

Progetto Automazione:
Modellazione, simulazione e realizzazione di un robot
autobilanciante su due ruote

Emanuele Coletta 2001600

January 15, 2024

0.1 Introduzione

Lo scopo del progetto è la realizzazione di un robot con due ruote indipendenti che sia in grado di rimanere in equilibrio autonomamente. Il problema è una variante del problema del pendolo inverso ed è intrinsecamente instabile. Il sistema ha un unico punto di equilibrio ed è necessario un controllore per mantenerlo attorno. È stato inizialmente derivato un modello semplificato del robot usando il formalismo Lagrangiano [1]. Il modello è stato successivamente linearizzato attorno al punto di equilibrio così da poter realizzare un controllore lineare di tipo PID. Il modello è stato successivamente simulato usando MatLab e Simulink così da poter calibrare i parametri del controllore PID e poter correttamente dimensionare i motori. Il controllore PID è di tipo discreto, data la successiva implementazione su microcontrollore. Infine è stato il robot è stato costruito usando un microcontrollore Raspberry Pi Pico (Arduino-compatibile), un motor driver per motori CC L298N, una IMU a 6 assi MPU6050 e una struttura fatta di pezzi stampati in 3D e di legno tagliato a laser.

0.2 Modellazione

È stato calcolato un modello semplificato del sistema tramite il formalismo Lagrangiano (fig. 1). Le equazioni ottenute modellano due gradi di libertà del sistema: la traslazione lungo l'asse di movimento (x) e il movimento rotatorio della massa del pendolo (pitch). Il modello prende in considerazione la presenza di entrambe le ruote ma non l'ulteriore grado di libertà di rotazione attorno all'altezza (yaw) che viene aggiunto quando esse ruotano a velocità diverse e/o in versi opposti. Queste differenze possono essere volute, se si vuole far ruotare il robot su se stesso, oppure possono essere dovute a differenze costruttive fra i due motori. Nella pratica è inverosimile pensare di poter avere due motori esattamente identici che girino sempre alla stessa velocità erogando sempre la stessa coppia, tuttavia per lo scopo di questo progetto si assume che le differenze siano trascurabili e viene ignorato il controllo dello yaw.

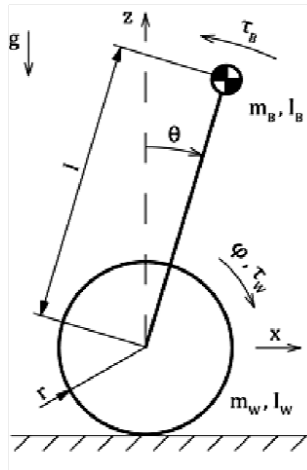


Figure 1: Schema fisico del robot

Sotto assunzione di moto di puro rotolamento si può scrivere che

$$dx = r d\theta \implies x = r\phi \implies \dot{x} = r\dot{\phi} \quad (1)$$

Per rispettare questa assunzione, durante la realizzazione ci si deve effettivamente assicurare che le ruote non slittino mai e facciano ben attrito sul terreno.

Denominando le componenti di spostamento e velocità del centro di massa

$$\begin{aligned}
x_{CM} &= x + l \sin \theta \\
z_{CM} &= l \cos \theta \\
v_{CM} &= \sqrt{\dot{x}^2 + \dot{z}^2}
\end{aligned} \tag{2}$$

Si ha che l'energia cinetica del corpo è composta da una componente traslazionale e una rotazionale.

$$T_1 = \frac{1}{2} m_b v_{CM}^2 + \frac{1}{2} I_b \dot{\theta}^2 \tag{3}$$

Dove I_b è l'inerzia del corpo escluse le ruote. Sviluppando i termini si ottiene che

$$\begin{aligned}
T_1 &= \frac{1}{2} I_b \dot{\theta}^2 + \frac{1}{2} m_b (\dot{x}_{CM}^2 + \dot{z}_{CM}^2) = \\
&= \frac{1}{2} I_b \dot{\theta}^2 + \frac{1}{2} m_b \left(\left[\frac{d}{dt} (x + l \sin \theta) \right]^2 + \left[\frac{d}{dt} (l \cos \theta) \right]^2 \right) = \\
&= \frac{1}{2} I_b \dot{\theta}^2 + \frac{1}{2} m_b ([\dot{x} + l \cos \theta \dot{\theta}]^2 + [-l \sin \theta \dot{\theta}]^2) = \\
&= \frac{1}{2} I_b \dot{\theta}^2 + \frac{1}{2} m_b (\dot{x}^2 + l^2 \cos^2 \theta \dot{\theta}^2) + 2 \dot{x} l \cos \theta \dot{\theta} + l^2 \sin^2 \theta \dot{\theta}^2 = \\
&= \frac{1}{2} I_b \dot{\theta}^2 + \frac{1}{2} m_b (\dot{x}^2 + l^2 \dot{\theta}^2) + 2 \dot{x} l \cos \theta \dot{\theta} = \\
&= \frac{1}{2} (I_b + m_b l^2) \dot{\theta}^2 + \frac{1}{2} m_b (\dot{x}^2 + 2 \dot{x} l \cos \theta \dot{\theta})
\end{aligned} \tag{4}$$

L'energia cinetica della ruota

$$T_2 = \frac{1}{2} m_w \dot{x}^2 + \frac{1}{2} I_w \dot{\phi}^2 = \frac{1}{2} m_w \dot{x}^2 + \frac{1}{2} I_w \frac{\dot{x}^2}{r^2} \tag{5}$$

L'energia potenziale del corpo

$$V_1 = m_b g l \cos \theta \tag{6}$$

L'energia potenziale della ruota

$$V_2 = 0 \tag{7}$$

Da cui la funziona Lagrangiana:

$$\begin{aligned}
L(x, \dot{x}, \theta, \dot{\theta}) &= T_1 - V_1 + 2T_2 - 2V_2 = \\
&= \frac{1}{2} (I_b + m_b l^2) \dot{\theta}^2 + \frac{1}{2} m_b (\dot{x}^2 + 2 \dot{x} l \cos \theta \dot{\theta}) + \frac{1}{2} m_w \dot{x}^2 + \frac{1}{2} I_w \frac{\dot{x}^2}{r^2} \\
&\quad - m_b g l \cos \theta
\end{aligned} \tag{8}$$

Da cui le equazioni che governano il moto del corpo, con τ_r, τ_l rispettivamente le coppie forzanti applicate da motori destro e sinistro (che si devono opporre al moto naturale del corpo) e D il termine di attrito rotazionale:

$$D = \frac{1}{2} b_w \dot{\phi}^2 = \frac{1}{2} b_w \frac{\dot{x}^2}{r^2} \tag{9}$$

$$\begin{cases}
(I_b + m_b l^2) \ddot{\theta} + m_b l \ddot{x} \cos \theta - m_b g l \sin \theta = -(\tau_r + \tau_l) \\
(m_b + \frac{2I_w}{r^2} + 2m_w) \ddot{x} + m_b l \cos \theta \ddot{\theta} - m_b l \sin \theta \dot{\theta}^2 = \frac{\tau_r + \tau_l}{r} - 2b_w \frac{\dot{x}}{r^2}
\end{cases} \tag{10}$$

Sono ovviamente non lineari, con non linearità di tipo trigonometrico, potenze delle derivate di variabili e moltiplicazioni fra variabili. Può tuttavia essere linearizzato attorno al punto di equilibrio $\theta = 0$ sotto ipotesi di piccole oscillazioni: per $\theta \ll 1 \text{ rad}$ si ha $\sin \theta \approx \theta$, $\cos \theta \approx 1$, $\dot{\theta}^2 \approx 0$. Il sistema linearizzato che si ottiene è

$$\begin{cases} (I_b + m_b l^2)\ddot{\theta} + m_b l \ddot{x} - m_b g l \theta = -(\tau_r + \tau_l) \\ (m_b + \frac{2I_w}{r^2} + 2m_w)\ddot{x} + m_b l \ddot{\theta} = \frac{\tau_r + \tau_l}{r} - 2b_w \frac{\dot{x}}{r^2} \end{cases} \quad (11)$$

La coppia forzante può essere riscritta come un'unica $\tau = \tau_r + \tau_l$, dato che si stanno ignorando eventuali differenze fra i motori che possano portare a spostamenti dallo 0 sullo yaw. Si può scrivere quindi $\tau_r = \tau_l$ e quindi $\tau = 2\tau_r$, avendo successivamente l'accortezza di pilotare ogni motore con solo metà della coppia restituita in uscita dal controllore.

Il modello linearizzato verrà utilizzato per sintetizzare un controllore PID. La simulazione verrà invece eseguita usando il modello non lineare.

Usando l'addon Symbolic Math Toolbox di Matlab, si possono agevolmente maneggiare le equazioni differenziali per ottenere l'espressione di una funzione di trasferimento in Laplace. Quest'ultima può essere convertita in un oggetto 'tf' di Matlab con il comando 'sym2tf' dell'addon MIMO Toolbox.

Con i seguenti parametri costruttivi

Nome	Parametro	Valore
b	coefficiente di attrito delle ruote	0.001
g	accelerazione di gravità	9.81 m/s^2
r	raggio delle ruote	0.032 m
l	distanza fra centro delle ruote e centro di massa	0.035 m
M_{body}	massa del robot escluse le ruote	0.232 kg
M_{wheel}	massa delle ruote	0.012 kg
I_{body}	inerzia del corpo escluse le ruote	9.4733e - 05 $kg \cdot m^2$
I_{wheel}	inerzia delle ruote	1.2288e - 05 $kg \cdot m^2$

si ottiene la funzione di trasferimento

$$F(s) = -13288 \frac{s + 183}{(s + 921.7)(s + 14.31)(s - 14.68)} \quad (12)$$

che ha due poli a parte reale negativa e uno a parte reale positiva, che rende il sistema instabile.

0.3 Sintesi del controllore

Un controllore PID può essere calibrato sia analiticamente che per tentativi. Sono state provate entrambe le alternative. Questo passaggio, assieme a quello di simulazione, sono necessari a dimensionare correttamente i motori prima del loro acquisto per la realizzazione del robot.

Prendendo in considerazione la funzione di trasferimento del processo nella forma

$$F(s) = -K \frac{s + z_1}{(s + p_1)(s + p_2)(s - p_3)}, K \geq 0, z_1 \geq 0, p_1 \geq 0, p_2 \geq 0, p_3 \geq 0 \quad (13)$$

e quella di un controllore PID ideale

$$G(s) = \frac{K_d s^2 + K_p s + K_i}{s} \quad (14)$$

Il controllore a dimensione minima che può stabilizzare il processo si ottiene raccogliendo al numeratore il fattore K_d

$$G(s) = K_d \frac{s^2 + \frac{K_p}{K_d} s + \frac{K_i}{K_d}}{s} \quad (15)$$

e usando il numeratore di $G(s)$ per semplificare i due poli a parte reale negativa del processo, ottenendo la funzione di trasferimento in catena aperta

$$F(s) = -K_d K \frac{s + z_1}{s(s - p_3)} \quad (16)$$

e le seguenti condizioni sul numeratore di $G(s)$

$$\frac{K_p}{K_d} = p_1 + p_2 \quad (17)$$

$$\frac{K_i}{K_d} = p_1 \cdot p_2 \quad (18)$$

Quindi la funzione ad anello chiuso è

$$W(s) = \frac{N_F}{N_F + D_F} = -KK_d \frac{s + z_1}{s(s - p_3) + -K_dK(s + z_1)} \quad (19)$$

che secondo il criterio di Routh è stabile per

$$K_d = -\frac{p_3}{K} \quad (20)$$

Questo dimostra che il sistema è stabilizzabile con un controllore PID e fornisce dei valori dei parametri K_p, K_i, K_d per cui esso è stabile.

0.4 Simulazione

L'intero sistema processo+controllore è stato simulato in Matlab Simulink.

Il sistema è stato realizzato come un blocco 'Funzione Matlab' di Simulink, che permette di esprimere il calcolo delle funzioni differenziali del processo usando codice Matlab invece che il sistema a blocchi di Simulink. Nel caso, come questo, in cui le equazioni siano composte da molti termini, l'uso del blocco 'Funzione Matlab' permette di avere un sistema molto più agevole da modificare e su cui fare debugging.

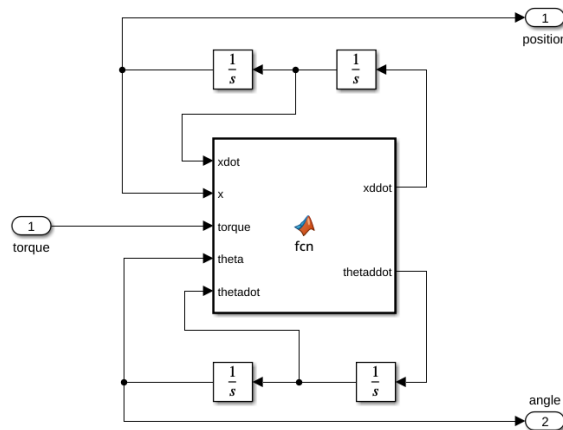


Figure 2: Sottosistema non lineare

Un *live script* di Matlab chiama la simulazione Simulink dopo aver impostato le costanti dei parametri fisici del robot. Questo permette di fare diverse simulazioni velocemente sperimentando l'effetto della variazione dei parametri sul comportamento del sistema. Al contrario, calibrare il PID direttamente sul microcontrollore richiederebbe molto più tempo per la necessità di dover ricompilare e ricaricare il codice e reimpostare tutto il sistema ad ogni nuovo tentativo.

La simulazione permette inoltre di calibrare il PID per tentativi, tramite il metodo di Ziegler-Nichols. Solo l'angolo di pitch viene usato ai fini del controllo, ma ai fini della simulazione è interessante vedere il comportamento della posizione al variare dei differenti parametri del robot.

È stato inizialmente realizzato un controllore PID ideale in tempo continuo, provando una leggera variazione dei parametri ottenuti analiticamente.

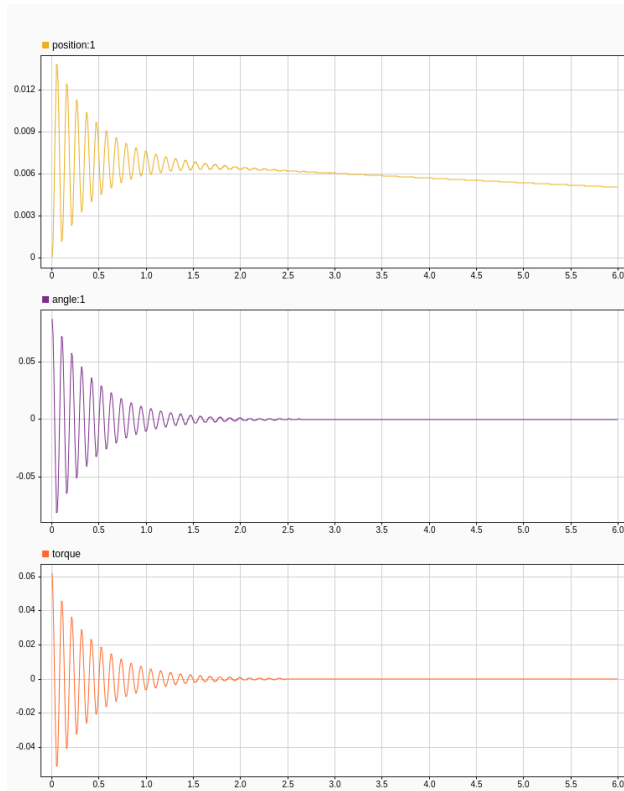


Figure 3: Comportamento del robot con $K_p=-0.6$, $K_i=-0.2$, $K_d=-0.001$ rispetto uno spostamento di 5 gradi dall'equilibrio

Dato che il controllore verrà implementato su microcontrollore, un PID a tempo continuo è di poco uso. Il blocco di controllore PID tempo continuo è stato dunque sostituito con uno a tempo discreto, tempo di campionamento di 1 ms, e nuovamente calibrato. Il microcontrollore è abbastanza potente da poter eseguire i calcoli necessari con questa frequenza. Al PID segue un organo di ritenuta di ordine zero, così che il controllore discreto possa controllare il processo a tempo continuo. Equivalentemente, una discretizzazione del controllore a tempo continuo può essere ottenuta con la funzione `c2d` di MatLab, indicando lo ZOH come metodo di discretizzazione e tempo di campionamento di 1 ms. È interessante notare come il tempo di campionamento sia abbastanza basso da non far differire i valori né il comportamento del controllore PID a tempo discreto da quello a tempo continuo.

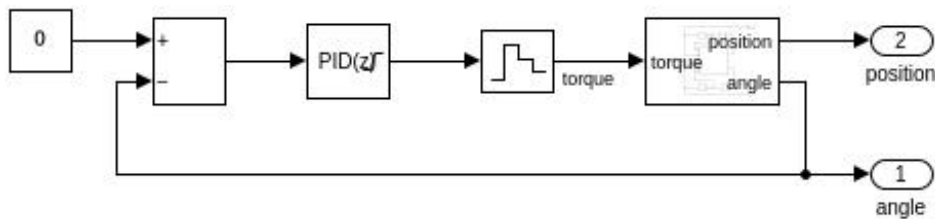


Figure 4: Scheda di simulazione del controllore PID discreto

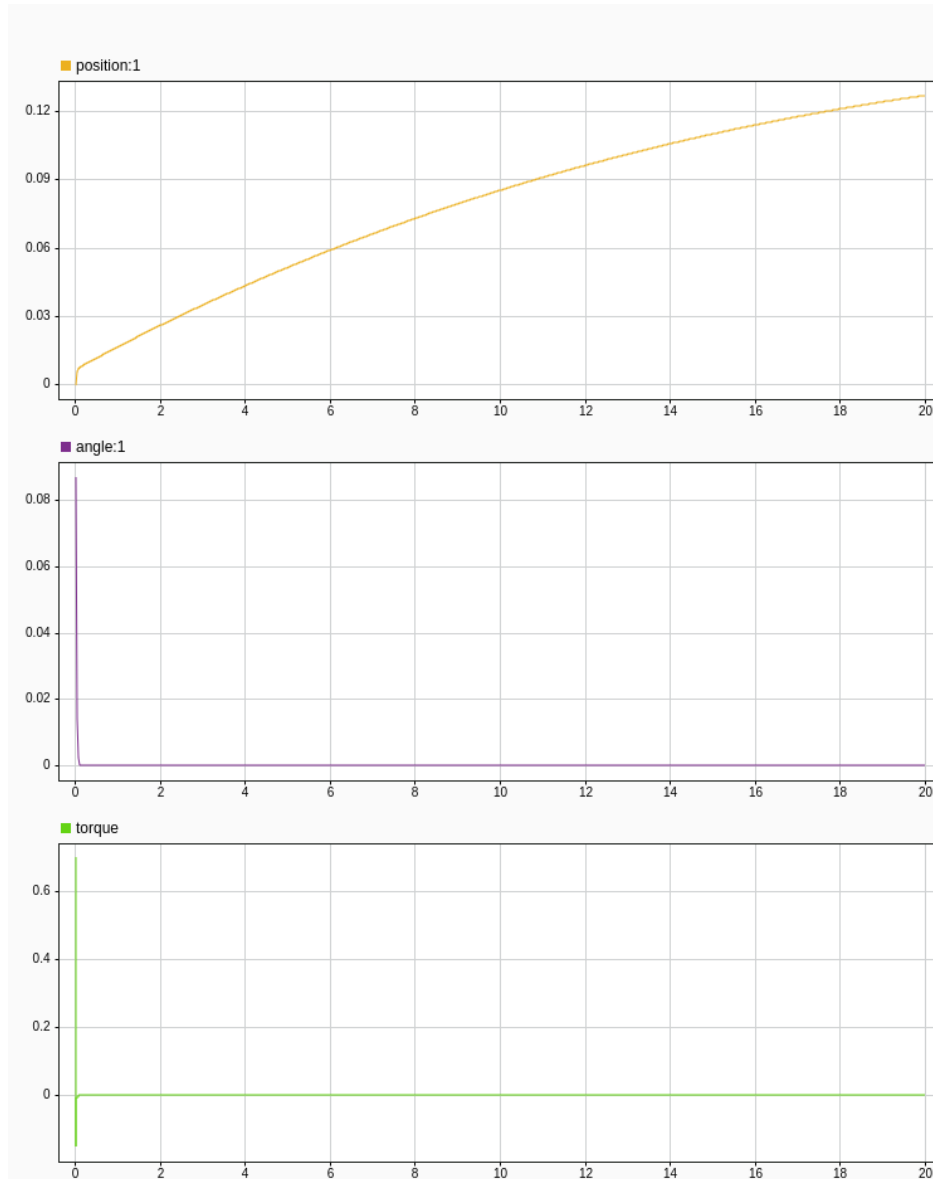


Figure 5: Comportamento del robot con PID a tempo discreto e $K_p=-5.8$, $K_i=-0.2$, $K_d=-0.1$ rispetto ad uno spostamento dall'origine di 5°

È interessante inoltre notare come in entrambi i casi, al fine di tenere il robot in equilibrio, il robot rimane in perenne movimento in una direzione. Questo avviene perché non viene effettuato nessun controllo né sulla posizione né sulla velocità di movimento, ma solo sulla coppia da far erogare ai motori. Questo mette in luce una limitazione di questo sistema: nella pratica è impossibile avere infinito spazio a disposizione del robot perché possa rimanere in equilibrio.

0.5 Realizzazione

0.5.1 Struttura

Una prima versione della struttura del robot è stata realizzata "ad occhio", cercando comunque di limitare peso ed ingombro per evitare inerzie inutili. I parametri sono stati successivamente inseriti nella simulazione per meglio calibrare il controllore PID. Iterativamente la struttura è stata modificata (alleggerendola,

diminuendo l'altezza) secondo i risultati delle simulazione e nuove simulazioni eseguite per verificare il comportamento della nuova struttura.

È da notare come nelle simulazioni sia difficile la stima esatta dei valori di inerzia delle varie masse. Sono stati valori approssimati a dischi pieni vincolati al centro per ruote e motori e di una sbarra rettangolare di massa uniforme per il corpo del robot.

Avendo accesso sia ad una stampante 3D che ad una tagliatrice a laser, ho preferito realizzare la struttura in compensato da 3mm tagliato con quest'ultima, che permette tempi di lavoro molto minori rispetto a pezzi stampanti in 3D (nell'ordine dei minuti invece che delle ore) ma richiede alcuni accorgimenti in fase di progettazione per quanto riguarda l'assemblaggio e la solidità della struttura. Le ruote e gli ancoraggi delle schede elettroniche alle basi di legno sono comunque stati stampati in 3D per la complicatezza delle forme. Per assicurare l'attrito con il terreno, le ruote hanno intorno elastici di gomma. Inoltre, le ruote si avvitano direttamente sugli *horn* dei servomotori.

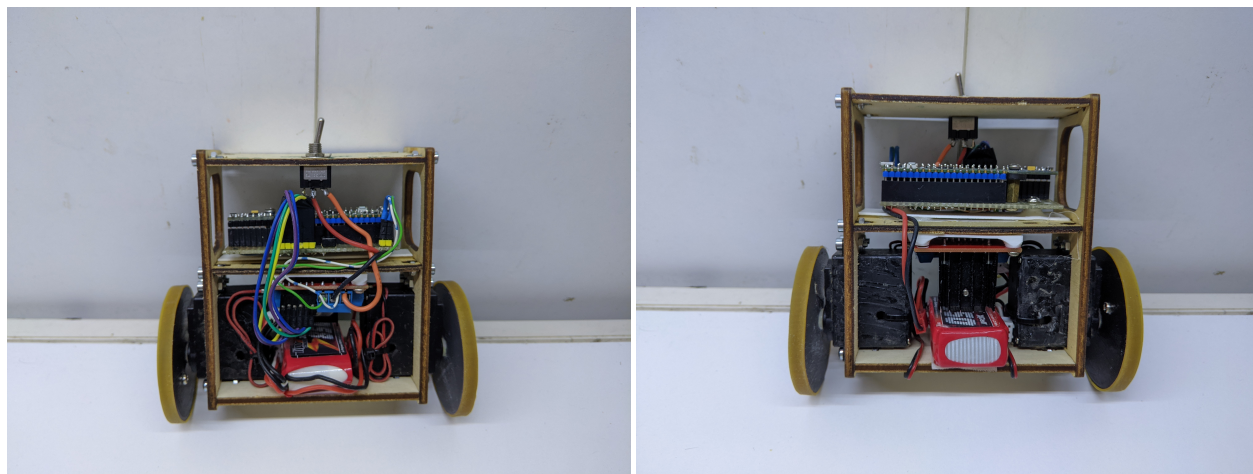


Figure 6: Versione finale del robot

0.5.2 IMU

Per valutare l'angolo di pitch (e anche quelli di yaw e roll, sebbene non usati) viene usata a IMU a 6 assi, 3 di giroscopio e 3 di accelerometro. Il giroscopio riporta le velocità angolari attorno i tre assi di rotazione, mentre l'accelerometro riporta l'accelerazione istantanea lungo i tre assi cartesiani. Per avere una misura degli angoli di rotazione è stato usato un semplice algoritmo di dead reckoning: usando la tecnica di Eulero all'indietro, i valori delle velocità angolari vengono integrati nel tempo per ottenere una posizione angolare relativa al punto in cui il sensore è stato acceso. Per evitare che il bias intrinseco del giroscopio porti a derive della misura, gli stessi angoli di rotazione vengono calcolati usando i dati provenienti dall'accelerometro, successivamente usando un filtro complementare per unire i due dati in percentuale 98% giroscopio e 2% accelerometro. Fare sensor fusion è necessario poiché i bias intrinseci del giroscopio porterebbero velocemente a derive della misura, ma comunque l'accelerometro è troppo suscettibile al rumore e alle altre forze applicate al corpo del robot (anche dal robot stesso) perché restituisca una misura affidabile. Tuttavia, l'accelerometro risponde bene a cambiamenti veloci degli angoli, ed è comunque necessario per ottenere buoni tempo di risposta a cambiamenti di posizione.

Dopo vari test, il filtro complementare è stato scartato e rimpiazzato con un filtro Madgwick, in quanto il primo soffriva comunque troppo del rumore dell'accelerometro per le rapide variazioni di angolo che si verificavano durante il controllo. Il filtro Madgwick restituisce un quaternion che può essere convertito negli angoli di Eulero, trattando con cura le singolarità che si verificano quando il pitch è intorno a $\pm \frac{\pi}{2}$. Tuttavia esso richiede un paio di secondi all'avvio per assestarsi, tempo in cui il PID non deve essere in funzione pena l'accumulo del termine integrale senza che ci sia un vero errore da correggere.

0.5.3 Motori

La simulazione riporta specifiche di coppia (max. ca. $0.6Nm$, che va diviso a metà fra i motori destro e sinistro) e velocità di rotazione dei motori (max. ca. 30 rpm), sono dunque stati scelti dei servomotori motoridotti HS-322HD, alimentati con una Lipo 2S da 7.4V. Per un motore CC, la coppia meccanica erogata dipende dalla corrente di armatura, che può a sua volta essere controllata dalla tensione di alimentazione se si conosce la resistenza di armatura. Il datasheet dei motori riporta una coppia massima di $0.3Nm$ a 6V di alimentazione. I motori vengono pilotati dal microcontrollore in tensione (PWM). Per stabilire un chiaro rapporto fra tensione applicata e coppia erogata, la coppia è stata misurata a diversi valori di PWM usando un dinamometro analogico e osservando una relazione praticamente lineare. Dunque, una volta che il microcontrollore calcola il valore di coppia necessario in un dato istante, esso viene convertito nel valore di tensione (PWM) corrispondente ed impostato sul motor driver.

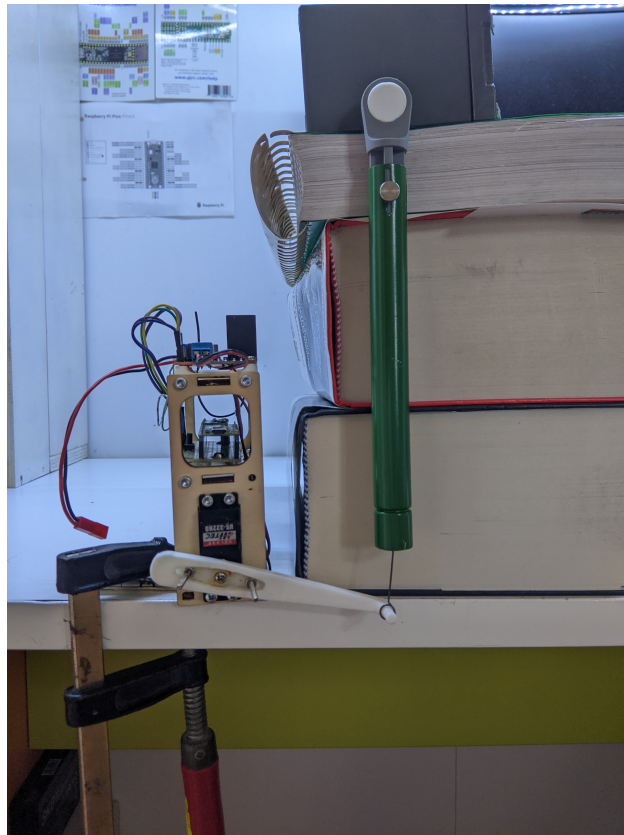


Figure 7: Setup per la misura della coppia. Entrambi i pezzi che collegano motore al dinamometro e fermano il dinamometro sulla "piattaforma" sono stati realizzati in stampa 3D.

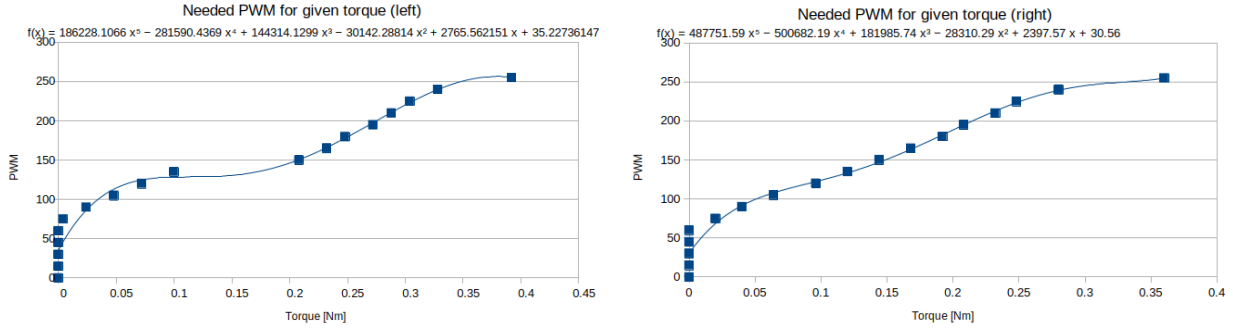


Figure 8: Rapporto fra tensione erogata (PWM) dal motor driver e coppia in uscita. La relazione è espressa da un polinomio di grado 5.

Dopo ripetuti test si è osservato che comunque i motori entrano in saturazione se il robot esce dall'intervallo di pitch di $\pm 5 \text{ rad}$. Questo accade anche in casi in cui il piano su cui poggia il robot è leggermente storto e il fenomeno può essere facilmente osservato vedendo che i valori di PWM calcolati in questi momenti eccede il valore massimo consentito da Arduino (che si traduce praticamente nella necessità di pilotare i motori ad una tensione più alta di quella possibile e quindi nella richiesta di applicazione di una coppia maggiore a quella massima erogabile dai motori). Questa è un'altra limitazione di questo robot ed in futuro i motori dovrebbero essere cambiati con motori leggermente sovradimensionati che permettano delle "manovre di emergenza".

0.5.4 Codice

Il microcontrollore Raspberry Pi Pico è programmato in C++ usando il framework Arduino. Il controllore PID è implementato dalla libreria ArduPID, che implementa fra le tante algoritmi di anti-windup, saturazione e si occupa di eseguire i calcoli con timestep adeguati. Il filtro Madgwick per la lettura della IMU è implementato dalla libreria FastIMU.

0.6 Ulteriori considerazioni

- Effettuare il controllo usando la coppia dei motori come forzante è estremamente sensibile a variazioni o imprecisioni dei parametri costruttivi e ad errori di misura dell'angolo. La sensibilità ad errori di misura era chiaramente visibile con l'uso del filtro complementare, come accennato prima. Questa imprecisione è dovuta anche al fatto che la coppia viene controllata solo in catena aperta. Per avere migliori prestazioni, il controllo dovrebbe avvenire ad anello chiuso, ad esempio usando la velocità angolare delle ruote o quella lineare di spostamento.
- Il controllo della posizione o della velocità richiederebbe un anello di controllo più esterno rispetto a quello di controllo del pitch e l'uso di encoder incrementali sui motori.
- Il controllo dello yaw, anche solo per eliminare le differenze fra i due motori, richiederebbe un ulteriore anello di controllo, più interno a quello del pitch, che mantenga la sincronia fra le ruote. Lo schema a blocchi apparirebbe così:

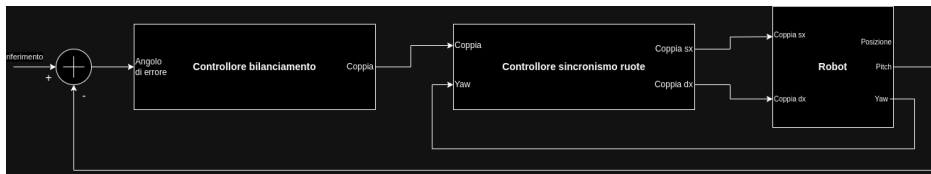


Figure 9: Anello di controllo interno dello yaw

Alternativamente può essere realizzato in catena aperta, sfruttando la differenza di coppia massima fra i due motori, applicando un fattore correttivo a uno dei due: > 1 per il più lento, < 1 per il più veloce. Applicare un fattore correttivo al motore più veloce eviterebbe di mandare quello più lento in saturazione, ma potrebbe risultare nell'applicare una coppia non sufficiente a mantenere il bilanciamento, nuovamente mostrando la necessità di usare motori che siano sovradimensionati e possibilmente una forzante diversa dalla coppia che possa essere controllata con un anello più interno di feedback, per esempio la velocità dei motori[2].

- I file di progetto della struttura, le simulazione Matlab e Simulink, i grafici di misura della coppia e il codice del microcontrollore sono disponibili sotto licenza copyleft GPLv3 su una repository git (mirror GitHub). Video del robot in funzione sono disponibili su YouTube

Riferimenti

- [1] P. Frankovský, L. Dominik, A. Gmiterko, and I. Virgala. Modeling of two-wheeled self-balancing robot driven by dc gearmotors. *Int. J. of Applied Mechanics and Engineering*, 22(3):793–747, 2017.
- [2] Felix Grasser, Aldo D’Arrigo, Silvio Colombi, and Alfred C. Rufer. Joe: A mobile, inverted pendulum. *IEEE Transactions on industrial electronics*, 49(1):107–114, 2002.