prof.ssa Battipede, per avermi guidato e supportato nella fase più importante del mio percorso accademico, nonchè tutti i ragazzi di Stratobotic che non hanno esitato un attimo a farmi sentire parte del loro affascinante progetto.

Grazie.

"Vivere come volare, ci si può riuscire soltanto poggiando su cose leggere." Brunori Sas

#### Sommario

Questo documento si concentrerà sullo uno studio riguardante i palloni stratosferici ed il miglioramento di un simulatore numerico, sviluppato in Python ed implementato con Javascript per l'interfaccia online "end-user", specificamente progettato per i palloni meteorologici. Questi palloni sono fondamentali per la riuscita di esperimenti nello spazio vicino, poiché qualsiasi altro mezzo sarebbe impossibile o molto più costoso da realizzare per eseguire tale tipo di compiti.

L'obiettivo del progetto è il miglioramento di un simulatore esistente (ASTRA [28]) al fine di calcolare correttamente il punto di rottura del pallone durante il suo volo e affinare la previsione del punto di atterraggio della piattaforma. Questo simulatore tiene conto di diverse variabili che svolgono un ruolo importante nell'evoluzione della traiettoria del pallone, come temperature e pressioni, insieme alla variazione del raggio del pallone. In particolare, il software utilizzerà previsioni dei venti e delle temperature dell'ambiente (di conseguenza della quota) in cui si svolge la missione simulata, fornite dalla National Oceanic and Atmospheric Administration (NOAA).

Il cuore principale del lavoro sarà quello di affinare la stima predittiva del software, in particolare mediando i dati delle previsioni meteorologiche del NOAA ed i venti reali acquisiti tramite GPS interpolati per la fase di discesa. Per la fase di salita, il progetto prevede la realizzazione di un tool atto al calcolo preliminare di numerose traiettorie basate su dati variabili e al loro confronto con una stima preliminare fatta con i dati di posizione rilevati nei primi minuti della missione.

I risultati ottenuti dal simulatore saranno confrontati con i dati ottenuti da un paio di missioni precedenti di un pallone ad alta quota al fine di verificarne la validità ed efficienza. Inoltre, per completare le analisi effettuate, verrà tentata un'analisi sul campo per attestare la veridicità del miglioramento.

# Indice

| El       | Elenco delle figure 5 |          |                                 |    |  |  |  |
|----------|-----------------------|----------|---------------------------------|----|--|--|--|
| 1        | Introduzione          |          |                                 |    |  |  |  |
|          | 1.1                   | Motiva   | azione                          | 12 |  |  |  |
|          | 1.2                   | Obiett   | ;ivi                            | 12 |  |  |  |
|          | 1.3                   | State    | of the art del simulatore       | 13 |  |  |  |
| <b>2</b> | Stato dell'arte       |          |                                 |    |  |  |  |
|          | 2.1                   | Contes   | sto storico                     | 15 |  |  |  |
|          | 2.2                   | Indust   | ria attuale                     | 16 |  |  |  |
|          |                       | 2.2.1    | Loon                            | 17 |  |  |  |
|          |                       | 2.2.2    | NASA Balloon program            | 19 |  |  |  |
|          |                       | 2.2.3    | National Weather Service        | 24 |  |  |  |
|          |                       | 2.2.4    | ARHAB Program                   | 24 |  |  |  |
|          | 2.3                   | Missio   | ni future                       | 25 |  |  |  |
|          |                       | 2.3.1    | World View Enterprises          | 25 |  |  |  |
|          |                       | 2.3.2    | Zero 2 Infinity                 | 26 |  |  |  |
| 3        | Quadro Normativo 27   |          |                                 |    |  |  |  |
|          | 3.1                   | Requis   | siti operativi di base          | 28 |  |  |  |
|          |                       | 3.1.1    | Requisiti generali              | 28 |  |  |  |
|          |                       | 3.1.2    | Procedure operative             | 29 |  |  |  |
|          |                       | 3.1.3    | Performance e limiti operativi  | 29 |  |  |  |
|          |                       | 3.1.4    | Palloni liberi senza equipaggio | 30 |  |  |  |
|          |                       | 3.1.5    | Notifica di volo                | 31 |  |  |  |
| 4        | $\mathbf{Asp}$        | etti fis | sici ed aerodinamici            | 33 |  |  |  |
|          | 4.1                   | Genera   | alità del moto di un pallone    | 33 |  |  |  |
|          |                       | 4.1.1    | Forze di base e Portanza        | 33 |  |  |  |

|   |     | 4.1.2                 | Forza resistente                                    | 34 |  |  |  |  |
|---|-----|-----------------------|---|----|--|--|--|--|
|   |     | 4.1.3                 | Legge dei gas                                       | 36 |  |  |  |  |
|   |     | 4.1.4                 | Equazioni del moto                                  | 37 |  |  |  |  |
|   | 4.2 | ASTR                  | A - Equazioni del moto                              | 37 |  |  |  |  |
|   | 4.3 | ASTR                  | A - Stime della resistenza del pallone              | 39 |  |  |  |  |
|   | 4.4 | ASTR                  | A - Incertezze e simulazione con metodo Monte Carlo | 42 |  |  |  |  |
| 5 | Din | Dinamica del vento 49 |   |    |  |  |  |  |
|   | 5.1 | Equaz                 | ioni della dinamica del vento                       | 49 |  |  |  |  |
|   |     | 5.1.1                 | Equazione di continuità                             | 50 |  |  |  |  |
|   |     | 5.1.2                 | Equazione di bilancio della quantità di moto        | 50 |  |  |  |  |
|   |     | 5.1.3                 | Equazione di bilancio energetico                    | 50 |  |  |  |  |
|   | 5.2 | Orogra                | afia del terreno                                    | 51 |  |  |  |  |
|   |     | 5.2.1                 | Venti Foehn   | 52 |  |  |  |  |
|   |     | 5.2.2                 | Ciclogenesi   | 53 |  |  |  |  |
|   | 5.3 | Flusso                |   | 53 |  |  |  |  |
|   | 5.4 | Conclu                | usioni e modifiche da apportare                     | 55 |  |  |  |  |
| 6 | AS  | ASTRA & honevcomb 5'  |   |    |  |  |  |  |
|   | 6.1 | Work                  | environment   | 57 |  |  |  |  |
|   |     | 6.1.1                 | Sistemi operativi - Windows & Linux                 | 57 |  |  |  |  |
|   |     | 6.1.2                 | Python  | 57 |  |  |  |  |
|   |     | 6.1.3                 | JavaScript  | 59 |  |  |  |  |
|   |     | 6.1.4                 | File .json, .csv, .kml                              | 60 |  |  |  |  |
|   | 6.2 | ASTR                  | A - Cosa è e come funziona                          | 64 |  |  |  |  |
|   |     | 6.2.1                 | Simulator.py  | 65 |  |  |  |  |
|   |     | 6.2.2                 | GFS.py  | 66 |  |  |  |  |
|   |     | 6.2.3                 | interpolate.py                                      | 68 |  |  |  |  |
|   |     | 6.2.4                 | weather.py  | 68 |  |  |  |  |
|   |     | 6.2.5                 | global_tools  | 68 |  |  |  |  |
|   | 6.3 | Honey                 | rcomb   | 70 |  |  |  |  |
|   |     | 6.3.1                 | Installazione                                       | 71 |  |  |  |  |
|   |     | 6.3.2                 | App   | 71 |  |  |  |  |
|   | 6.4 | Interfa               | accia online  | 72 |  |  |  |  |
|   |     | 6.4.1                 | Planner   | 72 |  |  |  |  |
|   |     | 6.4.2                 | Pre-flight tool                                     | 74 |  |  |  |  |
|   |     | 6.4.3                 | Tracker   | 75 |  |  |  |  |

| 7            | Metodologia di lavoro |  |     |  |
|--------------|-----------------------|--|-----|--|
|              | 7.1 Problematiche     |  | 77  |  |
|              | 7.2                   | 7.2 Moto di discesa  |     |  |
|              |                       | 7.2.1 Media dei dati GFS e interpolati $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots$ | 81  |  |
|              | 7.3                   | Moto di ascesa   | 110 |  |
|              |                       | 7.3.1 Variazione della Nozzle Lift   | 110 |  |
|              |                       | 7.3.2 Stima errore e scelta migliore simulazione   | 111 |  |
| 8            | Lancio                |  |     |  |
|              | 8.1                   | Sicurezza  | 115 |  |
|              | 8.2                   | Setup di missione  | 118 |  |
|              | 8.3                   | Giorno del Lancio  | 122 |  |
| 9            | Con                   | aclusioni e Sviluppi futuri  | 125 |  |
| $\mathbf{A}$ | Not                   | te sui gas LTA   |     |  |
| в            | Interpolatori SciPy   |  |     |  |
|              | B.1                   | Akima1DInterpolator  | 129 |  |
|              | B.2                   | CubicSpline  | 129 |  |
|              | B.3                   | UnivariateSpline   | 130 |  |
| С            | $\mathbf{Sim}$        | nulazioni Varie  |     |  |
| Bi           | Bibliografia          |  |     |  |

# Elenco delle figure

| 2.1  | Missione Loon 2   |
|------|---|
| 2.2  | Pallone ARCADE 19   |
| 2.3  | Missione SuperTIGER   |
| 2.4  | Balloon della missione BARREL 21  |
| 2.5  | Missione HIWIND   |
| 2.6  | Missione ASCOT  |
| 2.7  | Stratollite   |
| 2.8  | Zero2Infinity   |
| 4.1  | Schema fisico semplificato pallone  |
| 4.2  | Impatto del numero di Reynolds 35   |
| 4.3  | Caption   |
| 4.4  | Tipico spettro di volo di un pallone. La linea tratteggiata demarca la  |
|      | tropopausa $\ldots \ldots 40$ |
| 4.5  | Modello della resistenza ottenuto sperimentalmente da 20993 voli 43   |
| 6.1  | Versione Python   |
| 6.2  | Requirements  |
| 6.3  | Requirements di ASTRA "base"  |
| 6.4  | post_processing_input.json  |
| 6.5  | out.csv   |
| 6.6  | out.kml   |
| 6.7  | Traiettoria visualizzata su Google Earth con file kml   |
| 6.8  | Moduli di ASTRA 64  |
| 6.9  | Equazione ordinaria differenziale   |
| 6.10 | GFS wind  |
| 6.11 | Struttura honeycomb   |
|      |   |
| 6.12 | Interfaccia online Stratologic  |

| 6.13 | Particolare del codice JavaScript che regola il funzionamento del planner        |
|------|--|
|      | e la chiamata della simulazione con i dati immessi                               |
| 6.14 | Interfaccia Planner aggiornata   |
| 6.15 | Preflight tool   |
| 6.16 | Output preflight tool  |
| 6.17 | Tracker Stratologic  |
| 7.1  | Flag gpsWind   |
| 7.2  | Struttura getwind  |
| 7.3  | Interpolazione   |
| 7.4  | Media pesata a step di 100 m in un range di 500 m 81                             |
| 7.5  | Atterraggio previsto ed effettivo missione $11/08/2022$ 82                       |
| 7.6  | Atterraggio previsto ed effettivo missione $19/10/2022$ 83                       |
| 7.7  | Function get_highest_elevation   |
| 7.8  | Calcolo self.iterAlt   |
| 7.9  | Simulazione dei tre casi   |
| 7.10 | Confronto errore traiettorie   |
| 7.11 | Confronto errore traiettorie $19/10/2022$  |
| 7.12 | Codice per media geometrica  |
| 7.13 | Chiamata media geometrica  |
| 7.14 | Confronto errore traiettorie $11/08/2023$ mediate geometricamente 91             |
| 7.15 | Confronto errore traiettorie $19/10/2022$ mediate geometricamente 92             |
| 7.16 | Atterraggio previsto ed effettivo missione $11/08/2023$ con dati interpolati     |
|      | per via geometrica   |
| 7.17 | Atterraggio previsto ed effettivo missione $19/10/2022$ con dati interpolati     |
|      | per via geometrica   |
| 7.18 | Codice per media armonica  |
| 7.19 | Chiamata media armonica  |
| 7.20 | Confronto errore traiettorie $11/08/2023$ mediate armonicamente $\dots \dots 96$ |
| 7.21 | Confronto errore traiettorie $19/10/2022$ mediate armonicamente $\dots 97$       |
| 7.22 | Atter<br>raggio previsto ed effettivo missione $11/08/2023$ con dati interpolati |
|      | per via armonica   |
| 7.23 | Atter<br>raggio previsto ed effettivo missione $19/10/2022$ con dati interpolati |
|      | per via armonica   |
| 7.24 | Codice per media quadratica 100  |
| 7.25 | Chiamata media quadratica  |
| 7.26 | Confronto errore traiettorie $11/08/2023$ mediate quadraticamente 101            |

| 7.27      | Confronto errore traiettorie $19/10/2022$ mediate quadraticamente 1                | 02 |
|-----------|--|----|
| 7.28      | Atter<br>raggio previsto ed effettivo missione $11/08/2023$ con dati interpolati   |    |
|           | per via quadratica   | 03 |
| 7.29      | Atter<br>raggio previsto ed effettivo missione $19/10/2022$ con dati interpolati   |    |
|           | per via quadratica   | 04 |
| 7.30      | Codice per media cubica  | 05 |
| 7.31      | Chiamata media cubica  | 06 |
| 7.32      | Confronto errore traiettorie $11/08/2023$ mediate cubicamente 1                    | 06 |
| 7.33      | Confronto errore traiettorie $19/10/2022$ mediate cubicamente 1                    | 07 |
| 7.34      | Atterraggio previsto ed effettivo missione $11/08/2023$ con dati interpolati       |    |
|           | per via cubica   | 08 |
| 7.35      | Atterraggio previsto ed effettivo missione $19/10/2022$ con dati interpolati       |    |
|           | per via cubica   | 09 |
| 7.36      | Function definita per salvare in locale le simulazioni con le nozzle lift variate1 | 11 |
| 7.37      | Main di Best_simulation.py   | 12 |
| 7.38      | Commento in output del modulo "Best_simulation.py"                                 | 13 |
|           |  |    |
| 8.1       | Simulazioni $11/10/2023$   | 16 |
| 8.2       | Simulazioni $12/10/2023$   | 17 |
| 8.3       | Simulazioni $13/10/2023$   | 17 |
| 8.4       | Safe zones $12/10/2023$  | 18 |
| 8.5       | Organizzazione di una missione tipo 1  | 19 |
| 8.6       | Organizzazione di una missione tipo 1  | 20 |
| 8.7       | Particolare delle antenne della ground-station                                     | 21 |
| 8.8       | Particolare delle antenne della ground-station                                     | 22 |
| $C_{1}$   | 11/10/2023 ore 10 1  | 33 |
| C.2       | 11/10/2023 ore 12  | 34 |
| C.3       | 11/10/2023 ore 14  | 34 |
| C 4       | 11/10/2023 ore 16  | 35 |
| C 5       | 12/10/2023 ore 10 1  | 35 |
| C 6       | 12/10/2023 ore 12 1  | 36 |
| с.<br>С.7 | 12/10/2023 ore 14  | 36 |
| C 8       | 12/10/2023 ore 16  | 37 |
| C.9       | 13/10/2023 ore 10  | 37 |
| C 10      | 13/10/2023 ore 12  | 38 |
| C.11      | 13/10/2023 ore 14  | 38 |
| C.12      | $13/10/2023 \text{ ore } 16 \dots 11$  | 39 |
|           |  |    |

### Acronimi

- AMC Acceptable Means Compliance
- **ANSP** Air Navigation Service Provider
- AOC Air Operator Certificate
- ATC Air Traffic Control
- **ATS** Air Traffic Service
- CAA Civil Aviation Agency
- **CS** Certification Specs
- **CSV** Comma Separated Values
- ${\bf EASA}$  European Aviation and Space Administration
- ECMWF European Center for Mid-range Weather Forecasting
- **ENAC** Ente Nazionale Aviazione Civile
- FAA Federal Aviation Administration
- GeoTIFF Geospatial Tagged Image File Format
- GFS Global Forecasting System
- **GPS** Global Positioning System
- GUI Graphical User Interface
- **IR** Implementation Rule
- HAPS High Altitude Pseudo Satellite
- HD High Definition
- **ISA** International Standard Atmosphere
- **JSON** JavaScript Object Notation
- **KML** Keyhole Markup Language
- **LSODE** Livermore Solver for Ordinary Differential Equations

 ${\bf NASA}\,$  National Aviation and Space Administration

 ${\bf NOAA}\,$  National Oceanic and Atmospheric Administration

NL Nozzle Lift

- ${\bf NWP}\,$  National Weather Program
- $\mathbf{ODE} \ \ \, \mathrm{Ordinary} \ \, \mathrm{Differential} \ \mathrm{Equations}$
- **SD** Standard Definition
- **UE** Unione Europea
- UKHAS United Kingdom High Altitude Society
- ${\bf UTC}~$  Coordinated Universal Time
- **VFR** Visual Flight Rules

### Capitolo 1

# Introduzione

- I palloni ad alta quota sono palloni che, una volta rilasciati, volano fino alla stratosfera, raggiungendo altitudini generalmente comprese tra 18 e 37 km sopra il livello del mare. Queste piattaforme ad alta quota possono essere pilotate o meno, e di solito il pallone che le trasporta è riempito con un gas più leggero dell'aria come elio (He), idrogeno (H2) o metano (CH4). Questi tipi di palloni, chiamati anche palloni scientifici quando utilizzati per scopi di ricerca, sono ideali per condurre non solo esperimenti atmosferici ma anche esperimenti nello spazio vicino ed una vasta gamma di altre missioni, come ad esempio indagini climatiche e atmosferiche attraverso l'osservazione terrestre condotta da Stratobotic.
- Oggi, i droni stanno sostituendo questo tipo di piattaforme in diverse applicazioni a quote inferiori, in quanto possono offrire un controllo più preciso sulla stabilità della missione e sono meno dipendenti dalle condizioni meteorologiche. Nonostante ciò, i palloni scientifici ad alta quota rappresentano ancora una risorsa preziosa per gli studi da condurre nell'ambiente dello spazio vicino.
- Pertanto, è importante esaminare la fisica e gli elementi coinvolti in questo tipo di voli al fine di cercare di ottimizzarli il più possibile migliorandone le prestazioni complessive. Ecco perchè il miglioramento del simulatore, e da un punto di vista predittivo e da un punto di vista concettuale, lo potrebbe rendere sia più affidabile che più appetibile per la realizzazione di missioni corte e ad ampio raggio.
- Infatti, il goal di Stratobotic è quello, nel breve termine, di migliorare il modello predittivo fino ad ottenere stime sul punto di atterraggio del payload molto precise e, nel lungo termine, quello di realizzare un velivolo stratosferico alimentato tramite pannelli solari per l'osservazione più duratura: short mission e long mission. Diversi risultati ma stesso principio di base: il volo stratosferico.

- Nei primi capitoli, perciò, verrà illustrato il quadro fisico della missione in particolare, con uno sguardo all'influenza della conformazione terrestre sul vento locale che modifica la traiettoria del pallone. Verranno inoltre illustrati gli ambienti di lavoro del simulatore, con un accenno alle librerie Python usate ed estensioni di file particolari.
- Nei capitoli 6 e 7 verranno infine esposti i passaggi effettuati a livello software ed alcuni risultati ottenuti dal modello "migliorato".

#### 1.1 Motivazione

Il motivo principale della realizzazione di questo progetto è la volontà di collaborare con i ragazzi di Stratobotic S.R.L. per avere una visione di insieme su quello che è il mondo del lavoro, quindi interfacciarsi con qualcosa di tangibile, ed avere esperienza di lavoro sul campo, mettendo in pratica lo sviluppo del codice del simulatore in una missione vera e propria. Ulteriore motivo di interesse per questa collaborazione è la volontà di formare la mia figura professionale in ambiti e skill che il corso di Laurea scelto non ha potuto fornirmi, ovvero la conoscenza di più linguaggi di programmazione come Python e Javascript, integrati però in un ambiente comunque familiare come quello del volo. Non solo studio aerodinamico, ma soprattutto studio esterno di programmazione per implementare soluzioni consone all'aerodinamica del pallone stratosferico.

Inoltre, obiettivo fondamentale di questa Tesi è quello di formare e arricchire una buona base statistica per quello che sarà lo sviluppo futuro del simulatore di volo e del suo modello dei venti.

#### 1.2 Obiettivi

Oltre alla suddetta base statistica sui dati di vento rilevati da lasciare per futuri sviluppi, obiettivo principale del progetto è quello di migliorare il comportamento predittivo del simulatore di volo (da ora in poi chiamato Honeycomb). **Honeycomb** si basa su un simulatore di volo esistente, chiamato ASTRA, sviluppato dall'università di Southampton nel 2013, integrandolo a un'interfaccia grafica online che rende il monitoraggio della missione più semplice.

Primo obiettivo del progetto è quello di migliorare la fase predittiva di discesa del pallone. Questo perchè i dati di previsione NOAA non sono esattamente precisi oltre una certa altitudine.

Infatti l'influenza della stratosfera sulle condizioni meteorologiche dello strato più alto

della Troposfera, rendono difficile la predizione della traiettoria proprio per questa imprecisione dei dati previsti.

Per questo motivo, come verrà spiegato più avanti, è stata adottata una soluzione che sfrutta le posizioni GPS mandate a terra da un Arduino montato a bordo del payload, denominato per l'appunto TRACKuino, che tiene costantemente aggiornata la posizione del pallone. Chiaramente, ciò non basta per prevedere il punto di atterraggio della sonda, visto il delay nel trasmettere i dati di posizione.

Secondo obiettivo è quello di migliorare la stima predittiva in base ai dati di input del simulatore. Infatti, non sempre i dati previsti sono effettivamente quelli reali, come ad esempio la nozzle lift. Questa viene calcolata da un tool esterno al simulatore prima della missione per riempire il pallone della quantità d'aria calcolata. Il problema nasce perchè la nozzle lift durante la missione si "sposta" dal valore nominale variando di qualche punto percentuale.

Per questo motivo è stato sviluppato un ulteriore codice in grado di simulare varie traiettorie e metterle a confronto con le posizioni reali del pallone rilevate durante i primi minuti della missione.

#### **1.3** State of the art del simulatore

Il simulatore utilizzato come base per lo sviluppo di questo progetto è stato implementato attraverso un codice Python modificato tramite piattaforma VSCode per Windows. Il codice di base ASTRA utilizza dati meteorologici scaricati prima dell'esecuzione del simulatore, fornendo principalmente informazioni sulle temperature e le velocità del vento in momenti specifici. Queste previsioni meteorologiche sono fornite dal NOAA (National Oceanic and Atmospheric Administration).

Questo simulatore ha permesso l'introduzione dei parametri iniziali della missione relativi al pallone e alla piattaforma per simulare il volo fino alla rottura del pallone, il cosiddetto Bursting point.

Tuttavia, il modello presenta alcuni errori di calcolo e semplificazioni nella fisica che governa l'ascesa del pallone, oltre a cattive regolazioni dei parametri, che saranno risolti attraverso lo sviluppo di questo progetto. Questi errori compromettono i risultati finali della simulazione, fornendo posizioni di atterraggio molto diverse dalla realtà.

### Capitolo 2

## Stato dell'arte

Questo capitolo verrà utilizzato come breve spiegazione dei principali progetti che coinvolgono palloni ad alta quota nell'industria attuale. Verrà inoltre fornito del contesto storico riguardo l'evoluzione di questo tipo di tecnologia e al suo utilizzo nel corso del tempo, oltre a nuove possibilità e idee attualmente in fase di sviluppo.

Queste informazioni sono raggruppate in questo capitolo al fine di creare un contesto più completo attorno allo sviluppo di questa tesi in relazione al mondo reale.

#### 2.1 Contesto storico

La storia dell'aerostatica risale a molto tempo fa. Il raggiungimento dei primi voli umani fu possibile solo grazie agli sviluppi legati alla portata dei palloni basati sulla spinta data dall'uso di idrogeno o aria calda per farli salire nell'atmosfera.

Jacques Charles e i fratelli Robert lanciarono il primo pallone ad idrogeno mai esistito il 27 agosto 1783 a Parigi. Questo pallone era piuttosto piccolo (circa 4 metri di diametro, realizzato fondamentalmente con una sfera di seta gommosa) e poteva sollevare un payload di circa 9 kg, con una durata di volo di quasi 5 giorni. Il suo volo durò 45 minuti e il pallone atterrò a 21 chilometri dal punto di lancio. Questo risultato rappresentò un primo passo verso i voli cosiddetti "manned".

Parallelamente, i fratelli Montgolfier fecero volare il loro pallone ad aria calda non pilotato (senza necessità di idrogeno) il 19 settembre 1783. Questo volo fu il primo a trasportare esseri viventi: una pecora, una papera e un gallo. Il problema fu che questo volo durò solo circa 8 minuti, anche se raggiunse un'altitudine di circa 460 metri. Il 21 novembre 1783 fu lanciato il primo volo umano, con un pallone progettato dai fratelli Montgolfier con l'aiuto dello scienziato Jean-François Pilatre de Rozier. Questo pallone poteva trasportare esseri umani utilizzando l'aria calda come mezzo per generare la portanza. Jacques Charles e i fratelli Robert continuarono le loro ricerche e lanciarono la prima missione di pallone ad idrogeno con equipaggio il 1° dicembre del 1783. Questo volo durò circa 2 ore, coprendo 36 km con un pallone di 380 metri cubi che si elevò fino a 550 metri. Ma non solo, questo fu anche il primo pallone a fornire misurazioni meteorologiche dell'atmosfera sopra la superficie terrestre, poiché i piloti portarono durante la missione un termometro e un barometro.

Solo nel 1785 vennero aggiunti sistemi di controllo rudimentali ai palloni, quando Jean-Pierre Blanchard attraversò la Manica. Questi mezzi di controllo furono migliorati, consentendo la comparsa dei primi palloni dirigibili nel 1852. Il primo fu volato da Henri Giffard ed era alimentato da un motore a vapore. Alla fine del XIX secolo, l'introduzione di motori a combustione interna per il controllo della direzione rese i dirigibili più pratici, poiché aumentava la loro velocità di viaggio. Dalla fine degli anni '90 fino agli anni '60, il tipo di palloni più comuni erano quelli a gas, il che significa che erano riempiti con gas più leggeri come l'idrogeno, invece di sollevarsi solo riscaldando l'aria. Tuttavia, il 6 maggio 1937, il dirigibile passeggeri tedesco LZ 129 Hindenburg fu distrutto nel tentativo di attraccare alla Naval Air Station Lakehurst (New Jersey). Durante questo incidente persero la vita 35 delle 97 persone a bordo. Questa tragedia ebbe così tante ripercussioni che l'industria del trasporto in pallone entrò in un rapido declino verso l'estinzione.

Questo declino fu accelerato anche dall'apparizione degli aeroplani come nuovo mezzo per volare nel cielo. La loro scoperta e rapido sviluppo li portarono ad assorbire il mercato dei trasporti e a relegare i palloni a scopi più ricreativi e di ricerca scientifica.

Si può anche notare l'uso di palloni ad alta quota pilotati per la ricerca di record di altitudine di volo. Uno dei record più notevoli è stato stabilito per il più alto lancio con paracadute da Felix Baumgartner nel 2012 a un'altitudine di 38.969 metri. Nel 2014 questo record è stato battuto da Alan Eustace con un'altitudine di 41.419 metri, diventando il record attuale fino ad oggi.

#### 2.2 Industria attuale

In questa sezione verranno evidenziati e brevemente descritti alcuni dei progetti e delle missioni più notevoli riguardanti i palloni ad alta quota attualmente in corso o sviluppati negli ultimi anni. Inoltre, verrà menzionato se si trovano alla fine della loro procedura di sviluppo.

#### 2.2.1 Loon

Loon LLC [3] è una sussidiaria di Alphabet Inc. (società madre di Google) che si occupa di facilitare l'accesso a Internet nelle aree remote o rurali. Questa società utilizza i palloni ad alta quota per creare una rete wireless che fornisce connessione Internet alle aree desiderate. Questo progetto è stato inizialmente sviluppato da Development X (precedentemente noto come Google X), ma nel luglio 2018 è diventato una società separata.



Figura 2.1. Missione Loon 2

Il sistema utilizzato da questa società si basa sulla distribuzione di una rete di palloni, come quelloi mostrato in Fig.[2.1], ad alta quota nella stratosfera, ad altitudini comprese tra 18 km e 25 km, ai limiti dello spazio, un po' sulla falsa riga di Starlink. I palloni Loon sono progettati e realizzati per resistere alle estreme condizioni stratosferiche. Questo strato di altitudine è considerato molto vantaggioso per il progetto a causa delle basse velocità del vento presenti e della limitata turbolenza. La loro manovrabilità è controllata regolando la loro altitudine al fine di cercare lo strato della stratosfera con le velocità e la direzione del vento desiderate. Queste informazioni meteorologiche sul vento sono reperite, come sempre, dal NOAA (National Oceanic and Atmospheric Administration). La tecnologia si basa su palloni in polietilene delle dimensioni di un campo da tennis, in grado di durare per oltre 100 giorni prima di atterrare in modo sicuro in una discesa controllata. Inoltre, tutta l'attrezzatura per il volo è progettata per essere energeticamente efficiente ed è alimentata da pannelli solari. Questa attrezzatura è composta principalmente da antenne che trasmettono la connettività dalle stazioni a terra agli utenti nell'area interessata, pannelli solari per alimentare tutti i sistemi, una batteria ricaricabile per consentire le operazioni durante la notte, una capsula di volo che sostiene l'elettronica che controlla il sistema e un paracadute per garantire un atterraggio sicuro quando la vita del pallone è terminata.

La prima connessione Internet alimentata da palloni di questo sistema è stata fornita nel 2013 in Nuova Zelanda. Nel 2017, il progetto Loon ha fornito connettività di emergenza di base a migliaia di persone nelle aree colpite dalle inondazioni in Perù, così come a Porto Rico dopo l'uragano Maria.

#### 2.2.2 NASA Balloon program

Il programma NASA è stato istituito oltre 30 anni fa ed è stato responsabile della fornitura e dello sviluppo di palloni scientifici ad alta quota e piattaforme per numerose ricerche al fine di comprendere meglio la Terra e l'universo.Alcune delle missioni sviluppate dalla National Aeronautics and Space Administration (NASA) sono qui di seguito brevemente descritte.

#### Missione ARCADE

ARCADE[1] sta per Absolute Radiometer for Cosmology, Astrophysics and Diffuse Emission ed è un programma con l'obiettivo di studiare l'universo primordiale attraverso l'uso di palloni ad alta quota, come quello di Fig.[2.2].



Figura 2.2. Pallone ARCADE

Consiste in un insieme di 7 radiometri di precisione raffreddati a 2,7 K e trasportati a altitudini superiori a 35 km grazie all'uso di un pallone stratosferico scientifico. In questo modo è possibile misurare lo spettro in frequenza dello Sfondo Cosmico a Microonde

(CMB) al fine di cercare i segnali lasciati dalle prime stelle apparse dopo il Big Bang, cercare reliquie della fisica delle particelle e ottenere una migliore comprensione della struttura e dell'energetica su larga scala della Galassia. ARCADE ha compiuto con successo un volo nel 2006 quando la piattaforma ad alta quota è atterrata in modo sicuro (con l'aiuto di un paracadute) dopo circa 4 ore di osservazione. I risultati ottenuti hanno mostrato una forte sorgente radio residua, che era circa sei volte più alta di quanto previsto teoricamente. Questo fenomeno in astrofisica è rimasto irrisolto fino ad oggi.

#### Missione Super-TIGER Balloon

Super-TIGER[4] (Super Trans-Ions Galactic Element Recorder Balloon) è stato lanciato nel 2012 in Antartide. Si tratta di uno strumento volante (Fig.[2.3]) con l'obiettivo di misurare gli elementi pesanti rari tra il flusso di raggi cosmici ad alta energia che giungono sulla Terra da altre parti della Via Lattea, al fine di comprendere meglio le loro origini (dove vengono prodotti questi nuclei energetici) e come diventano così ad alta energia.



Figura 2.3. Missione SuperTIGER

Questa missione ha infranto il record della durata più lunga di un volo di un pallone scientifico a portata pesante, con una durata di 55 giorni. Nel dicembre 2017, era previsto un nuovo lancio della piattaforma SuperTIGER, ma dopo diversi tentativi effettuati, la campagna è stata cancellata a causa delle avverse condizioni meteorologiche.

#### Missione BARREL

La missione Balloon Array for Radiation-belt Relativistic Electron Losses (BARREL[5) consiste in una missione per aumentare le misurazioni delle Sonde delle fasce di Van Allen, anche conosciute come Radiation Belt Storm Probes, attraverso l'uso di palloni ad alta quota, come in Fig.[2.4]. Questa missione è stata condotta dalla NASA ed è stata composta da due campagne, una nel 2013 e una nel 2014 rispettivamente. Ciascuna di queste missioni consisteva in 20 carichi utili di palloni lanciati fino a circa 30 km nella stratosfera durante un periodo di un mese al fine di stimare la precipitazione degli elettroni relativistici e la perdita totale dalle fasce di radiazioni.



Figura 2.4. Balloon della missione BARREL

#### Missione HIWIND

Il payload di osservazione High altitude InterferometerWind[6] (HIWIND) è stato lanciato nel 2018 dal Colorado al fine di raccogliere informazioni sui venti termosferici, con un pallone simile a quello della missione precedente, come visibile in Fig.[2.5]. Una migliore comprensione di questo fenomeno faciliterà la previsione dei cambiamenti ionosferici, con un grande impatto sui sistemi di comunicazione e navigazione.



Figura 2.5. Missione HIWIND

#### **Missione ASCOT**

Sempre nel 2018, la NASA ha lanciato l'Advanced Scintillator Compton Telescope [7] (ASCOT) con l'aiuto di un pallone scientifico (Fig.[2.6]). L'obiettivo era quello di ottenere immagini della Nebulosa del Granchio in un ambiente vicino allo spazio con nuove tecnologie legate ai telescopi gamma a energia media.



Figura 2.6. Missione ASCOT

#### 2.2.3 National Weather Service

I palloni meteorologici ad alta quota sono utili per registrare dati utilizzati nell'analisi delle attuali condizioni atmosferiche e sono anche molto utili per la previsione del tempo e la ricerca meteorologica. Regolarmente, due volte al giorno, vengono lanciati palloni meteorologici da oltre 800 località in tutto il mondo. Il National Weather Service (NWS) opera 92 radiosonde (strumenti telemetrici alimentati a batteria portati nell'atmosfera) in diverse località del Nord America e 10 nei Caraibi.

La radiosonda equipaggiata sul pallone misura profili di temperatura, pressione e umidità relativa. Tramite il monitoraggio della posizione della piattaforma, è anche possibile recuperare le velocità e le direzioni del vento. Questi dati vengono trasmessi a terra attraverso un trasmettitore radio. Il pallone si solleva a circa 35 km durante un volo che di solito dura 2 ore.

I palloni meteorologici hanno anche utilizzi più specifici, come quelli legati all'aviazione, fotografia o videografia, controllo dell'inquinamento e ricerca. Un esempio è il cosiddetto pallone pilota o "ceiling balloon", utilizzato dai meteorologi per determinare l'altezza delle nuvole sopra il livello del suolo.

#### 2.2.4 ARHAB Program

Questo programma (Amateur Radio High-Altitude Balloon: ARHAB) si concentra sullo studio dei palloni stratosferici ad alta quota per esplorare lo spazio vicino con l'applicazione di radio amatoriali analogiche e digitali in palloni meteorologici. Le piattaforme sono abbastanza semplici e includono il pallone, un paracadute per il recupero sicuro e il carico utile in cui sono collocati i sensori, le telecamere e i sistemi di tracciamento e comunicazione. A causa dei costi relativamente bassi dei sistemi GPS e di comunicazione richiesti, la comparsa della High Altitude Society britannica (UKHAS) ha contribuito alla creazione di guide che aiutano lo sviluppo, la costruzione e la preparazione di questo tipo di piattaforme. Il sito UKHAS, come accennato in precedenza, fornisce strumenti molto utili come sistemi di previsione e tracciamento online, nonché guide e suggerimenti che aiutano chiunque a entrare nel mondo dei palloni stratosferici ad alta quota. Tra le altre organizzazioni che hanno contribuito alla diffusione dei palloni meteorologici ad alta quota, va menzionata la cosiddetta Overlook Horizon Inc., un'organizzazione educativa pubblica americana nata nella primavera del 2016. Sul loro sito forniscono informazioni molto utili per condurre semplici missioni di volo con questi palloni, con un focus maggiore sulla parte sperimentale della missione (inviluppi dei palloni, paracadute, telecamere, sistemi di tracciamento...).

#### 2.3 Missioni future

Attualmente, alcune aziende stanno cercando di portare la tecnologia dei palloni ad alta quota un passo avanti. Hanno diversi progetti innovativi e idee da realizzare nei prossimi anni. Alcune di queste idee sono principalmente legate all'uso dei palloni per le telecomunicazioni e il turismo spaziale. Di seguito verranno menzionate e brevemente descritte due di queste aziende.

#### 2.3.1 World View Enterprises

Questa azienda, fondata nel 2012, è incentrata sulla sperimentazione nello spazio vicino con l'obiettivo di aumentare l'accessibilità e l'utilizzo della stratosfera per scopi commerciali o scientifici. Alla fine, mirano a lanciare carichi utili e persino persone a quote elevate con l'aiuto di palloni. Si concentrano sull'introduzione del veicolo di volo Stratollite [8] senza equipaggio e sull'essere pionieri in nuove prospettive con lo sviluppo del sistema di volo spaziale umano Voyager.

Il progetto Stratollite si basa su un veicolo di volo stratosferico senza equipaggio (Fig.[2.7]), telecomandato e navigabile, progettato per rimanere su aree specifiche per lunghi periodi di tempo. Utilizza il controllo dell'altitudine per sfruttare diversi strati di vento stratosferico per la navigazione e il volo stazionario. Lavora a quote di circa 30 km e sarà utilizzato principalmente per le comunicazioni, la meteorologia, il rilevamento remoto e la ricerca.



Figura 2.7. Stratollite

L'esperienza di volo spaziale umano chiamata Voyager è attualmente in fase di sviluppo con l'obiettivo finale di portare persone a circa 30 km di altitudine in un veicolo spaziale pressurizzato trasportato da un pallone ad alta quota. Questa esperienza promette di essere completamente sicura per i passeggeri, offrendo loro l'opportunità di osservare la curvatura della Terra nello spazio. Ciò rappresenterà un primo passo verso l'aspetto del turismo spaziale

#### 2.3.2 Zero 2 Infinity

Similmente alla precedente, questa azienda privata spagnola fondata nel 2009 è focalizzata sul miglioramento e sull'accesso agli ambienti spaziali vicini attraverso l'uso di palloni ad alta quota. Attualmente, ha tre principali linee di attività. Una consiste nello sviluppo di un lanciatore trasportato da pallone [9] (Bloostar in Fig.[2.8]) per trasportare carichi utili come nano-satelliti per metterli in orbita. La più innovativa consiste nella creazione di un veicolo basato su pallone (Bloon) con l'obiettivo di lanciare veicoli con equipaggio nello spazio vicino a scopo di ricerca o addirittura turistico. Questa azienda fornisce anche a diverse organizzazioni un servizio chiamato Elevate per volare e testare piattaforme in ambienti quasi stratosferici cercando di avere il minor impatto possibile sull'ambiente.



Figura 2.8. Zero2Infinity

### Capitolo 3

# Quadro Normativo

Questa sezione serve a dare una visione generale di quello che è il quadro normativo riguardante i palloni stratosferici, evidenziandone limiti e capacità.

L'Agenzia Europea per la Sicurezza dell'Aviazione (EASA) ha stabilito un insieme di regole semplici per l'adeguato utilizzo dei palloni e le ha raccolte in un unico documento. Questo documento è chiamato "Easy Access Rules for Balloons" [29] ed include le normative per le operazioni aeree con palloni attualmente in vigore. Questo documento è stato pubblicato nel marzo 2018 dalla Commissione Europea ed è entrato in vigore nell'aprile 2019. Queste regolamentazioni comprendono tutte le operazioni con palloni ad alta quota, comprese non solo le missioni ufficiali dell'EASA, ma anche il volo con palloni privati e commerciali. Attualmente, l'EASA sta lavorando alla semplificazione delle procedure di licenza e delle regole per i palloni. Questa legislazione aggiornata è supportata anche dall'Agenzia dell'Aviazione Civile (CAA).

Con questa nuova legislazione, gli operatori, anziché richiedere e ottenere un Certificato di Operatore Aereo (AOC), effettueranno semplicemente una dichiarazione ufficiale presso la CAA o le autorità competenti riguardo alla loro intenzione di far volare il pallone (notifica di volo). Ciò include le missioni per palloni ad alta quota senza equipaggio, che sono il principale focus di questa tesi. Inoltre, la semplificazione dell'autorizzazione pre-volo ha contribuito alla pianificazione delle missioni del prototipo della piattaforma sviluppato attraverso questo progetto.

Il Regolamento sui Palloni è suddiviso in diverse sezioni. In primo luogo, copre le normative della regola di attuazione (IR), per poi passare ai punti dell'allegato IR, ai mezzi di conformità accettabili (AMC) correlati a questo argomento e al materiale di orientamento (GM). Comprende anche le specifiche di certificazione (CS) relative all'idoneità iniziale all'aria.

#### 3.1 Requisiti operativi di base

Le disposizioni di questa sotto-parte del handbook devono essere rispettate da qualsiasi operatore di palloni, a meno che non siano ancorati a un punto fisso per l'intera missione. Di seguito saranno evidenziate alcune delle principali requisiti relativi alle operazioni di base con palloni.

#### 3.1.1 Requisiti generali

- Le operazioni di volo introduttive devono essere condotte secondo le regole di volo visivo (VFR) di giorno e supervisionate dall'organizzazione responsabile dell'operazione.
- L'operatore deve implementare misure di sicurezza, direttive di aeronavigabilità e informazioni obbligatorie per qualsiasi operazione effettuata al fine di garantire una reazione immediata a problemi di sicurezza.
- Il pilota designato dall'operatore come comandante di volo è responsabile della sicurezza del pallone, delle persone e delle proprietà trasportate durante il volo, evitando di volare oltre i limiti consentiti dalle condizioni meteorologiche del sito. Inoltre, il pilota è responsabile di assicurare, prima del volo, che tutti i limiti operativi siano soddisfatti e che siano effettuate le ispezioni necessarie. Ha l'autorità per impartire comandi e intraprendere azioni appropriate per garantire la sicurezza del pallone, delle persone e delle proprietà trasportate ed è tenuto a conoscere le normative, le leggi e le procedure relative ai palloni.
- Il trasporto di merci pericolose a bordo è regolamentato dall'Allegato 18 della Convenzione di Chicago. Il pilota al comando è responsabile di adottare tutte le misure necessarie per evitare il trasporto di merci pericolose a bordo.
- I dettagli del pallone e delle missioni devono essere registrati per ciascun volo in forma di registro di bordo o equivalente. Tale registro dovrebbe includere la nazionalità e la registrazione del pallone, una descrizione del viaggio (orari e luoghi di atterraggio e decollo), il tipo di operazione e eventuali incidenti o osservazioni notate.

#### 3.1.2 Procedure operative

- La missione verrà effettuata solo in siti operativi adeguati al tipo di pallone e operazione in questione.
- È richiesta una preparazione del volo al fine di ottenere informazioni sulle condizioni meteorologiche e aeronautiche durante la missione di volo. Prima del lancio, è necessario assicurarsi che le condizioni meteorologiche consentano una partenza e un arrivo sicuri.
- Sono previste alcune limitazioni operative di notte. I palloni ad aria calda, i palloni a gas (senza equipaggio) e i palloni misti (che combinano aria calda e gas più leggero dell'aria per generare portanza) non devono atterrare di notte a meno che non si verifichi una situazione di emergenza. I palloni ad aria calda a propulsione possono essere operati di notte in conformità alle loro limitazioni VFR approvate.

#### 3.1.3 Performance e limiti operativi

- Il pallone deve essere ancorato in modo da non superare alcuna delle sue limitazioni esposte nel suo AFM (Manuale di Volo dell'Aeromobile, documento che contiene le limitazioni operative applicabili e approvate e le informazioni sul pallone).
- Il pallone può essere utilizzato solo se le sue prestazioni sono adeguate per rispettare le restrizioni stabilite negli allegati al Regolamento di Attuazione (UE)
  n. 923/2012 (ulteriormente spiegati nella successiva sottosezione) e qualsiasi altro requisito applicabile alla missione.

#### 3.1.4 Palloni liberi senza equipaggio

Dall'Appendice 2 del Regolamento di Attuazione (UE, Unione Europea) n. 923/2012, saranno evidenziati di seguito i principali requisiti relativi ai palloni liberi non pilotati.

- Questi palloni possono essere classificati come leggeri (massa combinata inferiore a 4 kg), medi (massa combinata superiore a 4 kg e inferiore a 6 kg) o pesanti (massa combinata superiore a 6 kg o se includono un carico di oltre 3 kg, un pacchetto con densità di area superiore a 13 g/cm<sup>2</sup> o un dispositivo di sospensione che richiede una forza d'impatto superiore a 230 N per separare il suo carico).
- Questi palloni non devono essere utilizzati se il loro impatto, compresi i carichi, potrebbe creare situazioni di pericolo per persone o proprietà.
- Devono essere operati in conformità alle condizioni specificate dallo Stato in cui sono registrati e dagli Stati previsti per essere attraversati durante la missione. Tutti gli altri requisiti principali rimanenti sono applicabili ai palloni liberi pesanti non pilotati.
- Non devono operare sopra il mare senza coordinamento con gli ANSP.
- Dovrebbero essere equipaggiati con almeno due dispositivi di interruzione automatica o radiocomandata del volo del carico indipendenti. Il volo dovrebbe essere interrotto immediatamente quando le condizioni meteorologiche sono peggiori di quanto prescritto o in caso di malfunzionamento.
- Non devono essere lanciati in modo da volare a quote inferiori a 300 m sopra aree congestionate di insediamenti, città o centri abitati. Sono inoltre previste restrizioni di altitudine per il volo dei palloni pesanti durante la notte, poiché non devono essere operati al di sotto di 18000 m di altitudine di pressione, a meno che il pallone, gli attacchi e il carico non siano illuminati. La stessa restrizione di altitudine è applicabile quando la visibilità orizzontale è inferiore a 8 km o ci sono nuvole o fenomeni oscuranti.

D'altra parte, si potrebbe notare che le normative e le leggi relative ai palloni liberi non pilotati negli Stati Uniti (US) sono registrate dalla Federal Aviation Administration (FAA). Sono essenzialmente simili a quelle esposte sopra dall'EASA, ma includono anche una restrizione all'operazione di questi palloni al di sotto di 2000 ft sopra la superficie all'interno dei confini laterali delle principali aree di spazio aereo designate dell'aeroporto, a meno che non siano autorizzati dal corrispondente ATC (Air Traffic Control).

#### 3.1.5 Notifica di volo

Questa notifica prima del volo relativa alla missione, che deve essere presentata ai servizi del traffico aereo (ATS) e che è richiesta sia dalla FAA che dall'EASA, deve contenere informazioni sull'identificazione o il nome in codice del pallone, la sua descrizione e classificazione, nome e numero di telefono dell'operatore, luogo e ora di lancio, quote di crociera stimate e tempo impiegato per raggiungerle, nonché la direzione di ascesa stimata. Nel documento di notifica dovrebbero essere incluse anche il punto di atterraggio e la durata del volo previsti prima del lancio. Se il lancio della missione viene annullato, l'operatore deve notificarlo quanto prima ai servizi del traffico aereo appropriati. Inoltre, l'operatore di palloni liberi non pilotati pesanti ha l'obbligo di monitorare il progresso del percorso di volo seguito durante la missione ogni 2 ore se vola al di sotto di 18000 m o ogni 24 ore in caso contrario. Se una posizione non può essere registrata secondo quanto previsto dalla normativa precedente, deve essere notificata all'unità adeguata dei servizi del traffico aereo. Gli operatori di palloni pesanti devono anche informare prima dell'inizio della discesa sulla posizione attuale e sul tempo stimato e il punto di impatto. Infine, devono comunicare anche la fine della missione.

### Capitolo 4

# Aspetti fisici ed aerodinamici

In questo capitolo vengono esposte ed approfondite le questioni legate all'aspetto fisicomeccanico del simulatore, relative quindi alle equazioni impiegate da ASTRA per simulare i moti di ascesa e discesa del pallone.

#### 4.1 Generalità del moto di un pallone

In questa sezione si mettono in mostra le equazioni principali del moto libero di un pallone gonfiato con gas più leggero dell'aria. In particolare, si analizzano in prima battuta delle equazioni generalizzate e semplificate per poi andare a vedere nel dettaglio, nella sezione successiva, quali sono le leggi che regolano il simulatore "ASTRA".

#### 4.1.1 Forze di base e Portanza

Qui di seguito viene schematizzata (Fig.[4.1]) la situazione generica di volo libero di un pallone [10] assumendo una situazione stazionaria [11], priva cioè di vento verticale, basata principalmente sulle reviews storiche di Morris [12], Brock & Richardson [13] e Yajima [14].


Aspetti fisici ed aerodinamici

Figura 4.1. Schema fisico semplificato pallone

La forza ascensionale spingente o "Forza Lorda" è  $F_b = (m_a - m_g)g$  dove  $m_a$  è la massa atmosferica equivalente di riempimento del pallone che è spostata dalla massa del gas del pallone,  $m_g$ , e g= 9,81 $\frac{m}{s^2}$  è l'accelerazione di gravità. Il gas elio alle condizioni standard ha una densità di 0,164  $\frac{kg}{m^3}$  e l'aria ha una densità di circa 1,2  $\frac{kg}{m^3}$ , il che genera una portanza di circa 1,04 kg/m<sup>3</sup>. La massa lorda del pallone,  $m_G$ , include la massa del pallone in gomma,  $m_b$ , oltre la massa del carico,  $m_p$ , in modo che  $m_G = m_b + m_p$ . La Nozzle Lift equivale a  $F_b - m_b g$  mentre la portanza libera (o netta) è data da

$$F_L = F_B - W = F_B - m_G g.$$

### 4.1.2 Forza resistente

Quando il sistema del pallone viene lanciato, questo accelera rapidamente a una velocità costante verso l'alto basata sull'equilibrio tra la Forza di Portanza Libera (o netta) e la Forza di Resistenza  $(F_D), F_L = F_D$ .

La Forza di Resistenza

$$F_D = \frac{1}{2}\rho_a C_D A_b v^2$$

dove  $A_b = \pi \frac{D^2}{4}$  è l'area trasversale del pallone,  $\rho_a$  è la densità atmosferica, v è la velocità verso l'alto del pallone, e  $C_D$  è il coefficiente di resistenza ed è di solito circa 0.3.

La forza di resistenza sulla superficie del pallone  $F_D \cdot \Delta z$  è proporzionale all'energia cinetica impartita all'aria  $\frac{1}{2}m_av^2$  dove  $\Delta z$  è un incremento verticale dell'altezza. Se moltiplichiamo e dividiamo il lato dell'energia cinetica per  $\frac{V}{V}$ , dove V è il volume dell'aria spostata,  $V = A \cdot \Delta z$ , otteniamo che

$$F_D \cdot \Delta z \propto \frac{1}{2} \frac{V}{V} m_a v^2 = \frac{1}{2} \Delta z A \rho_a v^2$$

Quindi,  $F_D = \frac{1}{2}\rho_a C_D A_b v^2$ .

Questa formula può anche essere esplicitata per trovare la velocità durante la discesa con il paracadute prendendo A come l'area del paracadute,  $A_P$ . La velocità terminale di discesa può essere calcolata imponendo che la forza resistente sia uguale al peso totale del paracadute e del carico utile:

$$m_a g = F_D = \frac{1}{2} \rho_a C_D A_b v^2$$

Risolvendo per v si ottiene la velocità terminale di caduta,  $v_T$ .

$$v_T = \left(\frac{2mg}{(C_D\rho A)}\right)^{\frac{1}{2}}$$

Dove il coefficiente di resistenza,  $C_D$ , è compreso tra 0.5 e 0.8 (Seifert e Mcintosh [15]) per il paracadute.

 $C_D$  è una funzione del numero di Reynolds,  $Re = \frac{vD\rho}{\mu}$ , dove  $\mu$  è la viscosità dinamica. Dalla Fig.[4.2] [16]



Figura 4.2. Impatto del numero di Reynolds

si evince che il coefficiente di resistenza per un pallone è nella regione di interesse

mostrata dove il flusso può passare da laminare a turbolento. Comprendere la dinamica di volo e i parametri atmosferici durante un volo può aiutare a misurare indirettamente  $C_D$ . Il flusso bidimensionale attorno a un cilindro è più facile da visualizzare come mostrato in Fig.[4.3] ed è affermato essere molto simile al flusso tridimensionale intorno a una sfera.



Figura 4.3. Caption

Trovare modi intelligenti per registrare video del flusso aerodinamico nel laboratorio di volo in pallone ad alta quota ad alta quota sarebbe di grande aiuto per comprendere il comportamento del flusso e il coefficiente di resistenza. Valutazioni delle differenze aerodinamiche tra un pallone e una sfera utilizzando la modellazione fluidodinamica computazionale nel software Fluent sono state dimostrate da Scholes [17]. I risultati di questo studio mostrano che la resistenza su una forma di pallone realistica non è statisticamente diversa da una sfera, anche se sono stati investigati solo piccoli palloncini a basse altitudini con diametri inferiori a 1 metro e numeri di Reynolds di circa  $10^4 - 10^5$ .

# 4.1.3 Legge dei gas

Per un gas ideale, vale l'equazione dei gas perfetti PV = nRT, dove P è la pressione, V è il volume del pallone, n è il numero di moli di gas di sollevamento, R è la costante dei gas (RR = 8,31 J/(mole·K)), e T è la temperatura interna del pallone. Se moltiplichiamo entrambi

i lati di questa equazione per la massa molecolare media dell'aria (M = 28,97kg/mol), possiamo risolvere per la densità dell'aria come segue,  $\rho_a = \frac{M\rho_a P_a}{RT_a}$ . Si può anche risolvere per il volume del pallone se si usano i parametri del gas di riempimento designati dall'apice "g":

$$V_b = \frac{m_g R T_g}{M_q \rho_q}$$

 $V_b$  è una funzione del raggio del pallone quando approssimato al volume di una sfera,  $V_b = \frac{4}{3}\pi r^3$ .

### 4.1.4 Equazioni del moto

Utilizzando le leggi del moto di Newton, rappresentate dall'equazione F = ma, è possibile modellare il moto dinamico di un pallone aerostatico attraverso un'equazione differenziale del secondo ordine, introducendo alcune approssimazioni, come proposto da Bachman [11]. Nel caso di equilibrio stazionario, in cui l'accelerazione è nulla (a=0), la somma delle forze agenti sul pallone è pari a zero, con la forza resistente dell'aria uguale alla forza di sollevamento ( $F_L = F_D$ ). In questo contesto, la velocità di ascesa costante in stato stazionario può essere calcolata come

$$\mathbf{V_b} = \sqrt{\frac{\pi D^3 \left(\rho - \rho_B\right) g/6 - \left(m_B + m_P\right) g}{C_D \rho \pi D^2 / 8}}$$
(4.1)

Sostituendo valori realistici si stima che la velocità verticale del pallone sia dell'ordine di 5 metri al secondo. Interessante notare che in molte missioni, il pallone prosegue la sua ascesa per oltre 30 chilometri a una velocità approssimativamente costante di 5 m/s (circa 1000 piedi al minuto), nonostante le variazioni di numerosi parametri (densità atmosferica, dimensioni del pallone, resistenza aerodinamica, temperatura).

# 4.2 ASTRA - Equazioni del moto

Da questo punto in poi, verrà analizzato il modello fisico implementato nel simulatore sviluppato dall'università di Southampton.

E' conveniente separare il movimento di un pallone in una componente di deriva orizzontale in un piano definito da una latitudine x e una longitudine y e una componente di ascesa verticale (lungo l'asse z). Si calcola la deriva basandosi sull'assunzione lagrangiana che il movimento orizzontale del pallone è identico a quello del pacchetto d'aria che lo circonda, in modo che, in un sistema di coordinate terrestri,

$$\frac{dx}{dt} = -w_z \cos(\phi_z) \frac{dy}{dt} = -w_z \sin(\phi_z) \tag{4.2}$$

dove  $w_z$  e  $\phi_z$  sono la velocità e la direzione, rispettivamente, del vento a quota z. L'equazione della componente di ascesa è semplicemente un galleggiamento ("buoyancy") che si oppone al peso e alla resistenza,

$$(m_{\rm bal} + m_{\rm pay} + m_{\rm gas})\frac{d^2z}{dt^2} = B - (m_{\rm bal} + m_{\rm pay})g - D$$
(4.3)

dove le componenti di massa si riferiscono al pallone, al suo carico (compresi cavi, paracadute, derollatori, ecc.) e al gas di sollevamento. Si presume che siano costanti durante l'ascesa (cioè non viene rilasciata zavorra e il gas di sollevamento non si disperde). La massa del gas di sollevamento può essere determinata mediante misurazioni durante il processo di riempimento, sebbene il metodo più comune sia stimarla indirettamente dalla spinta (NL o Nozzle Lift) generata dal pallone gonfiato a terra (con NL definita come la forza verso l'alto misurata sulla bocca del pallone gonfiato, senza il carico attaccato). Se usiamo lo zero come apice per fare riferimento alle condizioni del sito di lancio,

$$m_{\rm gas} = \rho_{\rm gas}^0 \frac{\frac{N_L}{g} + m_{\rm bal}}{\rho_{\rm air}^0 - \rho_{\rm gas}^0}$$

dove le densità sono calcolate usando l'ipotesi dei gas perfetti, cioè  $\rho_{\text{gas}}^0 = p^0 \cdot M/R \cdot T^0$ (con i valori di M ponderati secondo i rapporti di contaminazione del gas applicabili, si vedano le equazioni in Appendice [A]). Il termine di galleggiamento di Eq.(4.3) varia con la quota z (poiché il volume, la pressione, la densità e la temperatura sono variabili con la quota) e viene calcolato come

$$B = V(z) \left[\rho_{\rm air}(z) - \rho_{\rm gas}(z)\right]$$

Forse la parte più interessante di Eq.(4.3) è il termine di resistenza. Per gli scopi di questa analisi, definiamo la resistenza D come la forza aerodinamica verticale che contrasta il galleggiamento del pallone e il coefficiente di resistenza  $C_D^b$  come il risultato adimensionale dato dalla divisione di questa forza con la pressione dinamica del flusso incontrata dal pallone e con la superficie frontale proiettata A(z) di una sfera con lo stesso volume del pallone (ipotizzato praticamente quasi sferico). Formalmente,

$$D = \frac{1}{2}\rho_{\text{air}(z)} \left(\frac{dz}{dt}\right)^2 C_D^b(Re)A(z)$$

Ancora una volta una forza che varia al variare della quota. Densità, temperatura, viscosità e pressione (e, di conseguenza, il volume e quindi l'area frontale del pallone) cambiano tutte, e le evidenze sperimentali suggeriscono che abbiano anche un effetto su  $C_D^b$  (più

facilmente riassunto da una variazione con il numero di Reynolds). L'equazione (4.3) ammette una soluzione analitica (si veda [14]) a condizione che sia ipotizzata una relazione  $C_D^b - Re$  sufficientemente semplice e che si assuma che la fisica dell'atmosfera sia rappresentata dal modello dell'Atmosfera Standard Internazionale (ISA). Tuttavia, quest'ultima ipotesi limiterebbe gravemente l'utilità pratica della previsione della traiettoria. Consideriamo solo le temperature atmosferiche effettive: esse deviano in modo molto significativo dalla ISA e questa deviazione varia da un giorno all'altro. Inoltre, stiamo cercando un modello in grado di simulare voli con una varietà di programmi di scarico del gas, un altro ostacolo significativo sulla strada delle soluzioni analitiche veloci. Il movimento verticale del pallone (4.3) deve quindi essere risolto numericamente dati i dati atmosferici previsti (o effettivi) del giorno, nonché tenendo conto di eventuali manovre di scarico che potrebbero essere necessarie. Allo stesso modo, le equazioni di deriva (4.2) devono essere risolte numericamente utilizzando venti previsti (o effettivi). L'integrazione nel dominio del tempo in entrambe le dimensioni, verticale (ascesa) e orizzontale (deriva), è spesso complicata in modo significativo dalla presenza di transitori di lancio e scarico, nonché da cambiamenti relativamente repentini nelle condizioni ambientali (velocità e temperatura del vento, ecc.), che possono rendere le equazioni "stiff". Unitamente alla necessità di una rapida integrazione (soprattutto alla luce delle incertezze che spesso richiedono insiemi relativamente ampi di simulazioni Monte Carlo), il modello ASTRA utilizza LSODA, un risolutore di equazioni differenziali ordinarie in grado di adattarsi a tali cambiamenti "in tempo reale". Nei casi difficili, gli schemi di differenziazione all'indietro, come quello implementato nel Livermore Solver for Ordinary Differential Equations (LSODE [19]), producono soluzioni stabili, con l'adattamento della dimensione del passo che assicura alta velocità. Ulteriori velocizzazioni sono possibili solo per casi "non-stiff" perchè LSODA riconosce il problema e passa automaticamente alla risoluzione tramite metodo Adams. Le condizioni al contorno per il modello sono stabilite da vari parametri atmosferici, tra cui temperatura, pressione, umidità e profili del vento. Queste condizioni sono ottenute ovviamente dal NOAA. Inoltre, le condizioni ambientali e la posizione geografica in cui avviene il processo di gonfiaggio del pallone contribuiscono anche a definire le condizioni al contorno per il modello.

# 4.3 ASTRA - Stime della resistenza del pallone

La ricchezza dei dati sperimentali disponibili sulla resistenza di oggetti sferici ha un'utilità limitata quando si tratta di modellare la resistenza che contrasta la salita di un pallone meteorologico. In primo luogo, si affronta un dominio del numero di Reynolds e del numero di Mach che ha visto sperimentazioni relativamente limitate, specialmente per quanto riguarda il volo nell'atmosfera libera (si faccia riferimento ai test aeroballistici condotti da Bailey & Hiatt [20]). In secondo luogo, la forma dei palloni meteorologici spesso si discosta in qualche misura dalla sferica. Terzo, questi risultati dovrebbero essere corretti per la resistenza di superficie di lattice sempre più teso.

Tuttavia, la dipendenza della resistenza dei palloni meteorologici dal numero di Reynolds sembra mostrare una caratteristica che condivide con le sfere lisce: una crisi di resistenza, tipicamente in qualche punto dell'intervallo di numeri di Reynolds compreso tra  $3 \times 10^5$  e  $4 \times 10^5$ .

Ad esempio, considerando il brusco calo della velocità di salita evidente nella Fig.[4.4] (che mostra il profilo di salita di un tipico volo di un pallone sonda)



Figura 4.4. Tipico spettro di volo di un pallone. La linea tratteggiata demarca la tropopausa

quando il pallone attraversa questa presunta regione di transizione la velocità verticale del pallone inizia a oscillare significativamente a un'altitudine di circa 7 km, prima di stabilizzarsi intorno a 4,5 m/s per il chilometro successivo circa. Questo periodo di variazione nella velocità di salita coincide approssimativamente con la tendenza generale del numero di Reynolds (derivato qui da una stima della variazione del diametro del pallone) che interseca i margini della banda  $3 \times 10^5$ - $4 \times 10^5$ . Ciò è coerente con una crisi di resistenza che si verifica in questo dominio, dalla quale il pallone esce con un coefficiente di resistenza significativamente aumentato.

Questo brusco cambiamento nella velocità di salita in questo particolare intervallo di numeri di Reynolds si verifica in modo coerente in un numero molto elevato di voli e quindi adottare un modello di resistenza che presenta una crisi di coefficiente di resistenza qui è, almeno in parte, praticamente utile nella previsione delle traiettorie dei palloni.

La fisica della crisi di resistenza potrebbe essere riassunta come segue: la variazione del numero di Reynolds che accompagna la salita di un pallone meteorologico è dominata dai cambiamenti nella densità dell'ambiente e nella viscosità, e quindi, nonostante il diametro del pallone aumenti a volte di un ordine di grandezza durante la salita, la tendenza generale del numero di Reynolds del pallone sarà decrescente. I numeri tipici visti all'inizio sono compresi tra  $6 \times 10^5$  e  $8 \times 10^5$ , nel limite di guota planetaria, e quindi lo strato limite del pallone sarà completamente turbolento. Quando il numero di Reynolds raggiunge l'intervallo di  $3 \times 10^5 \cdot 4 \times 10^5$  (tipicamente a circa metà della troposfera, anche se l'altitudine esatta può variare significativamente), inizierà a emergere una regione laminare nello strato limite vicino alla sommità del pallone. Ciò avrà l'effetto di avvicinare la fascia di separazione dello strato limite alla sommità e quindi aumenterà la resistenza. Nel tentativo di costruire un modello di volo di pallone generico, si parametrizza la variazione della resistenza delineata in precedenza e si cerca quindi una stima probabilistica di questi parametri. La complessità e la robustezza del processo di stima dei parametri (che verrà posta come un problema di ottimizzazione) aumenta (in modo esponenziale) con il numero di parametri.

Più formalmente, si ipotizza che che, nella regione di interesse dal punto di vista delle operazioni con palloni meteorologici, il coefficiente di resistenza del pallone in salita abbia la seguente forma:

$$C_d^b \left( \operatorname{Re}, C_d^l, \operatorname{Re}_c^l, \Delta \operatorname{Re}_c, C_d^t \right) = \begin{cases} C_d^l & \text{se } Re < Re_c^l \\ C_d^l - \left( C_d^l - C_d^t \right) \frac{Re - Re_c^l}{\Delta Re_c} & \text{se } Re_c^l \le Re \le Re_c^l + \Delta Re_c \\ C_d^t & \text{se } Re > Re_c^l + \Delta Re_c \end{cases}$$

$$(4.4)$$

La formula precedente descrive i parametri e la metodologia utilizzati per stimare il coefficiente di resistenza di un pallone meteorologico in ascesa. I parametri sono denotati con apici e pedici, dove "b" rappresenta il pallone, "l" rappresenta la regione laminare, "t" rappresenta la regione turbolenta e "c" rappresenta la regione di crisi. La stima di questi parametri è basata su un ampio set di dati empirici derivanti da più di 2000 voli di palloni sonda realizzati in lattice da 700 g riempiti con elio condotti dal Servizio Meteorologico

Nazionale degli Stati Uniti in otto diversi siti distribuiti negli USA. I parametri sono stimati minimizzando la differenza tra il profilo di velocità di ascesa previsto dall'Eq. (4.3) e il profilo di velocità di ascesa effettivo dei voli. L'ottimizzazione è stata effettuata dall'università di Southampton utilizzando un'euristica del tipo Nelder-Mead [22] [24]. La Fig.[4.5] mostra gli istogrammi di tutti i valori dei quattro parametri registrati durante i voli, con la zona centrale dell'immagine che rappresenta il modello di resistenza basato sulle medie stimate di ciascuna di queste distribuzioni. I risultati indicano che la posizione e l'ampiezza della regione di transizione critica sono molto simili ai dati sperimentali di Achenbach [18] e ai dati sperimentali recenti di Son [26]. La posizione della regione critica è influenzata dai livelli di turbolenza nel flusso libero, e in particolare, maggiore è la turbolenza nel flusso libero, più basso sarà il numero di Reynolds critico [23]. I dati suggeriscono che le intensità di turbolenza riscontrate dal pallone nei suoi voli ascendenti devono spesso essere inferiori al 0,45%.

È interessante notare che i valori medi della posizione di transizione e dell'ampiezza derivanti dall'analisi dei voli con pallone sono molto vicini ai dati di intensità di turbolenza al 0,45% di Achenbach [18]. Queste osservazioni sono generalmente considerate come un caso di riferimento di flusso "liscio". Tuttavia, i dati suggeriscono che le intensità di turbolenza viste dal pallone nei suoi voli ascendenti devono spesso essere inferiori al 0,45%. Questo potrebbe essere dovuto al fatto che le scale integrali di turbolenza nel flusso vista dal pallone sono molto più grandi del diametro del pallone stesso, rendendo il flusso essenzialmente quasi stazionario dal punto di vista dell'attività turbolenta nella sua regione di confine superficiale.

# 4.4 ASTRA - Incertezze e simulazione con metodo Monte Carlo

Oltre alle incertezze associate alla resistenza in ascesa, diverse altre fonti di incertezza ampliano i margini di errore intrinseco nella previsione della traiettoria di volo di un pallone. Esiste una certa tradizione storica nel calcolare l'impatto combinato di queste fonti di errore attraverso un framework di tipo Monte Carlo (vedi, ad esempio, Palumbo et al. [27]), e questo è l'approccio che è stato adottato con ASTRA. Quest'ultimo tiene conto di quattro di queste fonti nelle sue stime di errore: le discutiamo qui di seguito in ordine.

• Venti in quota e modello di deriva lagrangiano

La simulazione dell'elemento di deriva del movimento di un pallone meteorologico si



Figura 4.5. Modello della resistenza ottenuto sperimentalmente da 20993 voli

basa sull'assunzione postulata nell'Eq.(4.2) che le componenti orizzontali del moto siano le stesse di quelle della massa d'aria circostante. Forti gradienti di velocità del vento verticale possono congiurare con l'inerzia del carico utile per violare questa assunzione, ma l'esperienza indica che tali effetti hanno un impatto limitato sulla forma della traiettoria.

Una causa più significativa dell'incertezza nella modellizzazione della traiettoria risiede nelle incertezze associate al campo vettoriale del vento stesso. Due fonti comuni di dati sul vento utilizzati nella previsione dei voli con pallone sono i venti previsti in quota (estratti da un modello di previsione meteorologica numerica -NWP) e le osservazioni recenti effettuate nelle vicinanze del sito di lancio. Sebbene per la pianificazione a lungo raggio non vi siano alternative ai dati del modello NWP, per l'analisi del giorno del lancio i dati osservazionali presentano alcuni vantaggi, anche se sono stati registrati a distanza dal sito di lancio (sia in spazio che nel tempo). La cosa più importante è che le misurazioni si estendano molto nella stratosfera (tipicamente fino a circa 30-35 km, quando vengono effettuate con un pallone da 700 g), oltre il limite superiore di alcune griglie NWP. Inoltre, la loro risoluzione verticale è più elevata.

Consideriamo ora le incertezze associate all'uso dei dati osservazionali di una radiosonda per il calcolo della deriva in una simulazione di volo con pallone, derivanti dal tempo trascorso dall'ultima radiosonda disponibile, così come dalla distanza tra il sito di lancio del volo in programma e il luogo in cui è stata effettuata la rilevazione tramite radiosonda. Il modo, relativamente semplice, di tener conto di queste incertezze in un framework di tipo Monte Carlo adottato in ASTRA è il seguente: è stato costruito un database storico di radiosonde effettuate in diversi siti distanti tra loro a distanze mesoscalari. Sono state calcolate le differenze tra coppie di radiosonde vicine prese contemporaneamente, così come le differenze tra radiosonde consecutive (tipicamente prese a intervalli di 12 ore) effettuate nello stesso sito e poi per ogni volo nella simulazione Monte Carlo, sono state perturbate le radiosonde di base con una variazione selezionata casualmente da questo database di variazioni (scalate con la distanza effettiva tra il sito di lancio e la radiosonda di base, così come con il tempo trascorso tra quella radiosonda e l'ora del lancio). Infinte, è stato costruito il database a partire dai dati raccolti nell'arco di un anno da tre coppie di siti di rilievo atmosferico, garantendo così di avere dati rappresentativi di una gamma di condizioni orografiche, climatiche e meteorologiche. Le distanze tra i siti di ciascuna coppia erano rispettivamente di 150, 180 e 330 km.

Proprio questa ricostruzione rende affidabile il simulatore ma allo stesso tempo impreciso, data la natura delle osservazioni statistiche e la loro distanza dal sito di lancio del pallone di Stratobotic.

Dal punto di vista delle variazioni nel dominio delle distanze, la Figura 5 mostra la variazione con l'altitudine delle deviazioni quadratiche medie tra le radiosonde in ciascuna coppia, prese nello stesso momento. Proiettate le velocità del vento a livelli distanziati di 1 km fino a un'altitudine di 30 km sulle direzioni U e V; seguendo la convenzione meteorologica, U rappresenta la componente est-ovest (positiva a est) e V rappresenta la componente nord-sud (positiva a nord).

Come previsto, le deviazioni aumentano all'aumentare della distanza tra i siti. Inoltre, un'altitudine di circa 10 km segna una zona chiaramente distinguibile di maggiore incertezza associata alla corrente a getto (questo concorda con le osservazioni di Box et al. [15], che hanno condotto un esercizio simile come parte dello sviluppo di un codice di simulazione di volo in pallone).

### • Burst Diameter

In questa mini sezione, vengono esaminati e confrontati i diametri di "scoppio" dei palloni sperimentali sottoposti a diverse condizioni, dato utile per costruire un database empirico per palloni "futuri".

Nonostante gli elevati standard di qualità a cui sono sottoposti la maggior parte dei palloni meteorologici, è inevitabile una certa variabilità e la presenza di piccoli difetti (forse accentuati dai lunghi periodi di conservazione prima del volo). La metrica chiave sulla quale tali incertezze geometriche e chimiche hanno un impatto è il diametro di rottura del pallone. Naturalmente, questo è solo uno dei fattori che influenzano il diametro di rottura (sebbene probabilmente il più importante). Altri includono la tecnica di gonfiaggio, la contaminazione pre-lancio che porta a concentrazioni di tensione, il metodo di fissaggio del carico utile, il formarsi di ghiaccio, incontri con grandine e altre forme di precipitazione, forti gradienti di vento, ecc. Nel contesto generale dell'incertezza di un modello di previsione della traiettoria di volo con pallone, è quindi importante tenere conto della variabilità del diametro di rottura e, per stimare la funzione di densità di probabilità associata, sono stati esaminati 3199 voli condotti con palloni nominalmente identici da 700 g riempiti con elio (He) e lanciati da NOAA come parte delle normali operazioni di rilevamento della parte superiore dell'atmosfera. La Figura 6 è un istogramma dei diametri di rottura di questi palloni stimati dai dati chiave del lancio (temperatura di gonfiaggio e quantità di gas di sollevamento), dalle condizioni ambientali registrate dalla radiosonda trasportata dal pallone e dall'altitudine di rottura. Il set ha una media campionaria di 7,06 m (comodamente al di sopra del valore nominale del produttore di 6,53 m) e una deviazione standard di 0,72 m. La distribuzione di Weibull che meglio si adatta a questi dati ha i parametri k = 14,3577 e  $\lambda = 7,3293$ , in modo che il valore della funzione di densità di probabilità (mostrato anche nella Figura 6, sovrapposto all'istogramma) per un diametro d sia  $\binom{k}{\lambda} \left(\frac{d}{\lambda}\right)^{k-1} e^{-\left(\frac{d}{\lambda}\right)^k}$ .

Per consentire al modello di stimare distribuzioni anche per altre dimensioni di palloni, è stata adattata la distribuzione trovata per i palloni da 700 g in base ai diametri di rottura nominali di questi ultimi; questa dovrebbe essere un'approssimazione ragionevole basata sull'assunzione che l'architettura generale del pallone sia indipendente dalle sue dimensioni. La Figura 7 mostra alcune di queste distribuzioni, che coprono la gamma di masse dei palloni tra 100 e 3000 g.

### • Incertezze nella discesa

La maggior parte delle operazioni con pallone ad alta quota prevedono una discesa

con paracadute del carico utile. Questo aggiunge ulteriore incertezza alle stime del tempo di volo e amplia i margini di errore nella previsione della zona di atterraggio. Le dinamiche della disintegrazione del pallone giocano un ruolo importante in questo contesto. Nella maggior parte dei casi, il pallone si romperà in strisce lunghe e strette, ma la misura in cui queste strisce si separano dal collo del pallone (a cui sono collegati il carico utile e il paracadute) varia: in alcuni casi, tutta la gomma rimane attaccata, ma talvolta torna solo il collo. I dati disponibili sulle frazioni di ritorno sono estremamente scarsi e quindi il meglio che si può fare è assumere che il collo tornerà sempre (questo corrisponde a circa il 3% della massa complessiva del pallone) e che la probabilità di qualsiasi frazione del restante 97% che ritorni sia uguale. Per la maggior parte dei voli (specialmente quelli con carichi utili molto leggeri) questo rappresenterà una parte sostanziale della massa complessiva in discesa del treno.

Un'altra fonte di incertezza durante la discesa è la resistenza del paracadute. Sulla base dei dati disponibili, nonché delle osservazioni sperimentali, si stima un coefficiente di resistenza compreso tra 0,6 e 0,8; ancora una volta, si assume che i valori all'interno di questo intervallo siano ugualmente probabili (il valore effettivo è determinato dalla misura in cui il paracadute è stato dispiegato).

### • Insiemi Monte Carlo

Lo strumento di previsione del percorso di volo descritto qui è dotato della capacità di generare un insieme Monte Carlo di qualsiasi dimensione, basandosi sulla risoluzione delle eq.(4.2) e (4.3) con le distribuzioni postulate in precedenza. L'unico fattore che limita le dimensioni di questi insiemi è il budget computazionale dell'analista. Solitamente 50-100 simulazioni sono sufficienti per scopi operativi pratici (pianificazione dei voli), sebbene ne possano essere richieste migliaia per studi comparativi rigorosi.

Di seguito viene proposto un esempio di output generato con il codice ASTRA dall'università di Southampton, in Figura 8. Sono stati calcolati insiemi di traiettorie previste cinque giorni, tre giorni e un giorno prima del volo, oltre che il giorno del lancio. La traiettoria effettiva è mostrata nei pannelli sul lato sinistro della Figura 8, mentre sul lato destro sono rappresentate le curve di densità dei punti di atterraggio. Da un punto di vista della pianificazione operativa, queste curve possono essere considerate come surrogati delle probabilità di atterraggio. Le curve di densità sono state calcolate dagli insiemi Monte Carlo simulati mostrati in linee leggermente più sottili nei pannelli a sinistra. Il cerchio più vicino al margine sinistro di ciascun pannello rappresenta il sito di lancio. Gli algoritmi di stima della probabilità di posizione di atterraggio utilizzati in altre applicazioni spesso si basano sull'assunzione che la distribuzione dell'insieme dei punti di atterraggio sia probabilmente (vicina a) gaussiana. In particolare, le curve utilizzate nella pianificazione del volo sono quelle di una distribuzione normale adattata ai dati dell'insieme dei punti di atterraggio. Come mostra la Figura 8, tuttavia, questa non è un'opzione nel caso della pianificazione dei voli in mongolfiera meteorologica, dove le distribuzioni multimodali dei punti di atterraggio sono comuni negli insiemi Monte Carlo. Fattori come netti scambi di vento nell'atmosfera possono dividere le distribuzioni in questo modo.

# Capitolo 5

# Dinamica del vento

Lo studio della dinamica del vento e, in seguito, della sua evoluzione soprattutto nella zona delle Alpi aostane si è reso necessario perchè si potesse individuare un'altitudine "target" in cui la conformazione dei venti locali cambiasse drasticamente.

Il vento, infatti, costituisce una componente indispensabile dell'atmosfera terrestre. Esso è il risultato di complesse interazioni meteorologiche scaturite da fluttuazioni nella temperatura, pressione e umidità nell'ambiente atmosferico. L'analisi della dinamica del vento rappresenta una delle discipline di rilievo nell'ambito della meteorologia e della fluidodinamica, orientata allo studio del moto dell'aria all'interno dell'atmosfera. Questo fenomeno atmosferico segue un preciso insieme di principi fisici che possono essere formalmente espressi attraverso un sistema di equazioni matematiche ben strutturate e rigorosamente definite che verrano esposte nelle seguenti sezione prima di passare alle considerazioni prettamente geografiche relative alla zona di interesse di questo studio, ovvero le Alpi aostane.

# 5.1 Equazioni della dinamica del vento

Il vento è un movimento di massa d'aria che si verifica a causa di differenze di pressione atmosferica e, come tutti i flussi di materia è regolato dalle equazioni di Navier-Stokes. La pressione atmosferica è la forza che l'aria esercita su una superficie. Quando la pressione atmosferica è più alta in una zona, l'aria tende a fluire verso una zona con pressione atmosferica più bassa.

## 5.1.1 Equazione di continuità

Questa equazione si riferisce chiaramente alla conservazione della massa d'aria spostata dal vento:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0 \tag{5.1}$$

 $con \rho = densità dell'aria, u = velocità dell'aria e t = tempo.$ 

### 5.1.2 Equazione di bilancio della quantità di moto

Equazione fondamentale per la descrizione fluidodinamica del vento:

$$\rho\left(\frac{D\mathbf{u}}{Dt} + \mathbf{u} \cdot \nabla \mathbf{u}\right) = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{u}$$
(5.2)

I termini a sinistra rappresentano la variazione della quantità di moto del fluido in un volume e la forza viscosa che si oppone al movimento del fluido. I termini a destra rappresentano le forze esterne agenti sul fluido.

# 5.1.3 Equazione di bilancio energetico

Quest'ultima equazione è necessaria per definire il comportamento dei fluidi in condizioni di moto, sia laminare che turbolento.

$$\frac{\partial E}{\partial t} + \nabla \cdot \mathbf{q} = \rho \mathbf{u} \cdot \mathbf{g} + H \tag{5.3}$$

I termini dell'equazione possono essere scomposti come segue:

- Energia cinetica: L'energia cinetica è l'energia associata al movimento del fluido. È proporzionale alla massa del fluido e al quadrato della sua velocità.
- Energia potenziale: L'energia potenziale è l'energia associata alla posizione del fluido. È proporzionale alla massa del fluido e all'altezza della sua posizione.
- Energia interna: L'energia interna è l'energia associata al movimento delle particelle del fluido. È proporzionale alla temperatura del fluido.
- Flusso di calore: Il flusso di calore è il trasferimento di energia da un corpo caldo a un corpo freddo. È descritto dalla legge di Fourier.
- Lavoro svolto dalla forza di pressione: Il lavoro svolto dalla forza di pressione è il lavoro svolto dalla pressione del fluido per spostare un volume di fluido.

- Lavoro svolto dalla forza di viscosità: Il lavoro svolto dalla forza di viscosità è il lavoro svolto dalla viscosità del fluido per trascinare le particelle del fluido l'una contro l'altra.
- Produzione di calore: La produzione di calore è il calore generato dal fluido. È causato da una varietà di fattori, tra cui attrito, reazioni chimiche e radiazioni.

Viste le equazioni che regolano il comportamento fluidodinamico del vento, serve adesso individuare quali sono i fattori principali che ne modificano l'andamento. Infatti, il vento è un fenomeno complesso che è influenzato da una serie di fattori, tra cui:

- La temperatura dell'aria: L'aria calda è meno densa dell'aria fredda, quindi tende a salire. Questo crea un gradiente di pressione che può portare alla formazione di venti.
- La rotazione terrestre: La rotazione terrestre crea una forza chiamata forza di Coriolis, che devia i venti verso destra nell'emisfero nord e verso sinistra nell'emisfero sud.
- La topografia/orografia del terreno: La topografia del terreno può influenzare il vento in diversi modi, ad esempio creando venti locali o deviando i venti preesistenti.

Ed è proprio sull'ultimo punto che si concentra l'analisi di questo capitolo, vista la necessità di automatizzare il processo iterativo di interpolazione dei venti esposto nei capitoli successivi.

# 5.2 Orografia del terreno

La Terra ha una varietà di caratteristiche topografiche e orografiche, tra cui:

- Montagne: Le montagne sono rilievi terrestri elevati con una pendenza significativa. Le montagne più alte della Terra si trovano nella catena montuosa dell'Himalaya, che si trova tra l'Asia e l'Himalaya.
- Valli: Le valli sono depressioni nella superficie terrestre che sono solitamente circondate da montagne o colline. Le valli possono essere formate da erosione, accumulo o tettonica a placche.
- Pianure: Le pianure sono aree di terra relativamente piatte. Le pianure possono essere formate da erosione, accumulo o tettonica a placche.

- Deserti: I deserti sono aree di terra con scarse precipitazioni. I deserti possono essere formati da una varietà di fattori, tra cui clima, geologia e vegetazione.
- Foreste: Le foreste sono aree di terra ricoperte da alberi. Le foreste possono essere formate da una varietà di fattori, tra cui clima, geologia e topografia.

La scelta dei luoghi dei lanci di missione è, perciò, fondamentale e varia proprio in base alle caratteristiche territoriali circostanti.

Le missioni svolte finora da Stratobotic hanno mostrato evidenti risultati soddisfacenti quando effettuate in zone montanare; non a caso il Mission center si trova a Pont Saint-Martin, luogo da cui verranno eseguiti i prossimi lanci.

La difficoltà intrinseca della location montana sta proprio nella difficolà di predizione dei venti locali. Basti immaginare i venti di Fohn e la loro improvvisa variazione di difficile previsione. Nonostante ciò, è proprio il clima di montagna a favorire il moto di ascesa del pallone. Per questo motivo si è resa necessaria un'analisi approfondita dell'influenza dell'orografia del terreno sui venti locali, per poter predire al meglio l'eventuale comportameno in discesa del pallone.

# 5.2.1 Venti Foehn

Il vento foehn [32] è un vento discendente, tipico della zona Alpina, che si verifica quando l'aria calda e secca "cade" da una montagna. Il vento foehn è spesso associato a un fronte freddo, linea di separazione tra aria calda e fredda. Quando il fronte freddo passa, l'aria calda e umida viene spinta verso l'alto dalla montagna. L'aria si raffredda man mano che sale e l'umidità si condensa formando nuvole e precipitazioni.

L'aria che è stata raffreddata e scaricata dall'altra parte della montagna è più secca e calda di quanto non fosse quando è salita. Questo è dovuto al fatto che l'aria perde acqua quando si condensa. Il vento foehn può portare a un aumento della temperatura, una diminuzione dell'umidità e una diminuzione della visibilità, oltre a un'evidente variazione delle velocità dell'aria all'aumentare della quota [33].

Questo è uno dei principali attori nel cambiamento delle condizioni ambientali in cui vola il pallone. L'altitudine, come facile intuire, gioca un ruolo fondamentale nella creazione di questi venti e nella modifica delle condizioni locali, dato che è proprio il fronte freddo/ostacolo a variare la direzione e la velocità di questi.

# 5.2.2 Ciclogenesi

In meteorologia la ciclogenesi è il processo di formazione o rafforzamento della circolazione ciclonica nell'atmosfera, cioè la formazione di una zona di bassa pressione e il conseguente aumento turbolento delle raffiche di vento. L'altitudine, come spiegato da Schar [34] e approfondito nel documento di Buzzi e Davolio [38], gioca un ruolo importante nel fenomeno della cyclogenesi orografica. Le montagne creano un ostacolo al flusso d'aria, che viene forzato a salire. L'aria che sale si raffredda e condensa, formando nuvole e precipitazioni. La liberazione di calore latente associata alla condensazione fornisce energia per lo sviluppo di un ciclone, che ha effetti molto importanti ma soprattutto imprevedibili, rendendo una previsione di traiettoria quasi imprevedibile.

# 5.3 Flusso orografico

Il documento pubblicato dall'ECMWF nel 1992 [44] fornisce una panoramica del flusso orografico nelle vicinanze delle Alpi, zona di grande interesse per questo progetto. Il flusso orografico è il flusso d'aria che viene forzato a salire da una catena montuosa. L'aria che sale si raffredda e condensa, formando nuvole e precipitazioni. Il flusso orografico è influenzato da una serie di fattori, tra cui:

- L'altitudine delle montagne
- La forma delle montagne
- La direzione del vento
- La temperatura e l'umidità dell'aria

### Effetto dell'altitudine

L'altitudine delle montagne è il fattore che influenza maggiormente il flusso orografico. Le montagne più alte creano un ostacolo più significativo al flusso d'aria, favorendo lo sviluppo di un flusso orografico più intenso.

Questo è dovuto a due motivi principali. Innanzitutto, l'aria si raffredda a una velocità di circa 0,65 gradi Celsius per 100 metri di altezza. Pertanto, l'aria che sale su una montagna più alta si raffredda di più rispetto all'aria che sale su una montagna più bassa.

In secondo luogo, l'aria che sale su una montagna più alta ha un tempo di salita più lungo. Questo significa che ha più tempo per raffreddarsi e condensare, formando nuvole e precipitazioni.

### Effetto della forma delle montagne

La forma delle montagne può influenzare la turbolenza del flusso orografico. Le montagne con una forma più complessa possono creare un flusso orografico più turbolento.

Ci sono diverse ragioni per cui questo accade. Innanzitutto, le montagne con una forma più complessa creano più zone di ristagno e di separazione del flusso d'aria. Queste zone possono provocare la formazione di vortici e altri fenomeni di turbolenza.

In secondo luogo, le montagne con una forma più complessa possono creare più punti di riflessione del flusso d'aria. Questi punti possono provocare la riflessione dell'aria verso l'alto, aumentando la probabilità di formazione di nuvole e precipitazioni.

### Effetto della direzione del vento

La direzione del vento può influenzare l'intensità del flusso orografico. Il vento che soffia da una direzione più ortogonale alla catena montuosa crea un flusso orografico più intenso. Ci sono due motivi principali per cui questo accade. Innanzitutto, il vento che soffia da una direzione più ortogonale alla catena montuosa incontra una maggiore resistenza da parte delle montagne. Questa resistenza forza l'aria a salire più rapidamente, aumentando la sua intensità.

In secondo luogo, il vento che soffia da una direzione più ortogonale alla catena montuosa crea una maggiore differenza di pressione tra la zona di salita e la zona di discesa. Questa differenza di pressione aumenta la forza di sollevamento, che spinge l'aria verso l'alto.

#### Effetto della temperatura e dell'umidità dell'aria

La temperatura e l'umidità dell'aria possono influenzare l'intensità del flusso orografico. L'aria più calda e umida crea un flusso orografico più intenso.

Ci sono due motivi principali per cui questo accade. Innanzitutto, l'aria calda e umida contiene più vapore acqueo, che si condensa formando nuvole e precipitazioni.

In secondo luogo, l'aria calda e umida ha una densità inferiore rispetto all'aria fredda e secca. Questo significa che è più facile da sollevare, aumentando l'intensità del flusso orografico.

Il flusso orografico ha un impatto significativo sul clima delle regioni montuose. I venti forti causati dal flusso orografico, infatti, oltre a poter danneggiare le infrastrutture e creare condizioni pericolose per le attività all'aperto potrebbe causare evoluzioni turbolente ed improvvise della traiettoria di volo del nostro pallone, rendendone difficile la previsione accurata.

# 5.4 Conclusioni e modifiche da apportare

In questa sezione verranno esposte le modifiche scelte da apportare al simulatore in base a quanto studiato ed esposto finora. Risulta chiaro, anche in base agli articoli dei professori Schmid [41] e Schmidli [42], che la topografia del terreno circostante la zona di lancio influenzi in maniera evidente i venti cui è sottoposto il pallone.

Per questo motivo un ulteriore ed ultimo studio è stato condotto sui venti in Piemonte [43] e nelle zone limitrofe, per capire quanto fosse influenzata l'intensità delle velocità di ascesa e discesa al variare dell'altitudine. Infatti, oltre a dare una buona stima delle velocità dei venti locali, il documento riesce a fornire una panoramica al variare delle altitudini delle stazioni di terra. Ebbene, si conclude che sopra i 3000 m, i venti locali sono quasi per nulla influenzati da variazioni macroscopiche mostrando un comportamento quasi costante. Risulta anche vero, però, che essendo la zona di lancio vicino a una catena montuosa importante , l'altitudine per cui i venti locali non sono influenzati da conformazione orografica è variabile proprio al variare del profilo del terreno.

In conclusione, la scelta è quella di impostare un valore di altitudine per cui il simulatore inizi a mediare i dati di vento reali (raccolti durante la fase di ascesa) e i dati di vento previsti così che l'errore di predizione del landing point risuli sensibilmente minore.

Come verrà esposto nel dettaglio nella Sez.[7.2], la scelta è ricaduta sulla creazione di un codice che fosse in grado di rilevare l'altitudine maggiore in una determinata area circostante il punto di lancio con l'ausilio di un file GeoTIFF, ed usare questa stessa altiudine come punto in cui partire con la mediazione dei dati.

# Capitolo 6

# ASTRA & honeycomb

In questo capitolo verranno esaminati l'ambiente di lavoro insieme all'architettura del simulatore di volo ASTRA, il suo funzionamento e quello dell'interfaccia honeycomb, prima di focalizzarsi sulle modifiche apportate.

# 6.1 Work environment

In questa sezione verrà esposta in maniera più o meno dettagliati, quali mezzi e linguaggi di programmazione sono stati usati nel corso del progetto, supportando la descrizione con qualche screenshot.

## 6.1.1 Sistemi operativi - Windows & Linux

Fin dall'inizio il progetto è stato approcciato da un terminale Windows **■** built-in. Durante il lavoro, però, si è reso necessario l'uso di una "distro" Linux Å, in particolare Ubuntu Jellyfish, per far si che alcune feature di honeycomb potessero essere eseguite senza problemi o add-on particolarmente "noiosi" per la macchina Windows.

Il tutto è stato svolto su un solo computer, con la realizzazione di una macchina virtuale alternativa a quella di base fisica, tramite l'utilizzo dell'app VirtualBoxMachine.

### 6.1.2 Python

Il linguaggio di programmazione principale, se non unico, del simulatore di volo è Python. Ciò ha richiesto uno studio preliminare del suddetto linguaggio di programmazione, che è stato reso possibile dai corsi "SkillsForAll" di Cisco Networking Academy, in particolare "Python Essentials 1" e "Python Essentials 2", per un totale di 90 ore di corso. La versione utilizzata per lo sviluppo delle modifiche è stata Python 3.11.2 come visibile dalla riga del terminale qui di seguito.

```
PS <u>C:\Users\aless\Desktop\Stratobotic_Main\honeycomb-alessio\honeycomb-main</u>> python -V Python 3.11.2
```

Figura 6.1. Versione Python

Per quanto riguarda le fasi preliminari riguaranti l'installazione del software ASTRA, sono state installate le versioni delle seguenti librerie, necessarie a far girare il codice:



Figura 6.2. Requirements

Si possono notare numerose librerie base come "matplotlib", "numpy" e "scipy" necessarie per tutte le operazioni matematiche definite all'interno del simulatore; oppure "gevent", "grequests" utili per effettuare operazioni di debugging, vitali per la riuscita ottimale delle modifiche; o ancora "urllib3" fondamentale per permettere al simulatore di effettuare il download dei dati GFS dal NOAA. Maggiori informazioni e dettagli sono esposti nell'Appendice

Bisogna notare però che, a differenza del pacchetto base di requisiti richiesti dal simulatore ASTRA (la cui repo è possibile trovare al linkgithub  $\mathbb{Z}$ ),

| 1  | certifi==2020.11.8                |
|----|-----------------------------------|
| 2  | chardet==3.0.4                    |
| 3  | cycler==0.10.0                    |
| 4  | deap==1.3.1                       |
| 5  | gevent==20.9.0                    |
| 6  | greenlet==0.4.17                  |
| 7  | grequests==0.6.0                  |
| 8  | idna==2.10                        |
| 9  | kiwisolver==1.3.1                 |
| 10 | <pre>matplotlib==3.3.3</pre>      |
| 11 | numpy==1.19.4                     |
| 12 | pyparsing==2.4.7                  |
| 13 | <pre>python-dateutil==2.8.1</pre> |
| 14 | requests==2.25.0                  |
| 15 | scipy==1.5.4                      |
| 16 | six==1.15.0                       |
| 17 | urllib3==1.26.5                   |
| 18 | <pre>zope.event==4.5.0</pre>      |
| 19 | <pre>zope.interface==5.2.0</pre>  |

Figura 6.3. Requirements di ASTRA "base"

compare in più una libreria denominata Rasterio.

Questa libreria si è rivelata fondamentale per l'implementazione di un modulo in particolare ("flight\_tools.py") atto a rilevare il dato di altitudine maggiore in un determinato quadrante. Per più dettagli, fare riferimento alla sezione 7.2

# 6.1.3 JavaScript

Con l'implementazione di honeycomb, si è resa necessaria anche la conoscenza, ad alto livello, del linguaggio JavaScript. Infatti, per comprendere il funzionamento dell'interfaccia online del simulatore sviluppata da Stratobotic, quali funzioni richiamasse e quali file processasse, è stato utile far riferimento a una conoscenza di base del linguaggio ideato da Brendan Elch. Come per Python, il corso SkillsForAll di Cisco ha reso possibile uno studio iniziale di JavaScript con il corso "JavaScript 1".

Ultimato lo studio, è stato possibile comprendere come il simulatore e, di conseguenza, le simulazioni vengono processate e "lette" dall'interfaccia online.

## 6.1.4 File .json, .csv, .kml

I file citati in questa sezione sono di rilevante importanza nelle analisi effettuate dal simulatore e soprattutto per la definizione di errori e il plottaggio di grafici, utili soprattutto a dare un riscontro grafico esterno a quella che è la missione in sè.

Infatti, data la scarsità di archivi "storici" contenenti dati di missione, le analisi al di fuori della missione si sono concentrate su due missioni in particolare, quella del 19/10/2022 e quella del 11/08/2023. Solo successivamente è stato possibile provare ad effettuare una missione vera e propria, di cui si parla al Cap.[8].

## File .json

JSON (JavaScript Object Notation) è un formato standard aperto per la rappresentazione e lo scambio di dati. Si basa su testo leggibile e viene comunemente utilizzato per memorizzare e trasmettere dati. I file JSON hanno un'estensione .json. Rispetto a XML, JSON richiede una formattazione meno verbosa ed è spesso considerato un'alternativa più leggera. Nonostante derivi da JavaScript, JSON è un formato di dati indipendente dal linguaggio di programmazione e gode di ampio supporto per la generazione e l'analisi dei dati in molti linguaggi di programmazione moderni. Nel contesto delle comunicazioni via Internet, il tipo di supporto "application/json" viene utilizzato per identificare i dati JSON.

Il file .json più importante nella stima di errori è stato "post\_processing\_input.json" (Fig.[6.4]) derivante da un tool preflight che, dati in input velocità di ascesa desiderata, tipo di paracadute e pallone, restituisce in output la Nozzle Lift desiderata.

### File .csv

Un file CSV è definito dall'estensione ".csv", che indica un file contenente dati strutturati sotto forma di valori separati da virgola. Questa estensione di file è l'abbreviazione di "comma separated values" (valori separati da virgola) ed è comunemente utilizzata come formato di output da molte applicazioni di fogli di calcolo.

Nel contesto di un file CSV, i dati sono organizzati in modo che siano separati da virgole, come se fossero delle matrici. Quando questi dati vengono importati in un'applicazione di fogli di calcolo, vengono interpretati e suddivisi in celle, tabelle e colonne, rendendo più agevole la loro visualizzazione, manipolazione e analisi.

Si tratta dell'estensione prediletta usata nel confronto tra traiettorie, poichè il simulatore conserva i dati essenziali della traiettoria in un file .csv(Fig.[6.5]), così come segue:

| astra_s | <pre>imulator_stratobotic &gt; error_estimation_study &gt; {} post_processing_input.json</pre> |
|---------|--|
| 1       | {  |
| 2       | "Launch_Date_String": "110823",  |
| 3       | "Payload_Mass": 2.55,  |
| 4       | "Nozzle_Lift": 4.5,  |
| 5       | "Balloon_Model": "HW1200",   |
| 6       | "Parachute_Model": "RCK6",   |
| 7       | "Parachute_Coefficient": "1_4",  |
| 8       | "Launch_Elevation": 320,   |
| 9       | "Landing_Elevation": 186,  |
| 10      | "Step_File": 500   |
| 11      |  |

Figura 6.4. post\_processing\_input.json



Figura 6.5. out.csv

Come si può evincere dal formato visualizzato, non si tratta nient'altro che di una matrice di valori allocati nella rispettiva coordinata. Chiaramente, le colonne relative a latitudine, longitudine e altitudine sono le principali candidate ad essere adoperate come valori di confronto.

### File .kml

Un file con estensione ".kml" è un documento che segue il formato Keyhole Markup Language (linguaggio di marcatura Keyhole). I file KML sono strutturati utilizzando la sintassi XML e sono progettati per rappresentare dati geografici, comprese annotazioni e visualizzazioni. Questi dati possono includere informazioni come posizioni geografiche, sovrapposizioni di immagini, collegamenti multimediali come video e dati di modellazione, ovvero linee, forme, immagini tridimensionali e punti.

L'utilità principale dei file KML, come quello mostrato in Fig.[6.6] è quella di fornire una struttura standard per la rappresentazione di dati geospaziali in modo che possano

essere facilmente compresi e utilizzati da diversi programmi software e servizi web.

| astra_output > 🔊 out.kml |  |  |
|--------------------------|--|--|
| 24                       | <style id="burstPin"></td></tr><tr><td>25</td><td><IconStyle></td></tr><tr><td>26</td><td><scale>1.0</scale></td></tr><tr><td>27</td><td><Icon></td></tr><tr><td>28</td><td><pre><href>http://maps.google.com/mapfiles/ms/micons/yellow-dot.png</href></pre></td></tr><tr><td>29</td><td></Icon></td></tr><tr><td>30</td><td></IconStyle></td></tr><tr><td>31</td><td></style> |  |
| 32                       | <style id="stratoLine"></td></tr><tr><td>33</td><td><LineStyle></td></tr><tr><td>34</td><td><width>1.0</width></td></tr><tr><td>35</td><td></LineStyle></td></tr><tr><td>36</td><td></style>   |  |
| 37                       | <placemark></placemark>  |  |
| 38                       | <name>Simulation 1</name>  |  |
| 39                       | <styleurl>#stratoLine</styleurl>   |  |
| 40                       | <linestring></linestring>  |  |
| 41                       | <coordinates></coordinates>  |  |
| 42                       | 3.13735,45.59385,370.00000   |  |
| 43                       | 3.13734,45.59384,370.93881   |  |
| 44                       | 3.13732,45.59382,378.10948   |  |
| 45                       | 3.13725,45.59377,406.02987   |  |
| 46                       | 3.13698,45.59358,516.85980   |  |
| 47                       | 3.13655,45.59327,692.18489   |  |

Figura 6.6. out.kml

Questa interoperabilità consente a varie applicazioni geospaziali di scambiare e condividere dati in modo efficace. Uno dei primi utilizzi noti dei file KML è stato con il software Keyhole Earth Viewer sviluppato da Keyhole, Inc., prima che fosse acquisito da Google nel 2004 e successivamente integrato in Google Earth. Qui di seguito (Fig.[6.7]) viene mostrata la finestra di google earth dopo aver importato il file .kml qui sopra:



Figura 6.7. Traiettoria visualizzata su Google Earth con file kml

# 6.2 ASTRA - Cosa è e come funziona

Come si evince dai capitoli precedenti, si parla di un simulatore di volo sviluppato interamente in Python, in grado di predire la traiettoria di un pallone aerostatico dati in input dati atmosferici e fisici relativi al pallone.

Il simulatore è in grado di predire la traiettoria del pallone anche in base a diverse condizioni di volo, che sia "Floating" o meno (semplicemente propulso), o a diversi tratti del volo; infatti lo strumento è utile anche per fare previsioni di traiettoria solo in discesa o solo in salita.

Per capire al meglio come funziona la fisica del simulatore, verranno esaminati i moduli più importanti tra quelli qui di seguito elencati (Fig.[6.8]) che formano il simulatore vero e proprio.



Figura 6.8. Moduli di ASTRA

# 6.2.1 Simulator.py

Il modulo principe di ASTRA è, ovviamente, "simulator.py". Al suo interno, infatti, sono definite praticamente tutte le funzioni matematiche che definiscono il moto del pallone. Ma non solo, perchè in questo modulo sono pure definite le funzioni atte a creare i file con estensioni .csv .json e .kml viste precedentemente.

Il modulo è fondamentalmente diviso in due classi principali: flightProfile e flight.

La prima rielabora tutti i dati posizionali di output della seconda per creare i file con le estensione di cui si è parlato nella Sez.[6.1.4]. La seconda, appunto, dà in output i dati posizionali dopo averli calcolati tramite il solver di ODE denominato "LSODA". Qui di seguito verrà mostrata la Fig.[6.9] con la parte di codice in cui è presente la definizione dell'equazione differenziale del moto del pallone:

| 1406 |  |
|------|--|
| 1407 | # External Forces  |
| 1408 | externalForces = -self.payloadTrainWeight * 9.81 * (                               |
| 1409 | 1 + selfballoonReturnFraction[flightNumber])\                                      |
| 1410 | + parachuteDrag + trainDrag  |
| 1411 |  |
| 1412 | # Derivatives  |
| 1413 | <pre>dvdt = externalForces / selftotalDescendingMass</pre>                         |
| 1414 | dhdt = ascentRate  |
| 1415 |  |
| 1416 | return numpy.array([dhdt, dvdt])   |
| 1417 |  |
| 1418 | # Define the initial conditions, the time vector at which we want                  |
| 1419 | # simulation data to be stored, and run the integration.                           |
| 1420 | # Note: the simulation carries on all the way to the maxFlightTime,                |
| 1421 | # even if the altitude becomes negative. Negative values of altitude               |
| 1422 | # will be trimmed later on.  |
| 1423 | <pre>initialConditions = numpy.array([self.launchSiteElev, 0.0])</pre>             |
| 1424 |  |
| 1425 | logger.debug('Beginning integration.')   |
| 1426 |  |
| 1427 | ### INTEGRATION ###  |
| 1428 | if selfusingGFS:   |
| 1429 | solution = solve_ivp(fun= ode, method= 'LSODA' ,t_span= (0, self.maxFlightTime), \ |
| 1430 | y0= initialConditions, vectorized= True,\  |
| 1431 | <pre>max_step = self.samplingTime, rtol=1e-3, atol=1e-3)</pre>                     |
| 1432 | else:  |
| 1433 | solution = solve_ivp(fun= ode, method= 'LSODA' ,t_span= (0, self.maxFlightTime), \ |
| 1434 | y0= initialConditions, vectorized= True,\  |
| 1435 | <pre>max_step = self.samplingTime)</pre>   |
| 1436 | ************   |
| 1437 |  |
| 1438 | <pre>logger.debug('Integration completed. Post-processing')</pre>                  |

Figura 6.9. Equazione ordinaria differenziale

Dalla linea 1408 fino alla 1414 si può notare la formulazione esplicita della stessa equazione differenziale di cui si è parlato nella Sez.[4.2]. Dalla linea 1429 invece si riconosce la chiamata del solver di ODE "LSODA". Proprio nella parte antecedente a questa del codice si svilupperà una delle modifiche principali apportate al simulatore, che cambia il modo di calcolare i venti da cui si estrapolano poi le posizioni di longiudine e latitudine istante per istante del pallone, che dal simulatore di base viene svolto usando esclusivamente le previsioni del GFS nel modo illustrato in Fig.[6.10]:



Figura 6.10. GFS wind

## 6.2.2 GFS.py

Il modulo GFS.py contiene le classi necessarie a interagire con il Global Forecasting System del NOAA; di tutte le classi che verrano elencate, la primaria è la GFS\_Handler che effettua download e rielabora le previsioni GFS.

### GFS\_Handler

Effettua il download dei dati meteorologici previsionali per un determinato luogo sulla Terra, con un intervallo massimo di 10 giorni rispetto al ciclo temporale più prossimo. Tali dati possono essere ottenuti sia tramite il servizio HD, che offre informazioni dettagliate su una griglia di dimensioni pari a 0,25 gradi di latitudine per 0,25 gradi di longitudine, suddivisa in 26 livelli altimetrici, sia tramite il servizio SD, il quale utilizza una griglia di risoluzione inferiore. È importante notare che, anche quando si opta per il servizio HD, i dati relativi alle altitudini più elevate vengono richiesti in formato SD, poiché la copertura del servizio HD si estende solamente fino a un'altitudine corrispondente a 10 millibar di pressione atmosferica.

Tra gli input di questa classe troviamo dati fisici come quelli posizionali di latitudine e longitudine, ma anche operatori booleani uilizzati come flag tipo HD settato a True di default che impone, appunto, il download dei dati GFS in HD.

Nota di riguardo per il NOAA che pubblica nuovi insiemi di dati (cicli) alle ore 00:00, alle 06:00, alle 12:00 e alle 18:00 UTC ogni giorno, fornendo previsioni meteorologiche per un periodo di 10 giorni.

È importante notare che poiché non è possibile conoscere in anticipo quale sia l'ultimo ciclo disponibile (a causa di un ritardo tra l'orario di pubblicazione del ciclo e la sua effettiva disponibilità), è opportuno limitare il periodo di tempo per il download dei dati dal Global Forecast System (GFS) a un massimo di 8 giorni. Questa precauzione è necessaria per garantire l'uso di dati aggiornati e affidabili nelle analisi e nelle previsioni meteorologiche.

### get\_urldict\_async

Funzione ausiliaria asincrona per effettuare richieste url multiple per ogni key in url\_dict. Utilizza grequests per generare le richieste in modo concorrente. Parametri:

- url\_dict : dizionario di liste (key: [url1, url2...]) coppie di url logicamente correlati.
- hooks\_dict : dizionario

### Output:

• results\_dict: dizionario di liste

### GFS\_Map

Classe usata per salvare il mapping in 4D dei dati scaricati per un parametro specifico.

#### GFS\_data\_interpolator

Classe usata per interpolare i dati. Funziona da interfaccia capace di usare le coordinate corrette quando richieste dal GFS Handler come dati interpolati. In pratica, una volta richiamato con le coordinate richieste, converte l'altitudine in pressione usando il pressure inerpolator e interpola i dati richiesti per poi ritornarli come output.

# 6.2.3 interpolate.py

All'interno di questo modulo è definita soltanto una funcion, "Linear4DInterpoaltor".

Il Linear4DInterpolator è un interpolatore lineare in quattro dimensioni progettato per l'uso su griglie rettangolari, anche se non necessariamente uniformi.

In sostanza, il Linear4DInterpolator identifica l'iper-tetraedro in quattro dimensioni in cui si trova il punto richiesto e quindi esegue un'interpolazione quadrilineare utilizzando solamente i 16 vertici di quell'iper-tetraedro.

Questo metodo è stato implementato perchè, come spiegato da Niccoloò Zapponi nella sua descrizione, "si è dimostrato significativamente più efficiente rispetto all'interpolazione lineare in quattro dimensioni basata sul metodo di triangolazione di Qhull utilizzato da scipy, il quale risulta particolarmente lento quando applicato a griglie rettangolari".

# 6.2.4 weather.py

Questo modulo contiene le classi necessarie per modellare il meteo e l'atmosfera. La classe environment passa al simulatore tutti i dati atmosferici richiamando le sottoclassi "forecastEnvironment" e "soundingEnvironment".

**soundingEnvironment** è la classe che genera il modello amosferico a partire dai file con estensione .ftr o .sounding.

**forecastEnvironment** è invece la classe responsabile del download delle previsioni meteo dal GFS e genera un modello atmosferico basato sulle previsioni.

In parole povere svolgono lo stesso compito ma a partire da situazioni diverse, in cui nella prima si conosce già la situazione meteorologica della zona di lancio; nella seconda, invece, c'è la necessità di interrogare i server GFS. All'interno di questo modulo verranno poi applicate delle modifiche necessarie per l'interpolazione dei dati di vento rilevati in ascesa.

# 6.2.5 global\_tools

Questo modulo contiene i tool necessari per effettuare le dovute conversioni e/o calcoli. Di tutte le function al suo interno, ce ne sono alcune degne di nota oltre le funzioni basilari di conversione. Tra queste vi sono:

- haversine : La formula del semisenoverso calcola la distanza del cerchio massimo tra due punti su una sfera dati in input longitudini e latitudini[53].
- dirspeed2uv : Converte la direzione e la velocità del vento in componenti u e v (componenti universali)

- **uv2dirspeed** : Converte le componenti u e v (componenti universali) del vento in direzione e velocità.
- **ISAatmosphere** : fornisce le condizioni atmosferiche ISA per i dati di input forniti (almeno una tra altitudine e temperatura deve essere fornita).

In Fig.[6.8] sono riconoscibili altri moduli quali mean\_tools.py e regression\_tools.py, i quali però sono stati implementati successivamente per le modifche apportate, perciò verranno discussi nel Cap.[7].
# 6.3 Honeycomb

La struttura di honeycomb, la repo sviluppata e fornitami da Stratobotic, è invece molto più complessa, in quanto il simulatore ASTRA ne rappresenta "soltanto" il core numerico. Infatti, come visibile da Fig.[6.11]:

| >   | astra_output                   |
|-----|--------------------------------|
| >   | astra_simulator_stratobotic    |
| >   | image_visualizer               |
| >   | position_visualizer            |
| >   | preflight_tool                 |
| >   | simulation_visualizer          |
| >   | workspaces                     |
|     | .gitignore                     |
| {}  | astra_output_progress.json     |
| Ξ   | astra_py_error.log             |
| \$  | cleaner.sh                     |
| \$  | cleanRestart.sh                |
| \$  | clearlog.sh                    |
| {}  | config.json                    |
| ٠   | infrastructure_architecture.md |
| \$  | init.sh                        |
| \$  | launch.sh                      |
| -{} | package-lock.json              |
| {}  | package.json                   |
| {}  | prova.json                     |
| ٢   | README.md                      |
| \$  | stop.sh                        |

Figura 6.11. Struttura honeycomb

vi sono molte più cartelle al cui interno sono definiti numerosi moduli (Python e non) per l'interfaccia grafica online.

Presente il preflight\_tool che, come spiegato in sez.[6.4.2], serve a ricevere in output i parametri di volo necessari a raggiungere i valori desiderati immessi come input (ad esempio, la nozzle lift necessaria per raggiungere i 5 m/s di velocità di ascesa).

#### 6.3.1 Installazione

Essendo una repo abbastanza complessa, essa richiede l'installazione di alcuni pacchetti preliminari per poi "runnare" i file .sh. Proprio nel caso di questi ultimi, si è rivelato necessario l'intervento della macchina virtuale Linux, in quanto, pur avendo installato Git Bash, Windows non è in grado di leggere propriamente suddetti file, rendendo impossibile il lavoro con l'interfaccia grafica.

Molto brevemente, dopo aver installato "Node.js", un handler di JavaScript, è stato necessario lanciare "init.sh" per installare correttamente la suite honeycomb.

La suite funziona proprio automaticamente richiamando i file con estensione .sh, ed in particolare di seguito vengono espresse le loro funzioni:

- **init.sh** : inizializza la suite honeycomb. Necessaria per installare tutte le dependencies.
- launch.sh : inizializza tutti i moduli.
- **stop.sh** : arresta tutti i moduli.
- cleaner.sh : cancella tutti i log dai moduli, rimuove i dati immagazzinati nel position\_visualizer e tutti i log relativi ad ASTRA.
- **clearlog.sh** : cancella soltanto i log dai moduli. Una sorta di versione "lite" della precedente

## 6.3.2 App

- ASTRA Simulator : come visto fin qui, il core principale del calcolo numerico della simulazione.
- Image Visualizer : WebApp usata per il post-mission processing. Fa uso dei software "Metashape" e "Geoserver geospatial storage" per permettere all'utente di vedere su mappa il mosaico di immagini creato dalla missione.
- **Position Visualizer** : WebApp usata per il planning della missione. Permette all'utente di vedere le posizioni del pallone in tempo reale, interagendo continuamente con ASTRA, calcolando continue simulazioni di discesa basate sui dati in realtime. App usata principalmente per apportare le modifiche in questo progetto e vederne i risultati durante una missione.
- Simulation Visualizer : WebApp simile alla precedente. Praticamente una GUI creata per ASTRA e permettere all'utente di eseguire e vedere numerose simulazioni sulla mappa.

# 6.4 Interfaccia online

Come anticipato nelle righe sopra, il simulatore di volo si avvale anche, se non soprattutto, di un'interfaccia online utile ad automatizzare il processo di visualizzazione delle traiettorie simulate. Di seguito viene fornita una analisi sulle feature accessibili dal planner e il loro funzionamento.

## 6.4.1 Planner

La sottostante schermata del tool riguarda il "Planner" dell'app "Stratologic"



Figura 6.12. Interfaccia online Stratologic

Da qui si può pianificare la missione inserendo i dati richiesti e/o voluti, come ad esempio la nozzle lift, la data di lancio e la posizione, oltre il modello di paracadute e di pallone. Sempre su questa schermata comparirà, poi, la traiettoria simulata calcolata. Per il funzionamento di questo tool si fa riferimento, oltre ai codici html e css necessari per la configurazione di web-design, al modulo "simulation\_server.js" scritto in JavaScript, visibile in Fig.[6.13].



Figura 6.13. Particolare del codice JavaScript che regola il funzionamento del planner e la chiamata della simulazione con i dati immessi

Tale codice è in seguito stato cambiato, insieme al modulo html relativo al design dell'inerfaccia web, per inserire delle caselle di scelta per l'interpolazione dei dati di cui si parla nel Cap.[7].

Viene quindi di seguito proposta la schermata aggiornata del planner in cui figurano tre modalità di scelta in più:

- Iteration mode: per scegliere quale tipo di media utilizzare per i dati di vento interpolati e GFS
- Do you want to interpolate ascent winds?: per scegliere se interpolare o meno i dati di vento raccolti durante il volo
- Do you want to use the newer simulation?: per scegliere se utilizzare anche la media dei dati di vento interpolati e GFS



Figura 6.14. Interfaccia Planner aggiornata

# 6.4.2 Pre-flight tool

Questo tool, sempre accessibile su Stratologic, chiamato "Pre-flight tool" è basato sul modulo qui sotto riportato.

|  |   | Balloon Burst Calculator |
|--|---|--------------------------|
| Payload Mass (g)<br>2850 C<br>Balloon Model<br>(HW1200 V | Minimum Burst Altitude (m)<br>Target Ascent Rate (m/s)<br>s |                          |
| Sut  | omit  |                          |

Figura 6.15. Preflight tool

Questo è utile per calcolare la nozzle lift necessaria a soddisfare le richieste di payload e velocità di ascesa, ottenendo in outpu un risultato come in Fig.[6.16].

|                        |               |                     |                        |   | Balloon Burst C | alculator |
|------------------------|---------------|---------------------|------------------------|---|-----------------|-----------|
|                        |               |                     |                        |   |                 |           |
|                        |               |                     |                        |   |                 |           |
| Payload Mass (g)       |               | Minimum Bu          | urst Altitude (m)      |   |                 |           |
| 2850                   | 0             | 10000               |                        |   |                 |           |
| Balloon Model          |               | larget Asce         | nt Rate (m/s)          |   |                 |           |
| HW1200                 | v             | 5                   |                        | v |                 |           |
|                        |               |                     |                        |   |                 |           |
|                        | Sub           | mit                 |                        | _ |                 |           |
| Nozzle lift (g)        | Balloon volum | e (m <sup>3</sup> ) | Balloon gas mass (g)   |   |                 |           |
| 4124                   | 5.6           | 240                 | 1213                   |   |                 |           |
| Final speed RCK5 (m/s) | Final speed   | RCK6 (m/s)          | Final speed RCK7 (m/s) |   |                 |           |
| 7.39                   | 6.4           | 44                  | 5.35                   |   |                 |           |

6.4 – Interfaccia online

Figura 6.16. Output preflight tool

## 6.4.3 Tracker

Infine, l'ultima sezione dell'app, necessaria per visualizzare sia le traiettorie simulate nel Planner sia le posizioni ricevute in tempo reale dal "TrackUINO" montato a bordo del pallone.



Figura 6.17. Tracker Stratologic

Come si nota da Fig.[6.17], spunta a sinistra la scritta "No CubeHAPS connected". Infatti, non vi è obbligo di segnale affinchè il tracker possa essere utilizzato, rivelandosi un ottimo strumento per le predizioni pre-lancio e per tutte le operazioni offline, in particolare per la scelta del giorno per una eventuale missione. Proprio da quest'ultimo punto partirà la discussione affrontata nel Cap.[8], per la scelta del giorno ottimale per il lancio di ottobre 2023, previsto al momento della stesura di questo documento per la prima metà del mese ma soggetto a possibili variazioni per impegni di lavoro con clienti esterni per l'azienda Stratobotic.

# Capitolo 7

# Metodologia di lavoro

In questo capitolo si affrontano nel dettaglio le modifiche apportate al simulatore e il motivo per cui si sono rese necessarie.

Il lavoro è stato diviso principalmente in due macro-aree: Discesa e salita, volutamente affrontati in quest'ordine.

# 7.1 Problematiche

Il motivo fondamentale di questa tesi, come già precedentemente esposto è quello di migliorare il sistema predittivo del simulatore, agendo sui fattori che già Sobester [28] aveva messo in evidenza nel documento in bibliografia, ovvero la necessità di migliorare la traiettoria di discesa implementando dati di vento reali, e valutando meglio il coefficiente di resistenza, agendo di fatto sulla nozzle lift.

Nelle sezioni successive sarà più chiaro il metodo con cui quest'ultima modifica è stata affrontata.

# 7.2 Moto di discesa

Il moto di discesa è chiaramente influenzato non più da parametri come la nozzle lift o la massa del gas di gonfiaggio in quanto, da ipotesi del simulatore, il pallone ha raggiunto il bursting point ed è esploso. Il problema principale rilevato durante la discesa è la difficoltà di predire con assoluta accuratezza la traiettoria poichè oltre una certa altezza i dati del GFS non sono del tutto accurati. Ecco che allora, come prima modifica sostanziale, è stato ipotizzato un range di altezze in cui l'unico dato di vento considerato fosse quello rilevato in ascesa dal tracker GPS arduino (cosiddetto "TRACKUino"). Per poter far ciò è necessario, dopo aver registrato le posizioni istante per istante, conservare i dati in un

file .csv e interpolare i dati così da poterli manipolare e trasformare nello stesso formato del vento estrapolato dai dati GFS.

Di seguito viene mostrata la parte di codice in cui si richiama l'interpolazione dei venti con il flag creato appositamente "gpsWind".



Figura 7.1. Flag gpsWind

Il calcolo è reso possibile da un'ulteriore function definita in "environment" di "weather.py" di ASTRA, ovvero getwind, mostrata nella pagina seguente in Fig.[7.2]



Figura 7.2. Struttura getwind

L'interpolazione adottata per i venti rilevati è una semplice interpolazione lineare di primo grado, come visibile in Fig.[7.3]

```
def calculateInterpFunc(interpFlag):
   """Returns a list containg two functions which interpolate
   windLat and windLon according to the altitude.
   interpFlag is a flag for interpolation"""
   here = os.path.dirname(os.path.abspath( file ))
   filenameWind = os.path.join(here, '../examples/ascent_data', 'winds_measured.csv')
   elevVec = []
   windLatVec = []
   windLonVec = 🚺
   if interpFlag:
       with open(filenameWind, "r") as fileWind:
           line = fileWind.readline() #skip heading
           for line in fileWind:
               splittedLine = line.split(",")
               elevVec.append(float(splittedLine[0]))
               windLatVec.append(float(splittedLine[1]))
               windLonVec.append(float(splittedLine[2]))
       # fill value=extrapolate is helpful for evaluate values above
       windLatFunc = interp1d(elevVec,windLatVec,fill value="extrapolate")
       windLonFunc = interp1d(elevVec,windLonVec,fill value="extrapolate")
       windLatFunc = None
       windLonFunc = None
   return [windLatFunc,windLonFunc]
```

Figura 7.3. Interpolazione

Questa scelta è stata effettuata dopo aver eseguito diverse prove con altri interpolatori built-in della libreria scipy come: interpolatore Akima ("Akima1dInterpolator"), una classica spline cubica e la UnivariateSpline, interpolatori che vengono approfonditi in App.[B].

Effettivamente i suddetti interpolatori non hanno dato assolutamente nessuna miglioria rispetto all'interpolatore lineare, tanto da appesantire soltanto il calcolo, rallentare la simulazione e non dare neanche l'opportunità di mettere in evidenza le differenze.

#### 7.2.1 Media dei dati GFS e interpolati

Fatta la prima modifica, e notato anche un buon miglioramento per la missione del 11/08/2023, non si può dire lo stesso per la missione del 19/10/2022, anzi l'errore nel punto di atterraggio sembra esplodere.

Per questo motivo, la soluzione messa in atto è stata quella di mediare i dati GFS e quelli interpolati, ovvero farne una media per un certo range di altezze. Questa decisione è dovuta in gran parte a ciò che è stato esposo nel Cap.[5], ovvero l'influenza dell'orografia del sito di lancio sui venti locali. Infatti, come nella soluzione senza media dei dati, si ipotizza inizialmente un'altezza standard fissata a 3000 m in cui si inizia questa sorta di fusione dei dati per evitare che proprio a questa altezza avvenga un distacco netto che porti a errori esagerati. La prima soluzione implementata è stata quella di fare una media pesata in un range di 500 m a scalare ogni 100 m, così programmata:



Figura 7.4. Media pesata a step di 100 m in un range di 500 m

La soluzione non sembrava dare migliorie evidenti, ma in realtà svolge al meglio il suo ruolo: media gli errori e rende il simulatore leggermente più affidabile, vista l'imprevedibilità dei dati GFS sulla traiettoria. L'aver lavorato su due missioni così diverse fra loro, infatti, ha permesso di fare un'analisi ai due casi estremi e opposti in cui il GFS prima si rivela il miglior strumento (missione 11/08/2023) e poi la fonte maggiore di errore

(missione 19/10/2022). La soluzione mediata rappresenta un'ottima stima a metà. Ma questo comunque non basta, come visibile dalla grafica di Google Earth in Fig.[7.5] e Fig.[7.6], visto il poco distaccarsi dal dato interpolato, agendo effettivamente in un range di altitudini ristretto rispetto a quelle raggiunte dal pallone.



Figura 7.5. Atterraggio previsto ed effettivo missione 11/08/2022



Figura 7.6. Atterraggio previsto ed effettivo missione 19/10/2022

Per questo motivo, si è dapprima inizializzato un flag (newsim) per richiamare la media dei venti e, poi, prima di entrare nel ciclo for di simulazione di posizione, si è inserita una function esterna al simulatore per rilevare l'altitudine maggiore del quadrante in cui si lancia. Prima di vedere nel dettaglio suddetta function, bisogna giustificare la scelta riprendendo la questione dell'orografia e ipotizzando che non può esistere un'altezza uguale per tutti i tipi di terreno per cui l'influenza dei venti locali non incide più e le correnti macroscopiche hanno la meglio. Esempio: se lanciassimo da un'altitudine di 1700 m ma ci trovassimo in una vallata in cui le montagne che ci circondano raggiungessero altezze di 4000-5000 m, viene difficile pensare che i venti non vengano influenzati da tale conformazione.

Ed è in questo scenario che è stata scritta ed implementata la function "get\_highest\_elevation" all'interno di "flight\_tools.py" mostrata in Fig.[7.7]:



Figura 7.7. Function get\_highest\_elevation

Il funzionamento della function di per sè è abbastanza intuitivo e semplice. Tramite l'uso della libreria "rasterio", la funzione riesce a leggere un file GeoTIFF di un quadrante di Terra.

Il GeoTIFF è uno standard di metadati di dominio pubblico che consente l'incorporazione di informazioni di georeferenziazione all'interno di un file TIFF. Queste informazioni aggiuntive possono includere parametri come proiezioni cartografiche, sistemi di coordinate, ellissoidi, datums e tutti gli elementi necessari per stabilire con precisione il riferimento spaziale esatto del file. Fondamentalmente una foto. La potenzialità di rasterio e di conseguenza della function sta nel saper leggere questa "foto in altezza"; ovvero, una volta aperto in "read" il file si divide il quadrante in piccoli pezzettini al variare della dimensione del pixel e di un input dato da linea di comando che riguarda la lunghezza del lato del quadrato di file da esaminare. Una volta definiti i lati del quadrato, trova il centro e ne calcola gli angoli che saranno i vertici di una sorta di matrice di controllo. Definiti i pezzettini della matrice, la funcion legge le altezze relative e immagazzina i dati in un array. Infine, con l'ausilio della funzione max di numpy, trova il valore massimo del quadrate.

Definito il metodo della function, si mostra qui di seguito come è stata richiamata ed utilizzata per la mediazione dei dati:



Figura 7.8. Calcolo self.iterAlt

Analizzando meglio, si apre un file csv di traiettoria simulata, viene letta l'ultima riga di posizione per avere latitudine e longitudine, stimate, del punto di atterraggio e si richiama la function sopra esposta. Da qui, deriva il parametro iterAlt che, come visto nella Fig.[7.4], viene usato come variabile di controllo per entrare nel "if" del calcolo dei venti mediati.

Per dare anche una stima grafica della miglioria, è stato scritto un modulo esterno standalone chiamato "error\_estimation\_analysis.py" che chiama la simulazione dei tre casi differenti (Fig.[7.9]) qui elencati:

- 1. simulazione traiettoria con solo dati GFS
- $2.\,$ simulazione traiettoria con dati mediati
- 3. simulazione traiettoria con stacco netto tra dati interpolati e GFS

e ne mette a confronto le traiettorie stimate rispetto al vero punto di atterraggio, grazie a un altro modulo esterno "error\_estimation.py".



Figura 7.9. Simulazione dei tre casi

L'output di questa analisi è un grafico, tra gli altri, di questo tipo:



Figura 7.10. Confronto errore traiettorie

Il grafico va letto da destra verso sinistra, ovvero al calare dell'altitudine posta sull'asse delle ascisse, visto che stiamo parlando del moto di discesa. Sull'asse delle ordinate, invece, vi è l'errore orizzontale, ovvero di quanto si è "sbagliata" la simulazione nel predire il punto di atterraggio.

Evidentemente, il GFS forecast si è rivelata una simulazione leggermente più affidabile rispetto ai dati interpolati e mediati. Si può anche notare che, verso un'altitudine di circa 3000 m, la curva arancione che rappresenta i dati mediati, si discosta da quella verde dei soli dati di vento in ascesa interpolati, segno che la media pesata sia andata a buon fine. Nel particolare, l'altitudine a cui inizia la media dei dati di circa 2877 m, come rilevato dalla function sopra citata nel quadrante di Terra in cui è avvenuto il lancio.

Questo risultato è in linea con le aspettative visto che si tratta di un'interpolazione molto semplice.

Per infoltire il contesto, di seguito, viene mostrato il grafico di errore relativo alla missione del 19/10/2022 in Fig.[7.11]:



Figura 7.11. Confronto errore traiettorie 19/10/2022

In questo caso, invece, il GFS forecast si è rivelato fonte di errore enorme, mentre i dati inerpolati e mediati sono sembrati più affidabili.

Per questo motivo, sembra necessaria la media dei dati, ovvero per dare anche una buona stima ed avere una sorta di safe zone non troppo variabile in base alle previsioni. Risulta infatti fondamentale, per l'appetibilità del servizio fornito, la capacità di recupero del pallone senza correre rischi di sorta o perdite di tempo nella ricerca dello stesso. Pur sembrando questi dei buoni risultati, il vero punto fondamentale della modifica è stato quello di scegliere e integrare altri tipi di medie algebriche. Qui di seguito verranno, perciò, analizzati i risultati di entrambe le missioni con i vari tipi di medie elencati e i codici sviluppati per il calcolo. Per far ciò si è inizializzato un flag denominato "itermode" che' al variare del numero associato ad esso da 1 a 5, richiama il metodo di media; in particolare:

- 1. Media pesata a "scalini"
- 2. Media cubica
- 3. Media armonica
- 4. Media quadratica
- 5. Media geometrica

#### Media geometrica

La seconda soluzione ad essere stata implementata, dopo quella a step, con il seguente codice (Fig.[7.12]) nel modulo "mean\_tools":

```
def geometric_mean(array):
"""Calculates the geometric mean of an array of values.
Args:
 array: The array of values.
Returns:
 The geometric mean of the array.
if all(value >= 0 for value in array):
 return np.prod(array) ** (1 / len(array))
elif all(value <= 0 for value in array):</pre>
  positive values = [abs(value) for value in array]
  return -np.prod(positive_values) ** (1 / len(positive_values))
else:
      positive values = []
      negative_values = []
      for value in array:
          if value >= 0:
              positive values.append(value)
              negative values.append(value)
      positive_mean = np.prod(positive_values) ** (1 / len(positive_values))
      negative_mean = -np.prod(negative_values) ** (1 / len(negative_values))
      positive weight = len(positive values) / len(array)
      negative_weight = len(negative_values) / len(array)
      return positive_mean * positive_weight + negative_mean * negative_weight
```

Figura 7.12. Codice per media geometrica

è stata la media geometrica.

In realtà esistono numerose librerie in grado di fornire function capaci di calcolare questo tipo di media ma, come la teoria impone, sono tutte definite positive. I dati in nostro possesso, invece, sono anche negativi perchè, per l'appunto, parliamo di un moto di discesa. Perciò si è resa necessaria questa manipolazione per poter trattare i nostri dati con questi strumenti matematici che, come si vedrà, risulteranno utili.

Nel simulatore, la media geometrica dei dati viene richiamata così come segue:



Figura 7.13. Chiamata media geometrica

Di seguito invece, in Fig.[7.14] e Fig.[7.15], vengono mostrati, rispettivamente, gli errori della missione del 11/08/2023 e del 19/10/2022.



Figura 7.14. Confronto errore traiettorie 11/08/2023 mediate geometricamente



Figura 7.15. Confronto errore traiettorie 19/10/2022 mediate geometricamente

Risulta evidente che, al variare delle missioni, varia il risultato. Infatti per la missione dell'ottobre dell'anno scorso la media geometrica sembra essere un'ottima soluzione per ottere un dato quantomeno affidabile (dato mediato rappresentato sempre da curva arancione). Nella missione dello scorso agosto, invece, è un disastro, perchè non solo non riesce ad essere una soluzione di mezzo, ma addirittura aumenta sensibilmente l'errore del punto di atterraggio. Per poter dare ulteriore contesto al confronto, vengono mostrate di seguito delle schermate (Fig.[7.16] e Fig.[7.17]) ottenute con l'ausilio dell'app desktop Google Earth.

#### 7.2-Motodi discesa



Figura 7.16. Atter<br/>raggio previsto ed effettivo missione11/08/2023 con dati interpolati per via geometrica



Figura 7.17. Atter<br/>raggio previsto ed effettivo missione19/10/2022 con dati interpolati per via geometrica

### Media armonica

La terza soluzione ad essere stata implementata, con il seguente codice (Fig.[7.18]) nel modulo "mean\_tools":

```
def harmonic mean(array):
 """Calculates the harmonic mean of an array of values.
 Args:
  array: The array of values.
 Returns:
  The harmonic mean of the array.
 if all(value \geq 0 for value in array):
   reciprocal_sum = sum(1 / value for value in array)
   n = len(array)
   return n / reciprocal sum
 elif all(value <= 0 for value in array):</pre>
   positive values = [abs(value) for value in array]
   reciprocal sum = sum(1 / value for value in positive values)
   n = len(positive_values)
   return -abs(n / reciprocal sum)
 else:
   positive values = []
   negative values = []
   for value in array:
       if value >= 0:
           positive values.append(value)
       else:
           negative values.append(value)
   reciprocal sum pos = sum(1 / value for value in positive values)
   n = len(positive values)
   reciprocal sum neg = sum(1 / value for value in negative values)
   m = len(negative_values)
   positive_mean = n / reciprocal_sum_pos
   negative mean = -abs(n / reciprocal sum neg)
   positive weight = len(positive values) / len(array)
   negative_weight = len(negative_values) / len(array)
   return positive mean * positive weight + negative mean * negative weight
```



è stata la media armonica.

Anche in questo caso esistono numerose librerie in grado di fornire function capaci di calcolare la media armonica ma, come anche in questo caso la teoria impone, sono tutte definite positive. I dati in nostro possesso, invece, essendo anche negativi ci impongono un'ulteriore manipolazione per poter trattare i nostri dati con questo tipo di media. Nel simulatore, la media armonica dei dati viene richiamata così come segue:

| if | self.iterMode == 3:  |
|----|--|
|    | wind_array_speeds = [float(wind_value) for wind_value in wind_array]   |
|    | gfs_array_speeds = [float(gfs_value) for gfs_value in gfs_array]   |
|    | array1 = [wind_array_speeds[0],gfs_array_speeds[0]]  |
|    | array2 = [wind array speeds[1],gfs array speeds[1]]  |
|    | <pre>wind_extrapolated = [mt.harmonic_mean([wind_array[0],gfs_array[0]]),mt.harmonic_mean([wind_array[1],gfs_array[1]])]</pre> |
|    |  |

Figura 7.19. Chiamata media armonica

Di seguito invece, in Fig.[7.20] e Fig.[7.21], vengono mostrati, rispettivamente, gli errori della missione del 11/08/2023 e del 19/10/2022.



Figura 7.20. Confronto errore traiettorie 11/08/2023 mediate armonicamente



Figura 7.21. Confronto errore traiettorie 19/10/2022 mediate armonicamente

Anche in questo caso, al variare delle missioni, varia il risultato. Ma, rispetto alla media geometrica, la media armonica sembra essere molto più affidabile, rendendo l'errore della missione dell'agosto di quest'anno minore: un sensibile miglioramento rispetto alla media precedente. Per la missione dell'ottobre dell'anno scorso, invece, nessun miglioramento rilevante.



Anche qui viene proposta una visione 3D della questione in Fig. [7.22] e Fig. [7.23].

Figura 7.22. Atterraggio previsto ed effettivo missione11/08/2023 con dati interpolati per via armonica



Figura 7.23. Atter<br/>raggio previsto ed effettivo missione19/10/2022 con dati interpol<br/>ati per via armonica

#### Media quadratica

La quarta soluzione ad essere stata implementata, con il seguente codice (Fig.[7.24]) nel modulo "mean\_tools":

```
def quadratic mean(array):
 """Calculates the quadratic mean of an array of values.
 Args:
  array: The array of values.
 Returns:
  The quadratic mean of the array.
 ......
 if all(value >= 0 for value in array):
 return np.sqrt(np.mean(np.power(array,2)))
 elif all(value <= 0 for value in array):</pre>
   positive values = [abs(value) for value in array]
   return -abs(np.sqrt(np.mean(np.power(positive values,2)))
 else:
   positive values = []
   negative values = []
   for value in array:
       if value >= 0:
           positive values.append(value)
       else:
           negative_values.append(value)
   positive mean = np.sqrt(np.mean(np.power(positive values,2)))
   negative_mean = -abs(np.sqrt(np.mean(np.power(negative_values,2))))
   positive weight = len(positive values) / len(array)
   negative weight = len(negative values) / len(array)
   return positive_mean * positive_weight + negative_mean * negative_weight
```

Figura 7.24. Codice per media quadratica

è stata la media quadratica.

Come per le altre due medie precedentemente esposte, la manipolazione del codice si è resa necessaria perchè, anch'essa, definita positiva .

Nel simulatore, la media quadratica dei dati viene richiamata così come segue:



Figura 7.25. Chiamata media quadratica

Di seguito invece, in Fig.[7.26] e Fig.[7.27], vengono mostrati, rispettivamente, gli errori della missione del 11/08/2023 e del 19/10/2022.



Figura 7.26. Confronto errore traiettorie 11/08/2023 mediate quadraticamente



Figura 7.27. Confronto errore traiettorie 19/10/2022 mediate quadraticamente

Se in questo caso, per la missione dello scorso agosto abbiamo un comportamento affidabile come quello della media armonica, per la missione dell'otobre 2022 abbiamo, addirittura un miglioramento globale, rendendo chiaro che questa può essere la migliore scelta nel mediare i dati.



Anche qui viene proposta una visione 3D della questione in Fig. [7.28] e Fig. [7.29].

Figura 7.28. Atter<br/>raggio previsto ed effettivo missione11/08/2023 con dati interpolati<br/> per via quadratica



Figura 7.29. Atter<br/>raggio previsto ed effettivo missione19/10/2022 con dati interpol<br/>ati per via quadratica

### Media cubica

La quinta ed ultima soluzione ad essere stata implementata, con il seguente codice (Fig.[7.30]) nel modulo "mean\_tools":

```
def cubic mean(array):
 """Calculates the cubic mean of an array of values.
 Args:
  array: The array of values.
 Returns:
  The cubic mean of the array.
 .....
 if all(value >= 0 for value in array):
  return np.cbrt(np.mean(np.power(array,3)))
 elif all(value <= 0 for value in array):</pre>
   positive_values = [abs(value) for value in array]
   return -abs(np.cbrt(np.mean(np.power(positive values,3))))
 else:
   positive values = []
   negative values = []
   for value in array:
       if value >= 0:
           positive values.append(value)
       else:
           negative_values.append(value)
   positive mean = np.cbrt(np.mean(np.power(positive values,3)))
   negative_mean = -abs(np.cbrt(np.mean(np.power(negative_values,3))))
   positive weight = len(positive values) / len(array)
   negative_weight = len(negative_values) / len(array)
   return positive_mean * positive_weight + negative_mean * negative weight
```

Figura 7.30. Codice per media cubica

è stata la media cubica.

In questo caso la manipolazione non è stata necessaria perchè definita positiva, ma semplicemente per poter fare un blend di dati che potessero presentarsi sia positivi che negaivi
essendo molto vicini all'ordine dei decimali.

Nel simulatore, la media cubica dei dati viene richiamata così come segue:



Figura 7.31. Chiamata media cubica

Di seguito invece, in Fig.[7.32] e Fig.[7.33], vengono mostrati, rispettivamente, gli errori della missione del 11/08/2023 e del 19/10/2022.



Figura 7.32. Confronto errore traiettorie 11/08/2023 mediate cubicamente



Figura 7.33. Confronto errore traiettorie 19/10/2022 mediate cubicamente

Ed ecco che, infine, si rende manifesta la soluzione migliore. Perchè non solo, come quella quadratica, migliora globalmente l'errore nella missione del 2022, ma lo fa anche nella missione dello scorso agosto, dove a farla da padrona erano i dati GFS. Anche per la soluzione migliore, viene forita una visione 3D della miglioria su Google Earh in Fig.[7.34] e Fig.[7.35].



Figura 7.34. Atterraggio previsto ed effettivo mission<br/>e11/08/2023 con dati interpolati per via cubica



Figura 7.35. Atter<br/>raggio previsto ed effettivo missione19/10/2022 con dati interpolati p<br/>er via cubica

Per altre grafiche in 3D generate con l'ausilio di Google Earh si rimanda a App.[C]. Per questo motivo e per le considerazioni viste finora, come verrà suggerito in calce al simulatore con un semplice commento e con magari un avviso nella GUI "Stratologic", si raccomanda l'uso futuro della media cubica per quella che è la soluzione della prima macro-area di questa tesi.

### 7.3 Moto di ascesa

Il problema del moto di ascesa è stato affrontato solo in un secondo tempo, per dare spazio alle modifiche ritenute più "importanti".

Ciò non significa che la modifica al simulatore nel moto di ascesa sia banale, anzi, forse, si è rivelata la più complessa.

Infatti, come spiegato nei capitoli precedenti, il moto di discesa viene calcolato proprio a partire dal moto di ascesa e dei venti rilevati durante questa fase. Per questo motivo, si è pensato ad una soluzione che prendesse in considerazione più scenari possibili. Infatti, dopo vari colloqui con i responsabili del simulatore, si è messo in evidenza il fatto che, spesso, la nozzle lift reale fosse diversa di qualche punto percentuale da quella calcolata dal preflight tool. Non potendo agire su un calcolo già assodato come quello del preflight tool, si è deciso di agire proprio su un confronto in fase di ascesa tra traiettorie simulate e quella reale. Ma non solo, perchè dato il problema della nozzle lift, le traiettorie simulate sono state molteplici.

### 7.3.1 Variazione della Nozzle Lift

Innanzitutto, si è reso necessario lo sviullpo di un modulo che andasse a calcolare e salvare in locale le simulazioni al variare della nozzle lif nominale. Infatti, come si può evincere dalla Fig.[7.36], dopo aver preso in input la Nozzle Lift nominale, ovvero quella data in output dal preflight tool, questa viene variata del 10% in positivo e in negativo (una sorta di safet factor) ed successivamente utilizzata per il calcolo di altre simulazioni per ogni variazione. A differenza della simulazione nominale, però, queste ultime verrano semplicemente immagazzinate in locale senza essere mostrate sul planner e sul map visualizer per non appesantire il costo computazionale della simulazione stessa.



Figura 7.36. Function definita per salvare in locale le simulazioni con le nozzle lift variate

Come si vede dalla figura qui riportata, compaiono delle variabili denominate "sys.argv"; queste non sono nient'altro che le variabili passate da linea di comando a partire dal planner online, mostrato già in Fig.[6.12] di Sez.[6.4.1].

#### 7.3.2 Stima errore e scelta migliore simulazione

Una volta che le varie simulazioni sono state salvate, tramite appositi file .csv, bisogna confrontarle con quella che dovrebbe essere la vera traiettoria del pallone. Chiaramente, per effettuare ciò, bisognerebbe trovarsi nell'ambito di una missione, ma non avendo avuto la possibilità di testare sul campo, si è reso necessario lo sviluppo di un modulo esterno al tool online in grado di confrontare le traiettorie. Ciò si è reso possibile con il modulo "Best\_Simulation.py" il cui nucleo principale visibile di seguito in Fig.[7.37]:

```
main():
here = os.path.dirname(os.path.abspath(__file__))
jsonfileName = os.path.join(here, 'post_processing_input.json') # To be replaced by preflight-tool json output file
      with open(jsonfileName, 'r') as input:
        data = input.read()
      inputDict = json.loads(data)
      datestr = inputDict['Launch_Date_String']
      nozzleLift = inputDict['Nozzle_Lift']
      Nozzle_lifts = {0.8 * nozzleLift , 0.9 * nozzleLift , nozzleLift , 1.1 * nozzleLift , 1.2 * nozzleLift}
real_data_filename = os.path.join(here,'..','..','astra_output','out.csv') # da modificare con file csv da tracker GPS
for nl in (Nozzle_lifts):
 strName = 'NL_'+str(nl).replace('.','_')+'.csv'
sim_data_filename = os.path.join(here , '..' , '..'
                                                           'astra output'
                                                                           , strName)
 error.append(compare_trajectory_velocities(sim_data_filename , real_data_filename))
  j += 1
best simulation index = error.index(min(error))
print('La migliore simulazione è la numero {}'.format(best_simulation_index))
```

Figura 7.37. Main di Best\_simulation.py

Le prime righe del codice servono a ottenere il percorso locale in cui sono salvate le impostazioni di lancio. Successivamente queste stesse impostazioni vengono usate per il lancio delle simulazioni, e quindi della function nlvs vista in Fig.[7.36]. Una volta salvate, le simulazione vengono aperte in modalità "read" per essere esaminate con apposita function all'interno di Best\_simulation.py. Stessa cosa avviene con il file .csv contenente la traiettoria reale del pallone. Una volta che tutti i file sono aperti e le coordinate vengono estrapolate, si passa alla ricerca della migliore simulazione.

Per questo task, si è fatto uso di una semplice differenza in valore assoluto tra le velocità medie in ascesa del pallone delle varie simulazioni con quella reale. Inoltre, si ricorre all'errore quadratico medio, con apposita function, per avere una stima globale migliore dell'errore. Calcolati gli errori, questi vengono mostrati nel terminale in valore assoluto, così da poter estrapolare la migliore. Chiaramente questo modulo è stato ottimo per le analisi in locale, mentre per quanto riguarda le analisi sul campo ne è stato definito un altro, praticamente uguale, in cui le variabili impostate manualmente vengono rimpiazzate dalle variabili "sys.argv".

Per dare contesto al funzionamento di questo modulo, si mostra in Fig.[7.38] l'output mostrato sul terminale dal modulo dopo aver effettuato il confronto tra le traiettorie.

#### /honeycomb-main/astra\_simulator\_stratobotic/error\_estimation\_study/Best\_simulation.py La migliore simulazione è la numero 2

Figura 7.38. Commento in output del modulo "Best\_simulation.py"

Nel caso in particolare, questa frase si riferisce allo studio effettuato per la missione del 19/10/2022 in cui vengono confrontati i tre casi visti nella sez.[7.2]. Coerentemente con quanto visto nei grafici, è la simulazione con i dati di vento mediati con media cubica a risultare la più accurata, dando così l'evidenza della correttezza biunivoca del modulo "Best\_simulation.py" e della media cubica dei dati reali e interpolati.

Stesso ragionamento è stato applicato proprio a cinque simulazioni in cui a variare fosse la nozzle lift nominale, impostata la media dei dati cubica di default. Risultato interessante, sempre per la missione del 19/10/2022, quello dato dal modulo, perchè nonosante la simulazione col valore nominale di nozzle lift sembrasse abbastanza precisa, il valore più vicino alla realtà (e quindi la traiettoria più precisa) è risultato essere quello variato del 10% in negativo. Ciò fa pensare che la scelta di simulare per nozzle lift diverse sia un controllo non solo corretto ma obbligato dall'esigenza di dover affinare la capacità predittiva del simulatore. In pratica si tiene conto delle fluttuazioni e delle variazioni improvvise della nozzle lift che non sono simulabili e/o prevedibili, andando ad agire sul controllo periodico del moto in ascesa del pallone.

Al momento il controllo è un banale confronto tra le velocità in ascesa rilevate e quelle simulate, ma tanto basta per avere già una miglioria evidente.

Oltre la semplice frase di avviso su quale sia la simulazione migliore, però, vi è la necessità di operare con la traiettoria migliore ed i suoi dati; ed è per questo che il modulo mantiene la simulazione migliore in memoria per poi usare in seguito i dati di vento della stessa per fare la media in fase di discesa del payload, una volta che il pallone sarà scoppiato.

## Capitolo 8

## Lancio

Verso la fine del progetto di tesi, si è presentata l'occasione per lo svolgimento di una missione su campo. Perciò, come richiesto dall'azienda, è rientrato nei task di questo progetto la scelta del possibile giorno di lancio del pallone stratosferico. Per fare ciò, sono state necessarie numerose simulazioni nell'arco di tempo di tre giorni prescelti in base ai colloqui con ENAC e, soprattutto, in base alle condizioni meteo.

Infatti, per definire la missione un successo, è necessario che le immagini raccolte siano tali da poter ottenere elevata risoluzione ed ampio raggio. Risulta ovvio pensare a delle giornate di sole prive di qualsivoglia nuvola o fenomeno atmosferico avverso, per poter lanciare il pallone alla ricerca di immagini della Terra sottostante.

Dopo appositi colloqui con ENAC, valutazione approfondita delle condizioni meteo ed anche disponibilità materiale della strumentazione, l'analisi dettagliata si è concentrata sui giorni 11/10/2023, 12/10/2023 ed infine 13/10/2023. Dalle previsioni meteo, tre giornate ottimali per una buona riuscita della missione, ma la qualità delle immagini non è l'unica cosa che conta.

### 8.1 Sicurezza

Pur parlando di un payload "esiguo" (circa 2.85 kg, comprendente tutta la strumentazione elettronica di bordo e la videocamera), bisogna pur sempre tenere conto del fatto che il payload deve pur atterrare da qualche parte. Non essendo un velivolo manovrabile, e non potendo prevedere al millimetro il landing point, è stato punto fondamentale dell'analisi ricercare una zona sicura in accordo coi dati storici di errore del simulatore di volo e dunque di predizione del punto di atterraggio. A tal proposito visto l'errore massimo di circa 700 m (ipotizzando una circonferenza con tale raggio), dopo aver fatto tutte le simulazioni del caso è stata prevista la visualizzazione a schermo di una "safe zone" con

centro nel landing point previsto e raggio 1 km (considerando una sorta di safety factor di circa 1,5). Di conseguenza verranno mostrate di seguito immaginei riepilogative di tutte le simulazioni effettuate ai diversi orari per ogni giornata presa in considerazione per poi commentarle con la scelta effettuata. Oltre Fig.[8.1],Fig.[8.2] e Fig.[8.3] sono disponibili in App.[C] tutte le visualizzazioni grafiche su google earth di ogni singola simulazione,



Figura 8.1. Simulazioni 11/10/2023



Figura 8.2. Simulazioni 12/10/2023



Figura 8.3. Simulazioni 13/10/2023

Viste tutte le considerazioni fatte sul tipo di terreno in cui sarebbe atterrato il payload e la safe zone relativa, oltre le analisi sui venti e la durata di volo, è stata scelta la data del 12/10/2023. Di seguito in Fig.[8.4] vengono mostrate tutte le safe zone per ogni orario e per ogni variazione di nozzle lift ( $\pm 10\%$ ).

Appare evidente che tutte le zone di atterraggio del 12/10/2023 siano abbastanza sicure trattandosi di campagne e/o simili. Da questo punto, poi, è partita la disamina sull'orario. Perchè se le zone di sicurezza vanno bene, l'analisi si sposta sulla qualità delle immagini catturate dal payload.



Figura 8.4. Safe zones 12/10/2023

Infatti è chiaro convenga avere una buona esposizione solare ma è altrettanto vero che le ombre sul terreno renderebbero di difficile comprensione le immagini acquisite da elaborare successivamente. Per questo motivo viene scelto un orario il più possibile vicino a mezzogiorno, in modo che i raggi del sole siano perpendicolari al terreno e le ombre che si formano siano minime, in modo tale da non intaccare la buona riuscita della missione. Chiaramente, di fondamentale importanza che il cielo sia terso e privo di nuvole per evitare di acquisire foto spezzettate o disturbate.

Risultato: il giorno e l'orario prestabiliti per il lancio sono il 12/10/2023 intorno alle 12:00.

### 8.2 Setup di missione

Prima di passare all'analisi del tentativo di lancio, in questa sezione verrà messo in mostra il setup tecnico della missione, a partire dal pallone fino alla ground station. Essendo un prototipo innovativo, l'azienda si riserva dal fornire immagini troppo dettagliate, ma le seguenti saranno sufficienti a capire e carpire il funzionamento pratico di una missione tipo.



Figura 8.5. Organizzazione di una missione tipo

In Fig.[8.5], si possono notare tre parti differenti. Da sinistra verso destra: zona di gonfiaggio del pallone, Ground-Station e mission control. Come spiegato in sezioni precedenti, la ground-station è collegata al PC del mission control tramite un semplice cavo usb-to-usb.

#### Lancio



Figura 8.6. Organizzazione di una missione tipo

In Fig.[8.6], invece, viene mostrato un particolare del pallone dopo esser stato gonfiato e pronto per il rilascio. Non viene mostrato il payload, contenente tutta la strumentazione di bordo e collegato al pallone tramite una fune. Questa fune viene poi sganciata nel momento del "Burst" per far sì che il payload torni giù. Non si vede ma si può intuire che il payload sia fornito di paracadute per la discesa.

#### 8.2 – Setup di missione



Figura 8.7. Particolare delle antenne della ground-station

In Fig.[8.7], viene mostrato un particolare delle antenne montate sulla ground-station; queste sono però soltanto la prima versione, in quanto l'idea è di avere in futuro una ground-station molto più grande e fissa e un'altra "portatile" montate sul veicolo di recupero. In Fig.[8.8], invece, un particolare dal basso della configurazione del pallone e della stazione di terra.



Figura 8.8. Particolare delle antenne della ground-station

### 8.3 Giorno del Lancio

Il giorno prestabilito, la missione sarebbe dovuta partire dalla sede di Stratologic a Pont-Saint-Martin, presso l'incubatore Pépinières d'Entreprises[60]. Svolti nuovamente i controlli delle traiettorie con apposite simulazioni, nel frattempo la missione veniva preparata. Il payload, con all'interno la telecamera, il tracker GPS, i motorini di avviamento e le batterie sono stati posizionati all'interno del "cube" che sarebbe in seguito stato collegato tramite delle funi al pallone e al paracadute. Proprio il pallone veniva esaminato e preparato per la "inflation" di Elio con apposita pompa pressurizzata.

Una volta effettuati i pre-test, la ground station composta da antenna, TRACKuino (Arduino[59] atto alla ricezione delle posizioni del GPS) e treppiedi, è stata posizionata nel parcheggio all'esterno dell'incubatore, luogo da cui sarebbe partito in volo il pallone. Purtroppo, prima di procedere con il gonfiaggio e dare il via alla missione, è stata notata una falla nel payload: la ground station non riceveva le posizioni dal tracker. Fortunatamente notato in tempo, l'errore non è stato di facile riconoscimento, visto che il payload era effettivamente sigillato; una volta aperto è stato chiaro che l'errore non fosse altro che un banale cavo scollegato. Risolto il problema, e dopo aver dato una nuova carica al GPSTracker che nel frattempo si era scaricato, la ground station è stata nuovamente posizionata. L'orario di lancio è slittato alle 14:30, ma le condizioni climatiche permettevano ancora la buona riuscita apparente della missione.

Impostato il tutto e pronti per il gonfiaggio, sono stati eseguiti gli ultimi test come in precedenza per far si che tutto andasse per il meglio. Ancora una volta, però, si è presentato un problema, ancora più grosso del precedente perchè non dipeso da errore umano. Infatti, pur funzionando tutto apparentemente, le simulazioni e gli aggiornamenti di traiettoria non venivano visualizzati nel tool online, rendendo impossibile il monitoraggio della missione. Problema notato anche nei giorni precedenti al lancio, è una sorta di blocco preventivo da parte del NOAA che è il fornitore delle previsioni meteo e dei dati del vento. Fondamentalmente, dopo numerose richieste di simulazione sembra mandare in blacklist l'IP richiedente.

Notato il tutto, il tecnico informatico del gruppo ha provato a risolvere il problema mentre nuove simulazioni venivano effettuate per indagare la possibilità di lanciare per le 17:00. Dopo alcune simulazioni e discussioni sulla scelta, il lancio sembrava ancora fattibile, col payload che sarebbe andato a finire nei pressi di Vinzaglio. Il problema col NOAA era stato anch'esso rimosso, perciò non restava altro che procedere all'inizio della missione.

Purtroppo, però, la giornata volgeva alla sera e le nuvole avevano ormai occupato gran parte del terreno che sarebbe stato fotografato, cosa che avrebbe reso la missione uno spreco totale.

Nonostante tutto, un altro tentativo era stato fatto con l'ENAC per il 15/10/2023, ma dopo esito positivo, le previsioni meteo hanno spento l'ultima speranza per il lancio di Ottobre, rimandando il tutto al mese successivo per il testing su campo delle modifiche del simulatore.

## Capitolo 9

## Conclusioni e Sviluppi futuri

Il lavoro svolto in questo progetto ha migliorato effettivamente il modello preditivo del simulatore usato da Stratobotic per le missioni col pallone stratosferico. Ma non è solo questo il vero lavoro, perchè sono state di fatto gettate le basi anche per un lavoro futuro a partire dai dati di vento archiviati utili per costruire un archivio storico e poterne fare un uso prezioso nell'ambito di un'analisi di regressione.

Infatti, è proprio in questo aspetto che il simulatore può essere ulteriormente migliorato, ovvero eliminare la totale dipendenza dai dati GFS per la predizione delle traiettorie e fare invece affidamento a una legge esponenziale storica che possa simulare fedelmente il comportamento del pallone stratosferico. Ma i possibili sviluppi non terminano qui. Oltre a poter lavorare sull'archivio storico, infatti, si propone a chi prenderà in mano il lavoro in futuro, di provare ad agire sul modello termodinamico definito da Sobester[28], evitando di perdere tempo con l'affinamento di calcoli aerodinamici che appesantiscono soltanto il costo computazionale del simulatore senza mostrare grandi migliorie a livello pratico.

Infine, si conclude anche guardando a possibili sviluppi futuri sulla messa in funzione del simulatore ma per voli con i cosiddetti HAPS. Infatti, l'obiettivo di Stratobotic è di riuscire ad ampliare la loro gamma di velivoli atti all'osservazione terrestre, andando ad esplorare il mondo degli High Altitude Pseudo Satellite. Per questo motivo, il miglioramento del simulatore oggetto di questa tesi non è altro che il punto di partenza per la realizzazione di un simulatore di volo per un aeromobile ad ala fissa per voli stratosferici.

## Appendice A

## Note sui gas LTA

I due gas più comunemente utilizzati nei lanci meteorologici come gas di sollevamento sono l'elio monoatomico (He) e l'idrogeno diatomico (H2). Questi gas sono entrambi caratterizzati da una massa molare notevolmente inferiore rispetto all'aria ambiente, con l'elio che presenta una massa molare di circa il doppio rispetto a quella dell'idrogeno. Tuttavia, questa differenza nella massa molare si traduce solo in un vantaggio marginale a favore dell'idrogeno, poiché la quantità di gas utilizzata nei palloni è generalmente ridotta rispetto alla massa complessiva del pallone stesso e del suo carico.

Considerando gas di qualità appropriata, è importante notare che essi possano contenere piccole quantità di aria contaminante, con frazioni di aria  $c_{He}^{air} = 0.055$  e  $c_{H2}^{air} = 0.015$ , rispettivamente [18]. Inoltre, è possibile applicare il modello del gas perfetto (con costante dei gas  $R \simeq 8,31447 \text{m}^3 \text{Pa/molK}$ ) in condizioni standard ISA s.l.m. (pressione  $p_{std} \simeq 101,325 \text{Pa}$ , temperatura  $T_{std} \simeq 288,15 \text{K}$ ) per calcolare le densità dei due gas come segue:

$$\rho_{\rm He} = p_{\rm std} \left[ \left( 1 - c_{\rm He}^{\rm air} \right) M_{\rm He} + c_{\rm He}^{\rm air} M_{\rm air} \right] / RT_{\rm std} = 0.22733 \text{ kg/m}^3 \tag{A.1}$$

е

$$\rho_{\rm H_2} = p_{\rm std} \left[ \left( 1 - c_{\rm H_2}^{\rm arr} \right) M_{\rm H_2} + c_{\rm H_2}^{\rm arr} M_{\rm air} \right] / RT_{\rm std} = 0.10235 \text{ kg/m}^3 \tag{A.2}$$

che restituiscono per un volume  $V = 1 \text{m}^3$  e densità  $\rho_{air} = 1.22479 \text{kg/m}^3$ , valori di portanza libera di:

$$L_{\rm He} = V \left( \rho_{\rm air} - \rho_{\rm He} \right) = 0.99746 \text{ kg}$$
 (A.3)

е

$$L_{\rm H2} = V \left( \rho_{\rm air} - \rho_{\rm H2} \right) = 1.12244 \text{ kg}$$
(A.4)

## Appendice B

# Interpolatori SciPy

### B.1 Akima1DInterpolator

L'interpolatore Akima1DInterpolator[56] è una classe nel modulo scipy.interpolate di Python che viene utilizzata per interpolare dati unidimensionali utilizzando polinomi cubici a tratti. L'interpolatore Akima è un metodo di interpolazione che produce una curva continua e differenziabile che passa attraverso tutti i punti dati forniti. **Parametri**:

- $\mathbf{x}$ : Un array unidimensionale di valori reali che aumentano monotonicamente.
- y: Un array N-dimensionale di valori reali. La lunghezza di y lungo l'asse di interpolazione deve essere uguale alla lunghezza di x. Il parametro axis può essere utilizzato per selezionare l'asse di interpolazione.
- **axis**: L'asse nell'array y corrispondente ai valori delle coordinate x. Il valore predefinito è axis=0.

L'interpolatore Akima1DInterpolator è utile per applicazioni in cui è necessario interpolare dati unidimensionali in modo accurato e mantenere la continuità e la differenziabilità della curva interpolata. Ad esempio, l'interpolatore Akima1DInterpolator può essere utilizzato per interpolare dati di temperatura, velocità o posizione in un'applicazione di controllo in tempo reale.

### B.2 CubicSpline

La classe CubicSpline[57] nel modulo scipy.interpolate di Python è un interpolatore basato su spline cubiche. Le spline cubiche sono polinomi cubici a tratti che garantiscono che la curva interpolata sia continua e differenziabile due volte.

### Parametri:

- $\mathbf{x}$ : Un array unidimensionale di valori reali che aumentano monotonicamente.
- y: Un array N-dimensionale di valori reali. La lunghezza di y lungo l'asse di interpolazione deve essere uguale alla lunghezza di x. Il parametro axis può essere utilizzato per selezionare l'asse di interpolazione.
- axis: L'asse nell'array y corrispondente ai valori delle coordinate x. Il valore predefinito è axis=0.
- **bc\_type**: Il tipo di condizione al contorno. Due equazioni aggiuntive, date dalle condizioni al contorno, sono necessarie per determinare tutti i coefficienti dei polinomi su ciascun segmento. Se bc\_type è una stringa, la condizione specificata verrà applicata a entrambe le estremità di una spline. Le condizioni disponibili sono:
  - 'not-a-knot' (predefinito): Il primo e il secondo segmento all'estremità di una curva sono lo stesso polinomio. È un buon valore predefinito quando non si hanno informazioni sulle condizioni al contorno.
  - 'periodic': Si presume che la funzione interpolata sia periodica di periodo x[-1] x[0]. Il primo e l'ultimo valore di y devono essere identici: y[0] == y[-1]. Questa condizione al contorno produrrà y'[0] == y'[-1] e y"[0] == y"[-1].
  - 'clamped': La prima derivata alle estremità delle curve è zero. Presupponendo una y 1D, bc\_type=((1, 0.0), (1, 0.0)) è la stessa condizione.
  - 'natural': La seconda derivata alle estremità delle curve è zero. Presupponendo una y 1D, bc\_type=((2, 0.0), (2, 0.0)) è la stessa condizione.
- extrapolate: Determina se extrapolare a punti fuori dai limiti in base al primo e all'ultimo intervallo o restituire NaN. Se True, l'estrapolazione viene utilizzata per i punti fuori dai limiti. Se None (predefinito), extrapolate è impostato su True.

La classe CubicSpline è un interpolatore potente e versatile che può essere utilizzato per interpolare dati unidimensionali in un'ampia gamma di applicazioni.

### B.3 UnivariateSpline

La classe UnivariateSpline[58] nel modulo scipy.interpolate di Python è un interpolatore basato su spline smoothing univariata. Le spline smoothing univariate sono spline che vengono adattate a un insieme di dati fornito minimizzando una funzione di costo che comprende sia la vicinanza ai dati che la liscezza della curva interpolata.

### Parametri:

- x: Un array unidimensionale di valori reali che aumentano monotonicamente.
- y: Un array N-dimensionale di valori reali. La lunghezza di y lungo l'asse di interpolazione deve essere uguale alla lunghezza di x. Il parametro axis può essere utilizzato per selezionare l'asse di interpolazione.
- w: Un array unidimensionale di pesi per l'adattamento della spline. I pesi devono essere positivi. Se w è None, tutti i pesi sono 1.
- bbox: Una sequenza di 2 elementi che specifica il confine dell'intervallo di approssimazione. Se bbox è None, bbox=[x[0], x[-1]].
- k: il grado della spline smoothing. Deve essere compreso tra 1 e 5. k = 3 è una spline cubica.
- s: Il fattore di smoothing positivo utilizzato per scegliere il numero di nodi. Il numero di nodi verrà aumentato fino a quando la condizione di smoothing non è soddisfatta.
- ext: Controlla la modalità di estrapolazione per gli elementi non nell'intervallo definito dalla sequenza di nodi:
  - Se ext=0 o 'extrapolate', restituisce il valore estrapolato.
  - Se ext=1 o 'zeros', restituisce 0.
  - Se ext=2 o 'raise', solleva un ValueError.
  - Se ext=3 o 'const', restituisce il valore al contorno.

Il valore predefinito è 0.

• check\_finite: Un valore booleano che indica se verificare che gli array di input contengano solo numeri finiti. La disabilitazione può migliorare le prestazioni, ma può causare problemi (arresti anomali, mancata terminazione o risultati non sensati) se gli input contengono infinite o NaN. Il valore predefinito è False.

La classe UnivariateSpline è un interpolatore potente e versatile che può essere utilizzato per interpolare dati unidimensionali in un'ampia gamma di applicazioni, come l'analisi dei dati, la modellazione matematica e la computer grafica.

# Appendice C



Figura C.1. 11/10/2023 ore 10



Figura C.2. 11/10/2023 ore 12



Figura C.3. 11/10/2023 ore 14



Figura C.4. 11/10/2023 ore 16



Figura C.5. 12/10/2023 ore 10



Figura C.6. 12/10/2023 ore 12



Figura C.7. 12/10/2023 ore 14



Figura C.8. 12/10/2023 ore 16



Figura C.9. 13/10/2023 ore 10



Figura C.10. 13/10/2023 ore 12



Figura C.11. 13/10/2023 ore 14



Figura C.12. 13/10/2023 ore 16

## Bibliografia

- [1] ARCADE 2006 Flight, NASA, 2006.  $https://asd.gsfc.nasa.gov/archive/arcade/flight\_2006.html$ [2] Loon balloon, Alphabet-Loon, 2020. https://www.datacenterdynamics.com/en/analysis/connecting-world-balloon/ [3] Loon balloon mission highlights, Alphabet-Loon, 2020.https://x.company/projects/loon/ [4] NASA Super-TIGER balloon, NASA, 2011. https://www.nasa.gov/../supertiger-record.html [5] NASA BARREL mission, NASA, 2013. https://www.nasa.gov/../barrel [6] HIWIND mission, NASA, 2018. https://www.nasa.gov/wallops/... [7] NASA's ASCOT mission, NASA, 2018. https://www.nasa.gov/../ascot-takes-flight-on-nasa-heavy-lift-balloon [8] World View Enterprise's Stratollite. NASA, 2017.https://www.nasa.gov/../world view [9] ESA's Zero2Infinity, ESA, 2017. https://www.eoportal.org/satellite-missions/zero-2-infinity [10] Voss, Hank D., et al. (2012) "Understanding High-Altitude Balloon Flight Fundamentals.", 3rd Annual Academic High-Altitude Conference, pp 69-70. https://www.iastatedigitalpress.com/ahac/.. [11] Bachman, R. A. "Idealized Dynamics of Balloon Flight." American Journal of Physics, vol. 52, no. 4, 1 Apr. 1984, pp. 309-312, doi: 10.1119/1.13879 [12] Morris, Alvin. NCAR-TN/IA-99 NCAR TECHNICAL NOTE Scientific Ballooning Handbook, 1975. https://opensky.ucar.edu/../PDF [13] Brock, Fred V., and Scott J. Richardson. Meteorological Measurement Systems. New York, Oxford University Press, 2001.
- [14] Nobuyuki Yajima, et al. Scientific Ballooning : Technology and Applications of Exploration Balloons Floating in the Stratosphere and the Atmospheres of Other Planets.
New York, Ny Springer New York Springer, 2016. https://doi.org/10.1007/978-0-387-09727-5

- [15] Seifert, Paul, and Gordon Mcintosh. Terminal Velocity of High-Altitude Balloon Payloads: Experiment versus Theory. Iowa State University Digital Press, 2011.https://www.iastatedigitalpress.com/ahac/..
- [16] Drag of a Sphere, Glenn Research Center, NASA. https://www1.grc.nasa.gov//drag-of-a-sphere
- [17] Scholes, Daniel. Evaluation of the Aerodynamic Differences of a Balloon Shape Evaluation of the Aerodynamic Differences of a Balloon Shape and a Sphere Using Computational Fluid Dynamic Modeling in and a Sphere Using Computational Fluid Dynamic Modeling in Fluent Fluent., 2011.https://digitalcommons.usu.edu/scholes
- [18] Achenbach, Elmar. Experiments on the Flow Past Spheres at Very High Reynolds Numbers. Journal of Fluid Mechanics, vol. 54, no. 3, 8 Aug. 1972, pp. 565–575, https://doi.org/10.1017/s0022112072000874.
- [19] Livermore, Lawrence, et al. NASA Reference Publication 1327 "Description and Use of LSODE, the Livermore Solver for Ordinary Differential Equations", 1993. https://computing.llnl.gov/../ODEPACK\_pub2\_u113855.pdf
- [20] BAILEY, A. B., and J. HIATT. Sphere Drag Coefficients for a Broad Range of Mach and Reynolds Numbers. AIAA Journal, vol. 10, no. 11, Nov. 1972, pp. 1436–1440 https://doi.org/10.2514/3.50387
- [21] Box, Simon, et al. A Stochastic Six-Degree-of-Freedom Flight Simulator for Passively Controlled High Power Rockets. Journal of Aerospace Engineering V, vol. 24, no. 1, 2011, pp. 31–45 https://doi.org/10.1061/ASCEAS.1943-5525.0000051
- [22] Lagarias, Jeffrey C., et al. Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. SIAM Journal on Optimization, vol. 9, no. 1, Jan. 1998, pp. 112–147 https://doi.org/10.1137/s1052623496303470
- [23] Moradian, Niloofar, et al. "The Effects of Freestream Turbulence on the Drag Coefficient of a Sphere." Experimental Thermal and Fluid Science, vol. 33, no. 3, Mar. 2009, pp. 460–471 https://doi.org/10.1016/..
- [24] Nelder, J. A., and R. Mead. A Simplex Method for Function Minimization. The Computer Journal, vol. 7, no. 4, 1 Jan. 1965, pp. 308–313. https://doi.org/10.1093/comjnl/7.4.308
- [25] Sato, Kaoru, et al. Gravity Waves and Turbulence Associated with Cumulus Convection Observed with the UHF/VHF Clear-Air Doppler Radars. Journal of Geophysical Research, vol. 100, no. D4, 1995, pp. 7111–7119 https://doi.org/10.1029/95jd00198

- [26] Son, Kwangmin, et al. Effect of Free-Stream Turbulence on the Flow over a Sphere. Physics of Fluids, vol. 22, no. 4, Apr. 2010, p. 045101. https://doi.org/10.1063/1.3371804
- [27] Palumbo, Roberto, et al. Effective Approach to Characterization of Prediction Errors for Balloon Ascent Trajectories. Journal of Aircraft, vol. 47, no. 4, July 2010, pp. 1331–1337 https://doi.org/10.2514/1.47005
- [28] Sobester, A. et al. (2014) "High-altitude gas balloon trajectory prediction: A Montecarlo model", AIAA Journal, 52(4), pp. 832-842. doi:10.2514/1.j052900
- [29] EASA (Revision from September 2020) "Easy Access Rules for Balloons", EASA eRules. https://www.easa.europa.eu/.../easy-access-rules-balloons
- [30] Cisco Networking Academy "Skills For All", Cisco https://skillsforall.com/
- [31] Chapter 6 "General Winds National Weather Service", National Weather Service. https://www.weather.gov/../FireWx\_General\_Winds.pdf
- [32] Foehn effect, UK Government. https://en.wikipedia.org/wiki/Foehn\_wind
- [33] Variazione dei venti alpini Foehn https://www.metoffice.gov.uk/weather/../foehneffect
- [34] M. Kurz, "Analysis of Lee Cyclogenesis". doi:10.1016/b978-0-12-382225-3.00290-5
- [35] Kurz, M. "Diagnosis of alpine lee cyclogenesis The complete picture kit." https://www.imk-tro.kit.edu/img/content/ICAM\_Innsbruck.pdf
- [36] Manishsiq, "Local winds of the world, types, map, list of major local winds", StudyIQ, 2023. https://www.studyiq.com/mountains..
- [37] Schmidli, J. & Quimbayo-Duarte, J., "Diurnal valley winds in a deep alpine valley: Model results", MDPI, 2023. https://www.mdpi.com/2674-0494/2/1/7
- [38] A. Buzzi & S. Davolio, "Effects of orographic representation on simulations of cyclone development and mesoscale aspects of IOP 15", Institute of Atmospheric and Oceanic Sciences ISAO-CNR. https://www.isac.cnr.it/dinamica/davolio/Paper/buzzi.pdf
- [39] "How Does Topography Affect Climate? A Complex Topic Made Simple", SpatialPost,7 Febbraio 2022. https://www.spatialpost.com/how-does-topography-affectclimate
- [40] "Why Is Topography Important? The Importance of Shaping the Landscape", SpatialPost, 3 Gennaio 2022. https://www.spatialpost.com/why-is-topography-important/
- [41] Schmid, F. et al., "Diurnal valley winds in a deep alpine valley: Observations", MDPI, 2020. https://doi.org/10.3390/atmos11010054

- [42] Schmidli, J., Böing, S. and Fuhrer, O., "Accuracy of simulated diurnal valley winds in the Swiss alps: Influence of grid resolution, topography filtering, and land surface datasets", MDPI, 2018. https://doi.org/10.3390/atmos9050196
- [43] "Il vento in Piemonte", ARPA Piemonte, 2007. https://www.arpa.piemonte.it/pubblicazioni/pdf-il-vento-in-piemonte
- [44] P. Rauh et. A. Rubli, "Orographic flow in the vicinity of Alps", Swiss meteorological institute, 1993. https://www.ecmwf.int/..orographic-flow-vicinity-alps.pdf
- [45] Holton, J.R. and Hakim, G.J., "An introduction to dynamic meteorology.", Amsterdam: Academic Press, 2013.doi:10.1016/B978-0-12-384866-6.00001-5
- [46] North, G.R., Pyle, J. and Zhang, F., Encyclopedia of Atmospheric Sciences., Amsterdam: Elsevier / Acad. Press, pp. 265-271, 2015. doi:10.1016/b978-0-12-382225-3.00138-9
- [47] Fields, Τ. In-flight  $\mathbf{et}$ al., landing location predictions using ascent for Altitude wind data High Balloons, AIAA, 2013.Available at: https://arc.aiaa.org/doi/abs/10.2514/6.2013-1294.
- [48] Ubuntu distro releases Ubuntu 22.04.3 LTS (Jammy Jellyfish) https://releases.ubuntu.com/jammy/
- [49] Oracle VirtualBox Virtual Machine generator https://www.virtualbox.org/
- [50] Visual Studio VSCode https://code.visualstudio.com/
- [51] National Oceanic and Atmospheric Administration https://www.noaa.gov/
- [52] NOAA's Global Forecasting System https://www.ncei.noaa.gov/../global-forecast
- [53] Formula semisenoverso. https://en.wikipedia.org/wiki/Haversine\_formula
- [54] File GeoTIFF https://en.wikipedia.org/wiki/GeoTIFF
- [55] Interpolazione trilineare https://en.wikipedia.org/wiki/Trilinear\_interpolation
- [56] "Akima1dInterpolator", comunità di sviluppo SciPy, forum SciPy, 2018 https://docs.scipy.org/../scipy.interpolate.Akima1DInterpolator.html
- [57] "CubicSpline", comunità di sviluppo SciPy, forum SciPy, 2017 https://docs.scipy.org/doc/../scipy.interpolate.CubicSpline.html
- [58] "UnivariateSpline", comunità di sviluppo SciPy, forum SciPy, 2018 https://docs.scipy.org/doc/../scipy.interpolate.UnivariateSpline.html
- [59] Open-source electronic platform "Arduino", Arduino, 2023 https://www.arduino.cc/
- [60] Incubatore aziendale Les Pépinièrs d'Entreprises, "Start-Alp", Pont-Saint-Martin https://www.pepinieresvda.eu/