

Obstacle Avoidance con Potenziali Artificiali per un quadrirotore controllato in Feedback Linearization

Davide Aversa - Vittorio Perera

3 dicembre 2011

Sommario

In questo progetto affronteremo la progettazione e la simulazione di un Quadrirotore UAV dotato di un sistema di navigazione autonomo basato su *Potenziali Artificiali* e controllato in *Feedback Linearization*.

Per prima cosa analizzeremo e simuleremo il controllore con l'ausilio di *MATLAB* e fornendo un modello *Simulink* del sistema.

Nella seconda parte convertiremo il modello in C++ e lo integreremo in *ROS* per fornire una simulazione 3D del quadrirotore all'interno del simulatore *Gazebo*.

Infine, aggiungeremo il modulo di Obstacle Avoidance basato sui Potenziali Artificiali valutandone le performance all'interno dell'ambiente simulato.

1 Modello Dinamico

Nel quadrirotore reale tutti i movimenti sono la conseguenza dalla velocità delle pale, due delle quali ruotano in senso orario (*pale rotanti*) mentre le altre ruotano in senso antiorario (*pale anti-rotanti*). Tali rotazioni generano quattro forze dirette lungo l'asse di rotazione e con modulo proporzionale alla velocità di rotazione

$$F_i = b\Omega_i^2 \quad (1)$$

dove b è il *coefficiente* di thrust delle eliche e Ω_i è la velocità di rotazione della pala i -esima.

Variando la velocità delle pale è possibile quindi variare il modulo di tali forze e, di conseguenza, impartire al quadrirotore i movimenti desiderati. Ad esempio, generare una differenza di velocità tra le pale poste sullo stesso asse causa la rotazione lungo l'asse opposto inclinando il mezzo (*roll* o *pitch*) con conseguente movimento in direzione della pala a velocità minore. Analogamente, tramite la differenza tra la velocità delle pale rotanti e quelle anti-rotanti è possibile far ruotare il quadrirotore lungo il suo asse verticale (*yaw*).

Dal punto di vista del controllo però risulta più semplice considerare queste quattro forze ascensionali come l'unione di una forza diretta verso l'asse verticale del quadrirotore applicata nel suo centro di massa (U_1) e di tre coppie di

rotazione lungo *roll*, *pitch* e *yaw* (U_2 , U_3 e U_4). La trasformazione in question è data dalle seguenti equazioni:

$$\begin{aligned} U_1 &= F_1 + F_2 + F_3 + F_4; \\ U_2 &= F_3 - F_1; \\ U_3 &= F_4 - F_2; \\ U_4 &= F_4 + F_2 - F_3 - F_1. \end{aligned} \tag{2}$$

Il mezzo si comporta allo stesso modo di un corpo rigido nello spazio e le nuove forze U_1 , U_2 , U_3 e U_4 rappresentano gli ingressi del nostro sistema. Questi ingressi possono poi essere riportati indietro alla velocità delle singole eliche tramite la seguente trasformazione inversa:

$$\begin{aligned} \Omega_1 &= 2\sqrt{\frac{U_1 - 2U_2 - U_4}{b}}; \\ \Omega_2 &= 2\sqrt{\frac{U_1 - 2U_3 + U_4}{b}}; \\ \Omega_3 &= 2\sqrt{\frac{U_1 + 2U_2 - U_4}{b}}; \\ \Omega_4 &= 2\sqrt{\frac{U_1 + 2U_3 + U_4}{b}}. \end{aligned} \tag{3}$$

A questo punto, senza entrare in dettaglio¹, possiamo esprimere il modello totale del quadrirotore come:

$$\begin{aligned} \dot{x}_0 &= v_x \\ \dot{y}_0 &= v_y \\ \dot{z}_0 &= v_z \\ \dot{\varphi} &= p + \sin \varphi \tan \theta q + \cos \varphi \tan \theta r \\ \dot{\theta} &= \cos \varphi q - \sin \varphi r \\ \dot{\psi} &= \frac{\sin \varphi}{\cos \theta} q + \frac{\cos \varphi}{\cos \theta} r \\ \dot{v}_x &= -\frac{1}{2m} \rho_{air} C_x v_x \sqrt{v_x^2 + v_y^2 + v_z^2} + \frac{1}{m} (\sin \varphi \sin \psi + \cos \varphi \cos \psi \sin \theta) U_1 \\ \dot{v}_y &= -\frac{1}{2m} \rho_{air} C_y v_y \sqrt{v_x^2 + v_y^2 + v_z^2} + \frac{1}{m} (\cos \varphi \sin \psi \sin \theta - \cos \psi \sin \varphi) U_1 \\ \dot{v}_z &= -\frac{1}{2m} \rho_{air} C_z v_z \sqrt{v_x^2 + v_y^2 + v_z^2} - g + \frac{1}{m} \cos \varphi \cos \theta U_1 \\ \dot{p} &= \frac{I_y - I_z}{I_x} q r - \frac{1}{2I_x} \rho_{air} C_p p \sqrt{p^2 + q^2 + r^2} + \frac{I_r}{I_x} q \Omega_m + \frac{l}{I_x} U_2 \\ \dot{q} &= \frac{I_z - I_x}{I_y} p r - \frac{1}{2I_y} \rho_{air} C_q q \sqrt{p^2 + q^2 + r^2} - \frac{I_r}{I_y} p \Omega_m + \frac{l}{I_y} U_3 \\ \dot{r} &= \frac{I_x - I_y}{I_z} p q - \frac{1}{2I_z} \rho_{air} C_r r \sqrt{p^2 + q^2 + r^2} + \frac{d}{I_z} U_4 \end{aligned} \tag{4}$$

¹La derivazione del modello totale è trattata in dettaglio nella relazione di Giovanni Mattei e Riccardo Sven Risuleo

Dove abbiamo le seguenti variabili di stato

1. x_0, y_0 e z_0 rappresentano la posizione del centro di massa del quadrirotore.
2. φ, θ e ψ rappresentano l'orientamento del quadrirotore nella terna RPY.
3. v_x, v_y e v_z rappresentano la velocità lineare del centro di massa del quadrirotore.
4. p, q e r è una terna che rappresenta la velocità angolare del quadrirotore.

e le seguenti costanti

1. m è la massa del quadrirotore.
2. ρ_{air} è il *coefficiente di densità dell'aria*.
3. C_x, C_y, C_z, C_p, C_q e C_r sono coefficienti aerodinamici dipendenti dalla forma del quadrirotore e che modellano la resistenza dell'aria che subisce il quadrirotore muovendosi rispettivamente lungo x, y e z e con velocità angolare p, q e r .
4. I_x, I_y e I_z sono i coefficienti di inerzia del quadrirotore.
5. l rappresenta la lunghezza delle braccia del quadrirotore.
6. d è il *coefficiente di drag* delle eliche.
7. g è semplicemente l'accelerazione gravitazionale terrestre.

A questo punto abbiamo tutto quello che ci serve per poter progettare il nostro controllore.

Riferimenti Bibliografici

Teoria sulla modellizzazione del quadrirotore [[6]]

2 Feedback Linearization

2.1 Analisi

La *feedback linearization* è un comune approccio per il controllo di *sistemi non-lineari* come, ad esempio, quello del quadrirotore. Dato un sistema del tipo

$$\begin{aligned}\dot{x} &= f(x) + g(x)u & x \in \mathbb{R}^n \ y \in \mathbb{R}^m \ u \in \mathbb{R}^p; \\ y &= h(x).\end{aligned}\tag{5}$$

lo scopo della feedback linearization è di trovare un ingresso

$$u = \alpha(x) + \beta(x)v\tag{6}$$

tale che il sistema (5) si comporti come un sistema lineare completamente raggiungibile del tipo

$$\begin{aligned}\dot{z} &= Az + Bv; \\ y &= Cx.\end{aligned}\tag{7}$$

dove v rappresenta il nuovo ingresso del sistema linearizzato e z rappresenta un nuovo stato ottenuto tramite $\Phi(x)$ ² una trasformazione di variabili completamente invertibile.

Per applicare la feedback linearization al nostro quadrirotore, per prima cosa, è necessario scegliere un sottoinsieme delle variabili di stato che definiscono il nostro *obiettivo di controllo*. Per evitare complicazioni, scegliamo un numero di uscite pari al numero di ingressi in modo da avere un sistema perfettamente controllabile. Nel caso del quadrirotore quindi sceglieremo di controllare la posizione (x, y, z) e l'angolo di *yaw*.

$$y = [x, y, z, \psi]^T\tag{8}$$

A questo punto consideriamo il vettore (r_1, r_2, r_3, r_4) contenente il *grado relativo del sistema per ogni uscita*. Tale valore indica il numero di volte che bisogna differenziare y_i per ottenere una dipendenza diretta dall'ingresso u .

Differenziando ogni elemento del vettore di uscita per il proprio grado relativo otteniamo

$$\begin{bmatrix} y^{(r_1)} \\ y^{(r_2)} \\ y^{(r_3)} \\ y^{(r_4)} \end{bmatrix} = l(x) + J(x)u\tag{9}$$

dove

$$J(x) = \begin{bmatrix} L_{g_1}(L_f^{r_1-1}(h_1)) & \cdots & L_{g_m}(L_f^{r_1-1}(h_1)) \\ \vdots & \ddots & \vdots \\ L_{g_1}(L_f^{r_m-1}(h_m)) & \cdots & L_{g_m}(L_f^{r_m-1}(h_m)) \end{bmatrix}\tag{10}$$

²In particolare Φ deve essere un *diffeomorfismo globale* ovvero una funzione invertibile di classe C^∞ in cui anche la sua inversa è di classe C^∞ .

e

$$l(x) = \begin{bmatrix} L_f^{r_1}(h_1) \\ \vdots \\ L_f^{r_m}(h_m) \end{bmatrix} \quad (11)$$

Invertendo la (9) otteniamo

$$\begin{aligned} \alpha(x) &= -J^{-1}(x)l(x); \\ \beta(x) &= J^{-1}(x). \end{aligned} \quad (12)$$

e, inserendo tutto in (6)

$$u = J^{-1}(x)[-l(x) + v] \quad (13)$$

È da notare che questa soluzione è ammissibile se e solo se $J(x)$ è invertibile³.

Sfortunatamente nel quadrirotore il vettore dei gradi relativi è $(2, 2, 2, 2)$ e la matrice $J(x)$ è della forma

$$J(x) = \begin{bmatrix} J_{1,1} & 0 & 0 & 0 \\ J_{2,1} & 0 & 0 & 0 \\ J_{3,1} & 0 & 0 & 0 \\ 0 & 0 & J_{4,3} & J_{4,4} \end{bmatrix} \quad (14)$$

Come si può facilmente intuire tale matrice *non* è invertibile per nessun valore di x . Per evitare che la matrice sia sempre singolare possiamo ritardare la comparsa di u_1 nelle componenti $y_1^{(2)}$, $y_2^{(2)}$ e $y_3^{(2)}$ applicando un *estensione dinamica* al sistema.

Aggiungiamo quindi alle 12 variabili di stato due variabili fittizie

$$\begin{aligned} \dot{U}_1 &= \eta \\ \dot{\eta} &= \tilde{U}_1 \end{aligned} \quad (15)$$

ottenendo in questo modo un nuovo sistema nella forma:

$$\dot{\tilde{x}} = \tilde{f}(\tilde{x}) + \tilde{g}(\tilde{x})\tilde{U} \quad (16)$$

³Da notare che il fatto di aver scelto un numero di stati controllabili uguale al numero degli ingressi rende $J(x)$ una matrice quadrata permettendoci di ignorare i problemi aggiuntivi che deriverebbero dall'uso di una pseudo-inversione.

dove

$$\begin{aligned}
\tilde{f}(\tilde{x}) &= \begin{bmatrix} v_x \\ v_y \\ v_z \\ p + q \sin \varphi \tan \theta + r \cos \varphi \tan \theta \\ q \cos \varphi - r \sin \varphi \\ \frac{q \sin \varphi + r \cos \varphi}{\cos \theta} \\ 0 \\ 0 \\ -g \\ \frac{(I_y - I_z)}{I_x} q r + \frac{I_r}{I_x} q \Omega_m \\ \frac{(I_z - I_x)}{I_y} p r - \frac{I_r}{I_y} p \Omega_m \\ \frac{(I_x - I_y)}{I_z} p q \end{bmatrix} \\
\tilde{g}(\tilde{x}) &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} (\cos \varphi \sin \theta \cos \psi + \sin \varphi \sin \psi) & 0 & 0 & 0 \\ \frac{1}{m} (\cos \varphi \sin \theta \sin \psi - \sin \varphi \cos \psi) & 0 & 0 & 0 \\ \frac{1}{m} \cos \varphi \cos \theta & 0 & 0 & 0 \\ 0 & \frac{l}{I_x} & 0 & 0 \\ 0 & 0 & \frac{l}{I_y} & 0 \\ 0 & 0 & 0 & \frac{d}{I_z} \end{bmatrix}
\end{aligned} \tag{17}$$

in cui il vettore di ingresso è formato da \tilde{U}_1 , U_2 , U_3 e U_4 .

Nel sistema esteso i gradi relativi di $[x, y, z, \psi]$ sono $[4, 4, 4, 2]$ e la corrispettiva matrice $\tilde{J}(x)$ risulta ora invertibile a patto che U_1 sia diverso da zero. La condizione non è restrittiva in quando è sempre verificata nel quadrirotore reale fintanto che le pale sono in movimento.

Al sistema lineare (7) così ottenuto si possono applicare tutti i metodi di controllo dei sistemi lineari tempo-invarianti. In particolare, nel caso di asservimento ad una traiettoria $y(t)$:

$$y_d(t) = \begin{bmatrix} y_{1d}(t) \\ y_{2d}(t) \\ y_{3d}(t) \\ y_{4d}(t) \end{bmatrix} \tag{18}$$

definiamo 4 errori di asservimento:

$$e_i = y_{id} - y_i \quad i = 1, \dots, 4 \tag{19}$$

e per ciascuna uscita un numero di coefficienti del polinomio di Hurwitz, pari al grado relativo:

$$c_0^i, \dots, c_{r_i-1}^i \quad i = 1, \dots, 4 \tag{20}$$

a questo punto abbiamo tutto ciò che ci serve per definire 4 leggi di controllo v_i che assicurano la convergenza asintotica delle quattro uscite alla traiettoria desiderata:

$$\begin{aligned}
v_1 &= y_{1d}^{(4)} + \sum_{j=1}^4 c_{j-1}^1 e_1^{(j-1)} \\
v_2 &= y_{2d}^{(4)} + \sum_{j=1}^4 c_{j-1}^2 e_2^{(j-1)} \\
v_3 &= y_{3d}^{(4)} + \sum_{j=1}^4 c_{j-1}^3 e_3^{(j-1)} \\
v_4 &= y_{4d}^{(2)} + \sum_{j=1}^2 c_{j-1}^4 e_4^{(j-1)}
\end{aligned} \tag{21}$$

Riferimenti Bibliografici

Teoria e condizioni di linearizzazione [4, 5, 3] e applicazione della stessa al modello del quadrirotore [6, 1]

3 Obstacle Avoidance

3.1 Analisi

Il campo potenziale artificiale può essere descritto da una funzione U somma di due campi, uno attrattivo e uno repulsivo.

$$U(\mathbf{p}) = U_{rep}(\mathbf{p}) + U_{att}(\mathbf{p}) \quad (22)$$

A partire da questo è possibile ricavare una forza F in grado di guidare il robot fino al minimo del campo seguendo il rispettivo *gradiente*.

$$F(\mathbf{p}) = F_{att}(\mathbf{p}) + F_{rep}(\mathbf{p}) = -\nabla U_{rep}(\mathbf{p}) - \nabla U_{att}(\mathbf{p}) \quad (23)$$

L'espressione del campo attrattivo è semplice e può essere espressa semplicemente come

$$U_{att}(\mathbf{p}) = \frac{1}{2} k_a |\mathbf{p} - \mathbf{p}_{goal}|^2 \quad (24)$$

dove p è la configurazione corrente del robot e \mathbf{p}_{goal} è la configurazione desiderata. Dall'equazione 24 ricaviamo F_{att}

$$F_{att}(\mathbf{p}) = -k_a (\mathbf{p} - \mathbf{p}_{goal}) \quad (25)$$

Per il campo repulsivo invece useremo la seguente formula

$$U_{rep}(\mathbf{p}) = \begin{cases} \frac{1}{2} k_r \left(\frac{1}{\rho} - \frac{1}{\rho_{max}} \right)^2, & \text{if } \rho \leq \rho_{max} \\ 0, & \text{if } \rho > \rho_{max} \end{cases} \quad (26)$$

dove ρ è espresso come

$$\rho = \underset{x \in \mathcal{CO}}{\operatorname{argmin}}(d(\mathbf{p}, x)) \quad (27)$$

ovvero la minima distanza fra la configurazione corrente ed ogni altro punto della frontiera dello spazio \mathcal{C} -object. Dall'equazione 26 ricaviamo

$$F_{rep}(\mathbf{p}) = \begin{cases} k_r \left(\frac{1}{\rho} - \frac{1}{\rho_{max}} \right) \frac{1}{\rho^2} \nabla \rho, & \text{if } \rho \leq \rho_{max} \\ 0, & \text{if } \rho > \rho_{max} \end{cases} \quad (28)$$

che è la nostra espressione per la forza repulsiva.

3.2 Potenziale artificiale applicato al quadrirotore

Lo spazio di di configurazione del quadrirotore può essere limitato alle sole variabili controllabili ovvero x , y , z e lo *yaw* ψ .

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \\ \psi \end{bmatrix} \quad (29)$$

Costruire la versione in \mathcal{CO} degli ostacoli avvolgiamo il quadrirotore in una sfera di raggio r (Fig. 3.2) e usiamo la stessa per estendere gli ostacoli reali.

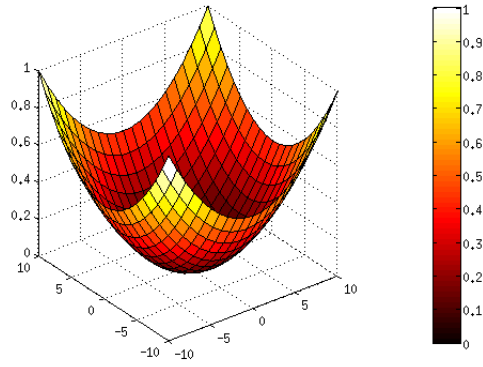


Figura 1: Un esempio di potenziale attrattivo per uno spazio di configurazione bidimensionale e $\mathbf{p}_{goal} = [0\ 0]^T$.

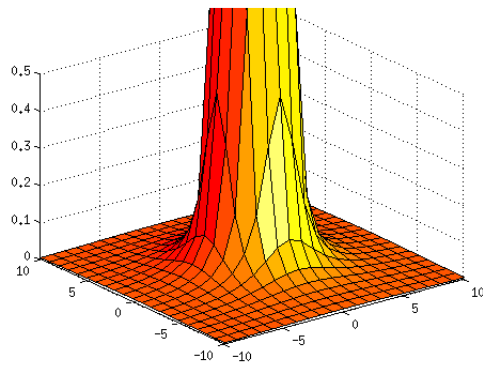


Figura 2: Un esempio di potenziale repulsivo per uno spazio di configurazione bidimensionale con un ostacolo circolare di raggio 2 centrato in $\mathbf{p}_{obs} = [0\ 0]^T$.

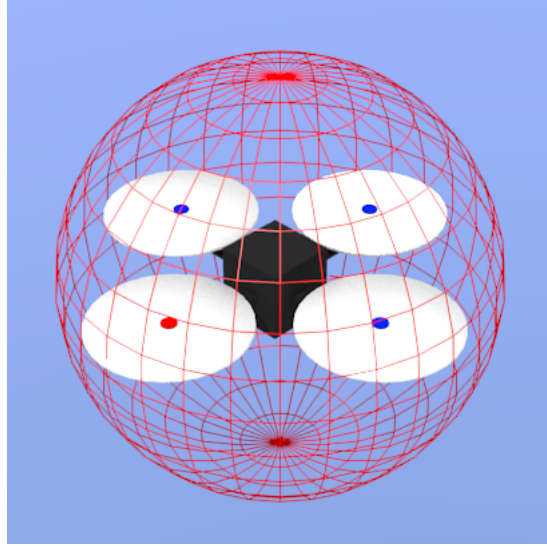


Figura 3: La sfera (*boundary*) teorica che avvolge il quadrirotore.

Come conseguenza di questa scelta abbiamo che la distanza minima del quadrirotore da un ostacolo nello spazio delle configurazioni dipenderà solamente dalla sua distanza dalla posizione xyz del quadrirotore e sarà valida la seguente relazione

$$\rho = \begin{cases} \rho_{reale} - r, & \text{if } \rho_{reale} \geq r \\ 0, & \text{if } \rho_{reale} < r \end{cases} \quad (30)$$

dove ρ_{reale} è la distanza minima del centro di massa del quadrirotore dall'ostacolo più vicino *nello spazio di lavoro*. Tale distanza è immediatamente ricavabile, ad esempio tramite un sensore *laser*.

Un'altra conseguenza di questa scelta è che gli ostacoli non influiscono sullo *yaw*. In ogni momento quindi la forza repulsiva sullo *yaw* è nulla

$$F_{rep}(\mathbf{p}) = \begin{bmatrix} F_{rep}^x \\ F_{rep}^y \\ F_{rep}^z \\ 0 \end{bmatrix} \quad (31)$$

Ora bisogna scegliere come applicare la forza derivante dal campo potenziale artificiale alla feedback linearization del quadrirotore. Partendo dalle 4 leggi di controllo descritte in (21) possiamo annullare tutti gli errori e imporre

$$\begin{aligned} v_1 &= y_{1d}^{(4)} = F_x(\mathbf{p}) + \delta_x \\ v_2 &= y_{2d}^{(4)} = F_y(\mathbf{p}) + \delta_y \\ v_3 &= y_{3d}^{(4)} = F_z(\mathbf{p}) + \delta_z \\ v_4 &= y_{1d}^{(2)} = F_\psi(\mathbf{p}) + \delta_\psi \end{aligned} \quad (32)$$

Tale assunzione è valida poiché, nel trajectory tracking basato sui potenziali artificiali, assumiamo che il robot si trovi *sempre* sulla traiettoria e, di conseguenza, l'errore è nullo. Il vettore di forze entra nel controllo quindi come *snap* desiderato per quanto riguarda la posizione e come *accelerazione* angolare desiderata dello *yaw*.

Come al solito, quando si imposta un riferimento di ordine superiore al primo, è necessario smorzare il comportamento oscillante o divergente del sistema aggiungendo alla forza un adeguato vettore di *damping* δ .

$$\delta = \begin{bmatrix} -k_1\dot{x} - k_2\ddot{x} - k_3\dddot{x} \\ -k_1\dot{y} - k_2\ddot{y} - k_3\dddot{y} \\ -k_1\dot{z} - k_2\ddot{z} - k_3\dddot{z} \\ -k_1\dot{\psi} \end{bmatrix} \quad (33)$$

Riferimenti Bibliografici

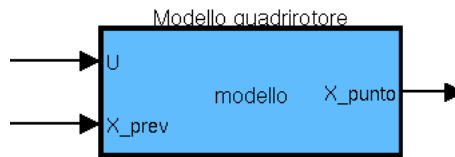
Teoria [2] ed esempi di applicazione [7].

4 Implementazione

4.1 Matlab

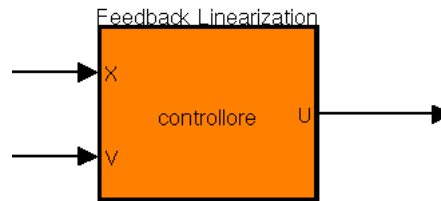
In prima istanza il modello del quadrirotore, il controllore tramite feedback linearization e l'obstacle avoidance tramite artificial potential field sono stati implementati in un modello Simulink.

4.1.1 Modello del quadrirotore



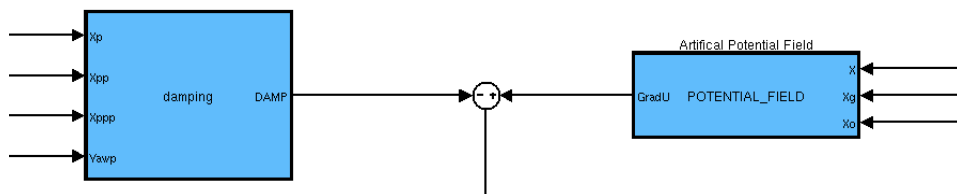
Questo primo blocco, ad ogni istante di tempo t , prende come input il vettore U_t , che rappresenta le variabili di controllo, e lo stato all'istante $t-1$, x_{t-1} . Esso calcola, tramite il modello teorico 4 del quadrirotore come un corpo rigido a 6 DOF, lo stato \dot{x}_t

4.1.2 Feedback linearization



Questo secondo blocco prende come input lo stato x ottenuto integrando l'uscita del blocco precedente ed il vettore v calcolato in accordo con l'equazione 21. All'interno del blocco vengono calcolate l'inversa della matrice di disaccoppiamento J e il vettore l . L'output di questo blocco, ovvero gli input di controllo del sistema, vengono ottenuti applicando la 13.

4.1.3 Campo Potenziale Artificiale



Per calcolare il campo potenziale artificiale abbiamo utilizzato due differenti blocchi. Il primo, sulla destra nell'immagine, prende in input tre vettori x , x_g e x_o che rappresentano, rispettivamente, lo stato del quadrotor, la posizione del goal e quella dell'ostacolo. Questo blocco calcola il valore dei potenziali in accordo con le definizioni delle equazioni 25 e 28.

Il secondo blocco, sulla sinistra nell'immagine, prende in input lo stato x e le derivate successive fino al terzo ordine. Esso è necessario per introdurre dei fattori di damping δ che stabilizzino asintoticamente la convergenza ai valori desiderati in accordo con l'equazione 33.

4.2 GAZEBO e ROS

Terminato lo sviluppo del modello Simulink, il quadrotor è stato implementato in ROS e GAZEBO.

ROS, acronimo di *Robot Operating system*, "è un sistema che offre librerie e strumenti atti ad aiutare gli sviluppatori di software a creare applicazioni per robot. In particolare offre un meccanismo di astrazione dell'hardware, drivers, librerie, visualizzatori, strumenti per la trasmissione di messaggi, per la gestione dei pacchetti e molto altro". Nel nostro lavoro ROS è stato utilizzato principalmente come piattaforma per lo scambio di messaggi tra le diverse componenti del sistema. GAZEBO è invece un simulatore multi-robot per ambienti aperti che ci ha permesso di visualizzare i risultati ma soprattutto di simulare le dinamiche dell'ambiente, del quadrotor e dei sensori.

In fig. 4 viene rappresentato lo schema a blocchi della simulazione implementata. Il blocco etichettato come GAZEBO simula l'ambiente, il quadrotor ed il sensore laser. Mentre l'ambiente e il quadrotor vengono visualizzati tramite un'interfaccia grafica i dati raccolti dal sensore vengono pubblicati, via ROS, su un apposito topic. Il blocco Feedback Controller, implementato in C++, calcola le forze e le coppie necessarie e le applica al quadrotor nell'ambiente simulato tramite le API di GAZEBO. Le forze e le coppie sono derivate dallo stato attuale del quadrotor, ottenuto direttamente da GAZEBO tramite API, dai rilevamenti del sensore, letti dall'apposito topic di ROS, e dalla posizione del goal. Il goal viene letto da un secondo topic, sempre implementato in ROS, sul quale pubblica il blocco Control Shell; quest'ultimo è una semplice shell implementata in C++ che permette di cambiare dinamicamente il goal del quadrotor.

Nelle simulazioni condotte tramite GAZEBO abbiamo però rilevato che il motore di integrazione su cui si basa crea un grosso rumore numerico. Questo comportamento è ben visibile analizzando l'andamento della velocità angolare dello ψ graficato in fig.5

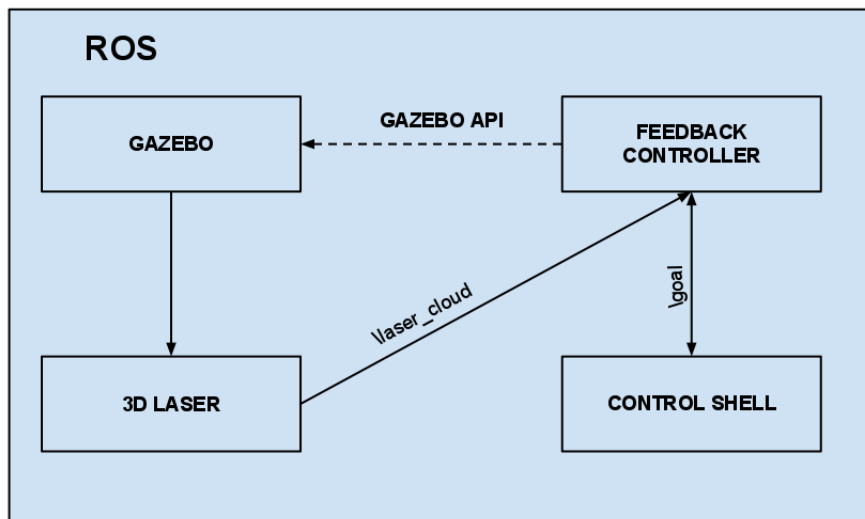
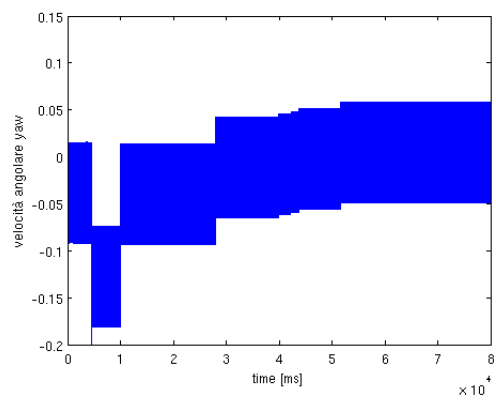
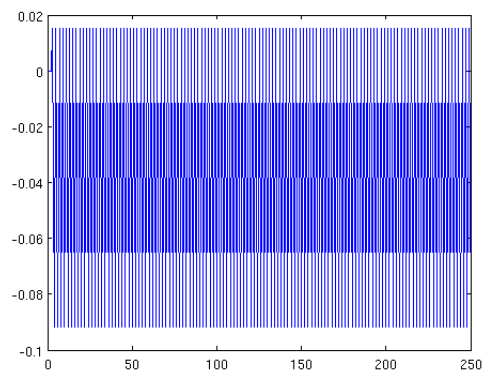


Figura 4: Schema a blocchi del modello simulato tramite GAZEBO e ROS



(a) Simulazione complessiva



(b) $t = (0, 250)$ ms

Figura 5: Rumore Numerico

5 Simulazioni

Nelle seguenti simulazione i parametri adottati sono i seguenti:

Guadagni

| | |
|---------------------|----|
| k_a | 3 |
| k_r | 5 |
| k_y | 3 |
| $k_{damping\ vel}$ | 40 |
| $k_{damping\ acc}$ | 20 |
| $k_{damping\ jerk}$ | 10 |
| $k_{damping\ \psi}$ | 8 |

Parametri Dinamici

| | |
|------------|-----------------|
| $mass$ | 0.64 |
| I_x | 0.2 |
| I_y | 0.2 |
| I_z | 0.4 |
| I_r | 10^{-3} |
| g | 9.81 |
| Ω_m | 0 |
| l | 0.165 |
| d | $4.5 * 10^{-7}$ |

Altri Parametri

| | |
|--------------|-----|
| ρ_{max} | 3 |
| r | 0.5 |

5.1 Matlab

Simulazione n°1

In questa prima simulazione lo stato iniziale del quadrotor è $[0, 0, 7, 0]$ mentre il goal è posto in $[7, 7, 7, -\pi/2]$, è inoltre presente un ostacolo di forma cilindrica e raggio unitario centrato in $[3, 5]$.

In fig. 6 è rappresentata la traiettoria seguita dal quadrirotore sul piano xy nell'aggirare l'ostacolo sia nel caso in cui lo stesso sia controllato in FL (in blu), sia nel caso di un sistema perfettamente linearizzato simulato con 4 integratori in catena (in rosso). In fig. 11 vengono invece presentati gli andamenti delle variabili controllate $[x, y, z, \psi]$ al variare del tempo.

A prima vista l'andamento è più brusco di quanto si potrebbe immaginare, notiamo infatti una rapida sterzata del quadrirotore in prossimità delle coordinate $< 3, 3 >$. Tuttavia è bene ricordare che le forze repulsive agiscono sul quadrirotore al livello della derivata quarta e che di conseguenza il loro effetto sulla posizione è piuttosto lento.

Tale comportamento può essere variato e reso più o meno *smooth* giocando con il rapporto fra k_a e k_r . Ad esempio, lasciando k_r fisso e diminuendo il guadagno attrattivo si ottiene una traiettoria molto più morbida al prezzo di

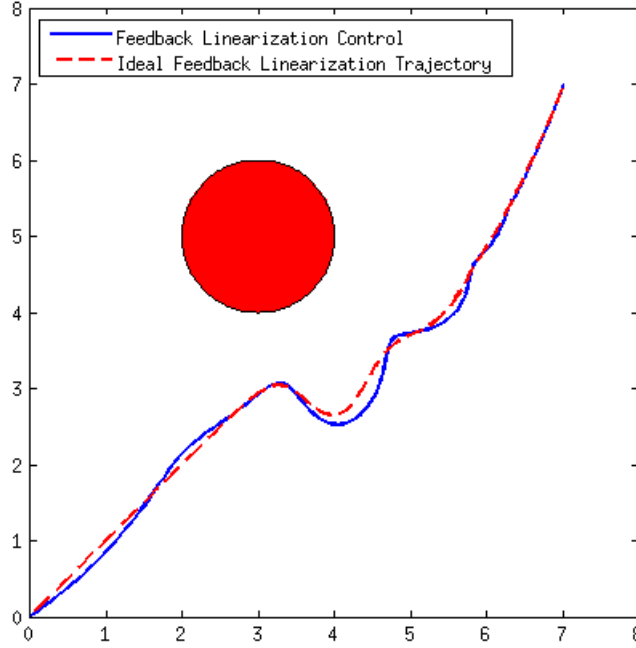


Figura 6: Simulazione n°1, Matlab, piano xy

una minore velocità del mezzo (con $k_a < 3$ il quadrotore non converge entro i 60 secondi).

Simulazione n°2

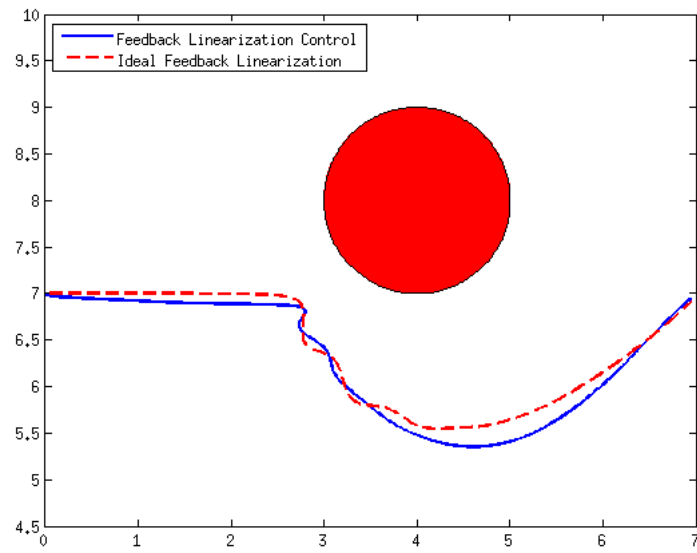
In questa seconda simulazione lo stato iniziale ed il goal sono i medesimi della precedente, l'ostacolo però è una sfera centrata in $[4, 4, 8]$. In fig. 7 vengono rappresentate le traiettorie del quadrotor sui piani xz ed yz . Come è facile notare, i due grafici presentano il medesimo andamento come era logico aspettarsi tenendo in considerazione la simmetria del quadrotor e la posizione dell'ostacolo.

Ancora una volta in fig. 12 vengono presentati gli andamenti delle variabili controllate al variare del tempo.

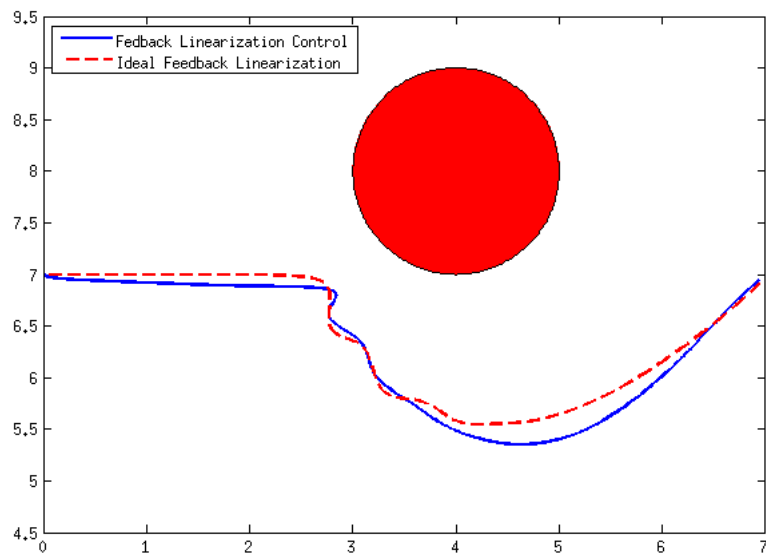
5.2 GAZEBO

Simulazione n°1

Le prima simulazione effettuata in GAZEBO riproduce nella maniera più fedele possibile la prima simulazione effettuata in Matlab: stato iniziale e goal sono i medesimi così come i guadagni dei campi attrattivi e repulsivi e il range of influence dell'ostacolo. Osservando fig. 8 possiamo notare che la traiettoria seguita manifesta un andamento oscillante, questo comportamento è da attribuirsi principalmente alla modellazione fisica del quadrotor che ne aumenta le dimensioni e di conseguenza l'ostacolo viene avvertito come più vicino. L'intensità di

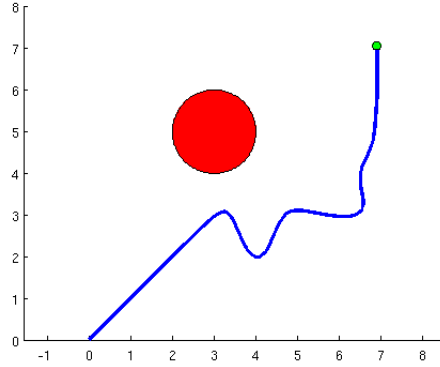


(a) piano xz



(b) piano yz

Figura 7: Simulazione n°2, Matlab



(a) piano xz

Figura 8: Simulazione n°1, GAZEBO

queste oscillazioni, infatti, dipende sia dal raggio di influenza dell'ostacolo che dal guadagno del campo potenziale repulsivo.

In fig. 11 vengono riportati gli andamenti delle variabili controllate in funzione del tempo.

Simulazione n°2

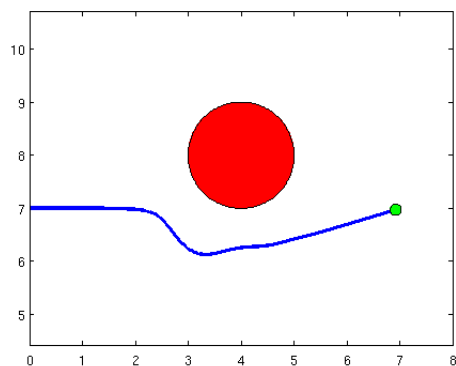
Anche nella seconda simulazione abbiamo ricalcato la situazione inizialmente presentata su Matlab in GAZEBO, ed anche in questo caso dalla fig. 9 osserviamo che la diversa modellazione fisica del quadrotor e gli errori di misurazione introdotti del laser risultano in una traiettoria differente.

In fig. 14 vengono presentati gli andamenti di tutte e quattro le variabili di controllo al variare del tempo.

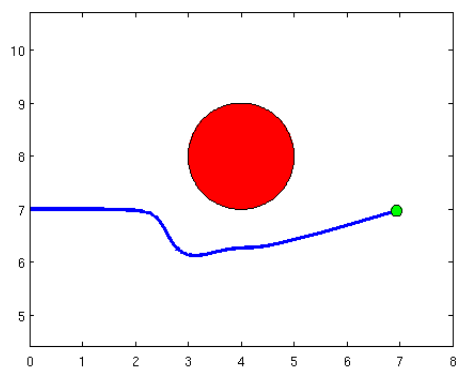
Simulazione n°3

In questa terza simulazione lo stato iniziale è $[0, 0, 7, 0]$ ed il goal $[18.5, 17.5, 7, -\pi/2]$. In questo caso sono stati posizionati due ostacoli, di forma cilindrica e raggio unitario, centrati rispettivamente in $[5, 3, 0]$ e $[13, 13, 0]$. In fig. 10 viene rappresentata la traiettoria seguita dal quadrotor sul piano xy ciò che colpisce è il comportamento osservato nell'evitare il secondo ostacolo: una traiettoria che lascia l'ostacolo sulla destra potrebbe sembrare più corretta ma, così facendo, il quadrotor si troverebbe in una posizione più lontana dal goal rispetto a quella raggiunta seguendo la traiettoria tracciata. Scegliere una traiettoria che lascia l'ostacolo sulla destra significherebbe, di fatto, andare in direzione opposta al gradiente.

Per completezza, in fig. 15 vengono riportati anche gli andamenti di $[x, y, z, \psi]$ al variare del tempo.



(a) piano xz



(b) piano yz

Figura 9: Simulazione n°2, GAZEBO

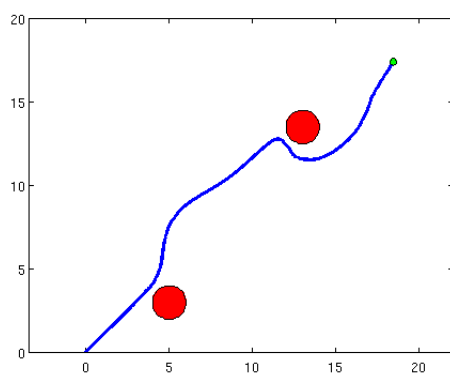


Figura 10: Simulazione n°3, GAZEBO, piano xy

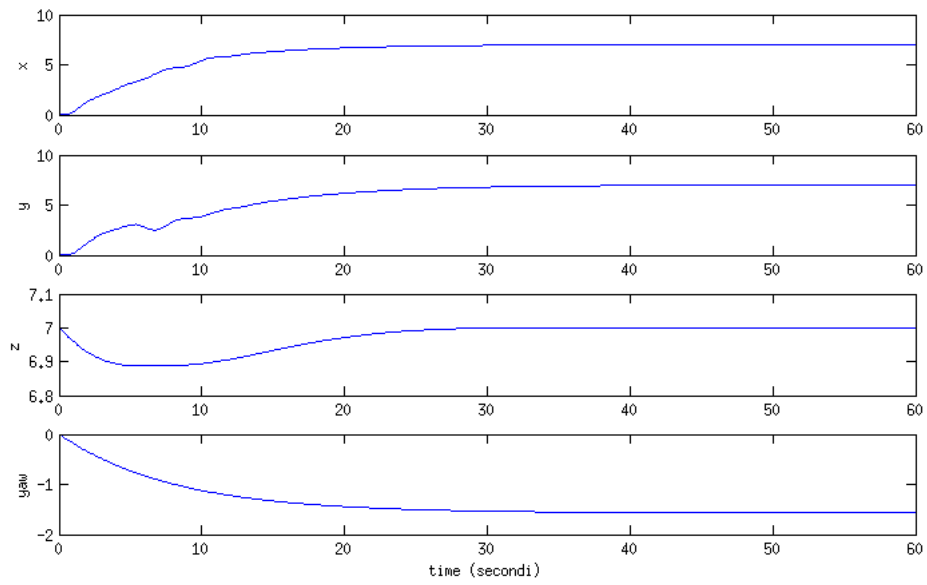


Figura 11: Simulazione n°1, Matlab

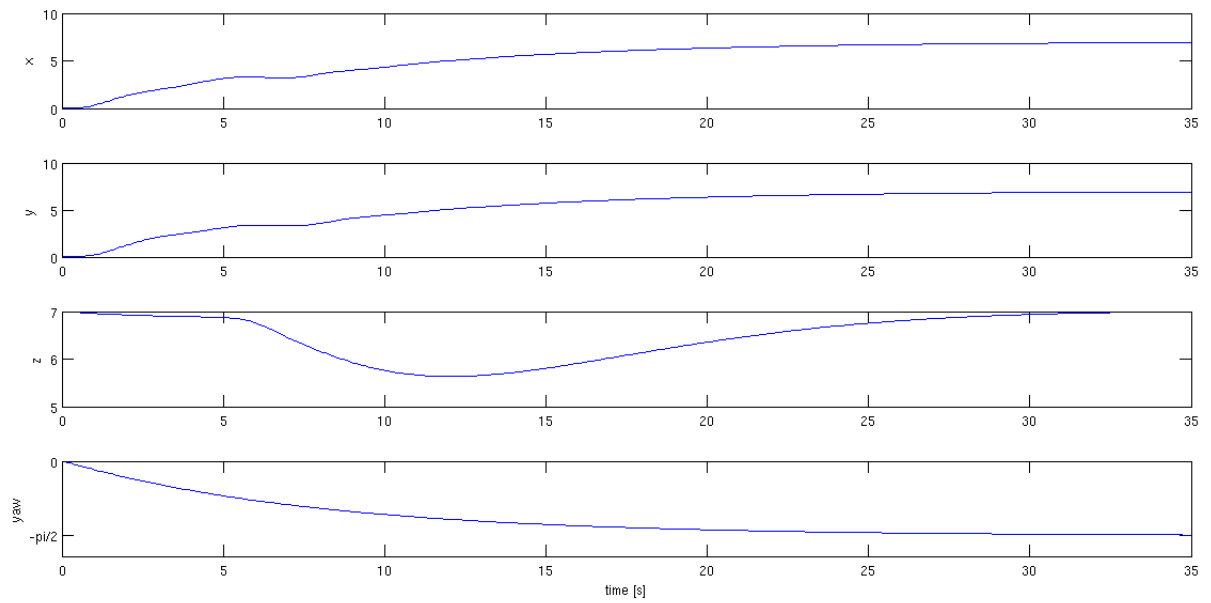


Figura 12: Simulazione n°2, Matlab

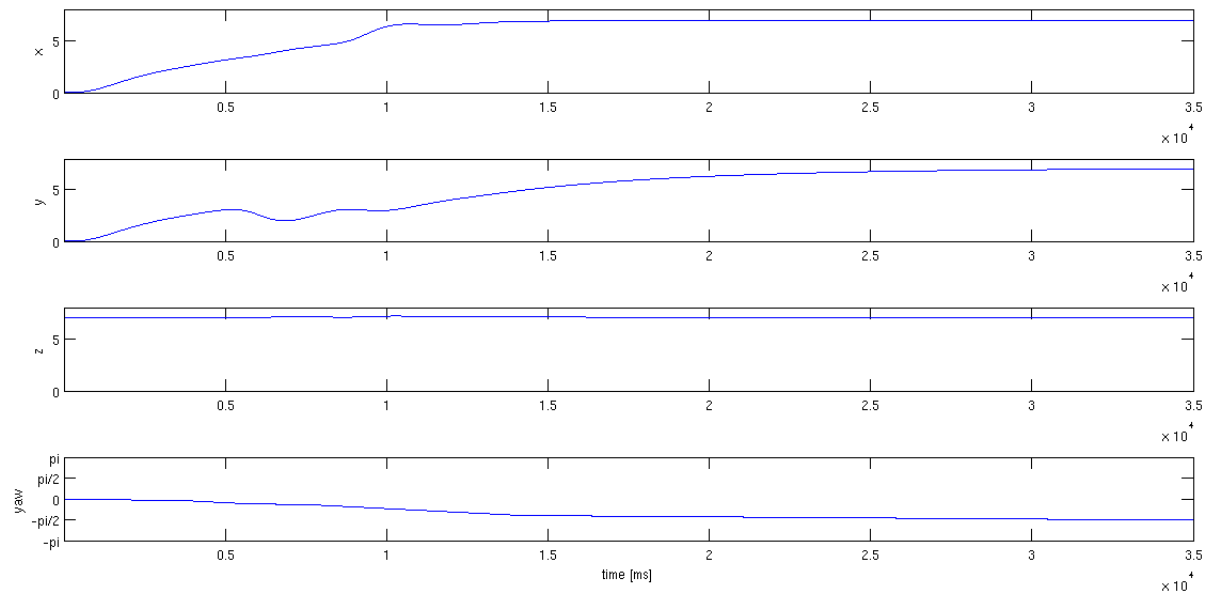


Figura 13: Simulazione n°1, GAZEBO

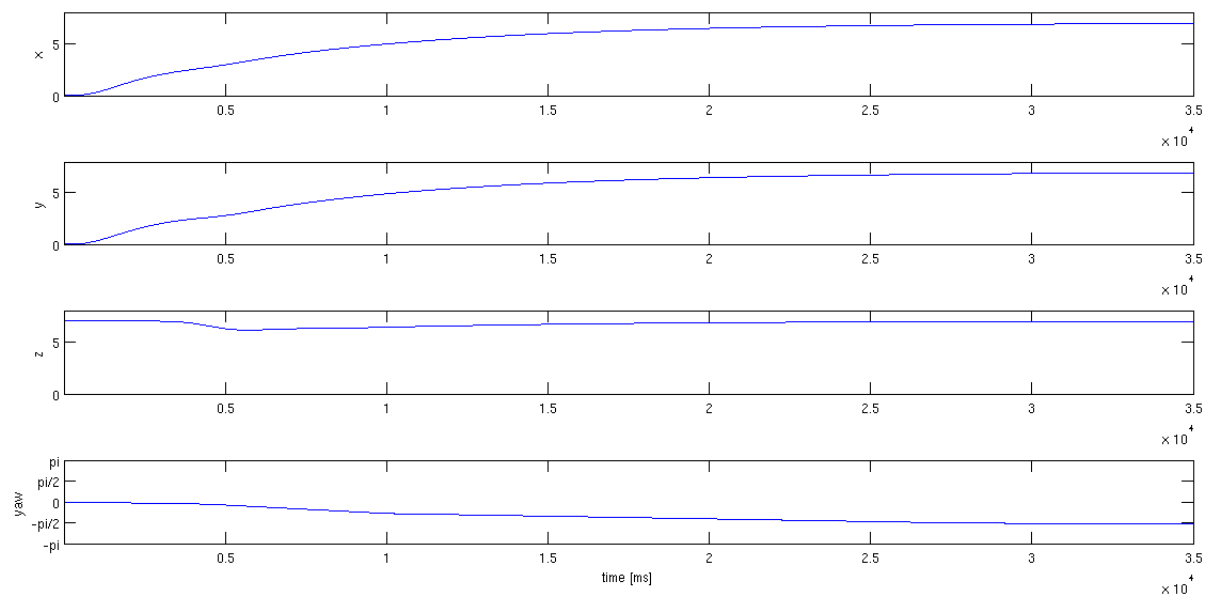


Figura 14: Simulazione n°2, GAZEBO

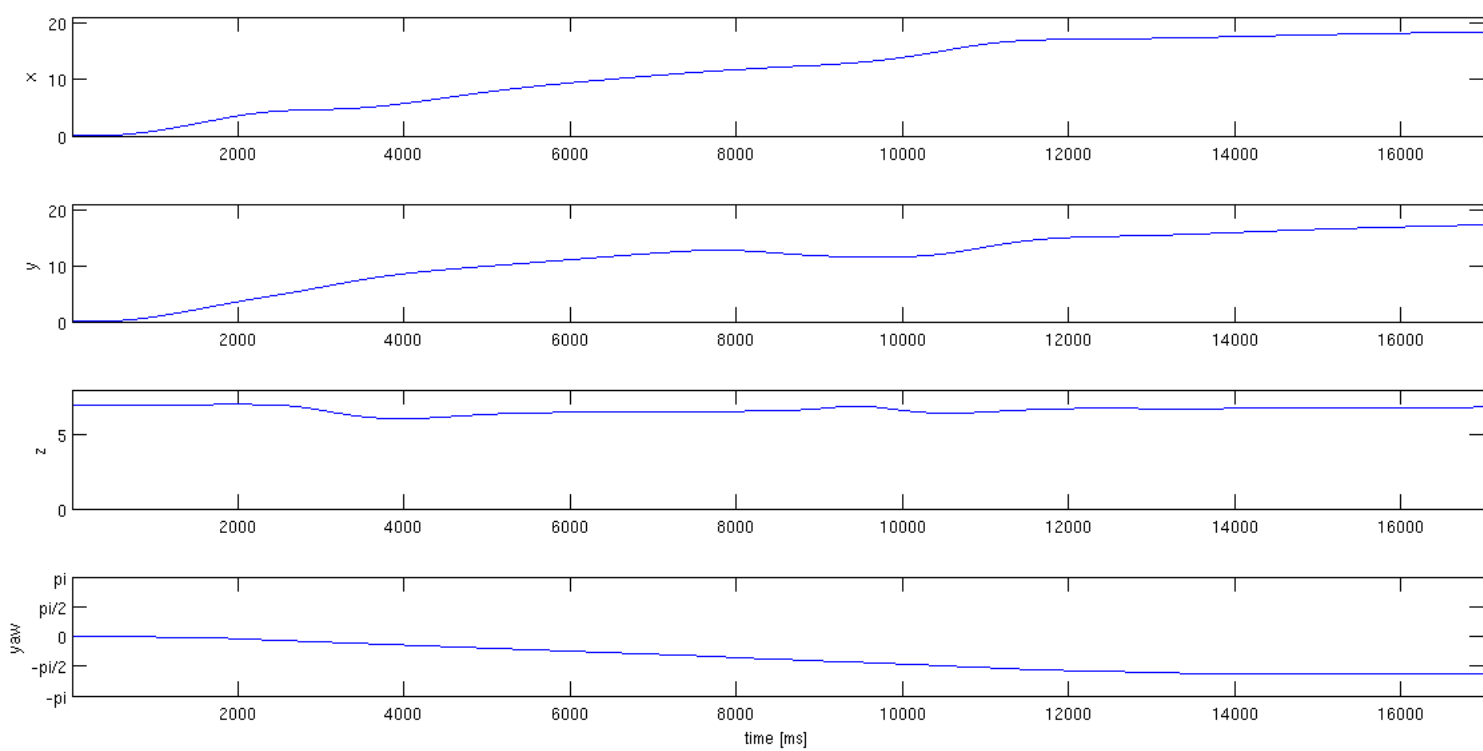


Figura 15: Simulazione n°3, GAZEBO

Riferimenti bibliografici

- [1] S.A. Al-Hiddabi. Quadrotor control using feedback linearization with dynamic extension. In *Mechatronics and its Applications, 2009. ISMA'09. 6th International Symposium on*, number 1, pages 1–3. IEEE, 2009.
- [2] L. Villani G. Oriolo B. Siciliano, L. Sciavicco. *Robotics Modelling, Planning and Control*. Springer, 2008.
- [3] A. Isidori. *Non-Linear Control System*. Springer, 1995.
- [4] A. Isidori, CH Moog, and A. De Luca. A sufficient condition for full linearization via dynamic state feedback. In *Decision and Control, 1986 25th IEEE Conference on*, volume 25, pages 203–208. IEEE, 1986.
- [5] H.K. Khalil. *Non-Linear System*. Prentice Hall, 1996.
- [6] V. Mistler, A. Benallegue, and N.K. M'Sirdi. Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback. In *Robot and Human Interactive Communication, 2001. Proceedings. 10th IEEE International Workshop on*, pages 586–593. IEEE, 2001.
- [7] M.G. Park, J.H. Jeon, and M.C. Lee. Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing. In *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on*, volume 3, pages 1530–1535. IEEE, 2001.