



UNIVERSITÀ POLITECNICA DELLE MARCHE  
FACOLTÀ DI INGEGNERIA

---

Corso di Laurea triennale in Ingegneria Informatica e dell'Automazione

**Implementazione numerica di una tecnica di rendezvous per droni quadricotteri basata su gruppi di Lie**

**Numerical implementation of a rendezvous technique for quadcopter drones based on Lie groups**

Relatore: Chiar.mo  
**Prof. Simone Fiori**

Tesi di Laurea di:  
**Matteo Losa**

A.A. 2022/2023



# Indice

<b>Introduzione</b>	<b>3</b>
<b>1 Consenso Multi-Agente</b>	<b>4</b>
1.1 Protocollo di consenso . . . . .	4
1.2 Valore comune di stato . . . . .	5
1.3 Traiettorie multi-agente . . . . .	7
1.4 Missione di rendezvous, rete non pesata . . . . .	9
1.5 Missione di rendezvous, rete pesata . . . . .	11
1.6 Missione di rendezvous, pesi tempo-varianti . . . . .	19
<b>2 Dinamica dei Quadricotteri</b>	<b>26</b>
2.1 Notazioni fisiche di base . . . . .	26
2.2 Forze e coppie esterne . . . . .	30
2.3 Pianificazione del movimento singolo-agente . . . . .	31
<b>3 Simulazioni Numeriche</b>	<b>33</b>
3.1 Metodo numerico . . . . .	33
3.2 Simulazione singolo-agente . . . . .	34
3.3 Simulazione multi-agente . . . . .	34
<b>4 Controllore per il Movimento del Drone</b>	<b>44</b>
4.1 Salita verticale (Heave) . . . . .	44
4.2 Orientazione (Yaw) . . . . .	45
4.3 Movimento orizzontale (Surge) . . . . .	45
4.4 Risultati . . . . .	46
<b>5 Conclusioni</b>	<b>54</b>

# Introduzione

I droni, e gli aeromobili a pilotaggio remoto in generale, stanno trovando sempre più impiego nelle missioni di ricerca, tracciamento, trasporto e monitoraggio di un certo obiettivo. L'intento è quello di far sì che dei droni facenti parte di una flotta collaborino tra loro per il raggiungimento di diversi obiettivi comuni; questi possono includere rendezvous, consenso, sincronizzazione e formazione. In questo caso ci focalizzeremo sulla realizzazione di una tecnica di controllo per una missione di rendezvous per dei droni quadricotteri. Per questo scopo, viene utilizzato un protocollo di consenso che tutti i droni, detti anche agenti, devono seguire: questo protocollo è realizzato senza input esterni, utilizzando solamente le informazioni disponibili sugli agenti.

I metodi utilizzati per la realizzazione del protocollo saranno quelli forniti dalla teoria dei grafi poiché ben si adattano alla scalabilità del numero di agenti e all'esistenza di una rete di comunicazione tra di essi, che è una delle supposizioni principali di cui ci serviremo. Per prima cosa, nel Capitolo 1, tramite la teoria dei grafi, definiremo il protocollo di consenso multi-agente indipendentemente dal tipo di velivolo considerato, poiché tratteremo i nostri agenti come delle particelle, trascurando la fisica del volo del velivolo.

Nel Capitolo 2, verrà introdotta e analizzata la dinamica di un drone quadricottero insieme a tre movimenti principali che esso effettuerà: salita verticale (*heave*), rotazione attorno all'asse verticale (*yaw*) e movimento orizzontale (*surge*). Nel Capitolo 3, dopo aver presentato i metodi numerici utilizzati per le simulazioni, verranno effettuate delle simulazioni numeriche di una missione di rendezvous multi-agente, utilizzando la dinamica realistica considerata in precedenza. Infine, verrà proposto, e poi applicato in una missione di rendezvous con tre agenti, un controllore per la pianificazione del movimento del drone verso la posizione finale, utilizzando solamente i tre movimenti citati sopra.

# Capitolo 1

## Consenso Multi-Agente

Consideriamo una missione al termine della quale gli agenti devono incontrarsi in una posizione comune nello spazio; ovvero, devono concordare una certa configurazione di stato in modo autonomo. Il nostro sistema sarà visto come una rete di agenti. In questo capitolo si vuole evidenziare la relazione presente tra le proprietà di convergenza del protocollo di consenso e la struttura delle connessioni tra gli agenti della rete.

### 1.1 Protocollo di consenso

**Definizione 1.** Definiamo una *rete* di agenti come un insieme finito di agenti i quali possiedono una certa interconnettività per lo scambio di informazioni. La rete viene rappresentata tramite un grafo, indicato con  $G(V,E)$ , dove  $V$  è l'insieme dei vertici ed  $E$  è l'insieme degli archi. Gli agenti corrispondono ai vertici del grafo  $G(V,E)$  e i collegamenti per lo scambio di informazioni sono rappresentati dagli archi. Indichiamo con  $V = \{v_1, \dots, v_n\}$  i vertici corrispondenti agli  $n$  agenti. L'insieme degli archi è un sottoinsieme  $E \subseteq V \times V$ ; ogni arco è indicato con  $(v_i, v_j) = v_i v_j$ . Inoltre, assumiamo che  $v_i v_j \in E$  implica  $v_j v_i \in E$  e che  $v_i v_i \notin E$ .

Il protocollo di consenso presuppone che il tasso di variazione dello stato di ogni agente è governato dalla somma degli stati degli agenti facenti parte del sottoinsieme di quelli ad esso adiacenti. Supponiamo che la funzione scalare  $q^i(t)$  indichi lo stato dell' agente  $i$ -esimo, con  $i = 1, \dots, n$ , e definiamo  $q(t) = (q^1(t), \dots, q^n(t))^T$ . Segue che

$$\dot{q}^i(t) = - \sum_{j \in N(i)} (q^i(t) - q^j(t)), \quad 1 \leq i \leq n, \quad (1.1)$$

dove  $N(i)$  è l'insieme dei vertici adiacenti al vertice  $v_i$  definito da  $N(i) = \{v_j \in V \mid v_i v_j \in E\}$ . Nel nostro caso assumiamo che l'adiacenza sia simmetrica, i.e.,  $v_i v_j \in E \iff v_j v_i \in E$ , e quindi possiamo dire che la rete è *non-orientata* e possiamo riscrivere (1.1) nella forma più compatta

$$\dot{q}(t) = -L(G)q(t), \quad (1.2)$$

dove  $L(G) \in \mathbb{R}^{n \times n}$  è detto *Laplaciano* del grafo. Quest'ultimo è determinato in modo univoco dalla struttura di  $G(V, E)$ .

**Definizione 2.** L'equazione 1.2 è il nostro *protocollo di consenso* o *dinamica del consenso*.

## 1.2 Valore comune di stato

Da qui in avanti assumeremo sempre che  $G(V, E)$  sia un grafo connesso e non-orientato per sfruttare le proprietà che derivano da questa assunzione.

**Proposizione 1.** *Supponiamo che  $G(V, E)$  sia un grafo connesso e non-orientato. Il Laplaciano  $L(G)$  è una matrice semidefinita positiva con autovalori  $0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_n$ . Inoltre, il vettore di dimensione  $n$  composto da tutti 1,  $\vec{1}$ , è un autovettore corrispondente all'autovalore  $\lambda_1$ , i.e.,  $L(G)\vec{1} = 0 \cdot \vec{1} = \vec{0}$ .*

*Dimostrazione.* Il risultato di questa dimostrazione può essere trovato in [10], [12], [13].  $\square$

Data  $\Lambda = \text{diag}(0, \lambda_2, \dots, \lambda_n)$ , sia  $U = [\hat{u}_1 \ \hat{u}_2 \ \dots \ \hat{u}_n]$  (il simbolo  $\hat{\cdot}$  indica un versore) una matrice  $n \times n$  che consiste negli autovettori di  $L(G)$  corrispondenti agli autovalori presenti nella diagonale di  $\Lambda$ , i.e.,  $U^T U = I$  e  $L(G)U = U\Lambda$ . Poiché possiamo riscrivere  $L(G)$  come  $U\Lambda U^T$ , la soluzione dell'equazione 1.2 con condizioni iniziali  $q(0) = q_0$  è data da

$$q(t) = e^{-L(G)t} q_0 = e^{-U\Lambda U^T t} q_0 = U e^{-\Lambda t} U^T q_0,$$

che può essere riscritta come,

$$q(t) = \sum_{i=1}^n e^{-\lambda_i t} (\hat{u}_i^T q_0) \hat{u}_i. \quad (1.3)$$

**Definizione 3.** Dato un insieme di  $n$  agenti, il suo *insieme di consenso*  $\mathbf{A} \subset \mathbb{R}^n$  è definito dal sottospazio  $\text{span}\{\vec{1}\}$ . Un *valore di stato comune di consenso* è un elemento di  $\mathbf{A}$ , e il valore comune dei suoi componenti (identici) è detto *valore di stato concordato*.

Con la prossima proposizione vedremo che quando lo stato di una rete multi-agente soddisfa il protocollo di consenso, lo stato converge al valore di stato comune di consenso il cui valore di stato concordato è ottenuto dalla media delle componenti iniziali degli stati.

**Proposizione 2.** *Sia  $G(V,E)$  una rete di  $n$  agenti il cui stato  $q(t)$  soddisfi il protocollo di consenso  $\dot{q} = -L(G)q$ . Allora  $\lim_{t \rightarrow +\infty} q(t) = \alpha \vec{1}$ , dove  $\alpha = \frac{\vec{1}^T q(0)}{n} \in \mathbb{R}$  è la media degli stati iniziali. Inoltre, poiché  $\lambda_2$  è l'autovalore più piccolo non nullo di  $L(G)$ , il suo modulo determinerà la velocità di convergenza a  $\alpha \vec{1}$  ([13], [10]).*

*Dimostrazione.* Il risultato può essere trovato in [8]. □

Il punto  $\lim_{t \rightarrow +\infty} q(t) = \frac{\vec{1}^T q_0}{n} \vec{1} = \alpha \vec{1} \in \mathbb{R}^n$  è precisamente la proiezione ortogonale di  $q_0$  sul sottospazio di consenso e quindi minimizza  $\|q - q_0\|$  su tutti i possibili  $q \in \mathbf{A}$ , dove  $\|\cdot\|$  è la distanza Euclidea standard. In più,  $\frac{\vec{1}^T q(t)}{n}$  è una quantità costante poiché

$$\frac{d}{dt} \left( \frac{\vec{1}^T q(t)}{n} \right) = \frac{\vec{1}^T \dot{q}(t)}{n} = \frac{\vec{1}^T (-L(G)q(t))}{n} = -\frac{q^T(t) L^T(G) \vec{1}}{n} \quad (1.4)$$

$$= -\frac{q^T(t) L(G) \vec{1}}{n} = -\frac{q^T(t) \vec{0}}{n} = 0. \quad (1.5)$$

per valori arbitrari di  $t$ . Quindi il valore di stato concordato  $\alpha = \frac{\vec{1}^T q_0}{n}$  è completamente determinato a  $t = 0$ .

Ora, supponiamo che gli agenti si debbano accordare su  $r$  valori scalari. Questo vuol dire che ad ogni agente  $i$  associamo una quantità vettoriale  $q^i(t) = (q_1^i(t), \dots, q_r^i(t))$ ; sia  $Q$  una matrice  $n \times r$  le cui righe sono determinate da  $q^i$ ,

$$Q = \begin{pmatrix} q_1^1 & \dots & q_r^1 \\ q_1^2 & \dots & q_r^2 \\ \vdots & \vdots & \vdots \\ q_1^n & \dots & q_r^n \end{pmatrix} \quad (1.6)$$

e possiamo riscrivere la dinamica di consenso come,

$$\dot{Q}(t) = -L(G)Q(t). \quad (1.7)$$

Ora introduciamo  $q^i(0) = (q_1^i(0), \dots, q_r^i(0))$ , ovvero il vettore riga dello stato iniziale dell'agente  $i$ , e  $q_k(0) = (q_k^1(0), \dots, q_k^n(0))^T$ , il vettore colonna di tutti

gli agenti per un certo valore scalare  $k$  di stato. Dato che  $L(G) = U\Lambda U^T$ , abbiamo che la soluzione di 1.7 è data da:  $Q(t) = e^{-L(G)t}Q(0) = e^{-U\Lambda U^T t}Q(0) = Ue^{-\Lambda t}U^T Q(0)$ , che, da [8], può essere riscritto come,

$$Q(t) = [\alpha_1 \vec{1} \dots \alpha_r \vec{1}] + \sum_{i=2}^n [e^{-\lambda_i t} \hat{u}_i^T q_1(0) \hat{u}_i \dots e^{-\lambda_i t} \hat{u}_i^T q_r(0) \hat{u}_i] \quad (1.8)$$

dove  $\frac{\vec{1}^T q_k(0)}{n} = \alpha_k \in \mathbb{R}$  è il valore di stato concordato per ogni stato scalare,  $1 \leq k \leq r$ .

Da notare che abbiamo assunto che gli stati  $q_k$ , misurati dagli agenti, siano condivisi sulla rete  $G(V, E)$ . Quindi, gli stati condividono un Laplaciano comune,  $L(G)$ .

### 1.3 Traiettorie multi-agente

In questa sezione,  $\hat{e}_j$  sta ad indicare il versore canonico relativo alla direzione  $j$ . Useremo  $u_1$  per riferirci al vettore  $\vec{1}$ , che, come abbiamo visto, è un autovettore di  $L(G)$  relativo all'autovalore  $\lambda_1 = 0$ . Seguendo questa notazione, abbiamo che  $\hat{u}_1 = \frac{\vec{1}}{\sqrt{n}}$ .

**Lemma 1.** *Supponiamo che l'agente  $j$  sia connesso ad ogni altro agente. Allora,  $u_1 - n\hat{e}_j$  è un autovettore per  $L(G)$  corrispondente all'autovalore  $\lambda_n = n$ :*

$$L(G)(u_1 - n\hat{e}_j) = n(u_1 - n\hat{e}_j). \quad (1.9)$$

*Dimostrazione.* La dimostrazione può essere trovata in [8]. □

**Definizione 4.** Diciamo che il vertice  $v_i$  è *completamente connesso* se esiste un arco tra  $v_i$  e ogni altro vertice  $v_j \in V$ ,  $j \neq i$ . Segue che un vertice completamente connesso ha  $n - 1$  archi.

Inoltre, è dimostrato in [1] che gli autovalori di  $L(G)$  sono al più uguali a  $n$  e che la molteplicità algebrica dell'autovalore  $\lambda = n$  è pari al numero di vertici completamente connessi.

**Teorema 1.** *Supponiamo che il vertice  $v_j$  sia completamente connesso e che  $q(t)$  soddisfi il protocollo di consenso  $\dot{q} = -L(G)q$ . Allora  $q^j(t)$  è una linea retta.*

*Dimostrazione.* Il risultato è dimostrato in [8]. □



Da notare che la dimostrazione è valida sia nel caso in cui  $q_j$  è una quantità scalare sia in quello in cui è un vettore riga di una matrice di stato  $Q(t) \in \mathbb{R}^{n \times r}$ , purché rispetti le premesse del teorema.

Consideriamo ora un grafo  $G_\omega$  con archi pesati, i.e., ad ogni arco è associato un peso numerico. Allora, in questo caso, gli elementi fuori dalla diagonale del Laplaciano  $L(G_\omega)$  non saranno solo limitati a 0 e  $-1$ . Infatti, i valori  $-1$  corrispondenti agli archi saranno sostituiti da  $-\omega_{ij} = -\omega_{ji}$ , ovvero i pesi.

I valori sulla diagonale saranno, anche in questo caso, i valori assoluti delle somme lungo le rispettive righe o colonne degli elementi fuori dalla diagonale. Le equazioni 1.1 e 1.2 diventano quindi:

$$\dot{q}^i(t) = - \sum_{j \in N(i)} \omega_{ij} (q^i(t) - q^j(t)) \quad (1.10)$$

$$\dot{q}(t) = -L(G_\omega)q(t) \quad (1.11)$$

Il fatto che, anche in questo caso, il valore comune di stato è dato da  $\alpha = \frac{\vec{1}q_0}{n} \vec{1}$  deriva dalla stessa logica utilizzata nel caso di rete non pesata, poiché  $L(G_\omega)$  è una matrice semidefinita positiva con autovalori  $0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_n$  e  $L(G_\omega)\vec{1} = \vec{0}$ . Questo risultato è una diretta conseguenza del fatto che il peso aggiunto agli archi preserva le proprietà di connessione e non-orientamento di  $G_\omega(V, E)$ . In più, la somma degli elementi lungo ogni riga e colonna di  $L(G_\omega)$  dà come risultato 0, indipendentemente dal fatto che gli archi siano pesati o no. Ciò comporta che i valori di  $\lambda_2, \dots, \lambda_n$  e  $\hat{u}_2 \dots \hat{u}_n$  cambieranno, ma avremo comunque che  $\lambda_1 = 0$  e  $\hat{u}_1 = \frac{\vec{1}}{\sqrt{n}}$ .

Infine, consideriamo il seguente teorema:

**Teorema 2.** *Consideriamo il sistema dinamico:*

$$\dot{q}^i(t) = - \sum_{j=1}^n \omega_{ij}(t) [q^i(t) - q^j(t)] \quad (1 \leq i \leq n) \quad (1.12)$$

con condizioni iniziali  $q(0) = (q^1(0), \dots, q^n(0))^T$ .

Per ogni  $i, j$ , assumiamo che  $\omega_{ij} : [0, \infty) \rightarrow [0, \infty)$  e che  $\omega_{ij}(t) = \omega_{ji}(t)$ .

Definiamo la matrice Laplaciana  $L(G_\omega)(t)$  nel seguente modo:

$$L(G_\omega)(t) = \begin{pmatrix} \sum_{j=1}^n \omega_{1j}(t) & -\omega_{12}(t) & \dots & -\omega_{1n}(t) \\ -\omega_{12}(t) & \sum_{j=1}^n \omega_{2j}(t) & \dots & -\omega_{2n}(t) \\ \vdots & \vdots & \ddots & \vdots \\ -\omega_{1n}(t) & -\omega_{2n} & \dots & \sum_{j=1}^n \omega_{nj}(t) \end{pmatrix} \quad (1.13)$$

Allora il sistema dinamico del primo ordine può essere riscritto come  $\dot{q}(t) = -L(G_\omega)(t)q(t)$ . Se il rango di  $L(G_\omega)(t)$  è pari a  $n - 1$  per ogni  $t$ , allora

abbiamo:

$$\lim_{t \rightarrow +\infty} q(t) = \left[ \frac{1}{n} \sum_{i=1}^n q^i(0) \right] \vec{1}. \quad (1.14)$$

*Dimostrazione.* Questo risultato può essere trovato in [8].  $\square$

## 1.4 Missione di rendezvous, rete non pesata

In questa sezione mostreremo alcuni risultati preliminari di una missione di rendezvous con 4 agenti. Lo stato di un agente  $i$ -esimo è rappresentato da  $q^i(t) = (x^i(t), y^i(t), z^i(t))$ , ovvero la sua posizione in  $\mathbb{R}^3$ . Faremo uso delle stesse condizioni (posizioni) iniziali degli agenti e la stessa configurazione di rete utilizzate in [8]:

$$q^1(0) = (4, 17, 24), q^2(0) = (18, 10, 32), q^3(0) = (15, 10, 26), q^4(0) = (4, 2, 35).$$

In questo primo esperimento, assumiamo che la rete di comunicazione tra gli agenti sia fissata e non pesata. Verranno effettuate le simulazioni con due configurazioni di rete differenti. Nel primo scenario, la rete è rappresentata dalla matrice Laplaciana

$$L_1 = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix} \quad (1.15)$$

Tradotto, abbiamo che l'agente 1 è connesso con l'agente 2; l'agente 2 è connesso con gli agenti 1, 3 e 4; l'agente 3 è connesso con gli agenti 2 e 4; l'agente 4 è connesso con gli agenti 2 e 3.

Mentre, nel secondo scenario, l'agente 1 è connesso con gli agenti 3 e 4; l'agente 2 è connesso con l'agente 3; l'agente 3 è connesso con gli agenti 1 e 2; l'agente 4 è connesso con l'agente 1. Quest'ultimo scenario è rappresentato dalla seguente matrice:

$$L_2 = \begin{pmatrix} 2 & 0 & -1 & -1 \\ 0 & 1 & -1 & 0 \\ -1 & -1 & 2 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix} \quad (1.16)$$

Prima di mostrare i risultati, mostreremo i metodi numerici utilizzati per effettuare la simulazione.

La legge di evoluzione è data dal seguente insieme di equazioni differenziali (non ci sono pesi),

$$q'_i = \sum_{j=1}^4 (q_j - q_i), \quad (1.17)$$

per  $i = 1, 2, 3, 4$ .

L'algoritmo utilizzato è il metodo fEul (*forward Euler method*) che prevede di risolvere numericamente le equazioni differenziali precedenti nel seguente modo:

$$q_{n+1}^i = q_n^i + h f_q^i, \quad i = 1, 2, 3, 4, \quad (1.18)$$

dove  $f_q^i = \sum_{j=1}^4 (q_{j,n} - q_{i,n})$  è la legge di evoluzione per l'agente  $i$  calcolata ad ogni istante  $n$  della simulazione; mentre  $h$  rappresenta il passo numerico. Questo metodo è ricavato direttamente dalla definizione di derivata destra di una generica funzione  $f(x)$ ,

$$f'_+(x) := \lim_{h \rightarrow 0^+} \frac{f(x+h) - f(x)}{h} \quad (1.19)$$

Approssimando numericamente, si può ottenere,

$$f'_+(t_n) \approx \frac{f(t_{n+1}) - f(t_n)}{t_{n+1} - t_n} \Rightarrow f'_n := \frac{f_{n+1} - f_n}{h} \quad (1.20)$$

Da cui si ricava,

$$f_{n+1} = f_n + h f'_n \quad (1.21)$$

Di seguito, è riportato il codice MATLAB dell'algoritmo numerico utilizzato per la simulazione di entrambi gli scenari:

```
% Simulazione numerica del consenso tra 4 droni con
    rete non pesata.

% Parametri utilizzati
t = 5; % intervallo temporale della simulazione
K = 2000; % numero di passi di calcolo
h = t/K; % passo numerico

% Stati iniziali
q1 = [4; 17; 24];
q2 = [18; 10; 32];
q3 = [15; 10; 26];
q4 = [4; 2; 35];
x1 = q1;
x2 = q2;
x3 = q3;
x4 = q4;
```

```

% Simulazione numerica
for k = 1:K-1
    % Primo scenario
    q1(:,k+1) = q1(:,k) + h*(q2(:,k) - q1(:,k));
    q2(:,k+1) = q2(:,k) + h*(q1(:,k) - q2(:,k)) + h
        *(q3(:,k) - q2(:,k)) ...
        + h*(q4(:,k) - q2(:,k));
    q3(:,k+1) = q3(:,k) + h*(q2(:,k) - q3(:,k)) + h
        *(q4(:,k) - q3(:,k));
    q4(:,k+1) = q4(:,k) + h*(q2(:,k) - q4(:,k)) + h
        *(q3(:,k) - q4(:,k));

    % Secondo scenario
    x1(:,k+1) = x1(:,k) + h*(x3(:,k) - x1(:,k)) + h
        *(x4(:,k) - x1(:,k));
    x2(:,k+1) = x2(:,k) + h*(x3(:,k) - x2(:,k));
    x3(:,k+1) = x3(:,k) + h*(x1(:,k) - x3(:,k)) + h
        *(x2(:,k) - x3(:,k));
    x4(:,k+1) = x4(:,k) + h*(x1(:,k) - x4(:,k));
end

```

Come previsto dai risultati teorici precedenti, il vettore dello stato concordato (ovvero la posizione di rendezvous) è dato da:

$$q^* = (10.25, 9.75, 29.25), \quad (1.22)$$

che è proprio la media degli stati iniziali degli agenti.

Nella Figura 1.1 è mostrato come differenti configurazioni della rete producono differenti traiettorie degli agenti durante la missione, nonostante la posizione finale di rendezvous sia la stessa, poiché essa dipende unicamente dalle posizioni iniziali degli agenti. Nelle Figure 1.2, 1.3, 1.4, 1.5 sono illustrate le traiettorie nei 2 scenari (linee continue per lo scenario 1 e linee tratteggiate per lo scenario 2) rispettivamente per l'agente 1, agente 2, agente 3 e agente 4.

## 1.5 Missione di rendezvous, rete pesata

Ora immaginiamo una missione di rendezvous multi-agente prendendo in considerazione la robustezza delle comunicazioni tra gli agenti.

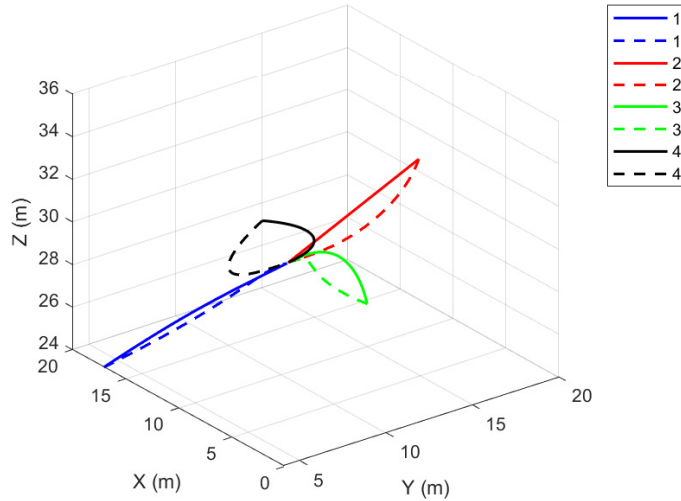


Figura 1.1: Missione di rendezvous con rete non pesata. Le linee continue rappresentano le traiettorie del primo scenario; le linee tratteggiate indicano le traiettorie degli agenti nel secondo scenario. La posizione di rendezvous coincidono ma le traiettorie sono differenti.

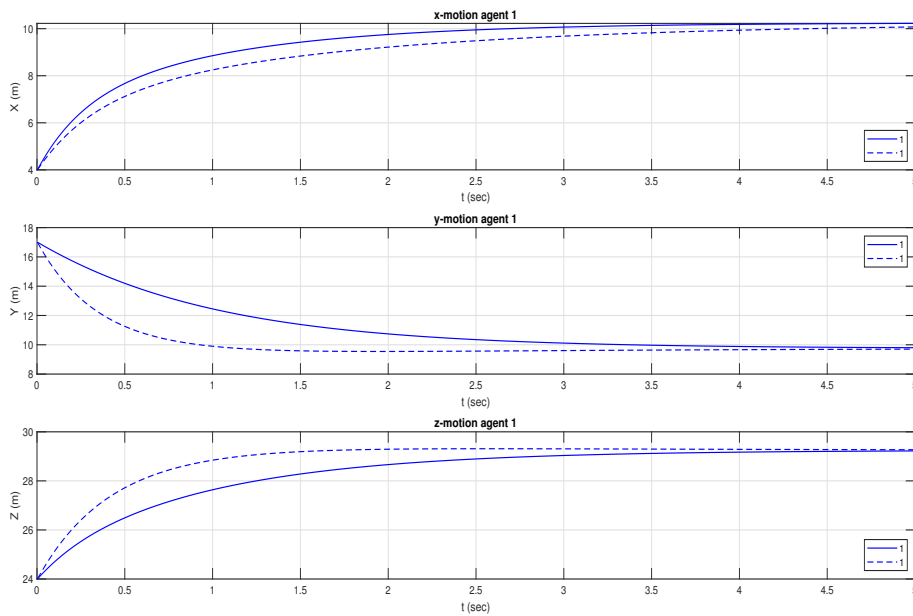


Figura 1.2: Confronto tra i movimenti lungo gli assi x,y e z dell'agente 1 corrispondente alla missione di rendezvous con rete non pesata in Figura 1.1

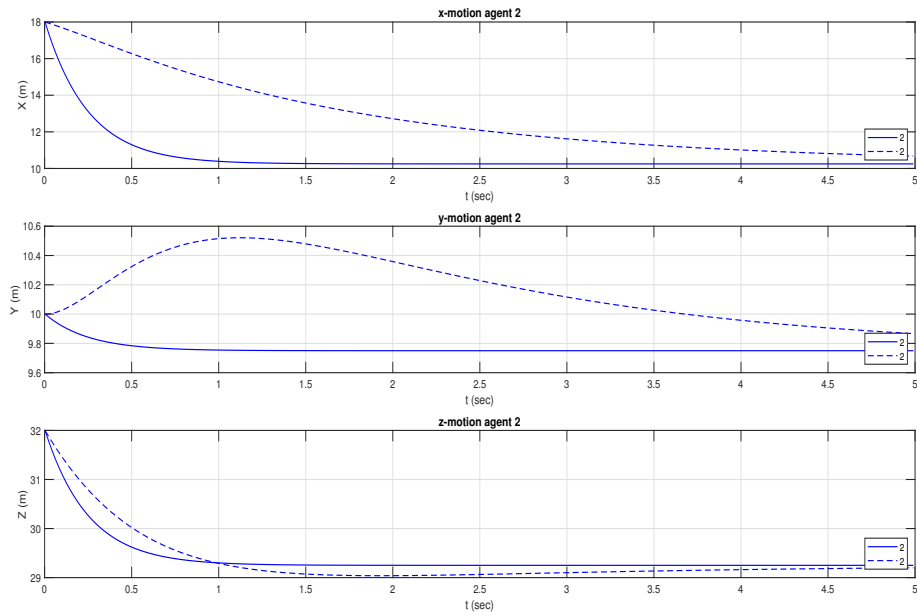


Figura 1.3: Confronto tra i movimenti lungo gli assi x,y e z dell'agente 2 corrispondente alla missione di rendezvous con rete non pesata in Figura 1.1

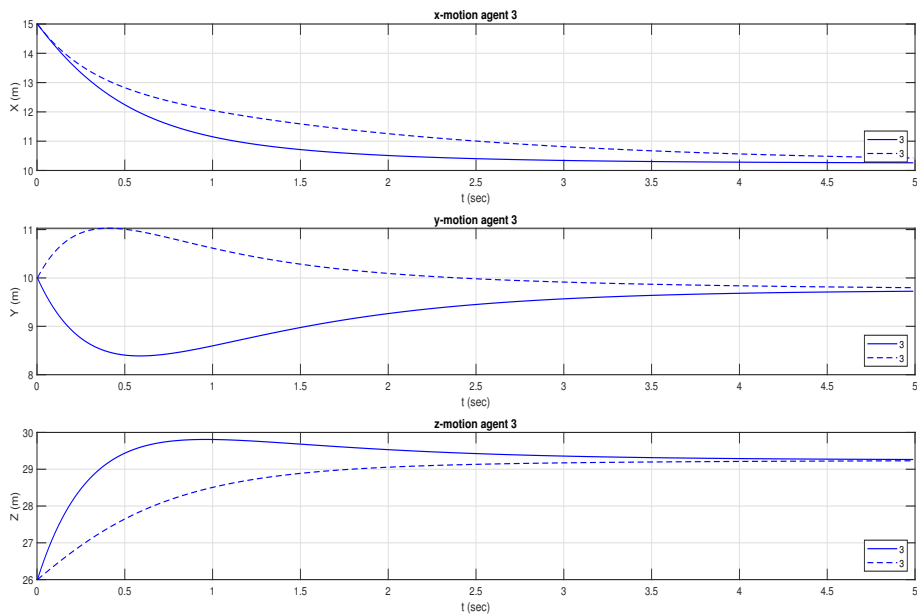


Figura 1.4: Confronto tra i movimenti lungo gli assi x,y e z dell'agente 3 corrispondente alla missione di rendezvous con rete non pesata in Figura 1.1

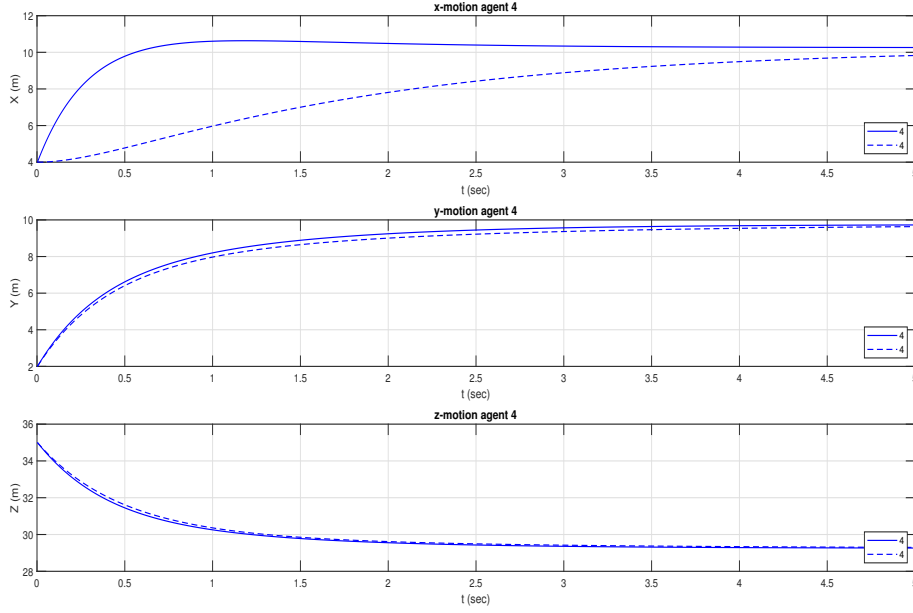


Figura 1.5: Confronto tra i movimenti lungo gli assi x,y e z dell'agente 4 corrispondente alla missione di rendezvous con rete non pesata in Figura 1.1

I 4 agenti sono inizialmente distribuiti nel seguente modo:

$$q^1(0) = (2, 16, 29), q^2(0) = (14, 20, 31), q^3(0) = (10, 1, 25), q^4(0) = (10, 14, 36).$$

Seguendo l'esperimento effettuato in [8], confronteremo direttamente 2 scenari; nel primo la rete non è pesata mentre lo è nel secondo.

Per entrambi gli scenari assumiamo che: l'agente 1 è connesso con l'agente 2; l'agente 2 è connesso con gli agenti 1, 3 e 4; l'agente 3 è connesso con gli agenti 2 e 4; l'agente 4 è connesso con gli agenti 2 e 3.

La Laplaciana del primo scenario è descritta da:

$$L_1 = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix} \quad (1.23)$$

Nel secondo scenario supponiamo che i pesi associati agli archi siano determinati dalla distanza iniziale nello spazio tra gli agenti:

$$L_2 = \begin{pmatrix} 12.8062 & -12.8062 & 0 & 0 \\ -12.8062 & 41.9036 & -20.3224 & -8.7750 \\ 0 & -20.3224 & 37.3518 & -17.0294 \\ 0 & -8.7750 & -17.0294 & 25.8044 \end{pmatrix} \quad (1.24)$$

L'algoritmo utilizzato per la simulazione è sempre il fEul, ma la legge di evoluzione nel secondo scenario deve prendere in considerazione i pesi,

$$q'_i = \sum_{j=1}^4 \omega_{ij}(q_j - q_i) \quad (1.25)$$

Di seguito, è riportato il codice MATLAB dell'algoritmo numerico utilizzato per la simulazione di entrambi gli scenari:

```

% Simulazione numerica del consenso tra 4 droni con
  rete pesata.

% Funzioni utilizzate
% distanza nello spazio
dist = @(v1,v2) sqrt((v2(1)-v1(1))^2 + (v2(2)-v1(2))^2 + (v2(3)-v1(3))^2);

% Parametri utilizzati
t = 5; % intervallo temporale della simulazione
K = 2000; % numero di passi di calcolo
h = t/K; % passo numerico

% Stati iniziali
q1 = [2;16;29];
q2 = [14;20;31];
q3 = [10;1;25];
q4 = [10;14;36];
x1 = q1;
x2 = q2;
x3 = q3;
x4 = q4;

% Pesi
w12 = dist(q1,q2);
w21 = w12;
w23 = dist(q2,q3);
w32 = w23;
w24 = dist(q2,q4);
w42 = w24;
w34 = dist(q3,q4);
w43 = w34;

```



```

% Simulazione numerica
for k = 1:K-1
    % Scenario con rete non pesata
    q1(:,k+1) = q1(:,k) + h*(q2(:,k) - q1(:,k));
    q2(:,k+1) = q2(:,k) + h*(q1(:,k) - q2(:,k)) + h
        *(q3(:,k) - q2(:,k)) ...
        + h*(q4(:,k) - q2(:,k));
    q3(:,k+1) = q3(:,k) + h*(q2(:,k) - q3(:,k)) + h
        *(q4(:,k) - q3(:,k));
    q4(:,k+1) = q4(:,k) + h*(q2(:,k) - q4(:,k)) + h
        *(q3(:,k) - q4(:,k));

    % Scenario con rete pesata
    x1(:,k+1) = x1(:,k) + h*w12*(x2(:,k) - x1(:,k));
    x2(:,k+1) = x2(:,k) + h*w21*(x1(:,k) - x2(:,k))
        + h*w23*(x3(:,k) - x2(:,k)) ...
        + h*w24*(x4(:,k) - x2(:,k));
    x3(:,k+1) = x3(:,k) + h*w32*(x2(:,k) - x3(:,k))
        + h*w34*(x4(:,k) - x3(:,k));
    x4(:,k+1) = x4(:,k) + h*w42*(x2(:,k) - x4(:,k))
        + h*w43*(x3(:,k) - x4(:,k));
end

```

Come ci aspettavamo, la posizione finale di rendezvous è la medesima in entrambi gli scenari e coincide con la media degli stati iniziali degli agenti,

$$q^* = (9, 12.75, 30.25) \quad (1.26)$$

Nella Figura 1.6 viene mostrato come i pesi aggiunti alla rete influiscano sulle traiettorie percorse dagli agenti. Inoltre, va notato che, nello scenario con reti pesate, nonostante l'agente 2 sia connesso a tutti gli altri agenti, esso non si muove più lungo una linea retta; cosa che accade per l'agente 2 nel primo scenario, come predetto dal Teorema 1.

Nelle Figure 1.7, 1.8, 1.9 e 1.10 sono confrontate le traiettorie dei due scenari rispettivamente dell'agente 1, agente 2, agente 3 e agente 4 lungo ogni coordinata. Da notare che nello scenario con rete pesata (linee tratteggiate) la velocità di convergenza alla posizione di rendezvous è più veloce.

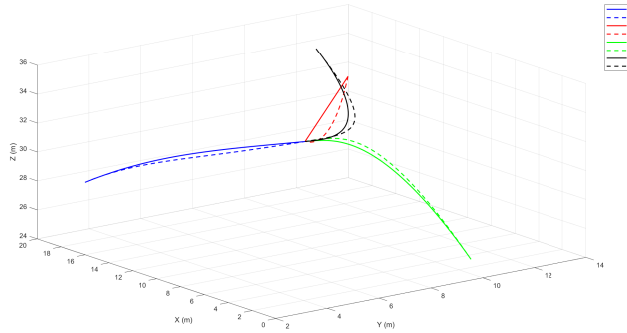


Figura 1.6: Confronto tra le traiettorie di una missione di rendezvous con rete pesata (linee tratteggiate) e non-pesata (linee continue). Le traiettorie convergono allo stesso punto ma le traiettorie differiscono tra i due scenari.

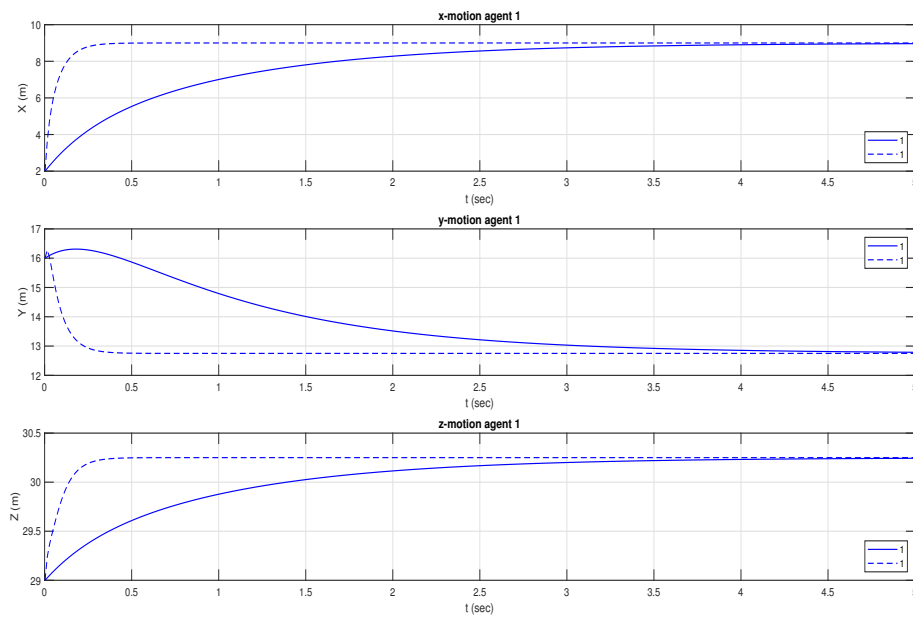


Figura 1.7: Confronto tra i movimenti lungo gli assi x, y e z dell'agente 1 corrispondente alla missione di rendezvous con rete sia pesata che non in Figura 1.6

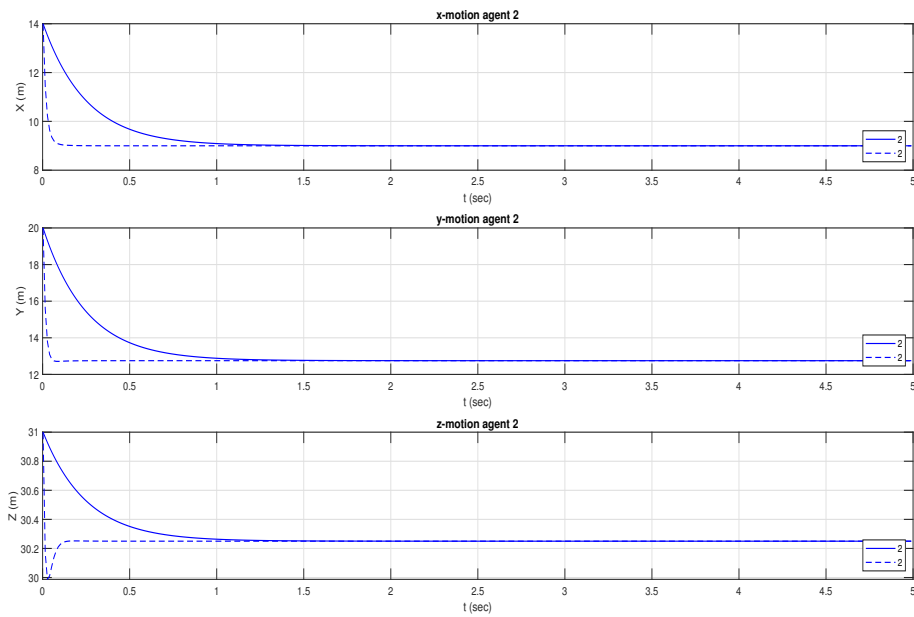


Figura 1.8: Confronto tra i movimenti lungo gli assi x, y e z dell'agente 2 corrispondente alla missione di rendezvous con rete sia pesata che non in Figura 1.6

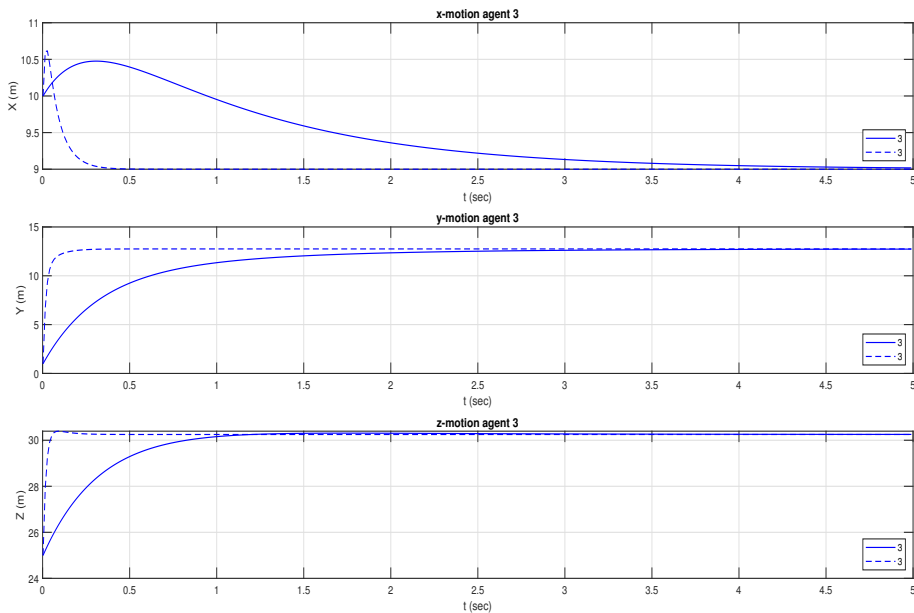


Figura 1.9: Confronto tra i movimenti lungo gli assi x, y e z dell'agente 3 corrispondente alla missione di rendezvous con rete sia pesata che non in Figura 1.6

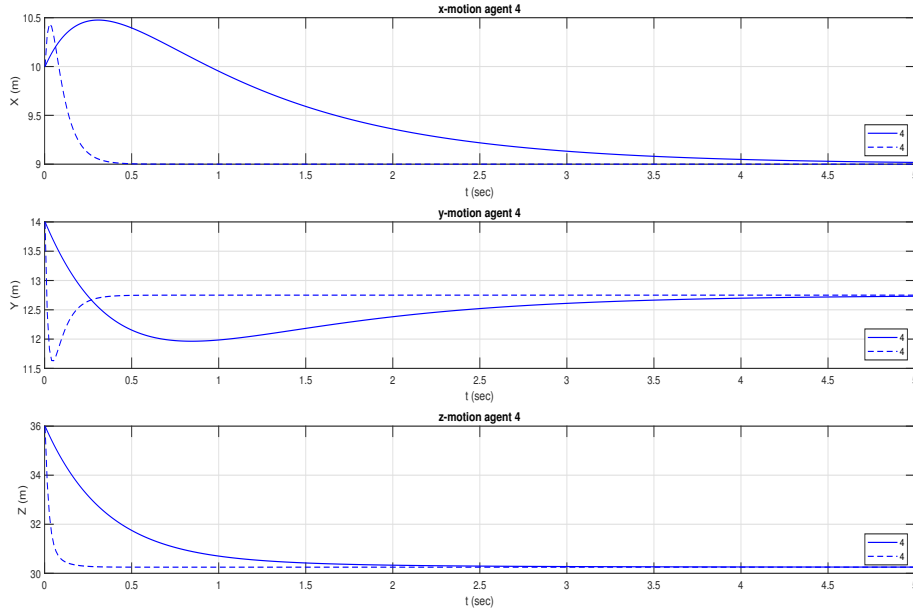


Figura 1.10: Confronto tra i movimenti lungo gli assi x, y e z dell'agente 4 corrispondente alla missione di rendezvous con rete sia pesata che non in Figura 1.6

## 1.6 Missione di rendezvous, pesi tempo-varianti

In quest'ultimo esperimento del capitolo vogliamo confrontare una missione di rendezvous con rete non pesata con un altro scenario in cui i pesi degli archi sono stabiliti dalla distanza nel tempo tra i singoli agenti.

Inizialmente, la rete non è completamente connessa, ma lo diventa una volta che una soglia di prossimità viene superata.

La formula utilizzata per calcolare la distanza nello spazio tra un agente  $i$  e un agente  $j$  è la seguente:

$$d(q_i, q_j) = \sqrt{(q_1^j - q_1^i)^2 + (q_2^j - q_2^i)^2 + (q_3^j - q_3^i)^2},$$

dove  $q_r^i$  e  $q_r^j$  sono le  $r$ -esime componenti dei vettori di stato, e quindi delle posizioni nello spazio, rispettivamente dell'agente  $i$  e dell'agente  $j$ .

In questo esperimento, la soglia sotto la quale la rete diventa completamente connessa è stata fissata a 15 metri. Anche per questa simulazione, utilizzeremo i dati dell'esperimento svolto in [8].

Gli agenti sono disposti inizialmente come di seguito:

$$q^1(0) = (16, 5, 36), q^2(0) = (19, 19, 29), q^3(0) = (12, 16, 33), q^4(0) = (14, 1, 26).$$

Per entrambi gli scenari, la rete è configurata nel seguente modo: l'agente 1

è connesso con gli agenti 3 e 4; l'agente 3 è connesso all'agente 3; l'agente 4 è connesso all'agente 1.

$$L_1 = \begin{pmatrix} 2 & 0 & -1 & -1 \\ 0 & 1 & -1 & 0 \\ -1 & -1 & 2 & 0 \\ -1 & 0 & 0 & -1 \end{pmatrix} \quad (1.27)$$

Per lo scenario con pesi tempo-varianti, la rete varia man mano che gli agenti si avvicinano tra loro. I pesi iniziali sono dati dalla distanza nello spazio tra gli agenti, come nell'esperimento precedente. Di seguito riportiamo la Laplaciana all'istante iniziale della missione:

$$L_{2i} = \begin{pmatrix} 23.0375 & 0 & -12.0830 & -10.9545 \\ 0 & 8.6023 & -8.6023 & 0 \\ -12.0830 & -8.6023 & 20.6854 & 0 \\ -10.9545 & 0 & 0 & 10.9545 \end{pmatrix} \quad (1.28)$$

Anche per questa simulazione abbiamo utilizzato l'algoritmo numerico fEul con la stessa evoluzione dell'esperimento con pesi tempo-invarianti.

Di seguito è riportato il codice MATLAB utilizzato per la simulazione numerica:

```
% Funzioni utilizzate
% distanza nello spazio
dist = @(v1,v2) sqrt((v2(1)-v1(1))^2 + (v2(2)-v1(2))^2 + (v2(3)-v1(3))^2);

% Parametri utilizzati
t = 5; % intervallo temporale della simulazione
K = 2000; % numero di passi di calcolo
h = t/K; % passo numerico
th = 15; % soglia di prossimita'

% Stati iniziali
q1 = [16;5;36];
q2 = [19;19;29];
q3 = [12;16;33];
q4 = [14;1;26];
x1 = q1;
x2 = q2;
x3 = q3;
```

```

x4 = q4;

% Pesì iniziali
w12 = zeros(1,K);
w21 = w12;
w13 = dist(x1,x3);
w31 = w13;
w23 = dist(x2,x3);
w32 = w23;
w14 = dist(x1,x4);
w41 = w14;
w24 = zeros(1,K);
w42 = w24;
w34 = zeros(1,K);
w43 = w34;

% Simulazione numerica
for k = 1:K-1

    q1(:,k+1) = q1(:,k) + h*(q3(:,k) - q1(:,k)) + h
        *(q4(:,k) - q1(:,k));
    q2(:,k+1) = q2(:,k) + h*(q3(:,k) - q2(:,k));
    q3(:,k+1) = q3(:,k) + h*(q1(:,k) - q3(:,k)) + h
        *(q2(:,k) - q3(:,k));
    q4(:,k+1) = q4(:,k) + h*(q1(:,k) - q4(:,k));

    if dist(x1(:,k),x2(:,k)) <= th
        w12(k) = dist(x1(:,k),x2(:,k));
        w21(k) = w12(k);
    end
    if dist(x2(:,k),x4(:,k)) <= th
        w24(k) = dist(x2(:,k),x4(:,k));
        w42(k) = w24(k);
    end
    if dist(x3(:,k),x4(:,k)) <= th
        w34(k) = dist(x3(:,k),x4(:,k));
        w43(k) = w34(k);
    end

    w13(k) = dist(x1(:,k),x3(:,k));
    w31(k) = w13(k);

```

```

w23(k) = dist(x2(:,k),x3(:,k));
w32(k) = w23(k);
w14(k) = dist(x1(:,k),x4(:,k));
w41(k) = w14(k);

x1(:,k+1) = x1(:,k) + h*w13(k)*(x3(:,k) - x1(:,k)
    ) + h*w14(k)*(x4(:,k) - x1(:,k)) ...
    + h*w12(k)*(x2(:,k)-x1(:,k));
x2(:,k+1) = x2(:,k) + h*w23(k)*(x3(:,k) - x2(:,k)
    ) + h*w21(k)*(x1(:,k)-x2(:,k)) ...
    + h*w24(k)*(x4(:,k)-x2(:,k));
x3(:,k+1) = x3(:,k) + h*w31(k)*(x1(:,k) - x3(:,k)
    ) + h*w32(k)*(x2(:,k) - x3(:,k)) ...
    + h*w34(k)*(x4(:,k)-x3(:,k));
x4(:,k+1) = x4(:,k) + h*w41(k)*(x1(:,k) - x4(:,k)
    ) + h*w42(k)*(x2(:,k)-x4(:,k)) ...
    + h*w43(k)*(x3(:,k)-x4(:,k));
end

```

Di seguito riportiamo la matrice Laplaciana alla fine della missione:

$$L_{2f} = \begin{pmatrix} 0.1451 & -0.0519 & -0.0477 & -0.0455 \\ -0.0519 & 0.149 & -0.0448 & -0.0523 \\ -0.0477 & -0.0448 & 0.1446 & -0.0521 \\ -0.0455 & -0.0523 & -0.0521 & 0.1499 \end{pmatrix} \quad (1.29)$$

La posizione di rendezvous è la medesima per entrambi gli scenari ed è pari alla media delle posizioni iniziali degli agenti, come ci si aspettava:

$$q^* = (15.25, 10.25, 31). \quad (1.30)$$

Nella Figura 1.11 sono mostrate le traiettorie seguite dagli agenti nello scenario con rete non pesata (linee continue) e nello scenario con pesi temporvarianti (linee tratteggiate).

Più nel dettaglio, nelle Figure 1.12, 1.13, 1.14, 1.15, sono illustrate le traiettorie dei due scenari lungo le tre coordinate rispettivamente dell'agente 1, agente 2, agente 3 e agente 4.

Notiamo come le traiettorie corrispondenti alla rete tempo-variante siano "più dritte" una volta superata la soglia di prossimità; questo è dovuto all'improvviso arrivo di informazione sulla posizione degli altri agenti che connette tutta la rete.

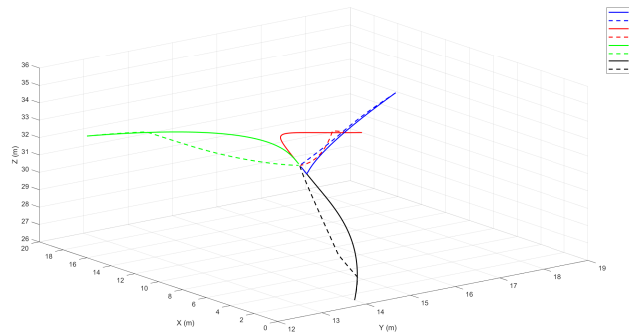


Figura 1.11: Confronto tra le traiettorie di una missione di rendezvous con rete con pesi tempo-varianti (linee tratteggiate) e non-pesata (linee continue). Le traiettorie convergono allo stesso punto ma le traiettorie differiscono tra i due scenari.

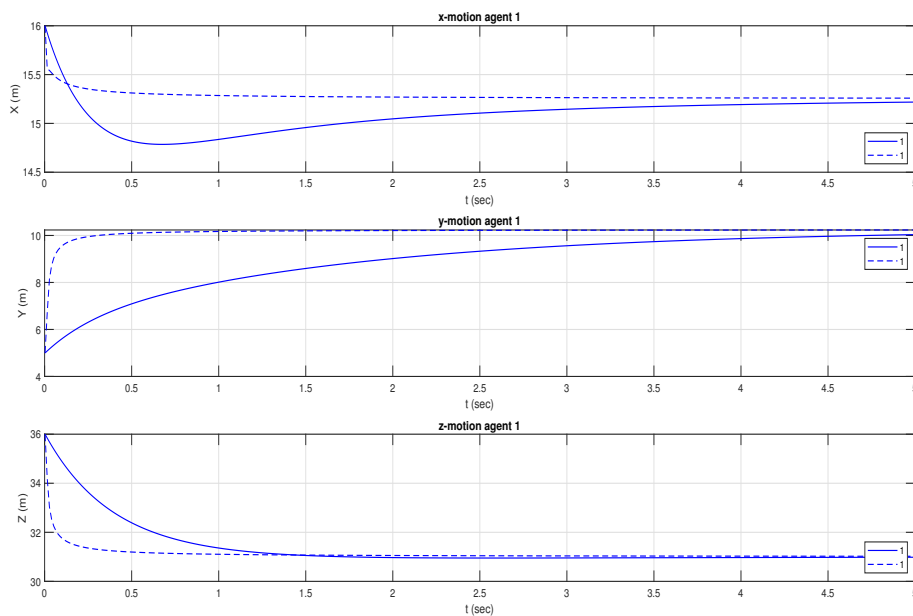


Figura 1.12: Confronto tra i movimenti lungo gli assi x, y e z dell'agente 1 corrispondente alla missione di rendezvous con rete non pesata e con pesi tempo-varianti in Figura 1.11



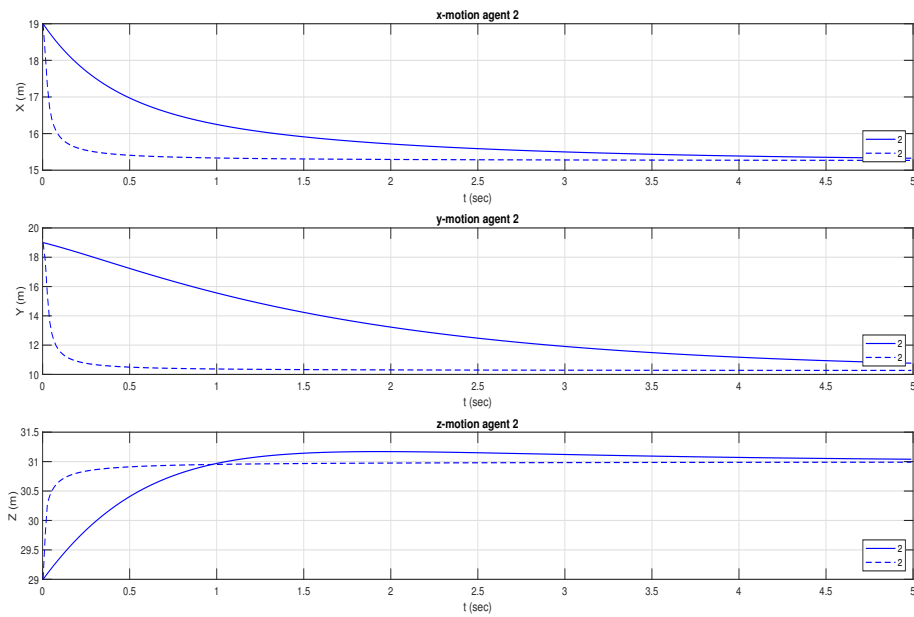


Figura 1.13: Confronto tra i movimenti lungo gli assi x, y e z dell'agente 2 corrispondente alla missione di rendezvous con rete non pesata e con pesi tempo-varianti in Figura 1.11

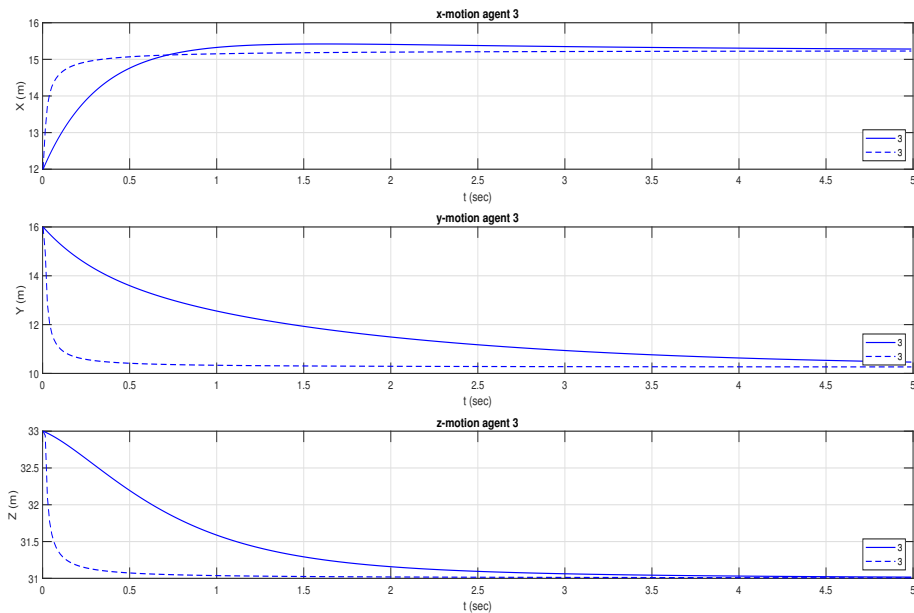


Figura 1.14: Confronto tra i movimenti lungo gli assi x, y e z dell'agente 3 corrispondente alla missione di rendezvous con rete non pesata e con pesi tempo-varianti in Figura 1.11

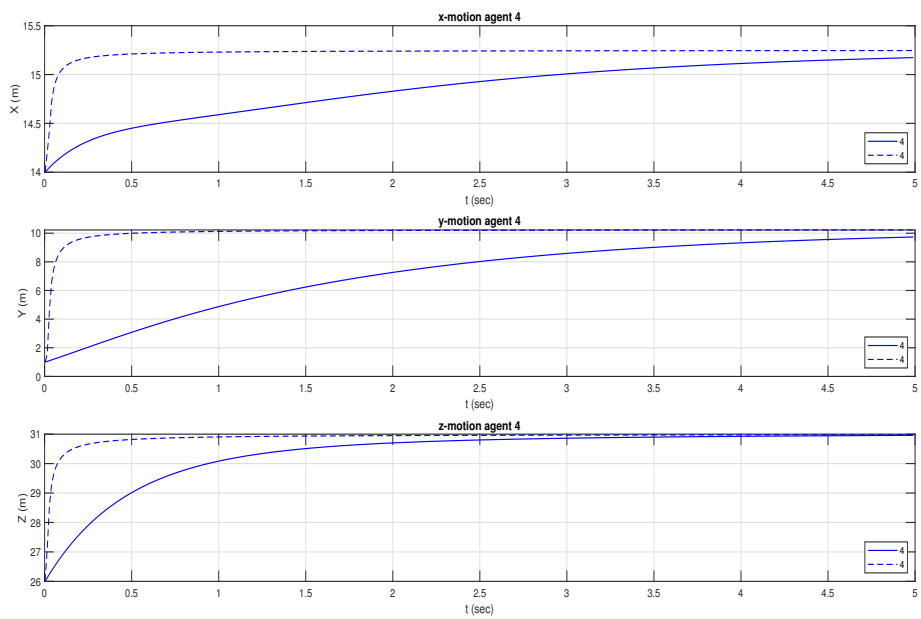


Figura 1.15: Confronto tra i movimenti lungo gli assi  $x$ ,  $y$  e  $z$  dell'agente 4 corrispondente alla missione di rendezvous con rete non pesata e con pesi tempo-varianti in Figura 1.11

# Capitolo 2

## Dinamica dei Quadricotteri

In questo capitolo verranno ricavate le equazioni del moto di un drone quadricottero grazie a certe semplificazioni realistiche riguardanti la forma del velivolo. Queste equazioni saranno necessarie per effettuare una simulazione di una missione di rendezvous che coinvolga dei droni quadricotteri realistici. Fornendoci del formalismo geometrico di [7], verranno presentate queste equazioni nella forma sia del primo che del secondo ordine. In questo capitolo ci avvarremo di notazioni differenti rispetto al Capitolo 1.

### 2.1 Notazioni fisiche di base

In questa prima sezione verrà fornita la derivazione standard delle equazioni di moto di un corpo rigido. Da [3], [4], un corpo rigido può modellare dei veicoli nello spazio o in un fluido che possono essere stabilizzati tramite eliche rotanti. Nel nostro caso, consideriamo il drone come nient'altro che un corpo rigido immerso in un fluido. Inoltre, seguiremo fortemente le derivazioni in [9], [14].

Lo spazio di configurazione per un corpo rigido in un fluido [2] è

$$Q = \mathbb{SE}(3) \cong \mathbb{R}^3 \times \mathbb{SO}(3),$$

ovvero lo *Special Euclidean Group* in 3 dimensioni. Questo spazio è un Gruppo di Lie, ovvero un manifold fornito di un'operazione binaria detta *moltiplicazione*, un *elemento neutro* rispetto alla moltiplicazione e un operazione di *inversione* rispetto alla moltiplicazione. Per l'elemento neutro, denotato con  $e \in \mathbb{SE}(3)$ , lo spazio tangente adesso associato  $T_e\mathbb{SE}(3)$  è detto *algebra di Lie*, e riveste un ruolo fondamentale nella teoria di Lie. Per  $\mathbb{SE}(3)$ , l'algebra di Lie si indica con

$$\mathfrak{se}(3) := T_e\mathbb{SE}(3)$$

Per semplicità, possiamo considerare questo gruppo come la combinazione tra lo spazio euclideo  $\mathbb{R}^3$  e il manifold  $\mathbb{SO}(3)$ , detto manifold delle matrici delle rotazioni, i cui punti sono delle matrici ortogonali di dimensione  $3 \times 3$ . Perciò, forniamo il nostro spazio  $Q$  delle coordinate  $(b, R)$ , dove  $b \in \mathbb{R}^3$  rappresenta la posizione del centro di massa del corpo rispetto ad un sistema di riferimento inerziale, e dove  $R \in \mathbb{SO}(3)$  è una matrice di rotazione che indica l'orientazione del sistema di riferimento fisso al corpo rispetto al sistema di riferimento inerziale.

Inoltre, faremo uso regolare della notazione dell'operatore *hat*, ovvero l'isomorfismo dell'algebra di Lie,

$$\hat{\cdot}: (\mathbb{R}^3, \times) \rightarrow (\mathfrak{so}(3), [,])$$

definito da  $\hat{y}z = y \times z$ , il che è equivalente a  $\hat{y} = \begin{pmatrix} 0 & -y_3 & y_2 \\ y_3 & 0 & -y_1 \\ -y_2 & y_1 & 0 \end{pmatrix}$ .

Seguiremo le seguenti notazioni [8]

- $v$  = velocità di traslazione rispetto sistema di riferimento fisso al corpo
- $\Omega$  = velocità angolare rispetto sistema di riferimento fisso al corpo
- $p$  = quantità di moto nel sistema di riferimento inerziale
- $\pi$  = momento angolare nel sistema di riferimento inerziale
- $P$  = quantità di moto nel sistema di riferimento fisso al corpo
- $\Pi$  = momento angolare nel sistema di riferimento fisso al corpo

Da [8], le equazioni della dinamica prendono la forma di un sistema di equazioni differenziali nelle coordinate  $(b, R, v, \Omega) \in TQ$ , dove  $TQ$  è l'insieme degli spazi tangenti relativi a  $Q$ , e sono:

$$\dot{b} = Rv \tag{2.1}$$

$$\dot{R} = R\hat{\Omega} \tag{2.2}$$

e trasformiamo i momenti tra i due sistemi di riferimento con:

$$p = RP \tag{2.3}$$

$$\pi = R\Pi + \hat{b}p \tag{2.4}$$

Siano ora

$f_i$  = forze esterne,  $i = 1, \dots, k$

$\tau_i$  = coppie esterne,  $i = 1, \dots, l$ ,

allora

$$E_F = \sum_{i=1}^k R^t f_i = \text{forza esterna totale nel sistema di riferimento fisso al corpo.} \quad (2.5)$$

$$E_T = \sum_{i=1}^l R^t \tau_i = \text{coppia esterna totale nel sistema di riferimento fisso al corpo.} \quad (2.6)$$

Differenziando le equazioni 2.3 e 2.4 otteniamo

$$\dot{p} = \dot{R}P + R\dot{P} \quad (2.7)$$

$$\dot{\pi} = \dot{R}\Pi + R\dot{\Pi} + \dot{b} \times p + b \times \dot{p} \quad (2.8)$$

Ora, nel sistema di riferimento inerziale la dinamica abbiamo

$$\dot{p} = \sum_{i=1}^k f_i \quad (2.9)$$

$$\dot{\pi} = \sum_{i=1}^k \hat{x}_i f_i + \sum_{i=1}^l \tau_i, \quad (2.10)$$

dove  $\hat{x}_i$  è il vettore dall'origine del sistema di riferimento inerziale al punto di applicazione di  $f_i$ . Da [8], risolvendo 2.7 e 2.8 rispetto a  $\dot{P}$  e  $\dot{\Pi}$  si ottiene

$$\dot{P} = \hat{P}\Omega + E_F \quad (2.11)$$

$$\dot{\Pi} = \hat{\Pi}\Omega + \hat{P}v + E_T + R^t \sum_i (x_i - b) \times f_i. \quad (2.12)$$

Per trovare delle equazioni solamente in termini di quantità di moto e momento angolare è possibile ricorrere alla trasformata di Legendre

$$P = \frac{\partial T}{\partial v}, \quad \Pi = \frac{\partial T}{\partial \Omega}.$$

Scrivendo in maniera esplicita l'energia cinetica  $T$ , introduciamo

$$T_{body} = \frac{1}{2} \begin{pmatrix} v \\ \Omega \end{pmatrix}^t \begin{pmatrix} mI_3 & -m\hat{r}_{CG} \\ m\hat{r}_{CG} & J_b \end{pmatrix} \begin{pmatrix} v \\ \Omega \end{pmatrix} \quad (2.13)$$

$$T_{fluid} = \frac{1}{2} \begin{pmatrix} v \\ \Omega \end{pmatrix}^t \begin{pmatrix} M_f & C_f^t \\ C_f & J_f \end{pmatrix} \begin{pmatrix} v \\ \Omega \end{pmatrix} \quad (2.14)$$

$$T = T_{body} + T_{fluid} = \frac{1}{2} \begin{pmatrix} v \\ \Omega \end{pmatrix}^t \begin{pmatrix} mI_3 + M_f & -m\hat{r}_{C_G} + C_f^t \\ m\hat{r}_{C_G} + C_f & J_b + J_f \end{pmatrix} \begin{pmatrix} v \\ \Omega \end{pmatrix} \quad (2.15)$$

dove  $m$  è la massa del drone,  $\hat{r}_{C_G}$  è il vettore dal centro di gravità all'origine del sistema fisso al corpo,  $J_b$  è il tensore di inerzia del corpo,  $J_f$  è il tensore di inerzia della massa aggiunta,  $M_f$  è la massa aggiunta,  $C_f$  sono i termini diagonali aggiunti e  $I_3$  è la matrice identità  $3 \times 3$ . Differenziando l'energia cinetica totale 2.15, abbiamo

$$P = \frac{\partial T}{\partial v} = (mI_3 + M_f)v + (-m\hat{r}_{C_G} + C_f^t)\Omega \quad (2.16)$$

$$\Pi = \frac{\partial T}{\partial \Omega} = (m\hat{r}_{C_G} + C_f)v + (J_b + J_f)\Omega \quad (2.17)$$

Queste equazioni trasformano le velocità di traslazione e angolari in rispettivamente quantità di moto e momenti angolari.

Faremo ora delle assunzioni realistiche sulla forma del drone.

**Assumption 1.** *Assumiamo che il corpo abbia tre piani di simmetria e che questi coincidano con gli assi del sistema di riferimento fisso al corpo stesso. Ciò implica che  $J_b$ ,  $J_f$  e  $M_f$  sono diagonali e che  $C_f = 0$ . Inoltre, estendiamo questa simmetria anche alla densità di massa del corpo.*

**Assumption 2.** *Supponiamo che il centro di gravità del corpo coincida con l'origine del sistema di riferimento fisso al corpo stesso. Allora  $\hat{r}_{C_G} = 0$ .*

Grazie a queste due assunzioni, le equazioni 2.16 e 2.17 diventano

$$P = Mv \quad (2.18)$$

$$\Pi = J\Omega, \quad (2.19)$$

dove  $M = mI_3 + M_f$  è la massa totale del sistema e  $J = J_b + J_f$  è il momento d'inerzia del sistema.

Ora, differenziando 2.18 e 2.19 utilizzando 2.11 e 2.12, abbiamo

$$m\dot{v} = Mv \times \Omega + E_F \quad (2.20)$$

$$J\dot{\Omega} = J\Omega \times \Omega + Mv \times v + E_T + R^t \sum_i (x_i - b) \times f_i, \quad (2.21)$$

ovvero le accelerazioni espresse solamente in termini di posizione e velocità, senza momenti.

Infine, poiché il fluido in cui è immerso il nostro drone è l'aria, facciamo la seguente assunzione:

**Assumption 3.** *Supponiamo che il fluido aggiunto sia trascurabile. Allora  $M_f = J_f = 0$ . Ciò implica  $J = J_b$ , e poiché  $M = mI_3$  è una matrice scalare, abbiamo  $Mv \times v = 0$ .*

Combinando questa semplificazione con 2.1, 2.2, 2.20 e 2.21, otteniamo il sistema di equazioni differenziali del primo ordine su  $TQ$  nelle coordinate  $(b, R, v, \Omega)$  che descrivono il movimento del drone immerso nell'aria:

$$\dot{b} = Rv \quad (2.22)$$

$$\dot{R} = R\hat{\Omega} \quad (2.23)$$

$$m\dot{v} = mv \times \Omega + E_F \quad (2.24)$$

$$J\dot{\Omega} = J\Omega \times \Omega + E_T + R^t \sum_{i=1}^k (x_i - b) \times f_i. \quad (2.25)$$

## 2.2 Forze e coppie esterne

In questa sezione vogliamo riscrivere in maniera esplicita le forze esterne in  $E_F$  e le coppie esterne in  $E_T$  nel sistema di riferimento fisso al corpo.

Il nostro drone quadricottero è equipaggiato di 4 motori, ognuno dei quali ha velocità angolare  $\omega_i$ ,  $i = 1, \dots, 4$ , che producono una spinta  $t_i = K_r \omega_i^2$ , dove  $K_r$  è il coefficiente di spinta [11]. Le velocità angolari  $\omega_i$  rappresentano le variabili di controllo del nostro sistema.

Per le forze esterne, seguiremo [15]. La forza dovuta all'attrito è

$$f_1 = -diag(v_1 |v_1|, v_2 |v_2|, v_3 |v_3|)C_D,$$

dove  $v_j$  è la  $j$ -esima componente della velocità di traslazione  $v$  e  $C_D$  è il vettore dei coefficienti d'attrito. La forza di gravità è data da

$$f_2 = -mgR^t e_3^I,$$

dove  $g$  è la costante gravitazionale e  $e_3^I$  è il versore con verso opposto rispetto all'attrazione gravitazionale. La forza indotta dalla spinta dei motori è

$$f_3 = \sum_{i=1}^4 e_3^B t_i,$$

dove  $e_3^B$  rappresenta la direzione della spinta prodotta da ogni motore.

Quindi abbiamo  $E_F = f_1 + f_2 + f_3$ .

Notiamo che le forze  $f_1$  e  $f_2$  non producono nessuna coppia poiché i vettori rappresentanti le due forze attraversano il centro di massa del corpo per via delle assunzioni sulla simmetria discusse nella sezione precedente.

Tuttavia, la forza  $f_3$  rappresenta la somma di quattro forze (una per ogni motore), ognuna delle quali genera una coppia. Da [8], la loro somma è data da

$$\tau_f = \begin{pmatrix} K_r d(\omega_3^2 - \omega_1^2) \\ K_r d(\omega_4^2 - \omega_2^2) \\ K_d \sum_{i=1}^4 (-1)^{i+1} \omega_i^2 \end{pmatrix},$$

dove  $K_d$  è il coefficiente di ritardo delle eliche e  $d$  è la distanza tra l'asse di rotazione del motore e il centro di massa del drone.

Esprimiamo le coppie esterne come [6], [15]. La coppia generata dall'attrito è

$$\tau_1 = -diag(\Omega_1 |\Omega_1|, \Omega_2 |\Omega_2|, \Omega_3 |\Omega_3|) C_\tau,$$

dove  $\Omega_j$  è la  $j$ -esima componente della velocità angolare  $\Omega$  nel sistema di riferimento non inerziale e  $C_\tau$  è il vettore dei coefficienti di attrito di rotazione nel sistema di riferimento non inerziale. La coppia dovuta all'effetto giroscopico è

$$\tau_2 = \sum_{i=1}^4 \Omega \times (-1)^{i+1} J_r(0, 0, \omega_i)^t,$$

dove  $J_r$  è il momento d'inerzia di ogni motore. Se assumiamo che ogni motore abbia un'inerzia non banale solo lungo l'asse verticale,  $J_r$  ha solo una componente non nulla che denotiamo con  $\bar{J}_r$ ; allora  $J_r = diag(0, 0, \bar{J}_r)$ . Quindi si ha che  $E_T = \tau_1 + \tau_2$ . Quindi, aggiornando le nostre equazioni della dinamica otteniamo

$$\dot{b} = Rv \tag{2.26}$$

$$\dot{R} = R\hat{\Omega} \tag{2.27}$$

$$m\dot{v} = mv \times \Omega + f_1 + f_2 + f_3 \tag{2.28}$$

$$J\dot{\Omega} = J\Omega \times \Omega + \tau_f + \tau_1 + \tau_2 \tag{2.29}$$

## 2.3 Pianificazione del movimento singolo-agente

In questa sezione verranno analizzati dei semplici movimenti per un quadricottero. Si utilizzeranno le coordinate  $\eta = (b_1, b_2, b_3, \phi, \theta, \psi)$  su  $Q = \mathbb{SE}(3)$  che rappresentano la posizione e l'orientazione del drone rispetto al sistema di riferimento inerziale. Le posizioni  $b_1, b_2, b_3$  sono le coordinate standard in  $\mathbb{R}^3$ .

Gli angoli  $\phi, \theta, \psi$  sono gli angoli di Tait-Bryan, conosciuti in ambito aeronautico come *roll* (*rollio*), *pitch* (*beccheggio*) e *yaw* (*imbardata*), rispettivamente. Nel nostro caso avremo  $(\phi, \theta, \psi) \in (-\pi, \pi) \times (-\pi/2, \pi/2) \times (-\pi, \pi)$ . Queste coordinate su  $Q$  inducono le coordinate naturali per le velocità  $(v, \Omega) = (v_1, v_2, v_3, \Omega_1, \Omega_2, \Omega_3)$ . I seguenti lemmi sono presi da [8].



**Lemma 2.** *Supponiamo che un quadricottero stia effettuando un movimento di imbardata (yaw), senza nessun altro movimento di traslazione o rotazione. Allora*

$$\omega_1 = \omega_3 = \sqrt{\frac{mg}{2K_r} - \omega_2^2} \quad e \quad \omega_4 = \omega_2,$$

dove  $\omega_2$  è un parametro libero.

*Dimostrazione.* La dimostrazione è data da [8]. □

**Lemma 3.** *Supponiamo che un quadricottero esegua il movimento detto "surge", ovvero che si muova solo nella direzione dettata dall'asse  $x$  del riferimento fisso al corpo. Allora*

$$\omega_2 = \frac{1}{3} \left( \sqrt{\frac{6a - 8K_r\omega_4^2}{K_r}} - \omega_4 \right) \quad e \quad \omega_1 = \omega_3 = \frac{1}{2}(\omega_2 + \omega_4),$$

dove

$$a = m \left( \frac{1}{\cos\theta} (\dot{v}_1 \sin\theta + v_1 \cos\theta \Omega_2 + v_3 \sin\theta \Omega_2) - v_1 \Omega_2 + g \cos\theta \right) + v_3 |v_3| C_{D_3},$$

e  $\omega_4$  è un parametro libero.

*Dimostrazione.* Questo risultato è dimostrato in [8]. □

# Capitolo 3

## Simulazioni Numeriche

In questo Capitolo, inizialmente, verranno effettuate delle simulazioni per dimostrare i movimenti descritti dal Lemma 2 e Lemma 3. Successivamente, verrà proposta una tecnica di controllo per il movimento di un quadricottero impegnato in una missione di rendezvous.

### 3.1 Metodo numerico

Prima di mostrare i risultati, verrà illustrato il metodo numerico utilizzato per la simulazione. Poiché in questo caso la dinamica avviene in un gruppo di Lie,  $\mathbb{SE}(3)$ , dobbiamo apportare una modifica all'algoritmo numerico basato sul fEul utilizzato nelle simulazioni del Capitolo 1.

In un gruppo di Lie  $\mathbb{G}$ , e nei manifold in generale, abbiamo a che fare con punti su  $\mathbb{G}$  e vettori tangenti a quel punto.

Dato che non possiamo basarci sulla definizione di derivata destra sui gruppi di Lie, occorre combinare diversamente un punto su  $\mathbb{G}$  e l'incremento che porta al punto successivo. Per i manifold, e quindi i gruppi di Lie, esiste una funzione detta *mappa esponenziale* definita da  $T\mathbb{G} \rightarrow \mathbb{G}$ .

Questa funzione, dato un punto  $x \in \mathbb{G}$  e un vettore tangente  $v \in T_x\mathbb{G}$ , restituisce un punto  $y = \exp_x(v)$  in  $\mathbb{G}$ .

Allora il metodo numerico basato sul fEul esteso ai gruppi di Lie diventa:

$$x_{n+1} = \exp_{x_n}(hf_x(x_n))$$

Per il manifold  $\mathbb{SO}(3)$ , la mappa esponenziale è definita da

$$\exp_R(V) := R \text{Exp}(R^T V),$$

con  $R \in \mathbb{SO}(3)$  e  $V \in T_R\mathbb{SO}(3)$ .

Sulla base di questi risultati, il sistema di equazioni su cui si baserà la simulazione è il seguente:

$$\begin{cases} b_{n+1} = b_n + hf_b(R_n, v_n), \\ R_{n+1} = \exp_{R_n}(hf_R(R_n, \Omega_n)), \\ v_{n+1} = v_n + hf_v(v_n, \Omega_n, R_n, \omega_n), \\ \Omega_{n+1} = \Omega_n + hf_\Omega(\Omega_n, \omega_n), \end{cases}$$

## 3.2 Simulazione singolo-agente

In questa sezione simuleremo il movimento di un drone quadricottero utilizzando il sistema di equazioni differenziali ricavato nel Capitolo 2, il Lemma 2 e il Lemma 3. Utilizzando le coordinate  $\eta = (b_1, b_2, b_3, \phi, \theta, \psi) \in \mathbb{SE}(3)$ , il nostro drone parte inizialmente a  $(0, 0, 0, 0, 0, 0) \in \mathbb{SE}(3)$  con i motori che producono la spinta necessaria per far rimanere il drone in bilico (*hovering*). In questo esperimento, per i primi 12 secondi il drone effettua una salita verticale di circa 150 metri lungo l'asse  $z$  del sistema di riferimento inerziale (e non), mantenendo l'orientazione iniziale giungendo di nuovo in bilico. Nei successivi 4 secondi, il drone mantiene la posizione ruotando di  $\pi/2$  radianti in senso antiorario attorno all'asse  $z$ , come descritto dal Lemma 2. Infine, nei 4 secondi finali, il drone si muove lungo l'asse  $x$  del sistema di riferimento fisso al corpo per circa 9 metri, come illustrato dal Lemma 3.

Nella Figura 3.1 è mostrato il percorso del drone nello spazio, il movimento lungo gli assi del sistema di riferimento inerziale e la sua orientazione nel tempo.

Infine, la Figura 3.2 mostra i controlli utilizzati per realizzare il movimento.

## 3.3 Simulazione multi-agente

In questa sezione insceniamo una missione di rendezvous tra tre droni quadricotteri andando, anche in questo caso, ad effettuare i movimenti definiti da Lemma 2 e Lemma 3. Utilizzando le stesse condizioni iniziali degli esperimenti in [8], ogni drone è inizialmente in bilico con velocità traslazionali e angolari nulle. Ciò implica che anche gli angoli di roll e pitch dei tre droni siano nulli, ma scegliamo angoli di yaw iniziali differenti. Tutti i droni si trovano nel piano  $z = 0$ , ma hanno coordinate  $x$  e  $y$  diverse.

Utilizzando il sistema di coordinate  $(b_1, b_2, b_3, \phi, \theta, \psi) \in \mathbb{SE}(3)$ , le condizioni iniziali dei droni sono  $(0, 0, 0, 0, 0, 0)$ ,  $(0, 9, 0, 0, 0, -\pi/4)$  e  $(15, 9, 0, 0, 0, \pi/2)$  rispettivamente per l'agente 1, agente 2 e agente 3.

Nella Figura 3.3 vengono mostrati i percorsi seguiti dai tre droni, mentre nelle Figure 3.4, 3.5 e 3.6 sono mostrati i movimenti lungo gli assi  $x$ ,  $y$  e  $z$  rispettivamente del drone 1, drone 2 e drone 3.

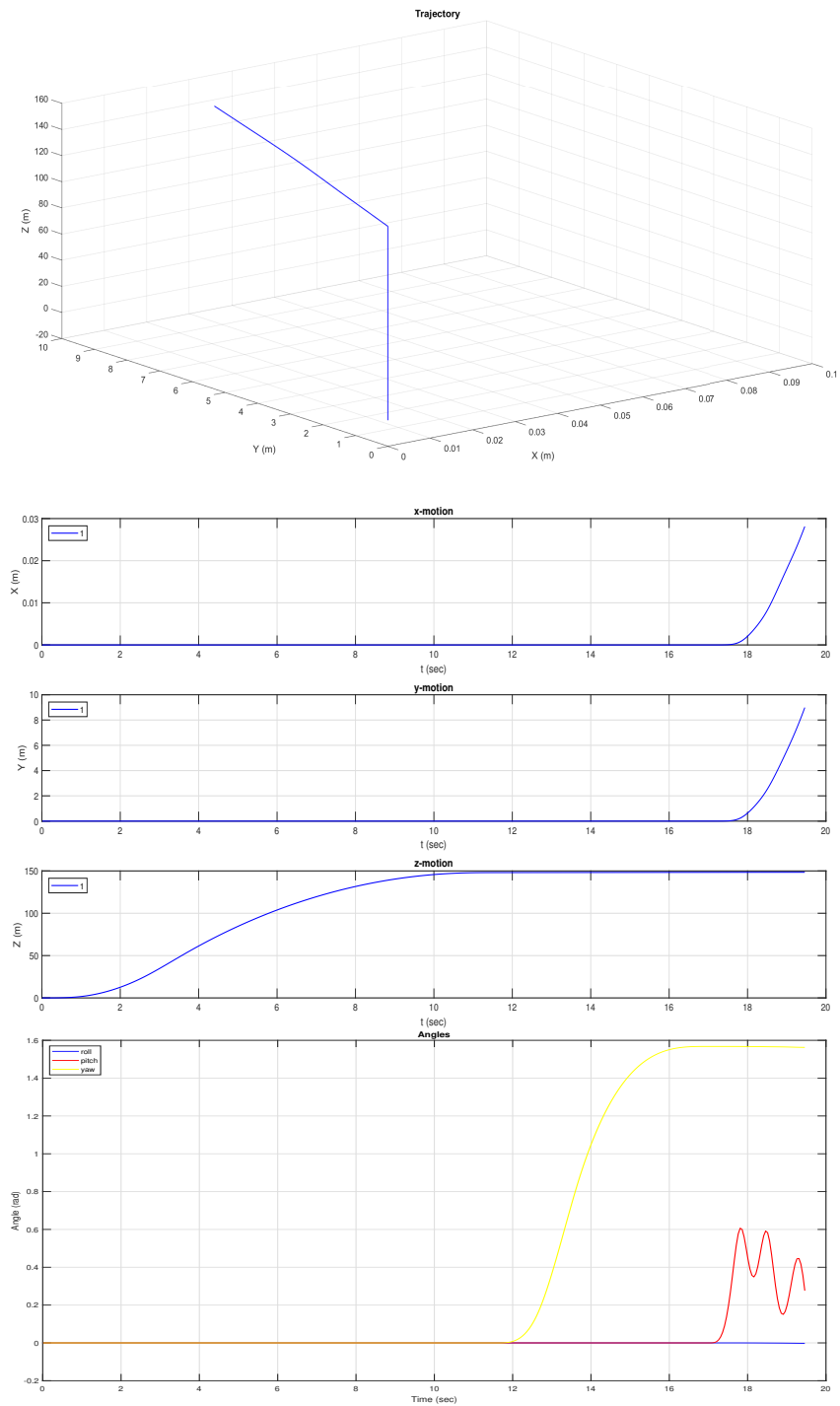


Figura 3.1: Traiettorie del quadricottero che sale verticalmente, effettua una manovra di yaw rimanendo in bilico e si muove orizzontalmente lungo l'asse  $x$  del corpo. In alto: posizione nello spazio. Al centro: movimenti lungo gli assi  $x,y$  e  $z$ . In basso: valore degli angoli roll, pitch e yaw.

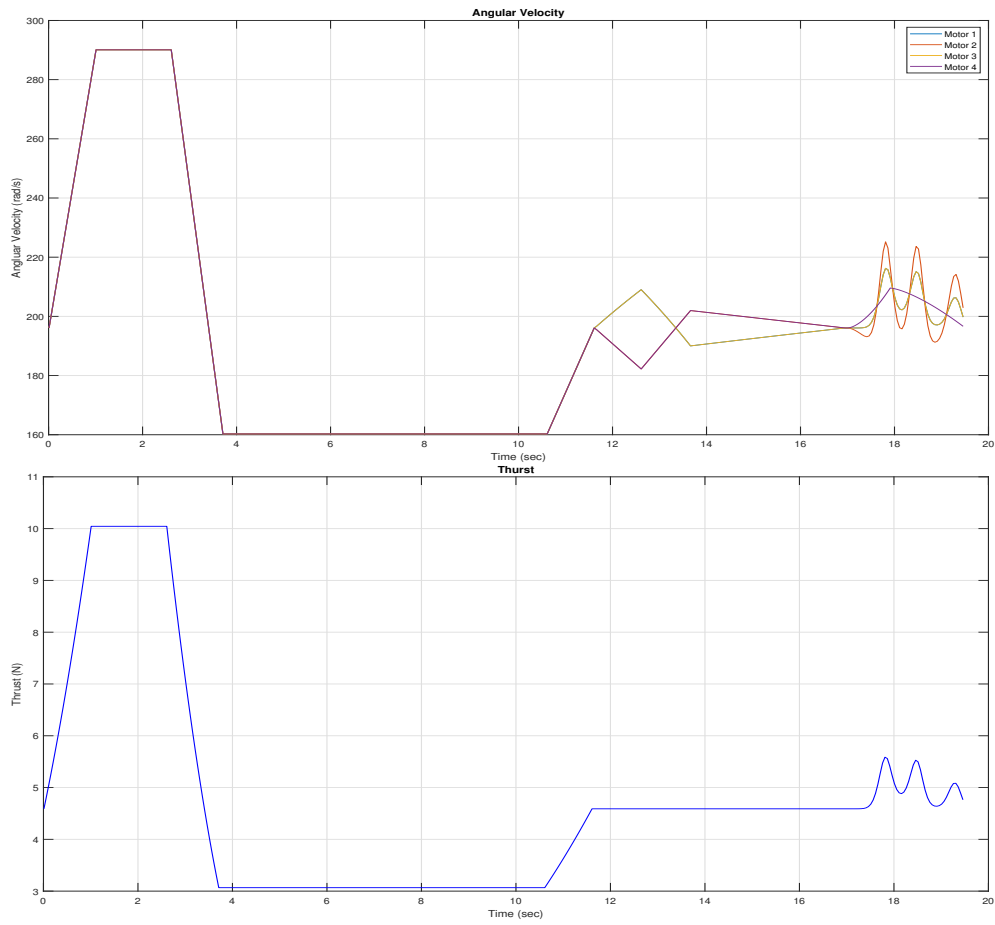


Figura 3.2: Controlli utilizzati per realizzare il movimento in Figura 3.1. In alto: velocità angolari dei quattro motori. In basso: spinta totale generata dai motori nel tempo.

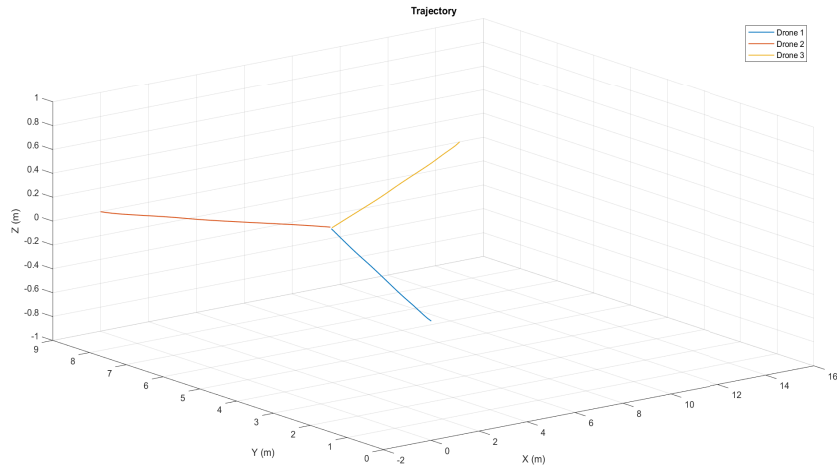


Figura 3.3: Traiettoria nello spazio dei tre droni che partono dalle posizioni  $(0,0,0)$ ,  $(0,9,0)$  e  $(15,9,0)$  con angoli di yaw iniziali  $0$ ,  $-\pi/4$  e  $\pi/2$ .

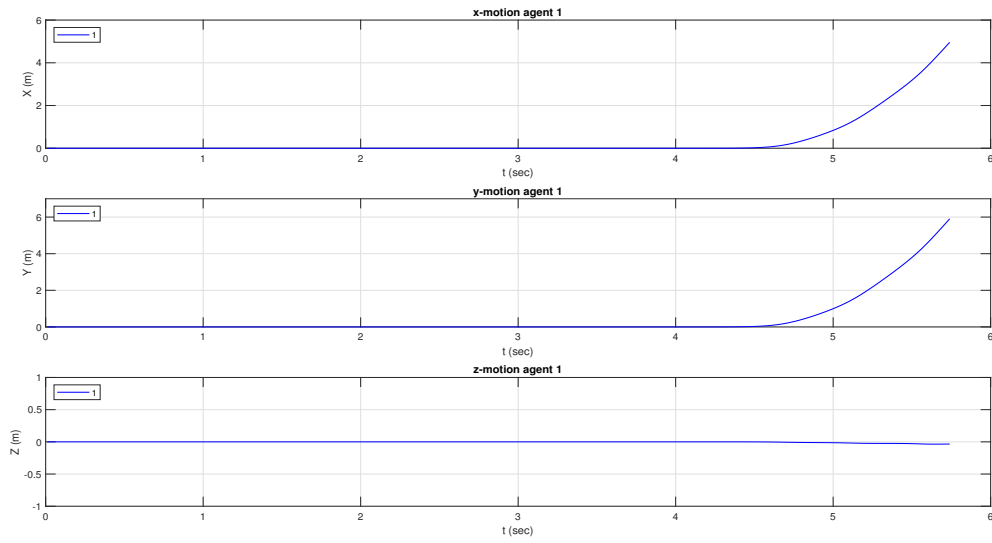


Figura 3.4: Movimento del drone 1 lungo gli assi x, y e z.

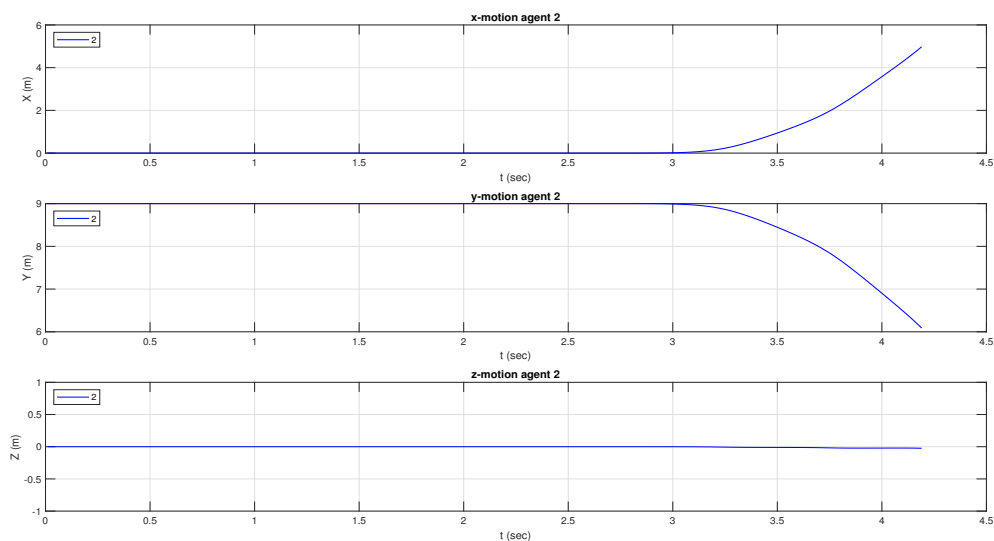


Figura 3.5: Movimento del drone 2 lungo gli assi x, y e z.

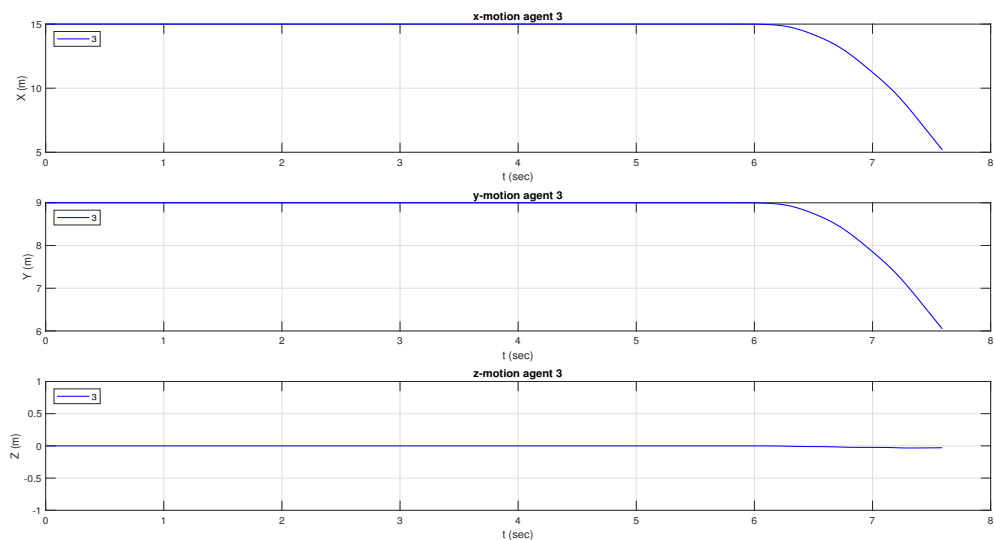


Figura 3.6: Movimento del drone 3 lungo gli assi x, y e z.

Ci sono, tuttavia, due metodi per pianificare il movimento dei droni. Il primo metodo deriva dai risultati teorici ottenuti nel Capitolo 2 e ci permette di ottenere traiettorie che convergono esponenzialmente alla posizione di rendezvous. Tuttavia, questo metodo trascura l'orientazione e la dinamica dei droni, fornendo dei percorsi non realistici. In questo Capitolo è stato invece studiato il secondo metodo; infatti, utilizzando Lemma 2 e Lemma 3, sono stati ricavati i controlli necessari per

Costante	Simbolo	Valore
massa del drone	$m$	0.468 Kg
inerzia del drone	$J$	$\text{diag}((3.8278, 3.8288, 7.6566) \cdot 10^{-3}) \text{ N} \cdot \text{m}^2$
inerzia delle eliche	$J_r$	$\text{diag}(0, 0, 2.8385 \cdot 10^{-5}) \text{ N} \cdot \text{m}^2$
distanza dal motore	$d$	0.25 m
coefficiente di spinta	$K_r$	$2.9842 \cdot 10^{-5}$
coefficiente di attrito di traslazione	$C_D$	$(5.5670, 5.5670, 6.3540) \cdot 10^{-4}$
coefficiente di attrito di rotazione	$C_\tau$	$(5.5670, 5.5670, 6.3540) \cdot 10^{-4}$
ritardo delle eliche	$K_d$	$3.2320 \cdot 10^{-7}$

Tabella 3.1: Parametri usati nella simulazione

permettere ad ogni singolo drone di percorrere la traiettoria desiderata verso la posizione di rendezvous in maniera tale da rispettare la dinamica realistica illustrata nel Capitolo 3. Ciò richiede che il drone esegua come prima cosa una manovra di yaw per orientare il proprio asse  $x$  verso la posizione di rendezvous, per poi volare orizzontalmente verso quest'ultima.

Nella Figura 3.7 sono mostrati gli angoli di yaw per ogni drone durante la simulazione; da notare che il tempo totale di volo differisce per ogni agente. Inoltre, come descritto dal Lemma 3, l'angolo  $\phi$  di roll è nullo per tutti i droni, mentre sono non nulli gli angoli  $\theta$  di pitch, poiché necessari per effettuare un movimento orizzontale lungo l'asse  $x$  del corpo.

I controlli utilizzati - ovvero le velocità angolari dei quattro motori - sono mostrati nella Figura 3.8. Nella Figura 3.9 sono mostrate le spinte totali generate dai 4 motori per ogni drone. Infine, i parametri utilizzati sono presi da [5] e sono mostrati nella Tabella 3.1



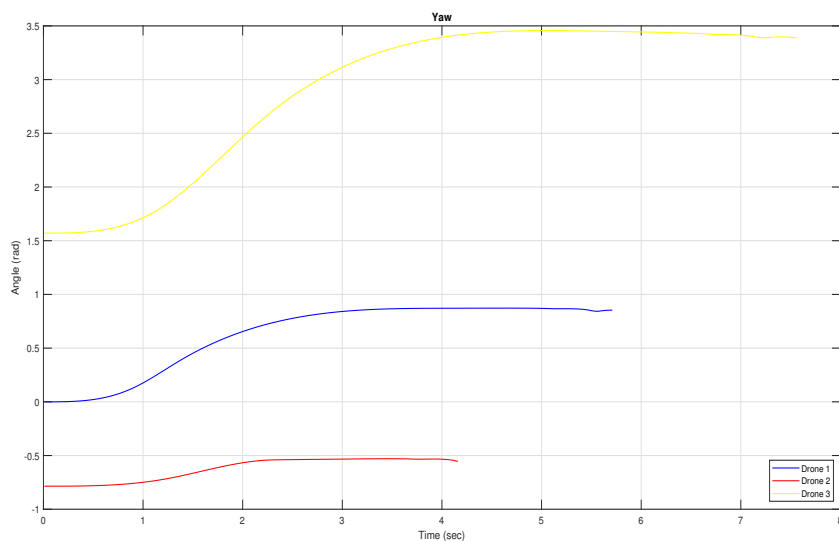


Figura 3.7: Angolo  $\psi$  di yaw nel tempo per ogni drone.

Infine, nella Figura 3.10 vengono confrontati i percorsi generati dai due metodi discussi in questa Sezione. Le curve continue rappresentano i percorsi non realistici derivati dal metodo illustrato nel Capitolo 2. Le curve tratteggiate fanno riferimento ai percorsi realistici in Figura 3.3; entrambi i metodi tracciano le stesse curve di questa figura, ma differiscono nella loro parametrizzazione. Dalla Figura 3.10, vediamo come le curve continue relative al movimento lungo l'asse  $x$  per i droni 1 e 2 coincidono poiché entrambi partono con un valore della coordinata  $x$  pari a 0. Analogamente, per il movimento lungo l'asse  $y$ , le curve relative ai droni 2 e 3 coincidono poiché entrambi partono con un valore iniziale della coordinata  $y$  pari a 9.

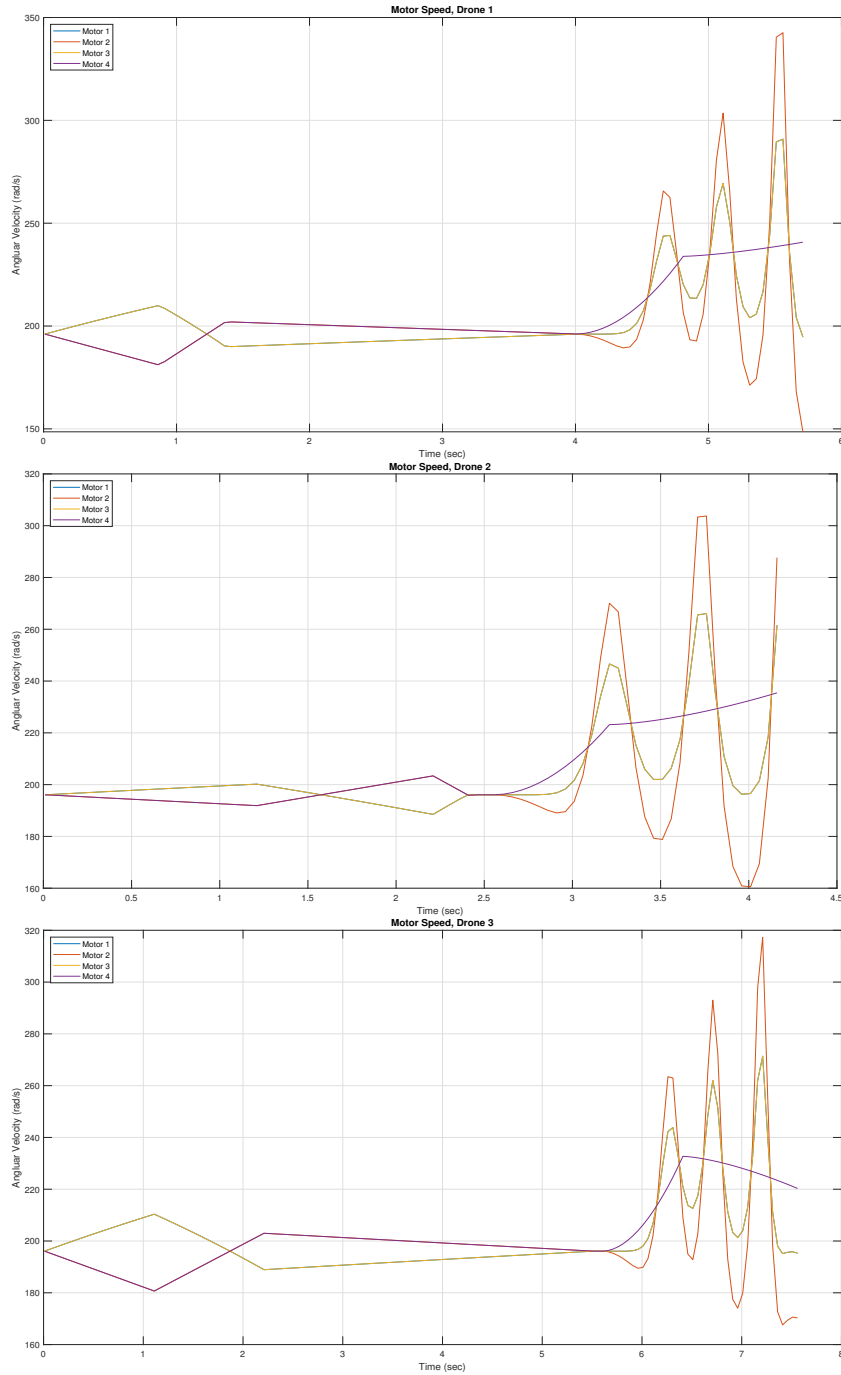


Figura 3.8: Controlli utilizzati per realizzare i movimenti in Figura 3.3. Sono mostrate le velocità angolari dei 4 motori per ogni drone.

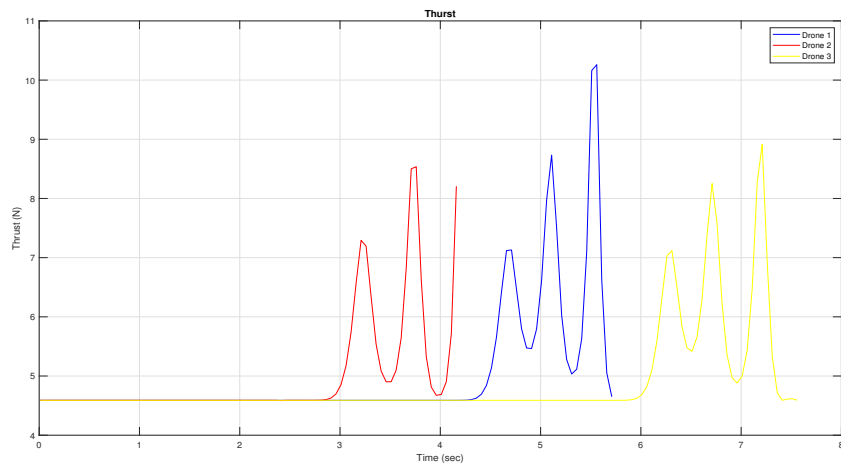


Figura 3.9: Spinta totale generata dai 4 motori per ogni drone, relativa alla Figura 3.8.

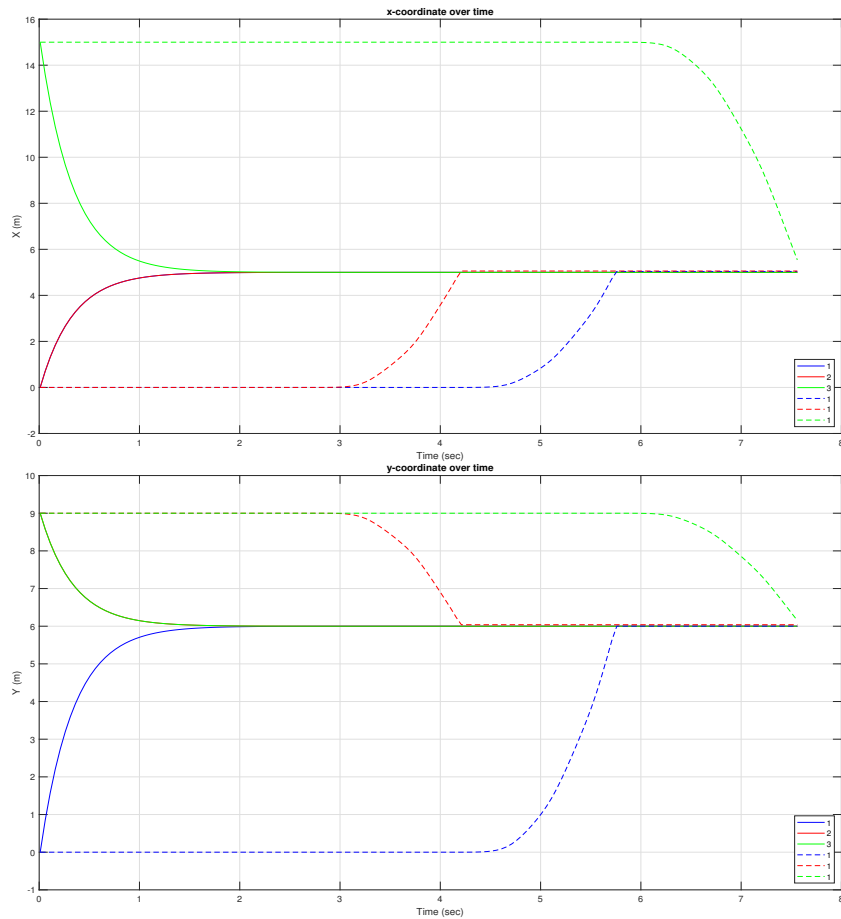


Figura 3.10: Confronto dei due metodi per la pianificazione del volo dei droni. Le curve tratteggiate tengono conto della dinamica realistica; le curve continue trascurano la dinamica. In alto: coordinata  $x$  nel tempo. In basso: coordinata  $y$  nel tempo.

## Capitolo 4

# Controllore per il Movimento del Drone

In questo capitolo verrà presentata una possibile implementazione di un controllore per la pianificazione del movimento del drone; salita verticale, yaw descritto dal Lemma 2 e movimento orizzontale lungo l'asse  $x$  del riferimento fisso al corpo descritto dal Lemma 3.

Si è voluto proporre questo controllore in quanto gli esperimenti del Capitolo 3 sono stati eseguiti conoscendo a priori la posizione finale di rendezvous, cosa che non avviene nelle reali missioni di questo tipo.

Infatti, supponendo che la rete sia completamente connessa, ogni agente conosce la posizione di tutti gli altri, oltre alla propria, ed effettua i movimenti sulla base di un vettore  $p \in \mathbb{R}^3$  che rappresenta la media non pesata delle posizioni nello spazio degli  $n$  agenti nel sistema di riferimento inerziale.

Per tutti e tre i movimenti, vogliamo imporre alle variabili controllate la dinamica del protocollo di consenso 1.1, presentato nel Capitolo 1.

Per realizzare ciò, interverremo sulle velocità angolari  $\omega_i$ ,  $i = 1, 2, 3, 4$ , dei motori, ovvero le nostre variabili di controllo, seguendo i Lemmi presentati nel Capitolo 3.

### 4.1 Salita verticale (Heave)

Nella salita verticale, la coordinata che vogliamo controllare è la componente  $b_3$  di  $b \in \mathbb{R}^3$ , che rappresenta la posizione del drone nel sistema di riferimento inerziale. In questo caso, supponendo che la rete sia composta da  $n$  agenti e completamente connessa, la dinamica che vogliamo imporre alla variabile

$b_{3,i}$  dell'agente  $i$ -esimo è

$$\dot{b}_{3,i} = \sum_{j \in N(i)} (b_{3,j} - b_{3,i}), \quad 1 \leq i \leq n, \quad (4.1)$$

dove  $N(i)$  è l'insieme degli agenti diversi dall'agente  $i$ .

Utilizzando il vettore  $p$  della media delle posizioni di tutti gli agenti introdotto in precedenza, la 4.1 diventa

$$\dot{b}_{3,i} = p - b_{3,i}, \quad 1 \leq i \leq n. \quad (4.2)$$

Il vettore  $p$  verrà aggiornato ad ogni passo della simulazione, e quindi varierà in base alle nuove posizioni degli agenti.

Il termine a destra dell'equazione precedente rappresenta l'errore di posizione del drone  $i$  lungo l'asse verticale del sistema di riferimento inerziale.

Questo errore verrà moltiplicato per una costante  $K_{GH}$  andando a realizzare un controllo proporzionale. Intervenendo sulle velocità angolari dei motori, verrà impressa una velocità verticale nel sistema inerziale che sarà pari a  $K_{GH}(p - b_3)$ .

## 4.2 Orientazione (Yaw)

Questa volta, nella rotazione attorno all'asse verticale del drone, la variabile controllata è rappresentata dall'angolo  $\psi$  di yaw necessario per orientare l'asse  $x$  del sistema di riferimento fisso al drone con il punto rappresentato dal vettore  $p \in \mathbb{R}^3$  introdotto nella sezione precedente.

Sulla base di  $p$  è possibile calcolare l'angolo di yaw desiderato, che indichiamo con  $\psi^*$ ; allora la dinamica che vogliamo imporre all'angolo di yaw  $\psi_i$  dell'agente  $i$ -esimo è

$$\dot{\psi}_i = \psi^* - \psi_i, \quad 1 \leq i \leq n. \quad (4.3)$$

Anche in questo caso, l'errore di orientazione rappresentato dal termine a destra dell'equazione precedente, viene moltiplicato per una costante  $K_{GY}$  andando a realizzare un controllo proporzionale. Agendo sulle velocità angolari dei quattro motori nella maniera descritta dal Lemma 2, si otterrà una velocità angolare attorno all'asse verticale nel sistema di riferimento fisso al corpo che sarà pari a  $K_{GY}(\psi^* - \psi)$ .

## 4.3 Movimento orizzontale (Surge)

Una volta che gli agenti sono allineati verticalmente e orientati con il proprio asse  $x$  verso la posizione di rendezvous, essi possono muoversi orizzontalmen-

te lungo questa direzione come descritto dal Lemma 3. In questo caso, le coordinate che vogliamo controllare solo le componenti  $b_1$  e  $b_2$  di  $b \in \mathbb{R}^3$  di ogni agente.

Assumendo che la rete sia completamente connessa e utilizzando il vettore della media delle posizioni degli agenti  $p \in \mathbb{R}^3$ , che viene aggiornato ad ogni passo della simulazione con le nuove posizioni degli agenti, la dinamica che vogliamo imporre alla posizione nel riferimento inerziale ad un agente  $i$ -esimo è

$$\dot{b}_i = p - b_i, \quad 1 \leq i \leq n. \quad (4.4)$$

Andando ad intervenire sulle velocità angolari dei motori di ogni agente come descritto dal Lemma 3, la velocità di traslazione di ogni drone sarà proporzionale all'errore  $p - b$ .

## 4.4 Risultati

In quest'ultimo esperimento andiamo a testare il controllore appena illustrato. Utilizzando le coordinate  $\eta = (b_1, b_2, b_3, \phi, \theta, \psi) \in \mathbb{SE}(3)$ , ipotizziamo una missione di rendezvous tra tre agenti con rete completamente connessa, non pesata e con le seguenti configurazioni iniziali degli agenti:

$(0, 0, 0, 0, 0, 0)$ ,  $(0, 10, 6, 0, 0, \pi/2)$  e  $(9, 9, 0, 0, 0, -\pi/4)$ .

In questo esperimento, rispetto ai precedenti, sono state effettuate le manovre di salita e di yaw contemporaneamente, per rendere il volo del drone più efficiente. Ciò è stato possibile poiché le costanti  $K_G$  che moltiplicano gli errori erano tali che le velocità angolari dei motori calcolate per le due manovre differivano tra loro di una quantità di due ordini di grandezza inferiori rispetto alle velocità angolari massime raggiunte dai motori dei moderni UAV.

L'algoritmo numerico utilizzato per questa simulazione è lo stesso di quello utilizzato nelle simulazioni del Capitolo 3. La posizione finale di rendezvous misurata è  $(3.02, 6.35, 2.55)$ , che è piuttosto vicina alla media delle posizioni iniziali degli agenti. Nella Figura 4.1 è mostrata la traiettoria percorsa dai tre droni durante la missione, e, di seguito, le posizioni lungo gli assi  $x$ ,  $y$  e  $z$ , gli angoli di roll, pitch e yaw e i controlli utilizzati nelle tre manovre per i tre droni.

I tre movimenti fondamentali appena riportati confermano la correttezza di quanto affermato dal Lemma 2 e dal Lemma 3, in quanto il drone riesce ad effettuare una salita verticale fino all'altitudine desiderata, ad orientarsi

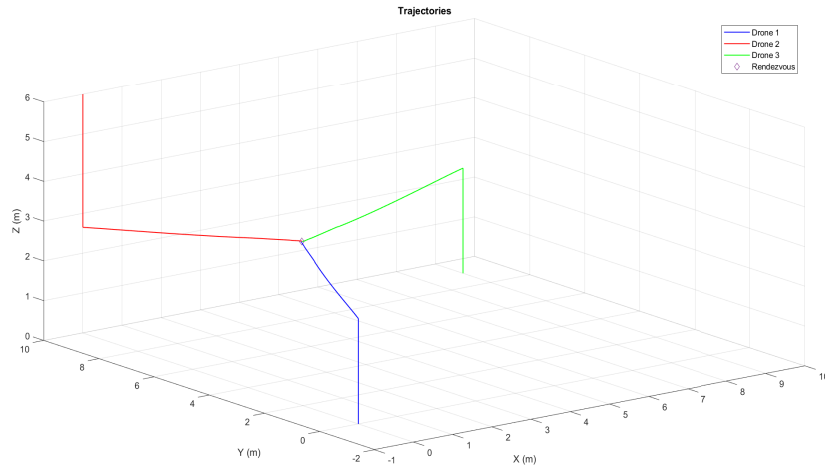


Figura 4.1: Traiettorie nello spazio dei tre droni che partono dalle posizioni  $(0,0,0)$ ,  $(0,10,6)$  e  $(9,9,0)$  con angoli di yaw iniziali  $0$ ,  $\pi/2$  e  $-\pi/4$ . Il rombo rappresenta la posizione finale di rendezvous ottenuta.

verso la posizione finale desiderata e a muoversi orizzontalmente verso di essa.

Tuttavia, nonostante siano state effettuate le manovre di salita e di yaw contemporaneamente, si è visto che non è possibile tramite l'utilizzo del Lemma 2 e del Lemma 3 fare lo stesso con una manovra di yaw e un movimento orizzontale, poiché verrebbero a mancare le premesse dei lemmi.



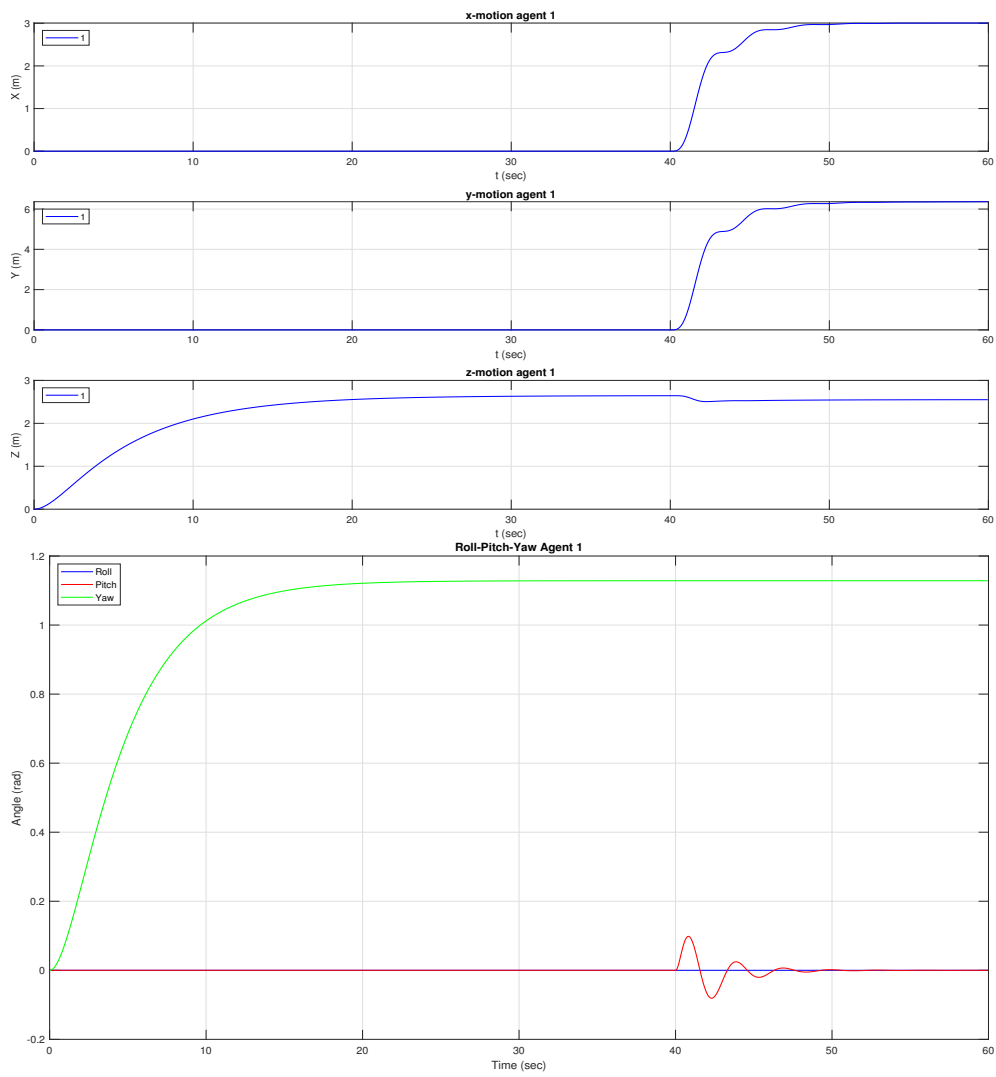


Figura 4.2: Coordinate x, y e z e angoli di roll, pitch e yaw del drone 1 nel tempo.

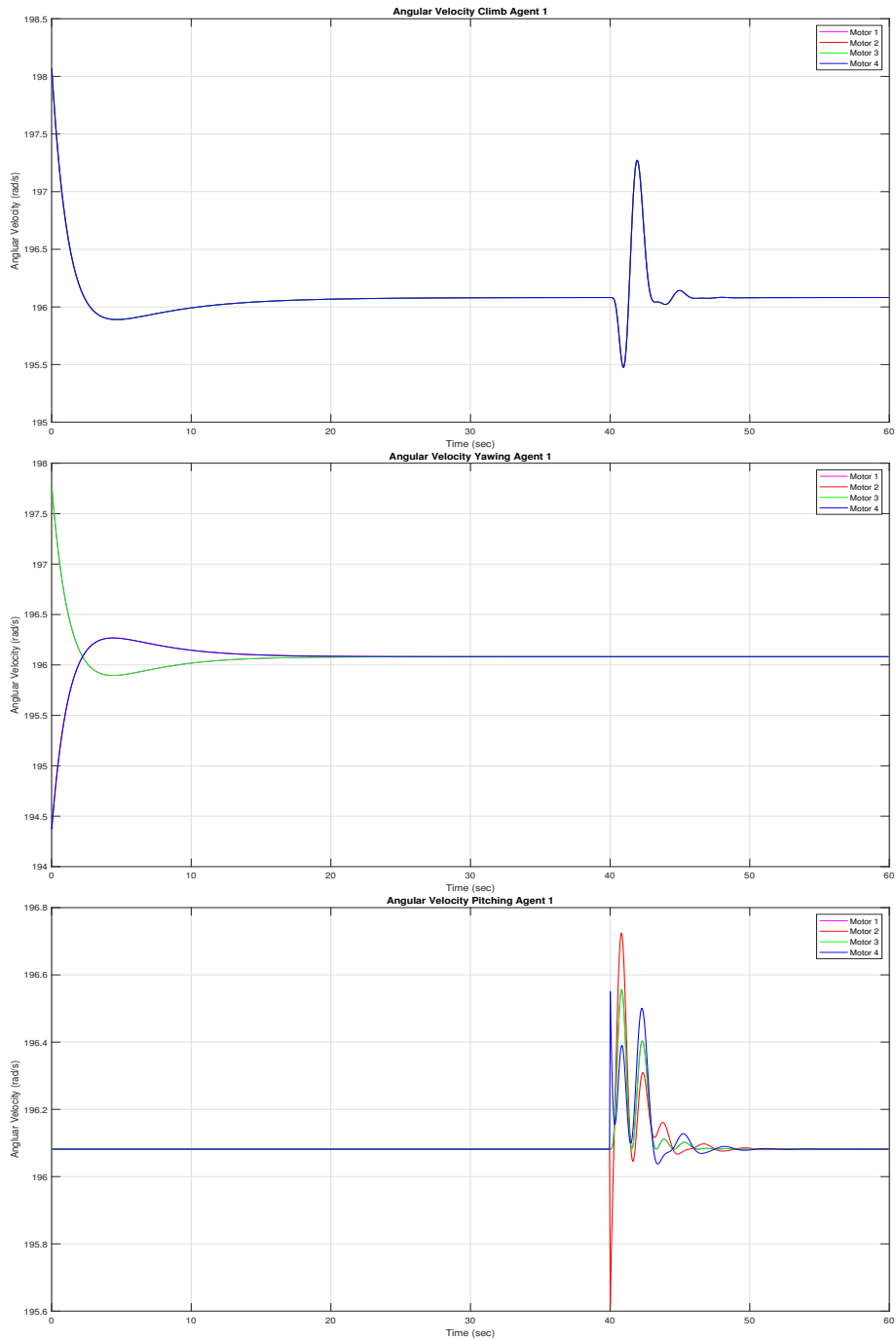


Figura 4.3: In alto: controlli utilizzati per la manovra di salita verticale del drone 1. Al centro: controlli utilizzati per la manovra di yaw del drone 1. In basso: controlli utilizzati per il movimento orizzontale del drone 1.

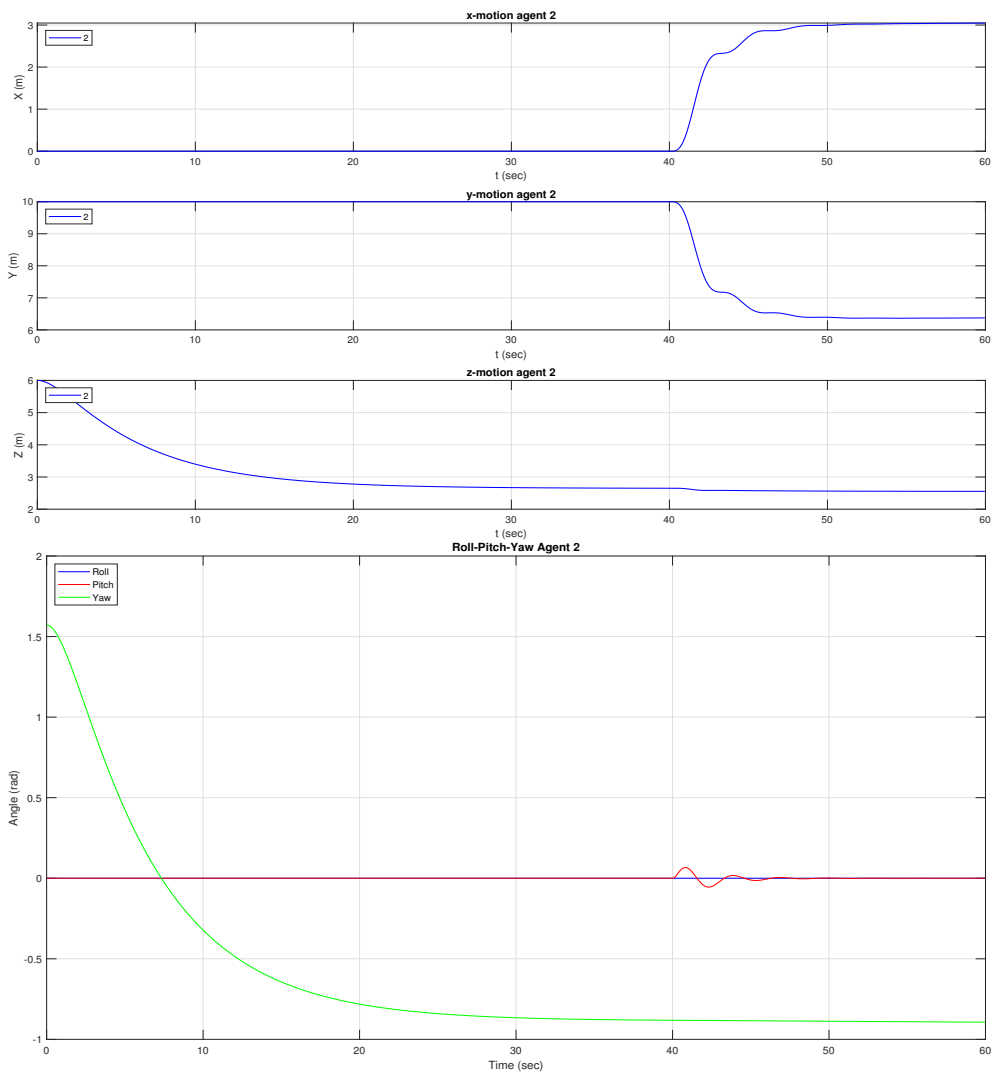


Figura 4.4: Coordinate x, y e z e angoli di roll, pitch e yaw del drone 2 nel tempo.

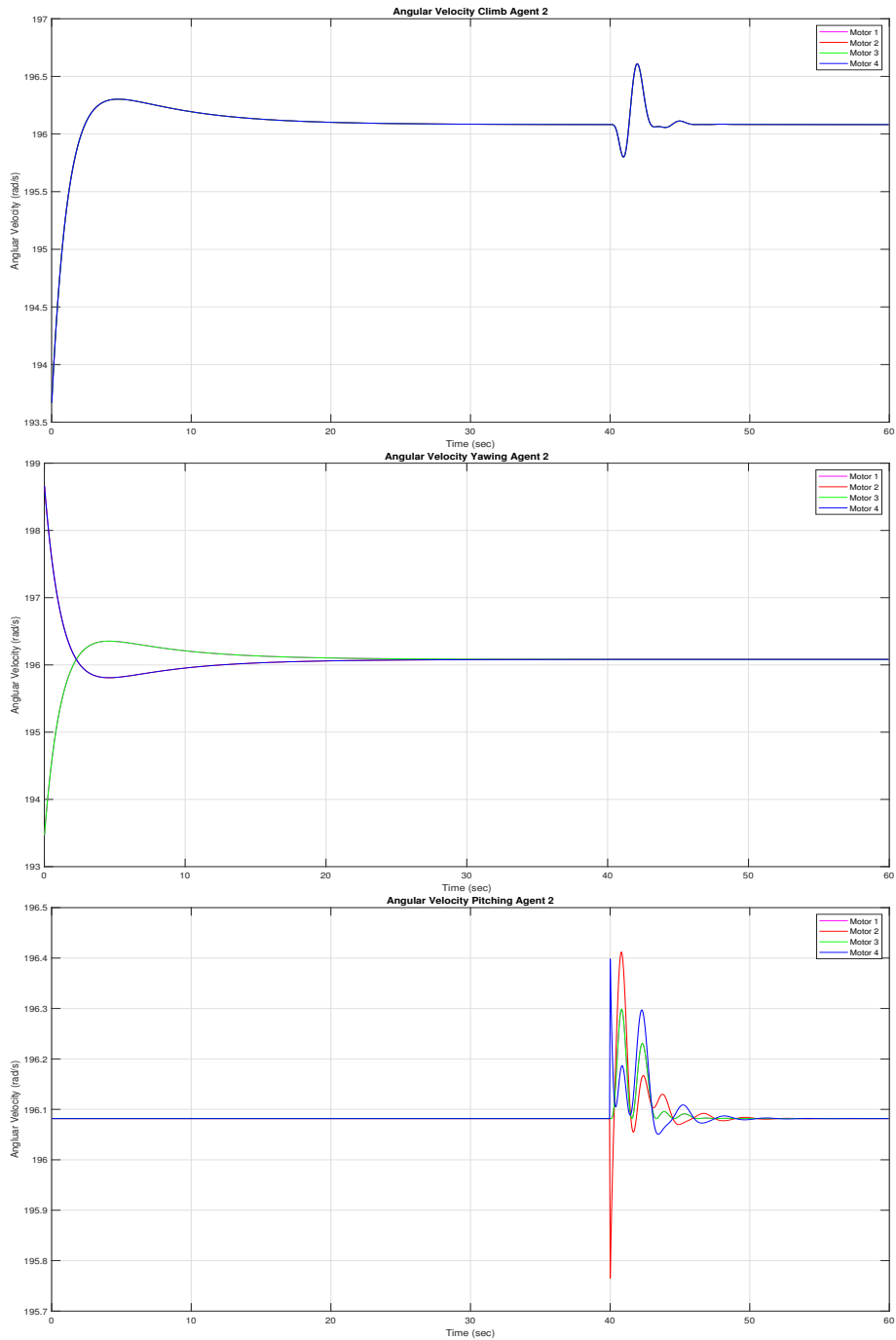


Figura 4.5: In alto: controlli utilizzati per la manovra di salita verticale del drone 2. Al centro: controlli utilizzati per la manovra di yaw del drone 2. In basso: controlli utilizzati per il movimento orizzontale del drone 2.

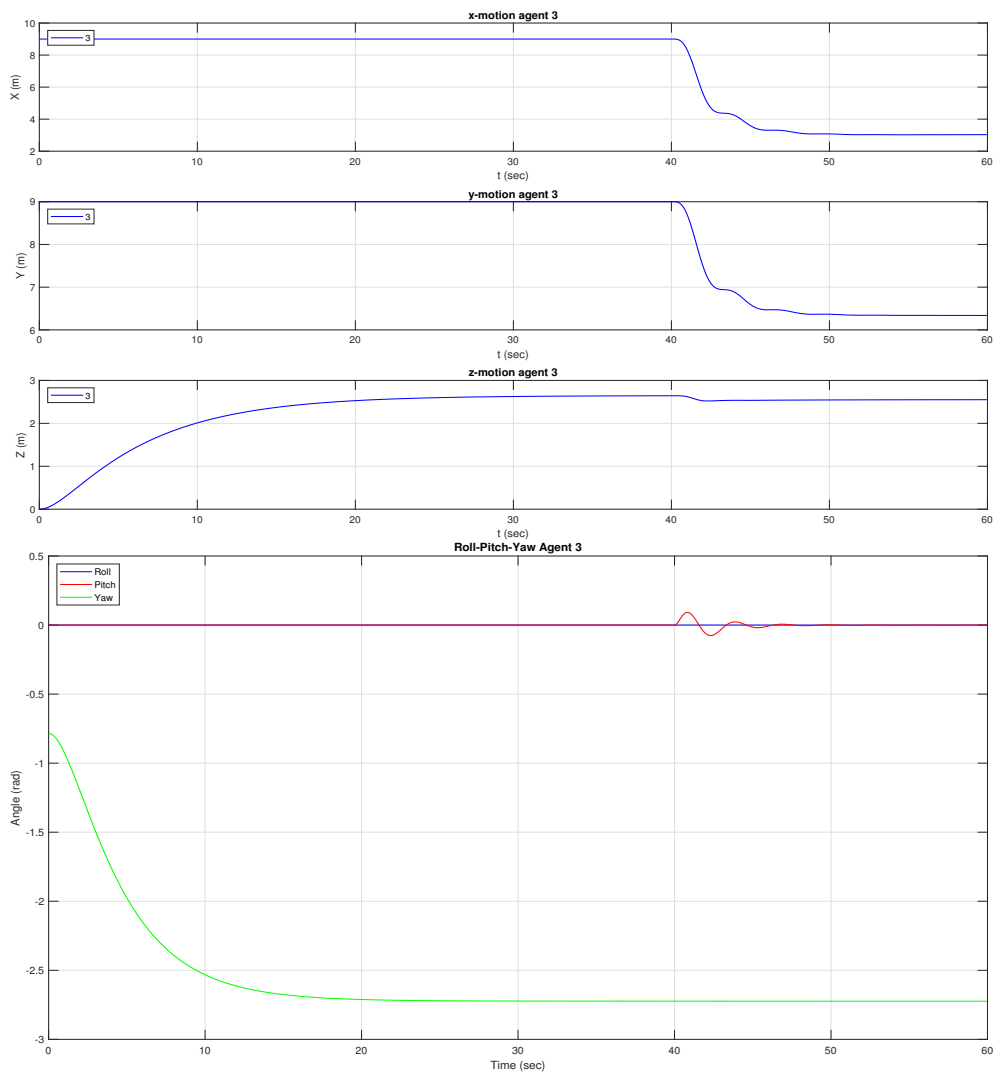


Figura 4.6: Coordinate x, y e z e angoli di roll, pitch e yaw del drone 3 nel tempo.

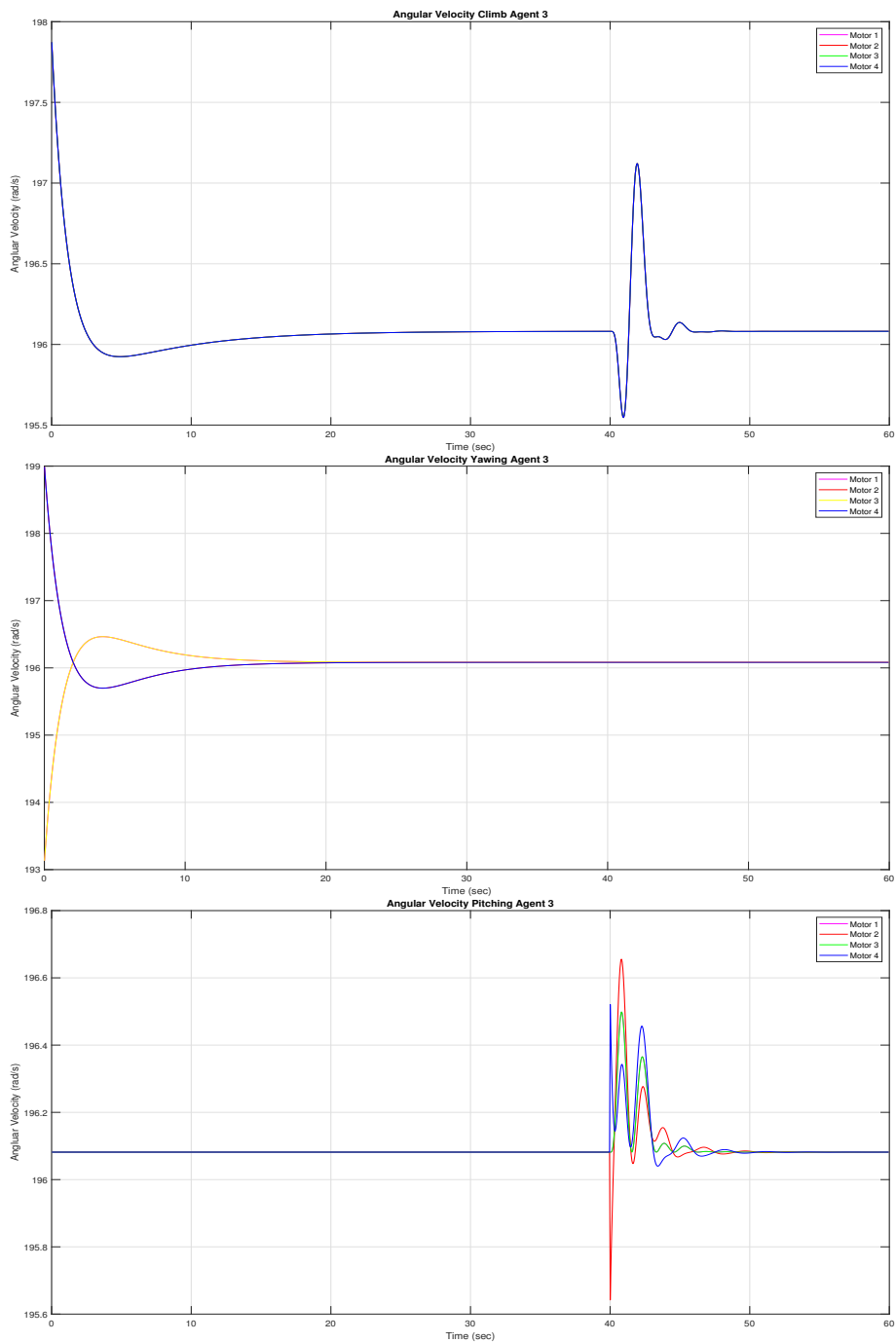


Figura 4.7: In alto: controlli utilizzati per la manovra di salita verticale del drone 3. Al centro: controlli utilizzati per la manovra di yaw del drone 3. In basso: controlli utilizzati per il movimento orizzontale del drone 3.

# Capitolo 5

## Conclusioni

Il primo degli obiettivi dello studio era quello di presentare il protocollo di consenso dal punto di vista teorico, trattando gli agenti come masse puntiformi e trascurando la fisica. In seguito, dopo aver ricavato le equazioni della dinamica per un quadricottero, abbiamo applicato il protocollo ad una missione di rendezvous realistica utilizzando solamente tre movimenti principali; salita, yaw e movimento orizzontale. Si è visto che è possibile ricavare i controlli necessari, ovvero le velocità angolari dei motori, per pianificare il percorso di un drone utilizzando il Lemma 2 e Lemma 3 del Capitolo 2, sia nel caso singolo agente che multi-agente. Infine, è stato presentato un controllore di tipo proporzionale che ha permesso ad una flotta di quadricotteri di raggiungere la posizione di rendezvous svolgendo le tre manovre principali, confermando il fatto che la posizione finale coincide con la media degli stati iniziali degli agenti.

Successivamente, si potrebbero confrontare le traiettorie ricavate dai risultati di questo studio con quelle ottenute nel caso di una rete di agenti non completamente connessa, o con pesi tempo-varianti, magari aggiungendo anche dei disturbi durante il volo dei droni. Un primo miglioramento da apportare potrebbe essere quello di velocizzare la convergenza degli stati degli agenti verso lo stato concordato e prevedere la realizzazione di un controllore che riesca a far compiere al drone tutti e tre i movimenti fondamentali contemporaneamente, come non si è riuscito a fare, o aggiungere movimenti più sofisticati e complessi per rendere la pianificazione del volo non più vincolata a tre soli tipi di movimento. Inoltre, il protocollo di consenso presentato potrebbe essere utilizzato per missioni diverse da quelle di rendezvous considerate in questo studio, come missioni di sincronizzazione e formazione o missioni più complesse come quelle di ricerca e salvataggio.

# Bibliografia

- [1] William N Anderson Jr and Thomas D Morley. Eigenvalues of the laplacian of a graph. *Linear and multilinear algebra*, 18(2):141–145, 1985.
- [2] AM Bloch and B Brogliato. Nonholonomic mechanics and control. *Appl. Mech. Rev.*, 57(1):B3–B3, 2004.
- [3] Anthony M Bloch, Perinkulam Sambamurthy Krishnaprasad, Jerrold E Marsden, and G Sánchez De Alvarez. Stabilization of rigid body dynamics by internal and external torques. *Automatica*, 28(4):745–756, 1992.
- [4] Anthony M Bloch, Naomi Ehrich Leonard, and Jerrold E Marsden. Controlled lagrangians and the stabilization of mechanical systems. i. the first matching theorem. *IEEE Transactions on automatic control*, 45(12):2253–2270, 2000.
- [5] Hakim Bouadi, M Bouchoucha, and M Tadjine. Sliding mode control based on backstepping approach for an uav type-quadrotor. *World Academy of Science, Engineering and Technology*, 26(5):22–27, 2007.
- [6] Hakim Bouadi and M Tadjine. Nonlinear observer design and sliding mode control of four rotors helicopter. *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, 1(7):354–359, 2007.
- [7] Francesco Bullo and Andrew D Lewis. Geometric control of mechanical systems. 2005.
- [8] Richard Carney, Monique Chyba, Chris Gray, Corey Shanbrom, and George Wilkens. Multi-agent systems for quadcopters. *arXiv preprint arXiv:2101.00033*, 2020.



- [9] Monique Chyba, Thomas Haberkorn, Ryan Smith, and George Wilkens. A geometric analysis of trajectory design for underwater vehicles. *Discrete and Continuous Dynamical Systems-Series B*, 11(2):233–262, 2009.
- [10] Gerardo Lafferriere, Alan Williams, J Caughman, and JJP Veerman. Decentralized control of vehicle formations. *Systems & control letters*, 54(9):899–910, 2005.
- [11] Mark W Mueller and Raffaello D’Andrea. Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 45–52. IEEE, 2014.
- [12] Innocent Okoloko. Path planning for multiple spacecraft using consensus with lmi avoidance constraints. In *2012 IEEE Aerospace Conference*, pages 1–8. IEEE, 2012.
- [13] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.
- [14] Ryan N Smith, Monique Chyba, George R Wilkens, and Christopher J Catone. A geometrical approach to the motion planning problem for a submerged rigid body. *Complex Variables and Elliptic Equations*, 82(9):1641–1656, 2009.
- [15] Vahram Stepanyan and Kalmanje S Krishnakumar. Estimation, navigation and control of multi-rotor drones in an urban wind field. In *AIAA information systems-AIAA infotech@ aerospace*, page 0670. 2017.