

POLITECNICO DI MILANO
Corso di Laurea in Ingegneria Informatica
Dipartimento di Elettronica e Informazione



Ballbot: studio e realizzazione di un prototipo di robot unicycle

Relatore: Prof. Matteo Matteucci

Correlatore: Ing. Roberto Sannino

Correlatore: Ing. Daniele Caltabiano

**Tesi di Laurea di:
Antonio Micali, mat. 750700**

Anno Accademico 2010-2011

Sommario

Questa tesi si colloca nell'ambito della robotica mobile. Questa si prefigge lo scopo di realizzare robot di piccole-medie dimensioni capaci di muoversi in spazi limitati, di divincolarsi tra gli ostacoli e di interagire con l'uomo. Negli ultimi anni, questo settore ha avuto una notevole espansione dovuta soprattutto all'invenzione di sensori sempre più evoluti. Gli ultimi robot realizzati assumono diverse forme e caratteristiche: robot umanoidi, robot marini, robot volanti, etc. Particolare attenzione è stata dedicata a robot in grado di rimanere bilanciati su due ruote e muoversi su di esse. Una possibile evoluzione, di questa tipologia, può essere realizzata passando da due punti di appoggio sul terreno ad uno solo. Utilizzando una comune palla come appoggio a terra, è possibile ideare un robot in grado di bilanciarsi e muoversi su di essa e che, rispetto ai primi, ha il vantaggio di potersi spostare in qualsiasi direzione senza bisogno di curvare.

L'obiettivo di questa tesi è quello di studiare, progettare e realizzare un prototipo di robot in grado di bilanciarsi e muoversi su una comune palla da calcio.

La parte di studio ha visto la modellazione del robot e la simulazione del sistema dinamico in ambiente MATLAB - Simulink. Sul modello simulato è stato applicato un controllore in modo da poterne verificare l'efficacia.

Nella fase di progettazione è stata scelta l'elettronica che permette al robot di azionare i motori, di conoscere la propria inclinazione e elaborare i movimenti da fare per stare in equilibrio. Parte importante del progetto è stata dedicata allo sviluppo di algoritmi capaci di stimare l'angolo di inclinazione e l'accelerazione da assegnare ai motori.

Durante la prototipazione si è invece disegnato e realizzato il telaio, assemblata la meccanica e l'elettronica e provato realmente l'intero sistema. Nelle

prove sperimentali si sono ottenuti degli ottimi risultati. In particolare il robot riesce a bilanciarsi sulla palla con piccoli movimenti, a compensare disturbi come spinte o variazioni di baricentro e anche a seguire traiettorie prefissate.

Ultima parte della tesi è stata la realizzazione di un programma che permettesse ad un qualsiasi utente di controllarne i movimenti.

Ringraziamenti

Un sentito ringraziamento a tutti i professori che mi hanno seguito in questi anni di studi trasmettendomi parte delle loro conoscenze.

Un particolare riconoscimento va al professor Andrea Bonarini, relatore della mia tesi di laurea triennale, che mi ha indirizzato presso la ST Microelectronics per la preparazione dell'attuale tesi.

Ringrazio il professor Matteo Matteucci, relatore dell'attuale tesi, e l'ingegnere Daniele Caltabiano, tutor aziendale presso la ST Microelectronics, per il supporto e gli insegnamenti che entrambi mi hanno impartito.

Desidero ringraziare anche tutte le persone che lavorano presso la divisione AST Robotics per la continua disponibilità e prontezza nei chiarimenti e suggerimenti, con particolare riferimento all'ingegnere Roberto Sannino, responsabile della divisione AST Robotics.

Infine un forte abbraccio ai miei familiari che non hanno mai smesso di avere fiducia in me spronandomi a fare sempre di più.

Indice

| | |
|---|-----------|
| Sommario | I |
| Ringraziamenti | V |
| 1 Introduzione | 1 |
| 1.1 Obiettivi | 2 |
| 1.2 Breve descrizione del lavoro | 2 |
| 1.3 Risultati | 3 |
| 1.4 Struttura della tesi | 4 |
| 2 Stato dell'arte | 7 |
| 2.1 Robot bilanciato su due ruote | 7 |
| 2.2 Prime versioni di Ballbot | 8 |
| 3 Studio del modello | 13 |
| 3.1 Angoli RPY | 13 |
| 3.2 Cinematica | 16 |
| 3.3 Dinamica | 18 |
| 3.3.1 Momenti di inerzia e legami cinematici | 20 |
| 3.3.2 Soluzione della dinamica mediante equazioni di Lagrange | 23 |
| 4 Algoritmi | 29 |
| 4.1 IMU: stima dell'inclinazione | 29 |
| 4.1.1 Filtro di Kalman | 29 |
| 4.1.2 EKF | 33 |
| 4.1.3 Accelerometri | 34 |
| 4.1.4 Giroscopi | 36 |

| | | |
|----------|---|-----------|
| 4.1.5 | Stima di Roll, Pitch e Yaw tramite EKF | 37 |
| 4.2 | Controllo della stabilità | 41 |
| 4.2.1 | Controllore PID | 42 |
| 4.2.1.1 | Tuning del controllore | 43 |
| 4.2.1.2 | Algoritmo PID | 46 |
| 4.2.2 | Simulazione del modello e del controllo | 47 |
| 4.2.2.1 | Risultati della simulazione | 48 |
| 4.2.3 | Controllo sperimentale | 49 |
| 5 | Prototipazione | 53 |
| 5.1 | Descrizione meccanica | 54 |
| 5.1.1 | Telaio | 54 |
| 5.1.2 | Motori | 54 |
| 5.1.3 | Ruote omnidirezionali | 57 |
| 5.2 | Descrizione elettronica | 58 |
| 5.2.1 | STM32 | 59 |
| 5.2.2 | iNemo 2 | 61 |
| 5.2.2.1 | IMU: Realizzazione pratica | 63 |
| 5.2.3 | Motori stepper | 65 |
| 5.2.3.1 | Principio di funzionamento | 65 |
| 5.2.4 | ST eval6470 | 67 |
| 5.2.4.1 | SPI | 68 |
| 5.2.5 | ST L5973 | 70 |
| 5.2.6 | Modulo XBee | 71 |
| 5.2.7 | Collegamenti tra i vari componenti | 71 |
| 5.3 | GUI | 72 |
| 5.3.1 | Comunicazione tra PC e Ballbot | 76 |
| 5.4 | Risultati ottenuti | 77 |
| 6 | Direzioni future e conclusioni | 83 |
| 6.1 | Valutazioni conclusive | 83 |
| 6.2 | Sviluppi futuri | 84 |
| | Bibliografia | 86 |

| | | |
|----------|-----------------------------------|-----------|
| A | Manuale di utilizzo | 89 |
| B | Collegamenti tra le board | 93 |
| C | Elenco delle abbreviazioni | 95 |

Capitolo 1

Introduzione

Lo sviluppo di nuove tecnologie rende possibili, in molti ambiti, applicazioni che, fino a poco prima, sembravano impensabili. Nella rivoluzione tecnologica in cui ci troviamo in questi anni, avvenimenti di questo genere sono quasi all'ordine del giorno.

Molti campi scientifici risentono positivamente di questa rivoluzione. Tra questi c'è sicuramente quello della robotica mobile che sta evolvendosi molto velocemente. Dopo i primi robot degli anni 80, molto grossi e dai pochi e buffi movimenti, si stanno sviluppando prototipi sempre più piccoli e sofisticati, capaci di compiere operazioni che nemmeno l'uomo è in grado di svolgere. Alcuni di questi robot sono in grado di entrare e interagire in luoghi dove, magari per pericolo di crolli o presenza di sostanze dannose alla salute, l'uomo non può entrare, oppure volare, dotati di ali, come se fossero uccelli.

Ultimamente, l'avvento della tecnologia MEMS ha dato una notevole spinta a questo mondo permettendo a un chip di pochi millimetri quadrati di effettuare misure che prima erano difficili da effettuare anche con grossi e costosi sensori. Due tipologie interessanti, realizzati anche con tecnologia MEMS, sono gli accelerometri e i giroscopi, che sono in grado di misurare accelerazione e velocità angolare. Questi due tipi di sensori hanno portato alla realizzazione, tra l'altro, di robot in grado di stare in equilibrio e di muoversi su due ruote oltre al famoso Segway [6].

1.1 Obiettivi

Lo scopo di questa tesi è quello di creare un robot in grado di stare in equilibrio e muoversi su una comune palla da calcio. Ricevendo i comandi da un utente, il robot deve potersi muovere in ogni direzione; quindi non solo avanti, indietro, destra e sinistra ma anche in diagonale. In questo modo è possibile fargli percorrere qualsiasi traiettoria nel piano 2D.

Come già accennato precedentemente, in quest'ultimo periodo c'è stato un elevato sviluppo, sia a livello amatoriale che commerciale, di robot capaci di equilibrarsi e muoversi su due ruote. Molti sono i pregi di questi sistemi ma contano come difetto l'impossibilità di muoversi lateralmente senza prima girarsi. La naturale evoluzione per risolvere questo problema è di sostituire le due ruote con una sfera che, rispetto ai primi, ha la possibilità di muoversi in ogni direzione.

Come è facile notare, il robot così creato risulterà instabile. Infatti, se venisse appoggiato, con i motori fermi, su una palla cadrebbe velocemente. Il problema complicato è quello di rendere stabile il sistema complessivo.

1.2 Breve descrizione del lavoro

Il lavoro si è diviso principalmente in tre parti: studio teorico, progettazione e prototipazione. Inizialmente si è studiato il problema dal punto di vista teorico scrivendo le equazioni che descrivono il modello dinamico. Il sistema è stato poi simulato e controllato in ambiente MATLAB - Simulink riuscendo ad ottenere il risultato voluto.

Nella seconda parte si sono decisi i componenti da usare, precisamente si è deciso di utilizzare quattro motori stepper con i quali muovere la palla, un'unità inerziale per la stima dell'inclinazione e un microcontrollore per la fusione dei dati provenienti dai sensori, l'elaborazione e l'invio dei comandi verso i motori.

tra i vari sensori è presente un'unità inerziale che si occupa di stimare l'inclinazione del robot. Per far ciò utilizza accelerometri e giroscopi, sensori che misurano grandezze diverse dall'inclinazione ma, con le quali è possibile

ricavarla con semplici passaggi matematici. Queste misure vengono elaborate tramite un filtro di Kalman per ottenere gli angoli di Eulero RPY che identificano l'inclinazione del robot.

Il microcontrollore implementa quattro controllori PID, due per ogni possibile direzione di movimento, il primo per gestire la stabilità, l'altro per controllare la velocità di movimento. In particolare, il microcontrollore acquisisce i dati provenienti dall'unità inerziale e, tramite i due controllori PID, li elabora per ottenere le accelerazioni con cui si devono muovere i motori. Le accelerazioni calcolate tramite i PID vengono integrate, in modo da ottenere le velocità di movimento delle ruote, e inviate alle schede di controllo dei motori che li attuano secondo le specifiche volute. Le coppie sviluppate dai motori vengono applicate alla palla tramite quattro ruote omnidirezionali, in questo modo è permesso il movimento della palla in ogni direzione.

Nella fase di prototipazione si è progettato il telaio e poi lo si è realizzato tramite fresatura e stampa 3D. Poi si sono montati i pezzi realizzati ed è stata posizionata opportunamente l'elettronica. Per rendere completamente mobile il robot, su di esso è stata aggiunta una batteria e un dispositivo radio che gli permetta di ricevere comandi da remoto.

1.3 Risultati

Dopo l'analisi teorica del problema si è progettato e realizzato un prototipo funzionante. Esso è in grado di restare in equilibrio su una palla spostandosi di pochi gradi dalla posizione di equilibrio.

Oltre al robot è stato sviluppato anche un software tramite l'utilizzo delle librerie QT sviluppate da Nokia. Questo software, tramite un'interfaccia intuitiva, consente di far seguire al robot le traiettorie volute, di modificare runtime alcuni parametri di controllo e di acquisire il log completo dello stato interno.

Il robot realizzato è completamente indipendente. Infatti, tramite l'applicazione di un dispositivo di comunicazione wireless e l'utilizzo di una batteria al piombo, può muoversi per circa una ventina di minuti.

1.4 Struttura della tesi

Di seguito viene elencata e descritta brevemente la struttura della tesi:

Capitolo 1 - Introduzione: riporta una breve introduzione al lavoro svolto, elencando i requisiti del sistema e i passi principali per giungere all'obiettivo finale.

Capitolo 2 - Stato dell'arte: presenta lo stato dell'arte descrivendo i robot simili già realizzati (sia su due ruote, sia sulla palla) e i vari algoritmi di controllo utilizzati.

Capitolo 3 - Studio del modello: descrive un possibile modello semplificato del robot, utilizzato per gran parte del progetto, e descrive i passaggi per derivarne la cinematica e la dinamica. Nel capitolo sono anche descritti gli angoli di Eulero RPY, estensivamente usati nella tesi.

Capitolo 4 - Algoritmi: presenta, dal punto di vista teorico e pratico, gli algoritmi usati durante le varie fasi del progetto: sono descritti gli algoritmi per il controllo e per la stima dell'angolo di inclinazione. Successivamente vengono descritti i sensori (accelerometri e giroscopi) le cui misure sono utilizzate in ingresso all'algoritmo per il calcolo dell'inclinazione. Nel capitolo sono anche mostrati i risultati di una simulazione al PC del modello dinamico del robot e del controllo.

Capitolo 5 - Prototipazione: descrive la fase di realizzazione del prototipo. In particolare viene tratteggiata la realizzazione delle parti meccaniche ed elettroniche e dell'interfaccia di comando. Riguardo alla parte elettronica, vengono descritti, nei dettagli, tutti i componenti utilizzati e il ruolo che hanno rappresentato nel progetto. Nella conclusione sono mostrati i risultati sperimentati ottenuti col prototipo realizzato.

Capitolo 6 - Direzioni future e conclusioni: presenta le valutazioni conclusive rispetto ai risultati ottenuti e alcuni possibili sviluppi futuri e migliorie del progetto.

Appendice A - Manuale di utilizzo: descrive come utilizzare il robot: dall'accensione, all'utilizzo dei comandi per i movimenti, ai settaggi dei parametri di configurazione.

Appendice B - Collegamenti tra le board: mostra come, in pratica, sono state collegate tra di loro le varie board costituenti l'elettronica del robot.

Appendice C - Elenco delle abbreviazioni: mostra una lista di acronimi e abbreviazioni usati nella tesi per meglio capirne il significato.

Capitolo 2

Stato dell'arte

L'idea di robot dotato di un unico punto d'appoggio è nata, come molte altre idee tecnologiche, in un film di fantascienza. In particolare, nella serie TV *Caprica* era presente un robot autonomo, Serge (Figura 2.1), in grado di muoversi su una sfera per svolgere diverse faccende domestiche.

Un primo tentativo di sviluppo di un robot unicycle [9] è stato realizzato utilizzando una ruota simile ad una palla da rugby. Il robot era dotato di un motore DC per muovere la ruota e di un giunto, posto tra la ruota e il corpo, per permettergli di inclinare il corpo a destra e sinistra. Per muoversi in avanti e indietro gli bastava inclinare il corpo verso la direzione di movimento invece, per curvare, era necessario piegare il corpo verso l'interno della curva.

2.1 Robot bilanciato su due ruote

I primi robot in grado di bilanciarsi autonomamente sono nati su due ruote coassiali. Questi sistemi sono in grado di risolvere il problema del pendolo inverso monoplanare utilizzando le due ruote per bilanciarsi attorno al punto di equilibrio. Avendo a che fare anche con disturbi esterni, questi sistemi non rimarranno immobili, ma subiranno sempre lievi oscillazioni. Per mantenere l'equilibrio sono state applicate diverse tecniche di controllo, dai classici algoritmi PID e LQR, all'apprendimento per rinforzo. Un approccio, sviluppato in [1], prevede l'utilizzo di due PID, uno per controllare la stabilità e un'altro per permettere i movimenti.



Figura 2.1: Serge, della serie TV Caprica

Su questa idea, si sono sviluppati anche altri sistemi in grado di svolgere i compiti più svariati come il trasporto di oggetti pesanti o la telepresenza. Il più famoso è sicuramente la piattaforma commerciale Segway (Figura 2.2) [3]. Presentato nel 2001 da Dean Kamen, è un mezzo di trasporto personale che può essere comandato semplicemente spostando il peso del corpo nella direzione di marcia. Il bilanciamento avviene tramite un controllo in retroazione (LQR) utilizzando giroscopi ed accelerometri come sensori.

2.2 Prime versioni di Ballbot

La prima versione reale di robot in grado di rimanere in equilibrio su una palla è stata realizzata dal Microdynamic Systems Laboratory della Carnegie Mellon University [11] (Figura 2.4(a)). Questa prima versione, puntando a interagire con persone umane, è stata realizzata con un'altezza che è circa quella dell'uomo. Il meccanismo di movimento è stato realizzato in modo inverso a quello di una classica pallina del mouse. Come visibile in Figura 2.3, la palla appoggia su delle sfere metalliche che sostengono il corpo del robot e viene mossa tramite dei rulli collegati a due motori DC. Con i due motori è possibile controllare solo i movimenti di tilt e roll; non c'è nessuna



Figura 2.2: Segway i2

possibilità di agire sul movimento di pan.

Per quanto riguarda il controllo, il sistema è stato modellato, in modo semplificato, tramite due unicicli lineari disaccoppiati (uno per il movimento di tilt, l'altro per il movimento di roll) come descritto in [2]. Questo disaccoppiamento ben si adatta al sistema di movimento usato perché ognuno dei due sistemi comanda, separatamente, il rullo relativo al moto controllato. Per il controllo di ogni sottosistema sono stati utilizzati due controllori PID messi a cascata, uno per la stabilità e uno per la velocità.

Come detto precedentemente, questa prima versione di Ballbot non è in grado di ruotare attorno al proprio asse verticale. Una versione che risolve questo problema, Figura 2.4(b) è stata realizzata nel 2009 da Masaaki Kumaga and Takaya Ochiai [7] utilizzando una particolare attuazione della palla proposta in [4]. Precisamente la palla appoggia su tre ruote omnidirezionali che, in base alle velocità che assumono, permettono il rotolamento della palla in ogni direzione e attorno al punto d'appoggio sul terreno. Anche questo Ballbot considera il sistema disaccoppiato lungo due piani verticali. Per ogni sottosistema il controllo è fatto semplicemente secondo un PD, in particolare

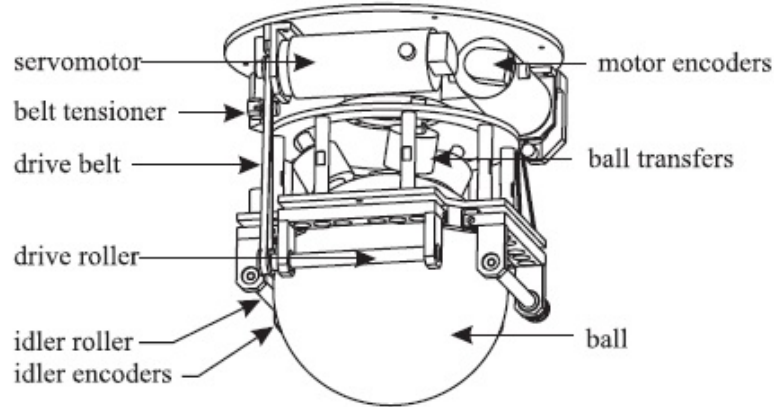


Figura 2.3: Comando palla come mouse

l'accelerazione che deve subire la palla lungo i due piani è così calcolata:

$$a_x = K_A \theta_x + K_{AV} \dot{\theta}_x + K_T (x - x_0) + K_V v_x \quad (2.1)$$

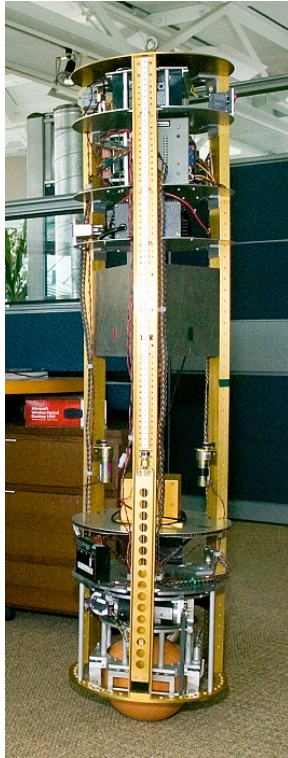
$$a_y = K_A \theta_y + K_{AV} \dot{\theta}_y + K_T (y - y_0) + K_V v_y \quad (2.2)$$

dove i guadagni proporzionali K_A e K_T e i guadagni derivativi K_{AV} e K_V sono stati trovati tramite esperimenti. Siccome sono stati usati dei motori stepper, Kumagai and Ochiai fanno anche notare come la variabile di controllo sia l'accelerazione e non l'usuale coppia. Una volta integrata l'accelerazione e trovata la velocità della palla nei due sottosistemi, la velocità da applicare alle ruote è ottenuta come:

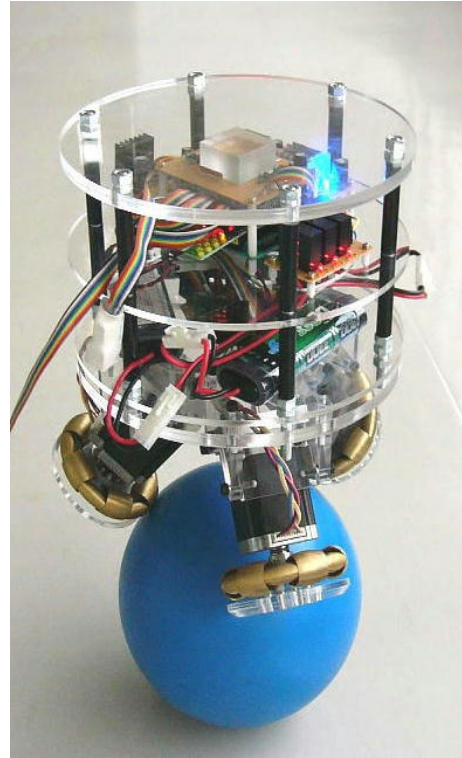
$$v_{s1} = -v_y \cos(\phi) + K_z \omega_z \quad (2.3)$$

$$v_{s2} = \left\{ \frac{\sqrt{3}}{2} v_x + \frac{1}{2} v_y \right\} \cos(\phi) + K_z \omega_z \quad (2.4)$$

$$v_{s3} = \left\{ -\frac{\sqrt{3}}{2} v_x + \frac{1}{2} v_y \right\} \cos(\phi) + K_z \omega_z \quad (2.5)$$



(a) Ballbot



(b) Ballbot



(c) Rezero



(d) Rezero applicazione

Figura 2.4: Diverse versioni di Ballbot

dove ϕ è l'angolo di appoggio della ruota sulla palla e $K_z = -R\sin(\phi)$ è il coefficiente di rotazione attorno all'asse verticale. L'ultima versione di Ballbot (Figura 2.4(c)) è stata realizzata dall'ETH Zurich. Anche questo robot utilizza il sistema di attuazione con tre ruote omnidirezionali e riesce a muoversi ad una velocità di 3,5 m/s. Su questo robot sono stati fatti anche degli studi applicativi come quello in Figura 2.4(d), in cui svolge la funzione di faro mobile.

Capitolo 3

Studio del modello

In questo capitolo studieremo il modello del Ballbot sia dal punto di vista cinematico che dal punto di vista dinamico. Per procedere nello studio è necessario introdurre un modello meccanico di base su cui lavorare. Questo modello, raffigurato in Figura 3.1, è composto solamente da parti meccaniche tali che una loro non considerazione renderebbe molto differente, dal punto di vista cinematico e dinamico, lo studio del modello.

Il modello di base si compone del corpo del robot e della palla. Il corpo del robot è formato da quattro ruote collegate a quattro motori uniti da un piano tondo. Su questo piano è fissata una batteria. Prima di proseguire nello studio del modello è necessario spiegare velocemente la teoria sugli angoli di Eulero RPY in quanto verranno molto usati nel proseguo.

3.1 Angoli RPY

Un corpo rigido è completamente descritto nello spazio conoscendo posizione e orientamento rispetto a una terna di riferimento. L'orientamento è possibile descriverlo tramite successioni di rotazioni elementari che vengono effettuate attorno a un asse arbitrario. Le rotazioni sono prese positive seguendo la regola della mano destra, o cavatappi.

La Figura 3.2 mostra le 3 usuali rotazioni elementari, attorno agli assi X, Y e Z. Ad ogni rotazione può essere associata una matrice di rotazione che per-

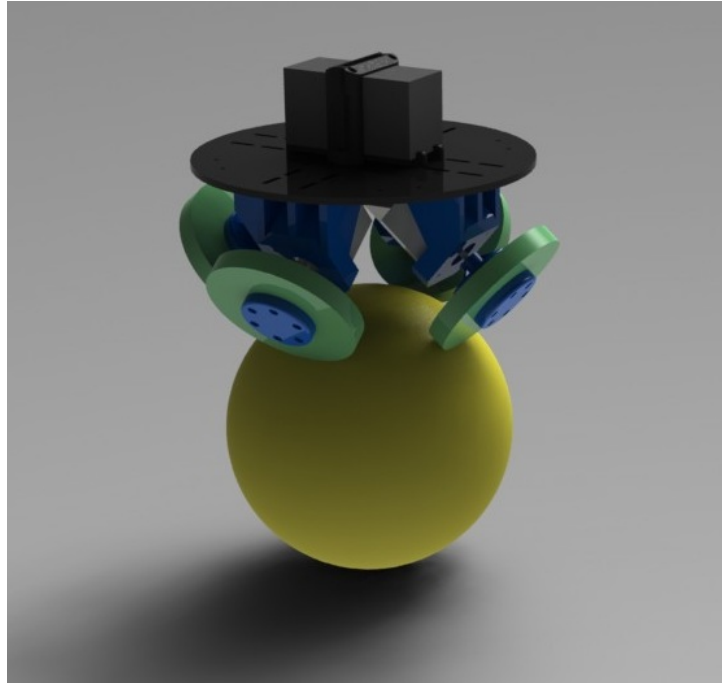
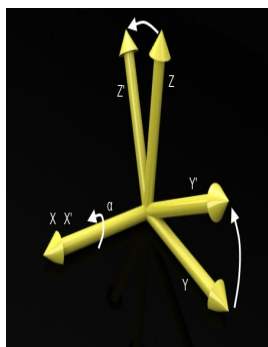
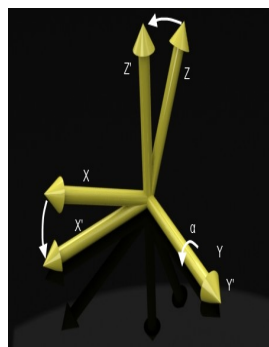


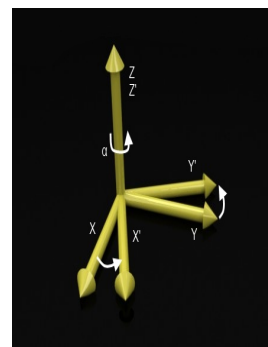
Figura 3.1: Modello di base di Ballbot



(a) Rotazione X



(b) Rotazione Y



(c) Rotazione Z

Figura 3.2: Rotazioni elementari

mette di ruotare i punti da una terna a un'altra. Per le rotazioni elementari mostrate abbiamo le seguenti matrici di rotazione:

$$R_X(\alpha) = \begin{bmatrix} c_\alpha & -s_\alpha & 0 \\ s_\alpha & c_\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

$$R_Y(\alpha) = \begin{bmatrix} c_\alpha & 0 & s_\alpha \\ 0 & 1 & 0 \\ -s_\alpha & 0 & c_\alpha \end{bmatrix} \quad (3.2)$$

$$R_Z(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha \\ 0 & s_\alpha & c_\alpha \end{bmatrix} \quad (3.3)$$

Nelle precedenti matrici è stata utilizzata la seguente notazione: c_x invece di $\cos(x)$, s_x al posto di $\sin(x)$.

Trasformazioni complesse, composte da più rotazioni elementari, si ottengono post-moltiplicando le matrici legate alle rotazioni elementari compiute.

Una rappresentazione minima di una terna nello spazio può essere ottenuta utilizzando un insieme di tre angoli. In campo aeronautico è frequentemente utilizzata una terna di angoli chiamata RPY, acronimo di Roll Pitch Yaw. La rotazione complessiva di cui gli angoli RPY sono rappresentativi viene così costruita:

- si ruota la terna origine dell'angolo ϕ attorno all'asse X; tale rotazione è descritta dalla matrice di rotazione $R_X(\psi)$ definita in (3.1)
- si ruota la nuova terna dell'angolo θ attorno all'asse Y; tale rotazione è descritta dalla matrice di rotazione $R_Y(\theta)$ definita in (3.2)
- si ruota la nuova terna dell'angolo ψ attorno all'asse Z; tale rotazione è descritta dalla matrice di rotazione $R_Z(\psi)$ definita in (3.3)

La rotazione globale è ottenuta nel seguente modo:

$$\begin{aligned}
R(\phi, \theta, \psi) &= R_X(\phi) \cdot R_Y(\theta) \cdot R_Z(\psi) \\
&= \begin{bmatrix} c_\phi c_\theta & c_\phi s_\theta s\psi - s_\phi c_\psi & c_\phi s_\theta c_\psi + s_\phi s\psi \\ s_\phi c_\theta & s_\phi s_\theta s\psi + c_\phi c_\psi & s_\phi s_\theta c_\psi - c_\phi s\psi \\ -s_\theta & c_\theta s\psi & c_\theta c_\psi \end{bmatrix} \quad (3.4)
\end{aligned}$$

3.2 Cinematica

In questa sezione viene ricavata la cinematica del robot tenendo conto che il robot si muove all'interno di un ambiente tridimensionale. Le equazioni cinematiche sono usate per descrivere la posizione la velocità e l'accelerazione del sistema rispetto all'ambiente.

Per iniziare, introduciamo i sistemi di riferimento necessari per la derivazione della cinematica del robot. Definiamo, come mostrato in Figura 3.3, una terna destrorsa E posizionata nell'ambiente e fissa rispetto a esso, una terna, sempre destrorsa, T posizionata sulla parte alta del robot e solidale a esso e una terna B posizionata nel punto di contatto tra palla e terreno e con assi paralleli a quelli della terna E. Le tre terne hanno centro, rispettivamente, O_e , O_t , O_b . Considerando la Figura 3.3 avremo quindi:

$$E = (X_e, Y_e, Z_e) \quad (3.5)$$

$$B = (X_b, Y_b, Z_b) \quad (3.6)$$

$$T = (X_t, Y_t, Z_t) \quad (3.7)$$

Il nostro scopo è quindi quello di trovare una trasformazione lineare che porti la terna E nella terna T (cinematica diretta) e la trasformazione opposta che trasformi la terna T nella terna E (cinematica inversa).

Iniziamo con l'osservare che la terna T può muoversi nell'ambiente con 6 DOF. Questi 6 gradi possono essere divisi in 3 di tipo traslatorio e altri 3 di tipo rotatorio.

I gradi di libertà traslatori rappresentano la posizione di O_t rispetto alla terna E, che chiamiamo Γ_E . Possiamo quindi scrivere questo vettore come:

$$\Gamma_E = (O_b O_e) + (O_t O_b) \quad (3.8)$$

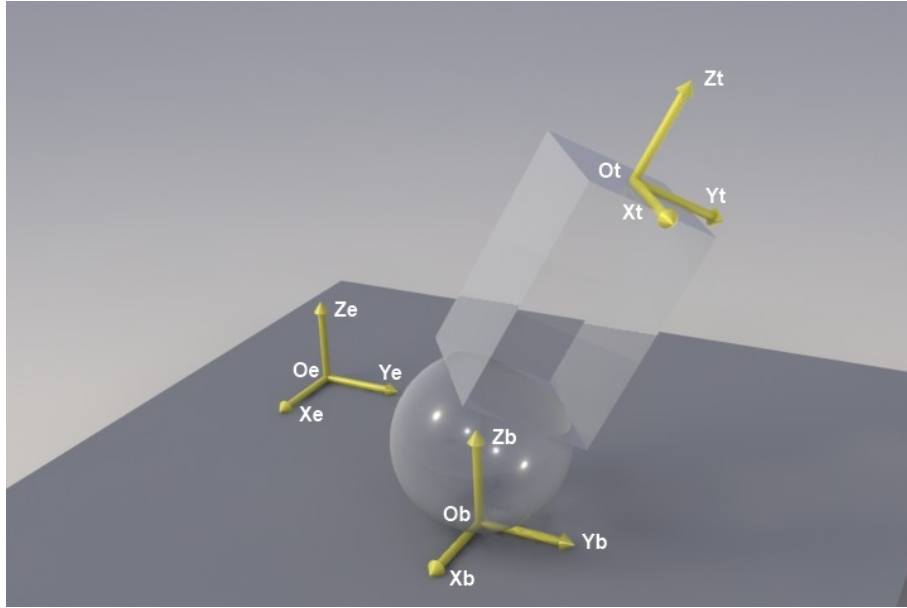


Figura 3.3: Sistemi di riferimento per il calcolo della cinematica

dove $(O_b O_e)$ è il vettore congiungente i centri delle terne E e B e $(O_t O_b)$ è il vettore congiungente i centri delle terne B e T.

Gli altri 3 DOF rappresentano le rotazioni da far compiere alla terna T per allineare gli assi con le altre due terne. Definiamo queste 3 rotazioni tramite angoli RPY (ϕ, θ, ψ) che raggruppiamo nel vettore Θ_E :

$$\Theta_E = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad (3.9)$$

Definiamo anche i vettori V_T e ω_T contenenti le componenti della velocità lineare e angolare rispetto alla terna T come:

$$V_T = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (3.10)$$

$$\omega_T = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (3.11)$$

Le velocità lineari rispetto ai due sistemi di riferimento E ed T sono legate dalla seguente relazione:

$$V_E = \dot{\Gamma}_E = V_{b,e} + R(\phi, \theta, \psi) \cdot V_T \quad (3.12)$$

con $V_{b,e}$ derivata del vettore $(O_b O_e)$, ossia la velocità lineare del punto O_b rispetto alla terna E.

Allo stesso modo, tramite una matrice $T(\phi, \theta, \psi)$, è possibile trovare un legame anche per le velocità angolari espresse nei due sistemi di riferimento E e T:

$$\dot{\Theta}_E = T(\phi, \theta, \psi) \cdot \omega_T \quad (3.13)$$

Ricordando l'ordine di come sono definite le rotazioni negli angoli RPY possiamo scrivere il legame tra ω_T e Θ_E nel seguente modo:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_X(\phi)^{-1} \cdot \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_X(\phi)^{-1} \cdot R_Y(\theta)^{-1} \cdot \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (3.14)$$

da cui è facile trovare che

$$T(\phi, \theta, \psi)^{-1} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\psi & c_\theta s_\psi \\ 0 & -s_\psi & c_\theta c_\psi \end{bmatrix} \quad (3.15)$$

e

$$T(\phi, \theta, \psi) = \begin{bmatrix} 1 & s_\psi t_\theta & c_\psi t_\theta \\ 0 & c_\psi & -s_\psi \\ 0 & s_\psi / c_\theta & c_\psi / c_\theta \end{bmatrix} \quad (3.16)$$

dove $t_x = \tan(x)$.

3.3 Dinamica

Nel seguito della tesi verrà proposta una simulazione al PC del sistema. Per poter effettuare la simulazione è necessario conoscere il modello matematico che descrive il robot. Tramite queste equazioni è possibile realizzare anche un controllore che permetta di stabilizzare il sistema simulato.

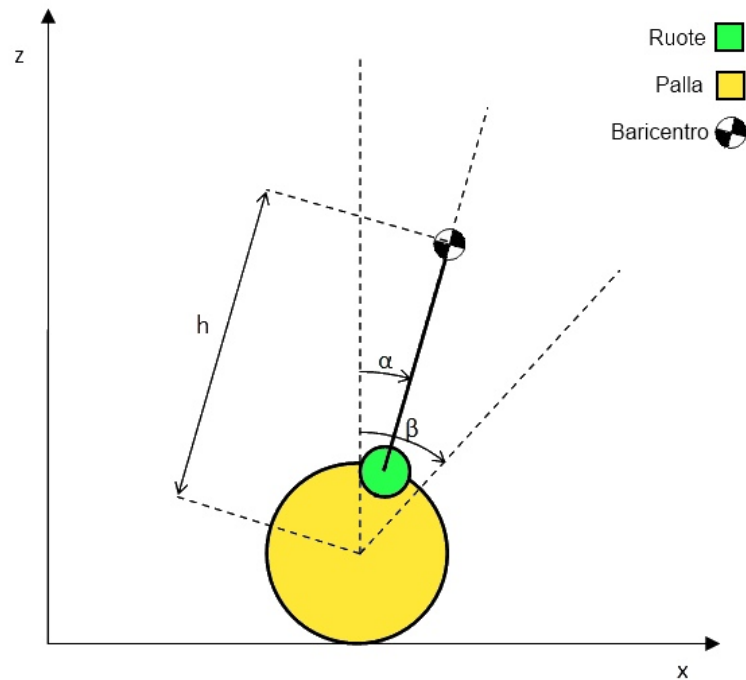


Figura 3.4: Modello semplificato

Il modello del sistema è ricavato da un sistema semplificato del robot. In particolare il sistema complessivo è visto come l'unione di due sottosistemi disaccoppiati tra loro. In questo modo, non influenzandosi a vicenda, è possibile trattare i due sottosistemi separatamente. I due modelli semplificati sono stati ottenuti sezionando il sistema tramite due piani verticali perpendicolari tra loro. I due sottomodelli così ottenuti risultano identici e per questo l'analisi viene effettuata su uno solo di questi, mostrato in Figura 3.4.

Il sottosistema utilizzato è composto da una palla, due ruote ogni una delle quali collegata a un motore senza l'utilizzo di alcun tipo di trasmissione e un corpo che collega i due motori. Per le successive analisi definiamo α l'angolo di inclinazione del corpo del robot rispetto alla verticale, β l'angolo di rotazione della palla e h la distanza del baricentro del corpo dal centro della palla.

3.3.1 Momenti di inerzia e legami cinematici

Per procedere allo studio della dinamica è necessario calcolare i vari momenti di inerzia e legare tra loro le velocità e le accelerazioni delle varie parti in movimento del sistema.

Il momento di inerzia rispetto ad un asse K di un sistema di n corpi puntiformi è definito come:

$$I_K = \sum_{i=1}^n m_i \cdot r_i^2 \quad (3.17)$$

dove m_i è la massa dell' i -esimo corpo e r_i è la minima distanza dell' i -esimo corpo dall'asse K . Similmente è definito anche il momento di inerzia di un corpo rigido di volume V rispetto a un asse K . In questo caso si considera il corpo come un insieme di punti materiali, ciascuno caratterizzato da un volume dV ed una massa $dm = \rho \cdot dV$ (dove ρ è la densità), ottenendo che:

$$I_K = \int_V \rho r^2 dV \quad (3.18)$$

Il corpo del robot lo assumiamo composto da un'asta di massa nulla su cui sono presenti due masse puntiformi M_m dovuta ai motori e alle ruote e M_p dovuta alla batteria. Il momento è calcolato rispetto all'asse passante dal baricentro del corpo, quindi perpendicolare al disegno di Figura 3.4:

$$I_{body} = M_m \cdot d_{mb}^2 + M_p \cdot d_{pb}^2 \quad (3.19)$$

dove d_{mb} e d_{pb} sono le distanze dei motori e della batteria dal baricentro del corpo.

Per il calcolo del momento di inerzia per le ruote si è assunto di considerarle come un corpo omogeneo di densità costante ρ , di raggio R_w e spessore H_w . Partendo dall'integrale della Formula 3.18 si ottiene:

$$\begin{aligned}
I_{wheel} &= \int_0^{R_w} \rho r^2 H_w 2\pi r \, dr \\
&= 2\pi \rho H_w \int_0^{R_w} r^3 \, dr \\
&= \frac{\pi \rho H_w R_w^4}{2} \\
&= \frac{1}{2} M_w R_w^2
\end{aligned} \tag{3.20}$$

dove con M_w è stata definita la massa della ruota.

Per il calcolo del momento di inerzia della palla, supponiamo che sia costituita da una sfera cava di raggio R_b e di densità superficiale, σ , uniforme e definita come:

$$\sigma = \frac{M_b}{4\pi R_b^2} \tag{3.21}$$

dove con M_b è stata definita la massa della palla. Utilizzando sempre la Formula 3.18 si ottiene:

$$\begin{aligned}
I_{ball} &= \int_0^{R_b} \sigma 8\pi R_b r^2 \, dr \\
&= \frac{2}{3} M_b R_b^2
\end{aligned} \tag{3.22}$$

Procediamo ora a descrivere i legami tra le velocità e le accelerazioni delle varie parti del sistema. Per far ciò osserviamo dalla Figura 3.5 come sono posizionate le ruote sulla palla. Sulla palla appoggiano quattro ruote a due a due parallele e inclinate di un angolo γ rispetto all'asse Z.

Iniziamo definendo:

$$r_b = R_b \cos(\gamma) \tag{3.23}$$

che rappresenta il raggio della sezione di palla su cui poggiano le ruote. A questo punto possiamo scrivere un'equazione sullo spazio percorso dalla ruota sulla palla (immaginiamo che non ci sia strisciamento tra ruota e palla):

$$R_w \cdot \theta_w = r_b \cdot (\beta - \alpha) \tag{3.24}$$

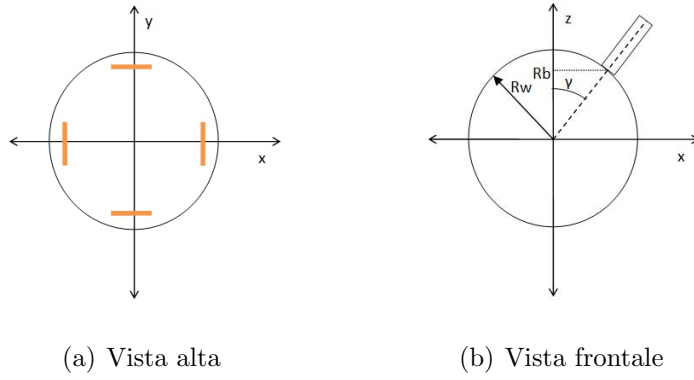


Figura 3.5: Viste del posizionamento delle ruote sulla palla

dove θ_w rappresenta l'angolo di rotazione della ruota e α e β sono gli angoli rappresentati in Figura 3.4. Possiamo poi calcolare lo spazio percorso dalla palla sul terreno nel seguente modo:

$$s_b = R_b \cdot \beta \quad (3.25)$$

Manipolando e derivando le (3.24) (3.25) e considerando che l'asse del motore è collegato direttamente alla ruota senza nessun tipo di trasmissione, possiamo scrivere le seguenti equazioni che legano tra di loro velocità e accelerazioni della palla, delle ruote e del motore:

$$\omega_m = \omega_w = \eta(\dot{\beta} - \dot{\alpha}) \quad (3.26)$$

$$\dot{\omega}_m = \dot{\omega}_w = \eta(\ddot{\beta} - \ddot{\alpha}) \quad (3.27)$$

$$v_b = R_b \cdot \dot{\beta} \quad (3.28)$$

$$a_b = R_b \cdot \ddot{\beta} \quad (3.29)$$

dove:

$$\eta = \frac{r_b}{R_w} = \frac{R_b \cos(\gamma)}{R_w} \quad (3.30)$$

Ulteriore necessità prima di ricavare le equazioni dinamiche è quella di calcolare la posizione del baricentro del corpo del robot. Come visto precedentemente, si è deciso di modellizzare il corpo come due masse puntiformi del valore di M_m e M_p poste, rispettivamente, a distanza l_m e l_p dal centro della palla. Dopo aver definito M_{body} , la massa totale del corpo, come:

$$M_{body} = M_m + M_p \quad (3.31)$$

possiamo trovare la posizione del baricentro rispetto al centro della palla nel seguente modo:

$$h_{body} = \frac{M_m \cdot l_m + M_p \cdot l_p}{M_{body}} \quad (3.32)$$

Per quanto riguarda invece la palla, il suo baricentro è posto nel centro della stessa.

3.3.2 Soluzione della dinamica mediante equazioni di Lagrange

Per il calcolo delle equazioni dinamiche si è scelto di procedere risolvendo le equazioni di Lagrange. Queste equazioni valgono solamente in presenza di vincoli lisci e bilateri. Nel caso in esame, essendo la coppia che esercitano i motori non conservativa, dobbiamo usare le equazioni di Lagrange nella forma:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_h} \right) - \frac{\partial T}{\partial q_h} - \frac{\partial U}{\partial q_h} = Q_h \quad h = 1, \dots, n \quad (3.33)$$

dove n è il numero di coordinate libere, q_h è la h -esima coordinata libera (detta coordinata lagrangiana), T è l'energia cinetica dell'intero sistema, U è l'energia potenziale e Q_h sono le componenti generalizzate di Lagrange che assumono la seguente espressione:

$$Q_h = \sum_i F_i \times \frac{\partial P_i}{\partial q_h} \quad (3.34)$$

in cui F sono le forze / coppie attive agenti sul sistema e P_i indica l'infinitesimo spostamento virtuale relativo all' i -esima forza / coppia attiva.

La prima grandezza da ricavare per procedere con l'impostazione delle equazioni di Lagrange è l'energia cinetica. Per un corpo puntiforme che si muove di moto traslatorio, l'energia cinetica è così definita:

$$T = \frac{1}{2} \cdot m \cdot v^2 \quad (3.35)$$

dove m è la massa del corpo e v è la sua velocità. Nel caso in cui il corpo si muova di moto rotatorio attorno a un asse K , l'energia cinetica è così calcolata:

$$T = \frac{1}{2} \cdot I_K \cdot \omega^2 \quad (3.36)$$

in cui I_K è il momento di inerzia calcolato rispetto all'asse K e ω è la velocità di rotazione attorno a K . Nel caso in cui siano presenti sia il contributo traslatorio che rotatorio, entrambe le componenti entrano nel calcolo dell'energia cinetica:

$$T = \frac{1}{2} \cdot m \cdot v^2 + \frac{1}{2} \cdot I_K \cdot \omega^2 \quad (3.37)$$

Nel nostro caso, l'energia cinetica relativa agli spostamenti traslatori, T_t , assume la seguente forma:

$$T_t = \frac{1}{2} M_b (\dot{x}_b^2 + \dot{z}_b^2) + \frac{1}{2} M_{body} (\dot{x}_{body}^2 + \dot{z}_{body}^2) \quad (3.38)$$

dove con \dot{x}_b e \dot{z}_b è indicata la velocità della palla in direzione orizzontale e verticale e con \dot{x}_{body} e \dot{z}_{body} è indicata la velocità del baricentro del corpo del robot in direzione orizzontale. Da notare che non si è inserito il contributo relativo a ruote e motori perché già compreso nel corpo. Osservando la Figura 3.4 possiamo trovare queste quattro velocità, infatti:

$$(x_b, z_b) = (R_b \cdot \beta, R_b) \quad (3.39)$$

$$(x_{body}, z_{body}) = (R_b \cdot \beta + h_{body} \cdot \sin(\alpha), R_b + h_{body} \cdot \cos(\alpha)) \quad (3.40)$$

che derivate diventano:

$$(\dot{x}_b, \dot{z}_b) = (R_b \cdot \dot{\beta}, 0) \quad (3.41)$$

$$(\dot{x}_{body}, \dot{z}_{body}) = (R_b \cdot \dot{\beta} + h_{body} \cdot \dot{\alpha} \cdot \cos(\alpha), -h_{body} \cdot \dot{\alpha} \cdot \sin(\alpha)) \quad (3.42)$$

Sostituendo le (3.41), (3.42) nella (3.38) otteniamo:

$$T_t = \frac{1}{2} M_b R_b^2 \dot{\beta}^2 + \frac{1}{2} M_{body} \left((R_b \dot{\beta} + h_{body} \dot{\alpha} \cos(\alpha))^2 + (h_{body} \dot{\alpha} \sin(\alpha))^2 \right) \quad (3.43)$$

Considerando ora le varie parti che subiscono un moto rotatorio (ruote, palla, motori, corpo) è possibile scrivere anche l'energia cinetica T_r dovuta alla rotazione:

$$T_r = \frac{1}{2} I_{ball} \dot{\beta}^2 + \frac{1}{2} I_{body} \dot{\alpha}^2 + \frac{1}{2} 2I_m \omega_m^2 + \frac{1}{2} 2I_w \omega_w^2 \quad (3.44)$$

dove I_m è il momento di inerzia dei motori (questo valore viene letto direttamente sul datasheet del motore scelto). Tenendo presente dell'equazione (3.26) si può riscrivere la (3.44) come:

$$T_r = \frac{1}{2} I_{ball} \dot{\beta}^2 + \frac{1}{2} I_{body} \dot{\alpha}^2 + \frac{1}{2} 2I_m \eta^2 (\dot{\beta} - \dot{\alpha})^2 + \frac{1}{2} 2I_w \eta^2 (\dot{\beta} - \dot{\alpha})^2 \quad (3.45)$$

Ulteriore termine da ricavare prima di procedere con la risoluzione delle equazioni di Lagrange è l'energia potenziale, che nel nostro caso sarà solo gravitazionale. L'energia potenziale U di un corpo puntiforme di massa m può essere calcolata nel seguente modo:

$$U = m \cdot g \cdot h \quad (3.46)$$

dove g è l'accelerazione di gravità e h è l'altezza rispetto ad un riferimento prefissato. Nel problema in esame, l'energia potenziale diventa:

$$U = M_b \cdot g \cdot z_b + M_{body} \cdot g \cdot z_{body} \quad (3.47)$$

e sostituendo le (3.39) e (3.40) otteniamo:

$$U = M_b \cdot g \cdot R_b + M_{body} \cdot g \cdot (R_b + h_{body} \cdot \cos(\alpha)) \quad (3.48)$$

Procediamo ora con lo sviluppare le equazioni di Lagrange. Il sistema studiato ha due gradi di libertà che abbiamo chiamato α e β e per questo motivo possiamo scrivere due equazioni di Lagrange:

$$\frac{d}{dt} \left(\frac{T}{\partial \dot{\alpha}} \right) - \frac{\partial T}{\partial \alpha} - \frac{\partial U}{\partial \alpha} = Q_\alpha \quad (3.49)$$

$$\frac{d}{dt} \left(\frac{T}{\partial \dot{\beta}} \right) - \frac{\partial T}{\partial \beta} - \frac{\partial U}{\partial \beta} = Q_\beta \quad (3.50)$$

I due termini Q_α e Q_β possono essere calcolati usando la definizione (3.34):

$$Q_\alpha = 2C \frac{\partial \beta}{\partial \alpha} - 2C \frac{\partial \alpha}{\partial \alpha} = -2C \quad (3.51)$$

$$Q_\beta = 2C \frac{\partial \beta}{\partial \beta} - 2C \frac{\partial \alpha}{\partial \beta} = 2C \quad (3.52)$$

dove con C è stata indicata la coppia generata dai motori; il 2 che moltiplica la C sta ad indicare i contributi di entrambi i motori.

A questo punto abbiamo ricavato tutto per poter risolvere le equazioni (3.49) e (3.50). Dopo aver calcolato le varie derivate dell'energia cinetica e potenziale, giungiamo a queste due equazioni:

$$\begin{aligned} & (M_{body} h_{body}^2 + J_{body} + 2\eta^2 (I_m + I_w)) \ddot{\alpha} + \\ & + (M_{body} h_{body} R_b \cos(\alpha) - 2\eta^2 (I_m + I_w)) \ddot{\beta} - M_{body} g h_{body} \sin(\alpha) = -2C \end{aligned} \quad (3.53)$$

$$\begin{aligned} & (M_{body} h_{body} R_b \cos(\alpha) - 2\eta^2 (I_m + I_w)) \ddot{\alpha} + \\ & + ((M_{body} + M_b) R_b + I_b + 2\eta^2 (I_m + I_w)) \ddot{\beta} - M_{body} h_{body} R_b \dot{\alpha}^2 \sin(\alpha) = 2C \end{aligned} \quad (3.54)$$

Per rendere più facile la simulazione, linearizziamo il sistema attorno a un punto di equilibrio. Scegliamo banalmente il punto di equilibrio per il quale $\alpha = 0$, ossia nel punto in cui il robot è perfettamente in equilibrio. Sostituiamo quindi $\sin(\alpha)$ con α e $\cos(\alpha)$ con 1 e $\dot{\alpha}$ con 0 e otteniamo:

$$\begin{aligned}
& (M_{body}h_{body}^2 + J_{body} + 2\eta^2 (I_m + Iw)) \ddot{\alpha} + \\
& + (M_{body}h_{body}R_b - 2\eta^2 (I_m + Iw)) \ddot{\beta} - M_{body}gh_{body}\alpha = -2C
\end{aligned} \tag{3.55}$$

$$\begin{aligned}
& (M_{body}h_{body}R_b - 2\eta^2 (I_m + Iw)) \ddot{\alpha} + \\
& + ((M_{body} + M_b) R_b + I_b + 2\eta^2 (I_m + Iw)) \ddot{\beta} = 2C
\end{aligned} \tag{3.56}$$

Queste equazioni approssimano bene il modello dinamico del robot nell'intorno del punto di equilibrio e rendono facile la simulazione del sistema. In seguito verranno utilizzate sia per simulare il sistema al PC che per trovare la coppia che i motori devono erogare per riuscire a mantenere in equilibrio il sistema.

Capitolo 4

Algoritmi

4.1 IMU: stima dell'inclinazione

La stima dell'inclinazione del robot è uno dei principali problemi da risolvere per riuscire a bilanciarlo al meglio. Infatti, un piccolo errore nell'identificazione di questo parametro potrebbe portare il robot a oscillare e, successivamente, a cadere. Tra varie possibilità, si è scelto di rappresentare l'inclinazione tramite gli angoli di Eulero RPY. Per stimare l'inclinazione si sono quindi utilizzati due tipi di sensori inerziali: accelerometri e giroscopi. Partendo dai dati provenienti da questi due sensori, la stima dei tre angoli (roll, pitch e yaw) è stata realizzata tramite un filtro di Kalman esteso.

4.1.1 Filtro di Kalman

Il filtro di Kalman [5] è un algoritmo matematico ricorsivo per la stima dello stato di un sistema dinamico, la cui formulazione è dovuta a Rudolf Emil Kalman, ingegnere e matematico statunitense.

Il filtro di Kalman è utilizzato per il filtraggio di parametri ottenuti tramite misure inaccurate e soggette a rumore. Spesso vengono effettuate più misure sullo stesso parametro in modo che ogni una abbia caratteristiche di rumore differenti a pari frequenze in modo da prendere in considerazione la frequenza corretta di ogni misura.

Per l'utilizzo del filtro è necessario conoscere:

1. il modello dinamico che descrive il sistema preso in considerazione;
2. il valore iniziale dello stato del sistema;
3. la descrizione statistica dei rumori di cui sono affetti i nuovi stati, le misure e lo stato iniziale.

Continuando con la trattazione, considereremo un sistema dinamico lineare di grado n , tempo discreto, con parametri invarianti nel tempo e soggetto a disturbi stazionari. Partiamolo quindi da un sistema nella seguente forma:

$$\begin{cases} x(t+1) = Fx(t) + Gu(t) + v_1(t) \\ y(t) = Hx(t) + v_2(t) \end{cases} \quad (4.1)$$

dove:

- $x(t) \in \mathfrak{R}^n$ rappresenta lo stato del sistema al tempo t ;
- $u(t) \in \mathfrak{R}^n$ è il vettore degli ingressi al tempo t ;
- $y(t) \in \mathfrak{R}^p$ è il vettore contenente le p misure al tempo t ;
- $F \in \mathfrak{R}^{n \times n}$ è la matrice di evoluzione del sistema;
- $G \in \mathfrak{R}^n$ è il vettore di moltiplicazione degli ingressi;
- $H \in \mathfrak{R}^p$ è il vettore di moltiplicazione delle uscite;
- $v_1(t) \in \mathfrak{R}^n$ è il rumore sullo stato al tempo t ;
- $v_2(t) \in \mathfrak{R}^p$ è il rumore sulle misure al tempo t .

Per quanto riguarda i due rumori presenti nel sistema, si presuppone che siano rumori bianchi, a valor medio nullo e incorrelati tra loro, vale a dire:

$$v_1(t) \sim WN(0, V_1) \quad (4.2)$$

$$v_2(t) \sim WN(0, V_2) \quad (4.3)$$

$$v_1 \perp v_2 \quad (4.4)$$

con $V_1 \in \mathfrak{R}^{n \times n}$ e $V_2 \in \mathfrak{R}^{p \times p}$ è indicata la varianza dei due rumori $v_1(t)$ e $v_2(t)$.

In Figura 4.1 è mostrato uno schema sintetico di descrizione dei passi per l'esecuzione del filtro di Kalman.

Le fasi principali dell'algoritmo sono quella di Predict e quella di Udate. Nella prima fase si stima lo stato del sistema, $\hat{x}(t+1|t)^-$, al tempo $t+1$ conoscendo tutte le misure fino al tempo t , come di seguito:

$$\hat{x}(t+1|t)^- = F\hat{x}(t|t-1) + Gu(t) \quad (4.5)$$

il termine $\hat{x}(t|t-1)$ è la stima corretta dello stato al tempo t conoscendo i dati fino al tempo $t-1$, che vedremo successivamente come calcolare.

Nella fase di Update, invece, si corregge la stima $\hat{x}(t+1|t)^-$ fatta al passo precedente tenendo conto delle uscite del sistema. In particolare, si inizia calcolando il vettore $e(t)$, che prende il nome di vettore innovazione, come:

$$e(t) = y(t) - \hat{y}(t|t-1) \quad (4.6)$$

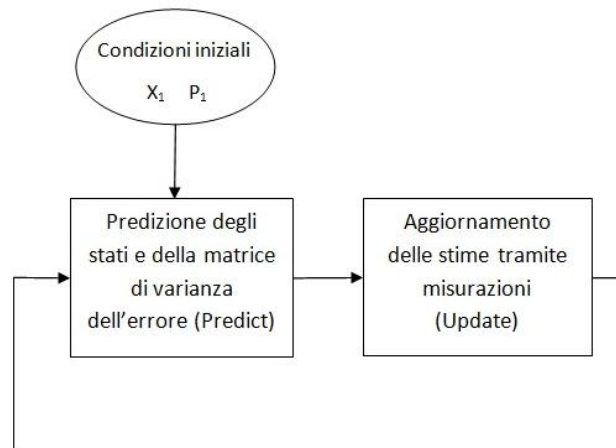


Figura 4.1: Schema algoritmo filtro di Kalman

dove $\hat{y}(t|t-1)$ è la stima della misura al tempo t ed è valutata nel seguente modo:

$$\hat{y}(t+1|t) = H\hat{x}(t+1|t)^- \quad (4.7)$$

Il vettore innovazione rappresenta il distacco tra lo stato stimato e lo stato reale del sistema, a meno di errori di misura. Grazie a questo vettore, si corregge la stima sullo stato calcolata precedentemente:

$$\hat{x}(t+1|t) = \hat{x}(t+1|t)^- + K(t)e(t) \quad (4.8)$$

In questa fase viene anche calcolata la matrice di varianza sullo stato al tempo $t+1$, $P(t+1)$, nel seguente modo:

$$P(t+1) = FP(t)F^T + V_1 - K(t) \cdot [HP(t)H^T + V_2] \cdot K(t)^T \quad (4.9)$$

la matrice $K(t) \in \mathfrak{R}^{p \times p}$ prende il nome di guadagno di Kalman e viene così calcolato:

$$K(t) = P(t)H^T \cdot [HP(t)H^T + V_2]^{-1} \quad (4.10)$$

Per quanto riguarda invece l'inizializzazione dei parametri, si assumono le seguenti ipotesi:

1. $x(1|0) = E[x(1)]$
2. $P(1) = Var(x(1))$

Il riassunto di quanto detto è possibile vederlo in Figura 4.2. Nella parte alta della figura è posto il sistema dinamico con i due rumori v_1 e v_2 ; invece, nella parte inferiore è posizionato il sistema usato come predittore di stato. In ingresso al sistema predittore è sommato il prodotto dell'innovazione per il guadagno di Kalman.

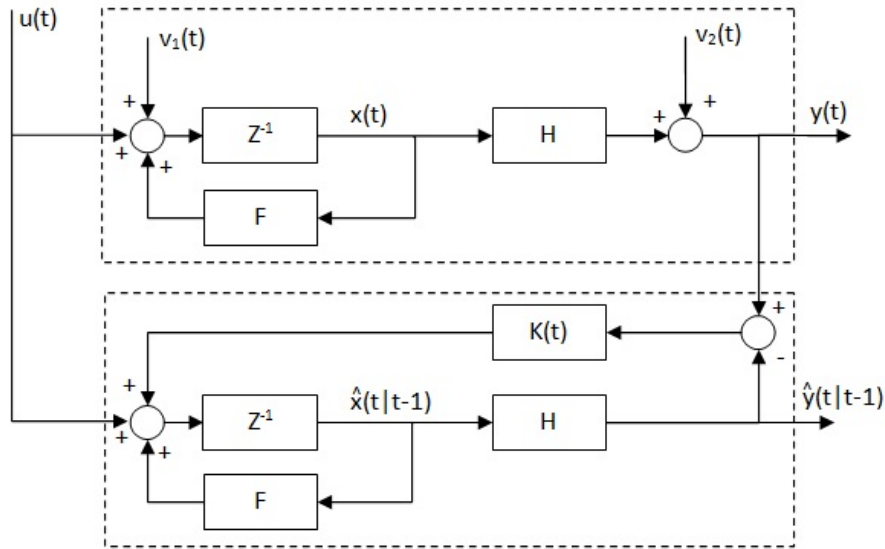


Figura 4.2: Rappresentazione filtro di Kalman

4.1.2 EKF

Il filtro di Kalman esteso (EKF) è una variante del filtro di Kalman usata quando il sistema di cui dover stimare lo stato non è lineare. Avremo quindi a che fare con un sistema nella seguente forma:

$$\begin{cases} x(t+1) = f(x(t), u(t), t) + v_1(t) \\ y(t) = h(x(t), t) + v_2(t) \end{cases} \quad (4.11)$$

in cui $f(\cdot)$ è la funzione di evoluzione dello stato del sistema e $h(\cdot)$ è la funzione per il calcolo del valore delle misure; tutti gli altri termini rimangono definiti similmente al caso del filtro di Kalman lineare.

Le equazioni che stanno alla base dell'EKF sono le stesse del filtro di Kalman lineare. Le matrici F e H vengono ottenute linearizzando le funzioni $f(\cdot)$ e $h(\cdot)$ intorno all'ultima stima dello stato:

$$\hat{F}(t|t-1) = \left. \frac{\partial f(x, u, t)}{\partial x} \right|_{x=\hat{x}(t|t-1), u=u(t)} \quad (4.12)$$

$$\hat{G}(t|t-1) = \left. \frac{\partial f(x, u, t)}{\partial u} \right|_{x=\hat{x}(t|t-1), u=u(t)} \quad (4.13)$$

$$\hat{H}(t|t-1) = \left. \frac{\partial h(x, t)}{\partial x} \right|_{x=\hat{x}(t|t-1)} \quad (4.14)$$

L'EKF è quindi composto dai seguenti passi:

1. stima dei parametri iniziali;
2. calcolo dello stato stimato $\hat{x}(t+1|t)$ tramite il sistema non lineare;
3. stima delle matrici $\hat{F}(t|t-1)$, $\hat{G}(t|t-1)$ e $\hat{H}(t|t-1)$ tramite le formule (4.12), (4.13) e (4.14);
4. calcolo del guadagno di Kalman $K(t)$ e della varianza sullo stato $P(t)$;
5. correzione dello stato tramite le misure;
6. si riprende dal punto 2.

Il sistema risultante risulta quindi essere il seguente:

$$\begin{cases} \hat{x}(t+1|t) = \hat{F}(t|t-1)\hat{x}(t|t-1) + \hat{G}(t|t-1)u(t) + K(t)e(t) \\ \hat{y}(t+1|t) = \hat{H}(t|t-1)\hat{x}(t+1|t) \end{cases} \quad (4.15)$$

4.1.3 Accelerometri

L'accelerometro è uno strumento di misura che permette di misurare l'accelerazione di un corpo. Nella maggiorparte degli accelerometri il principio di funzionamento si basa sulla rilevazione dello spostamento di una massa sospesa a un elemento elastico. In presenza di un'accelerazione la massa, dotata di una propria inerzia, si sposta dalla propria posizione di riposo in modo proporzionale all'accelerazione rilevata. Lo spostamento della massa, rilevato tramite un apposito sensore, viene convertito in un'accelerazione per poi essere convertito in un segnale elettrico. Esistono in commercio vari tipi di accelerometri:

- accelerometro estensimetrico;
- accelerometro piezoresistivo;
- accelerometro LVDT;
- accelerometro capacitivo;
- accelerometro piezoelettrico;
- accelerometro laser;
- gravitometro.

L'accelerometro usato nel progetto è di tipo capacitivo e è realizzato con tecnologia MEMS. Per rilevare lo spostamento della massa mobile, questo accelerometro misura la variazione di capacità elettrica di un condensatore al variare della distanza tra le sue armature. In pratica, la massa (realizzata con materiale conduttivo) costituisce un'armatura, mentre l'altra è realizzata sulla struttura fissa del dispositivo e posta nell'immediata prossimità della massa (Figura 4.3). La massa è sospesa su un elemento elastico abbastanza rigido in modo da potersi muovere senza allontanarsi eccessivamente dalla sua posizione di riposo. Un apposito circuito rileva la capacità del condensatore formato dalle due armature istante per istante, trovando quindi una misura proporzionale all'accelerazione subita dal dispositivo.

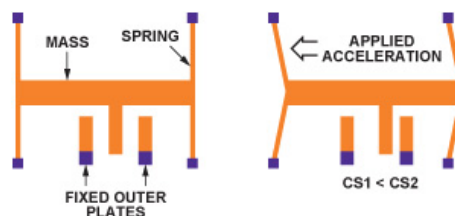


Figura 4.3: Principio di funzionamento di un accelerometro capacitivo

4.1.4 Giroscopi

Il giroscopio classico è un dispositivo fisico rotante che, per effetto della legge di conservazione del momento angolare, tende a mantenere il suo asse di rotazione orientato in una direzione fissa. Fruttando questo fatto è possibile misurare le velocità angolari. Nel realizzare i giroscopi con la tecnologia MEMS, si utilizza però un principio di funzionamento differente. La maggior parte dei giroscopi MEMS usano elementi meccanici vibranti per sentire le rotazioni. Questi sensori si basano sul trasferimento di energia tra due vibrazioni perpendicolari di una struttura provocato dall'accelerazione Coriolis.

I giroscopi a vibrazione possono essere modellizzati tramite un sistema massa-molla come mostrato in Figura 4.4. Il sistema è composto di una particella A dotata di una propria massa e quattro molle che collegano la particella A a un sistema fisso. La massa A ha due gradi di libertà e, in ciascun istante di tempo, la sua posizione è definita da uno spostamento x' lungo l'asse x e da uno spostamento y' lungo l'asse y .

Il sistema così definito permette di misurare la velocità angolare di rotazione, Ω , del piano xy attorno all'asse z . Per misurarla è necessario introdurre una vibrazione ad ampiezza costante della massa A lungo l'asse x . Questa oscillazione è chiamata moto primario del giroscopio e è tenuta perfettamente costante tramite un sistema a retroazione. Quando il giroscopio è posto in rotazione, sulla particella A risulta applicata la forza di Coriolis F_c che ha un valore proporzionale alla velocità di rotazione applicata e direzione perpendicolare al moto primario:

$$F_c = 2mv \times \Omega \quad (4.16)$$

dove m è il valore della massa della particella A e v è la velocità del moto primario. La forza di Coriolis F_c introduce quindi una vibrazione di A lungo l'asse y , che prende il nome di moto secondario del giroscopio. Osservando l'equazione 4.16 si nota che l'ampiezza di questo moto risulta essere proporzionale alla velocità angolare Ω . La velocità angolare può essere misurata attraverso un controllo ad anello chiuso sul moto secondario. La misura è utilizzata per generare una forza di controllo in grado di annullare il moto

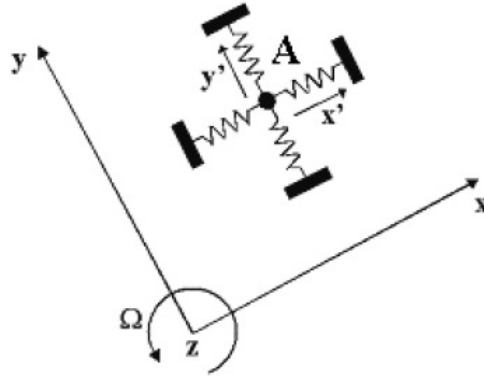


Figura 4.4: Modello giroscopio MEMS

lungo l'asse y . L'ampiezza della forza di controllo rappresenta la misura della velocità di rotazione.

4.1.5 Stima di Roll, Pitch e Yaw tramite EKF

In questa sezione vediamo come realizzare teoricamente un algoritmo per ricavare l'inclinazione di un oggetto. Più precisamente, date due terne destrorse con origine comune, troveremo il modo di stimare dei parametri che permettano di descrivere la posizione di una (terna mobile) rispetto all'altra (terna fissa).

Per l'identificazione della stima si è scelto di utilizzare gli angoli di Eulero RPY (Capitolo 3.1). I dati per la stima di questi tre angoli sono ricavati da tre accelerometri e tre giroscopi posti in modo tale da misurare le accelerazioni lungo gli assi della terna mobile e le velocità angolari attorno agli assi della stessa terna.

In prima analisi, potrebbe sembrare che sia con le misure provenienti dagli accelerometri che con quelle provenienti dai giroscopi sia possibile ricavare l'orientamento della terna mobile, e che quindi bastino solamente le misure provenienti da uno dei due sensori. Purtroppo non è così.

Supponendo di utilizzare soltanto i tre accelerometri e che il sistema sia sog-

getto alla sola accelerazione di gravità, è possibile trovare i tre angoli di Eulero (ϕ , θ e ψ) nel seguente modo:

$$\phi = \text{atan2}(A_y, A_z) \quad (4.17)$$

$$\theta = \text{atan2}(A_x, A_z) \quad (4.18)$$

$$\psi = \text{atan2}(A_y, A_x) \quad (4.19)$$

in cui A_x , A_y e A_z sono le componenti dell'accelerazione di gravità misurate dai tre accelerometri. Tali equazioni (4.17), (4.18) e (4.19) non sono utilizzabili da sole in quanto spesso il sistema, oltre all'accelerazione di gravità, è soggetto ad ulteriori accelerazioni non conosciute a priori.

Considerando questa volta solamente i tre giroscopi, risulta intuitivo ricavare gli angoli ϕ , θ e ψ integrando le velocità angolari misurate, nel seguente modo:

$$\phi(t) = \phi(t-1) + p \cdot T \quad (4.20)$$

$$\theta(t) = \theta(t-1) + q \cdot T \quad (4.21)$$

$$\psi(t) = \psi(t-1) + r \cdot T \quad (4.22)$$

dove p , q e r sono le velocità angolari misurate dai tre giroscopi e T è il periodo di campionamento dei giroscopi. Come si vede dalle formule (4.20), (4.21) e (4.22), gli angoli calcolati al tempo t dipendono dalle misure effettuate al tempo t , $t-1$, $t-2$, ... fino all'inizio delle misurazioni. Un errore di misura commesso nel passato rimane quindi presente nella stima degli angoli. Per questo motivo, nel tempo, gli angoli calcolati tramite le formule (4.20), (4.21) e (4.22) risultano subire un errore sempre maggiore, portando a valori divergenti degli angoli. In Figura 4.5 è mostrato l'andamento dell'integrazione dei valori restituiti da un giroscopio tenuto fermo. Dalla figura è possibile notare come già dopo 800 campioni si arrivi ad un errore di integrazione di circa 12° .

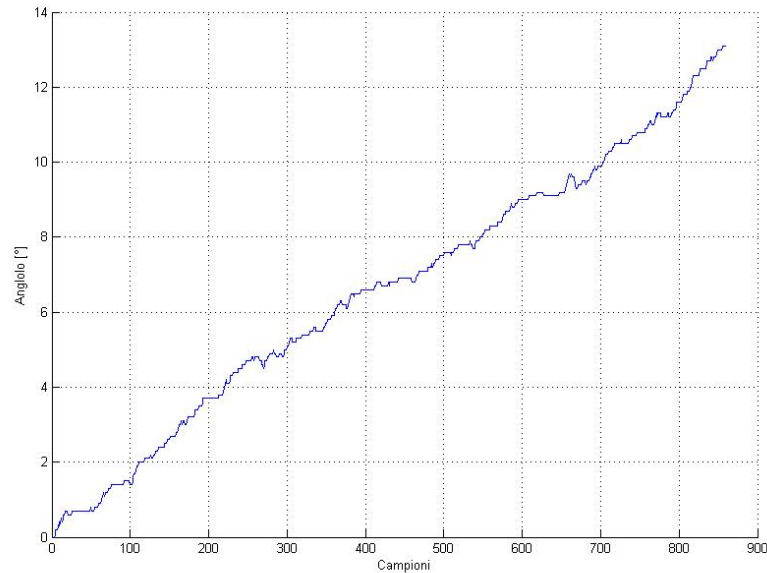


Figura 4.5: Deriva dell'angolo ricavato dall'integrazione di un giroscopio tenuto fermo

Per risolvere i problemi descritti sopra, sono stati utilizzate entrambe le tipologie di sensori fondendo assieme ambedue le misure effettuate tramite un filtro di Kalman esteso realizzato su microcontrollore. In questo modo è possibile fare affidamento alla misura pervenuta dai giroscopi per il breve periodo e utilizzare, invece, il valore ricavato dagli accelerometri per correggere lentamente le misure ottenute precedentemente.

L'EKF realizzato è composto dalle seguenti fasi:

fase di predict: vengono stimati gli angoli ϕ , θ e ψ integrando il valore misurato dai giroscopi;

fase di update: viene corretta la stima precedente utilizzando le accelerazioni lette dagli accelerometri.

Il modello da stimare nel filtro è stato ottenuto usando come vettore degli stati, $x(t)$, gli angoli ϕ , θ e ψ

$$x(t) = \begin{bmatrix} \phi(t) \\ \theta(t) \\ \psi(t) \end{bmatrix} \quad (4.23)$$

e utilizzando come ingressi i dati provenienti dai tre giroscopi

$$u(t) = \begin{bmatrix} p(t) \\ q(t) \\ r(t) \end{bmatrix} \quad (4.24)$$

in cui $p(t)$, $q(t)$ e $r(t)$ sono le misure dei giroscopi al tempo t attorno, rispettivamente, agli assi x , y e z della terna mobile.

I tre stati possono essere calcolati integrando le misure ottenute dai giroscopi. Come stato di partenza è stato preso il vettore

$$x(1) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.25)$$

in quanto si è posto che il robot parta inizialmente in equilibrio sulla palla. Per calcolare il valore degli stati non è sufficiente integrare il valore letto dai giroscopi in quanto, questi, misurano le velocità angolari attorno agli assi della terna mobile. Per questo motivo, è necessario per prima cosa convertire le velocità lette in modo che siano riferite alla terna fissa. Per far ciò utilizziamo la trasformazione lineare $T(\phi, \theta, \psi)$ ottenuta nel Capitolo 3.2:

$$T(\phi, \theta, \psi) = \begin{bmatrix} 1 & s_\psi t_\theta & c_\psi t_\theta \\ 0 & c_\psi & -s_\psi \\ 0 & s_\psi/c_\theta & c_\psi/c_\theta \end{bmatrix} \quad (4.26)$$

A questo punto gli stati del sistema possono essere calcolati nel seguente modo:

$$\begin{bmatrix} \phi(t+1) \\ \theta(t+1) \\ \psi(t+1) \end{bmatrix} = \begin{bmatrix} \phi(t) \\ \theta(t) \\ \psi(t) \end{bmatrix} + T \cdot \begin{bmatrix} 1 & s_\psi t_\theta & c_\psi t_\theta \\ 0 & c_\psi & -s_\psi \\ 0 & s_\psi/c_\theta & c_\psi/c_\theta \end{bmatrix} \begin{bmatrix} p(t) \\ q(t) \\ r(t) \end{bmatrix} \quad (4.27)$$

in cui T è il periodo di campionamento dei giroscopi. La (4.27) scritta in forma estesa diventa:

$$\begin{bmatrix} \phi(t+1) \\ \theta(t+1) \\ \psi(t+1) \end{bmatrix} = \begin{bmatrix} \phi(t) + T \cdot [p(t) + s_\psi t_\theta q(t) + c_\psi t_\theta r(t)] \\ \theta(t) + T \cdot [c_\psi q(t) - s_\psi r(t)] \\ \psi(t) + T \cdot [\frac{s_\psi}{c_\theta} q(t) + \frac{c_\psi}{c_\theta} r(t)] \end{bmatrix} = f(x(t), u(t)) \quad (4.28)$$

Com'è facile notare, le tre funzioni per il calcolo degli stati sono fortemente non lineari. Per questo motivo è necessario utilizzare le equazioni (4.12) e (4.13) per eseguire il filtro di Kalman esteso.

All'arrivo di ogni nuovo ingresso per il sistema, viene calcolata la nuova stima dello stato $x(t+1|t)^-$. A questo punto è necessario correggere la stima con l'aiuto delle misure ottenute dagli accelerometri. A tale scopo si trovano i valori dello stato $x(t+1)$, dai dati provenienti dagli accelerometri, tramite le formule (4.17), (4.18) e (4.19).

La formula di correzione dello stato stimato prende la forma:

$$\hat{x}(t+1|t) = \hat{x}(t+1|t)^- + K(t)(x(t) - \hat{x}(t|t-1)) \quad (4.29)$$

Si noti che per semplificare leggermente i conti è stata presa

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.30)$$

in quanto le uscite del sistema potevano essere scelte arbitrariamente.

L'ultima parte mancante, da descrivere prima di utilizzare il filtro, è la stima dei valori di $P(1)$, V_1 e V_2 . V_1 e V_2 sono stati calcolati tramite varianza campionaria di un numero elevato di misurazioni note, effettuate con il giroscopio, per V_1 , e con l'accelerometro, per V_2 . $P(1)$, che indica la varianza sullo stato iniziale, posto pari a V_2 in quanto lo stato iniziale è calcolato utilizzando gli accelerometri.

4.2 Controllo della stabilità

In questa sezione è descritto il processo che ha portato a controllare i movimenti del robot con lo scopo di renderlo stabile sulla palla.

Questo processo si è suddiviso in due parti:

- simulazione del modello dinamico del robot ricavato nel Capitolo 3.3 e la realizzazione di un controllo per il modello simulato;
- realizzazione del controllore su di un modello realizzato realmente.

Il controllore finale è stato realizzato tramite due PID in cascata e permette di mantenere in equilibrio il robot e di fargli seguire un profilo di velocità desiderato.

4.2.1 Controllore PID

Il controllore Proporzionale - Integrale - Derivativo, comunemente abbreviato con PID, è un controllore a retroazione negativa impiegato nel controllo automatico dei sistemi dinamici. Per la semplicità di taratura e per il gran numero di processi che è possibile controllare in maniera soddisfacente, il PID è di gran lunga il controllore più utilizzato in ambito industriale. Il lavoro che esegue il controllore è quello di calcolare il valore dell'errore (e), tra la misura di una variabile di processo (*output*) e un riferimento desiderato (*setpoint*), cercando di minimizzarlo aggiustando le variabili in ingresso al processo.

Il controllore PID (mostrato in Figura 4.6) utilizza tre azioni separate e poste in parallelo, chiamate: proporzionale, integrale e derivativa, ognuna delle quali utilizza un proprio parametro di controllo (o guadagno), che assumono i nomi K_p , K_i e K_d . Queste tre azioni possono essere interpretate in funzione del tempo: il valore risultante dalla parte proporzionale dipende dall'errore istantaneo, il valore calcolato dalla parte integrale dipende dall'accumulo degli errori passati, il valore calcolato dalla parte derivativa è basato sugli errori futuri. La somma di questi tre contributi costituisce la variabile di controllo u ed è usata per controllare il processo.

La variabile di controllo è, quindi, definita come segue:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad (4.31)$$

in cui è stata specificata la dipendenza dal tempo delle variabili u e e . Dall'equazione 4.31 è facile ricavare anche la funzione di trasferimento del controllore:

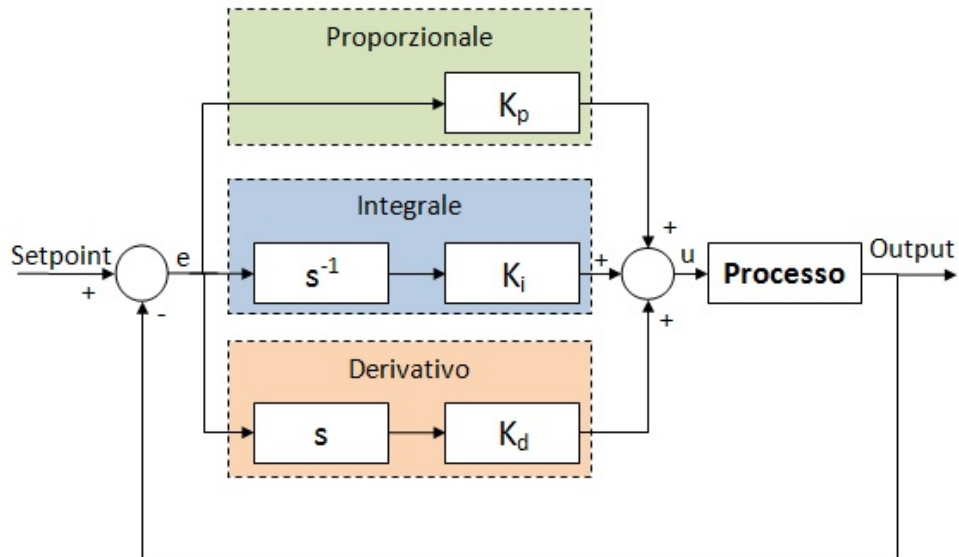


Figura 4.6: Schema generale di un controllore PID

$$G_{PID}(s) = K_p + s^{-1}K_i + sK_d \quad (4.32)$$

In alcune applicazioni possono essere necessarie solo una o due azioni per realizzare un opportuno sistema di controllo. Questo è ottenibile ponendo a zero i parametri superflui. In assenza di alcune azioni, il controllore PID prende il nome di P, PI, PD o I.

4.2.1.1 Tuning del controllore

La taratura del controllore viene realizzata agendo sui tre parametri di controllo K_p , K_i e K_d .

Esistono diversi metodi per eseguire il tuning dei tre parametri. Nei problemi teorici, in cui si conoscono perfettamente le equazioni alla base del sistema dinamico, il metodo più praticato sono i diagrammi di Bode verificando che il margine di fase sia positivo.

| Tipo Controllore | K_p | K_I | K_d |
|------------------|-----------|----------------------|---------------------|
| P | $0.50K_u$ | - | - |
| PI | $0.45K_u$ | $1.2\frac{K_p}{T_u}$ | - |
| PID | $0.60K_u$ | $2\frac{K_p}{T_u}$ | $\frac{K_p T_u}{8}$ |

Tabella 4.1: Relazioni di Ziegler e Nichol

Nei metodi reali, in cui non è possibile conoscere con esattezza le equazioni che regolano il modello, si usano metodi online; ossia che prevedono la modifica dei parametri mentre il sistema è attivo.

Il più famoso di questi metodi è quello di Ziegler e Nichols. Questo prevede di effettuare una serie di prove a ciclo chiuso, vale a dire con la presenza del controllore. Inizialmente si utilizza un controllore puramente proporzionale, con guadagno K_p relativamente basso, e si aumenta il guadagno fino a che non si innesca un regime di oscillazioni permanenti per il sistema a ciclo chiuso. Una volta raggiunto il regime di oscillazione, si registrano i valori del guadagno proporzionale critico K_u e del periodo di oscillazione T_u . In base alla stima di questi due parametri, è possibile progettare i guadagni del regolatore calcolando K_i e K_d secondo le relazioni presenti in Tabella 4.1. Si noti che le relazioni risultano differenti in base al tipo di controllore da tarare (P, PI, o PID).

In Figura 4.7 vengono mostrati i diversi effetti che i parametri K_p , K_i e K_d hanno sull'inseguimento del valore di riferimento. In Figura 4.7(a) è mostrato l'inseguimento di uno scalino per valori diversi di K_p . Il termine proporzionale modifica l'*output* in modo proporzionale a e . Un grande guadagno proporzionale comporta un'elevata modifica del valore di *output* anche per piccole variazioni di e . Valori troppo elevati di K_p possono portare il sistema in anello chiuso all'instabilità. Al contrario, un guadagno proporzionale troppo piccolo può portare a una risposta molto lenta anche a fronte di valori elevati dell'errore e . In questo caso possono essere necessari tempi molto lunghi per portarsi a regime e per contrastare i disturbi.

Il contributo al controllo dato dal termine integrale è proporzionale sia all'in-

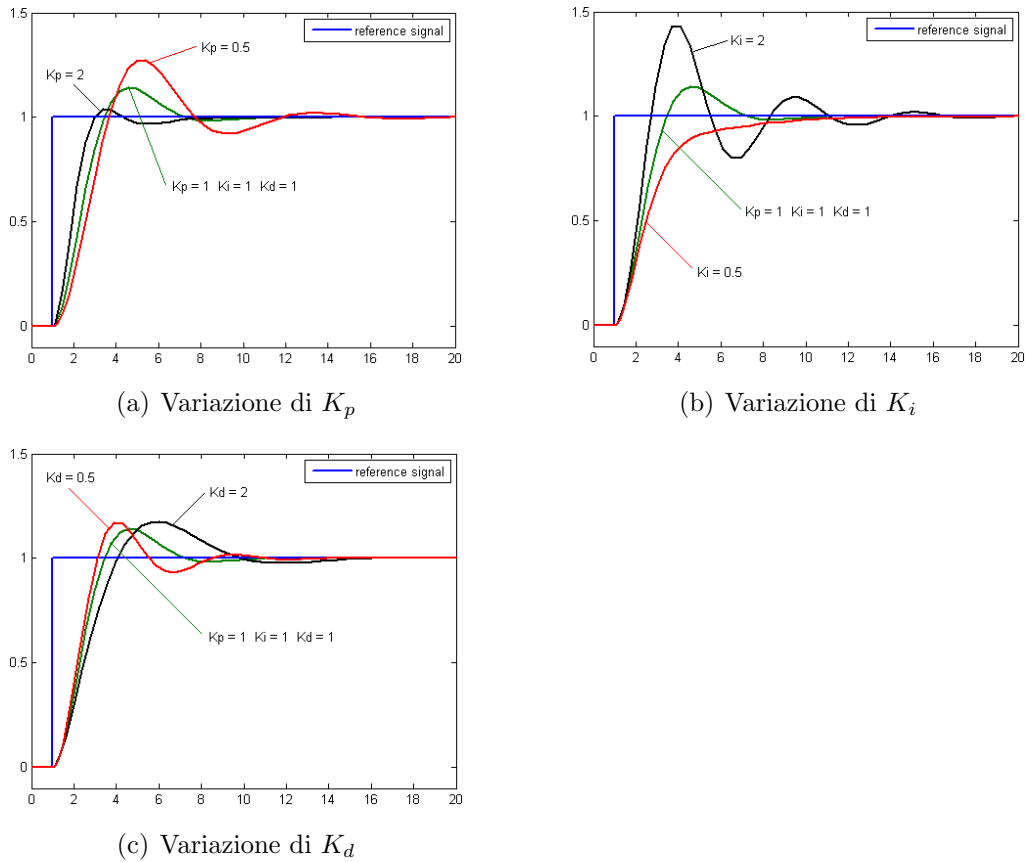


Figura 4.7: Variazione di risposta di un sistema con PID in relazione ai tre parametri del PID

tensità, che alla durata dell'errore. Questo termine accelera il movimento del sistema verso il *setpoint* ed elimina gli errori costanti (o offset) che rimangono con un controllore proporzionale. Tuttavia, come è possibile vedere in Figura 4.7(b), una K_i troppo elevata può generare notevoli oscillazioni nel sistema.

Nella Figura 4.7(c) è mostrato come il termine derivativo influenza l'inseguimento di un gradino come valore di *setpoint*. L'uscita prodotta da questo termine è proporzionale alla variazione dell'errore nell'istante in considerazione. Il contributo di K_d modifica la prontezza del sistema, rendendo più veloce l'inseguimento del riferimento. Questo fatto si verifica perché, appena l'errore sta aumentando, l'azione derivativa cerca di compensare il gap sull'u-

scita in modo proporzionale alla variazione di e , senza attendere che l'errore aumenti molto di ampiezza o persista per molto tempo. Valori troppo elevati di K_d potrebbero rendere troppo veloce il sistema, portandolo a inseguire anche i più piccoli disturbi.

4.2.1.2 Algoritmo PID

Il controllore PID, così per come è definito, si presta molto bene ad essere realizzato su PC, FPGA o microcontrollori, anche con poca potenza di calcolo. Implementando il controllore in un sistema digitale, è però necessario passare dal tempo continuo a quello discreto. Per questo motivo, il calcolo dell'errore integrale e derivativo vengono ridefiniti come segue:

$$\int_0^{t_k} e(\tau) d\tau \rightarrow \sum_{i=1}^k e(t_i) \Delta t \quad (4.33)$$

$$\frac{d}{dt} e(t_k) \rightarrow \frac{e(t_k) - e(t_{k-1})}{\Delta t} \quad (4.34)$$

dove ΔT indica il periodo di tempo che intercorre tra due esecuzioni successive dell'algoritmo. Note queste due correzioni, un algoritmo che svolge la funzione di un PID digitale è il seguente:

```
% Initializations
setpoint = read_from_user()
actual_position = read_from_sensors()
previous_error = setpoint - actual_position
integral = 0
% Main loop
start:
    setpoint = read_from_user()
    actual_position = read_from_sensors()
    error = setpoint - actual_position
    integral = integral + (error*dt)
    derivative = (error - previous_error)/dt
    output = (Kp*error) + (Ki*integral) + (Kd*derivative)
    previous_error = error
```

```
wait(dt)
goto start
```

4.2.2 Simulazione del modello e del controllo

Prima di realizzare il controllore su un sistema reale, si è proceduto alla realizzazione di un controllore su un sistema simulato al PC in modo da verificarne il corretto funzionamento. In questa fase è stato utilizzato il modello semplificato del robot esposto nel Capitolo 3.3.

Per ricavare l'espressione del sistema da simulare partiamo dalle equazioni dinamiche linearizzate ricavate nel Capitolo 3.3.2

$$\begin{aligned} & (M_{body}h_{body}^2 + J_{body} + 2\eta^2 (I_m + Iw)) \ddot{\alpha} + \\ & + (M_{body}h_{body}R_b - 2\eta^2 (I_m + Iw)) \ddot{\beta} - M_{body}gh_{body}\alpha = -2C \end{aligned} \quad (4.35)$$

$$\begin{aligned} & (M_{body}h_{body}R_b - 2\eta^2 (I_m + Iw)) \ddot{\alpha} + \\ & + ((M_{body} + M_b) R_b + I_b + 2\eta^2 (I_m + Iw)) \ddot{\beta} = 2C \end{aligned} \quad (4.36)$$

e le rappresentiamo come un sistema in forma normale, del tipo $\dot{x} = Ax + Bu$, da poter inserire nel simulatore. Per far ciò sono stati introdotti il vettore degli stati, x , e la variabile di ingresso, u , come

$$x = \begin{bmatrix} \beta \\ \alpha \\ \dot{\beta} \\ \dot{\alpha} \end{bmatrix} \quad (4.37)$$

$$u = C \quad (4.38)$$

Manipolando poi le (4.35), (4.36) arriviamo a scrivere la matrice A e il vettore B nella seguente forma:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & A_{3,2} & 0 & 0 \\ 0 & A_{4,2} & 0 & 0 \end{bmatrix} \quad (4.39)$$

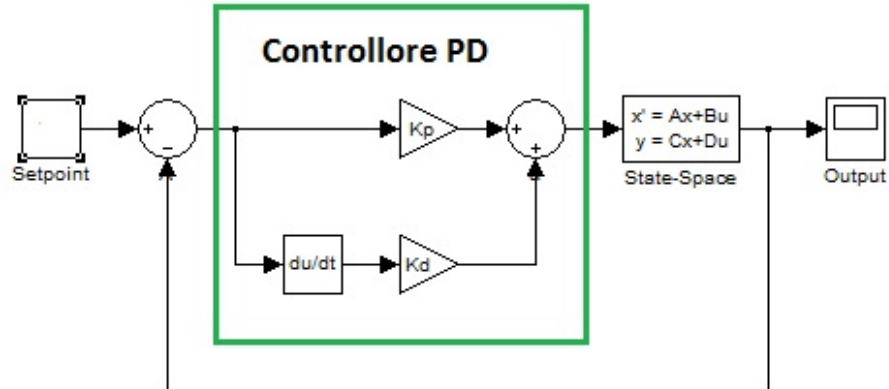


Figura 4.8: Schema del modello per la simulazione con Simulink

$$B = \begin{bmatrix} 0 \\ 0 \\ B_3 \\ B_4 \end{bmatrix} \quad (4.40)$$

dove $A_{3,2}$, $A_{4,2}$, B_3 e B_4 sono termini numerici che sono stati calcolati direttamente al computer.

Il modello dinamico risultante è stato simulato in ambiente Simulink utilizzando lo schema mostrato in Figura 4.8.

I valori utilizzati nella simulazione sono stati calcolati tenendo conto del modello realizzato realmente e descritto nella successiva sezione.

4.2.2.1 Risultati della simulazione

In Figura 4.9 è mostrata la simulazione di tre casi interessanti, utilizzando il controllore tarato ottimalmente. La prima simulazione, mostrata in Figura 4.9(a), è stata realizzata facendo partire il robot leggermente inclinato e verificando che riuscisse a portarsi nella posizione di equilibrio ($\alpha = 0^\circ$). Come si può vedere dal grafico, il robot è partito inclinato di circa 12° e dopo un paio di secondi è riuscito a bilanciarsi perfettamente.

La seconda simulazione, mostrata in Figura 4.9(b), è stata realizzata facendo partire il robot sempre da $\alpha = 0^\circ$, ma, al contrario della precedente, impostando un setpoint pari a circa 6° . In questo caso si può notare che il setpoint viene raggiunto dopo poco più di un secondo, ma la palla continua a rotolare sempre più velocemente. Questo fatto è dovuto al fatto che, per mantenere una certa inclinazione, il robot, è costretto ad accelerare in modo da creare, sul corpo, una forza d'inerzia pari ed opposta alla componente della forza di gravità che tenderebbe a farlo cadere.

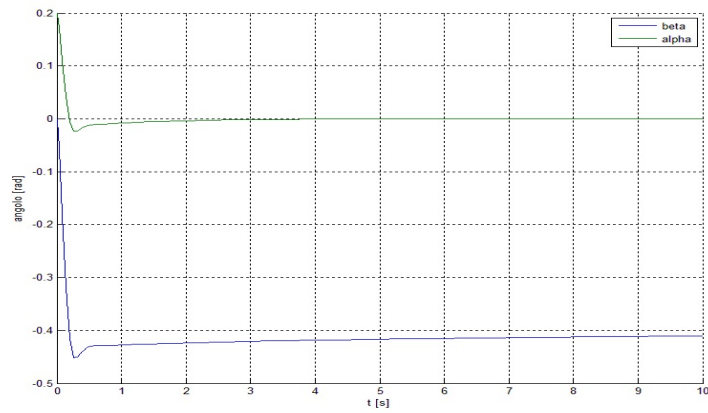
In Figura 4.9(c) è invece mostrata l'ultima simulazione effettuata. In questo caso si è partiti sempre col robot inclinato di circa 12° ma, una volta stabilizzato il sistema, si è introdotto un disturbo sull'inclinazione del corpo, che nella realtà può essere rappresentato come una spinta. Dopo cinque secondi si può notare il disturbo che porta il robot a sbilanciarsi ma, poco dopo, il sistema è riuscito a tornare nella posizione di equilibrio. Dal grafico è possibile anche notare come la posizione della palla sia variata a seguito del disturbo.

4.2.3 Controllo sperimentale

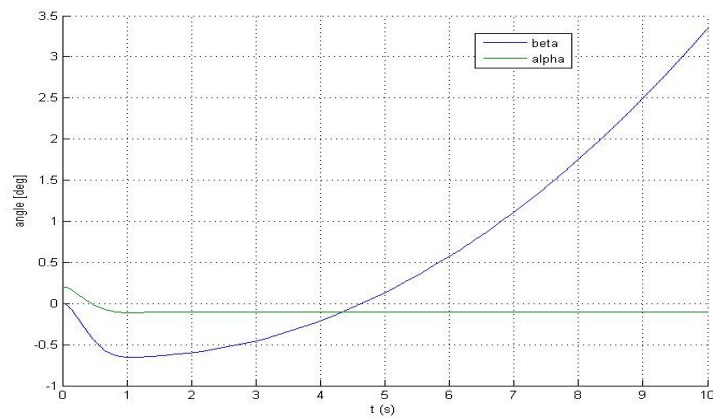
Una volta realizzato il controllo in fase di simulazione e verificato il corretto funzionamento, si è proceduto a realizzarlo sul microcontrollore che verrà posizionato sul sistema reale.

Nella realizzazione sperimentale del controllo ci si è trovati a dover gestire una problematica che nel sistema simulato non era presente: l'offset tra l'angolo di inclinazione del corpo misurato e quello reale per essere perfettamente in equilibrio. Questo offset non è ricavabile per via sperimentale in quanto, anche un piccolo errore nella stima, porterebbe il robot ad accelerare continuamente per inseguire un angolo che, seppur piccolo, non è nullo. Contrariamente, il problema è stato risolto introducendo un ulteriore loop di controllo (PID) esterno a quello di stabilità, per controllare l'angolo d'inclinazione.

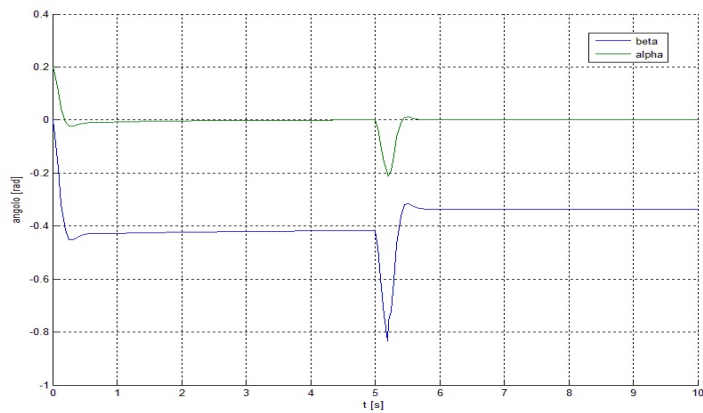
In Figura 4.10 è possibile osservare l'organizzazione dei due controllori PID, sempre per il modello del robot monoplanare. Il PID più interno si occupa della stabilità del robot, ossia cerca di inseguire il riferimento di inclinazione desiderato α° generando un comando in accelerazione per i motori, a .



(a) Simulazione in equilibrio



(b) Simulazione con riferimento non nullo



(c) Simulazione in equilibrio e con disturbo

Figura 4.9: Simulazione del Ballbot in tre casi interessanti

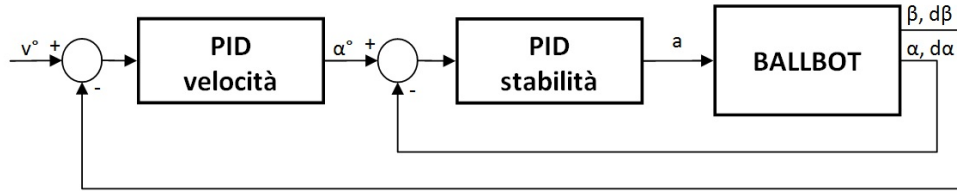


Figura 4.10: Schema dei due controllori PID per il controllo del robot

Il secondo PID ha, invece, il compito di scegliere il riferimento di angolazione, α° , per seguire il profilo di velocità v° . In questo modo, quando v° è pari a zero, il controllore più esterno genera un riferimento α° tale da mantenere il robot perfettamente in equilibrio. Quando, al contrario, v° assume un valore non nullo, il PID di velocità aumenta α° in modo da far aumentare la velocità al robot e, una volta giunti in prossimità della velocità voluta, riporta α° a zero.

Con questa configurazione del controllo, una variazione del baricentro del robot, dovuta ad esempio all'aggiunta di un carico da trasportare, non crea molti problemi alla stabilità del sistema. Uno spostamento del baricentro genera una maggior inclinazione del robot tale da far intervenire il controllore della stabilità facendo accelerare la rotazione della palla. La velocità della palla che si genera fa intervenire il PID di velocità che, piano piano, reimposta α° ad un nuovo valore tale da far tornare il baricentro sull'asse verticale del robot.

Capitolo 5

Prototipazione

La parte di prototipazione ha rivestito un ruolo di grande importanza nel progetto. Grazie a questa fase è stato possibile verificare che i modelli e gli algoritmi sviluppati teoricamente funzionassero realmente. Questa parte della tesi si è suddivisa in quattro sottosezioni:

progetto meccanico: progetto del telaio del robot in ambiente Solidworks, scelta delle ruote e dei motori;

progetto elettronico: scelta dell'elettronica su cui realizzare gli algoritmi studiati teoricamente, controllare i motori e alimentare tutti i sistemi correttamente;

sviluppo del software: realizzazione di un'interfaccia per permettere ad un utente di controllare il robot;

verifica dei risultati: prove sperimentali sul robot per verificarne il corretto funzionamento in ogni situazione.

5.1 Descrizione meccanica

In questa sezione sono descritte, in modo dettagliato, le varie parti che compongono la meccanica del robot e sono motivate le scelte progettuali che hanno portato al prototipo finale. In particolare, saranno descritti: il telaio, i motori e le ruote.

5.1.1 Telaio

Il telaio del robot, mostrato in Figura 5.1, è stato progettato in ambiente CAD 3D tramite la suite di programmi per il disegno meccanico Solidworks. La maggior parte dei pezzi meccanici sono stati realizzati in ABS tramite stampante 3D.

Il telaio è composto da tre piani paralleli, realizzati per fresatura e separati tra loro con dei distanziali esagonali in alluminio. Il piano inferiore, costruito a partire da un foglio di policarbonato nero dello spessore di 5mm, permette l'alloggiamento della batteria e il fissaggio dei supporti dei motori. I fori per il fissaggio di quest'ultimi sono stati realizzati in modo da poterne variare la posizione fino ad un massimo di 1.5cm; questo permette di poter poggiare il robot su sfere di dimensioni diverse (diametro da 18.1cm a 23.5cm). Il piano centrale, realizzato fresando un foglio di plexiglas di 5mm di spessore, è utilizzato per l'alloggiamento delle schede elettroniche. L'ultimo piano, costruito sempre da un foglio di plexiglas, ha l'unica funzione di proteggere l'elettronica da eventuali urti o cadute ed, eventualmente, favorire l'appoggio di oggetti per il trasporto.

In Figura 5.2 è mostrato il supporto dei motori. Questo pezzo è stato progettato in modo che le ruote siano inclinate rispetto alla verticale di un angolo di 40°.

5.1.2 Motori

Il robot è attuato da quattro motori stepper. I motori sono stati scelti sulla base di queste due caratteristiche:

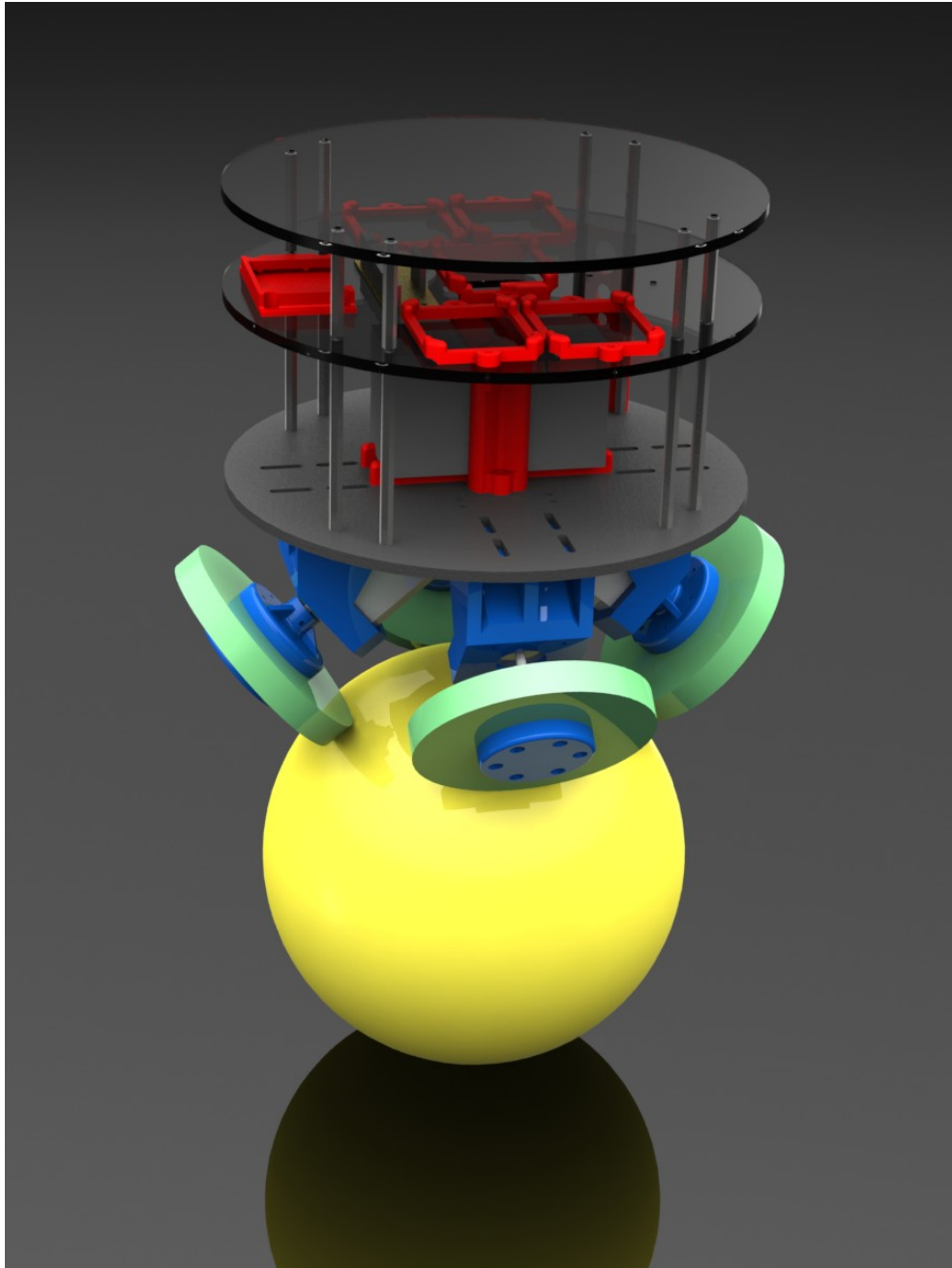


Figura 5.1: Disegno 3D del telaio del robot

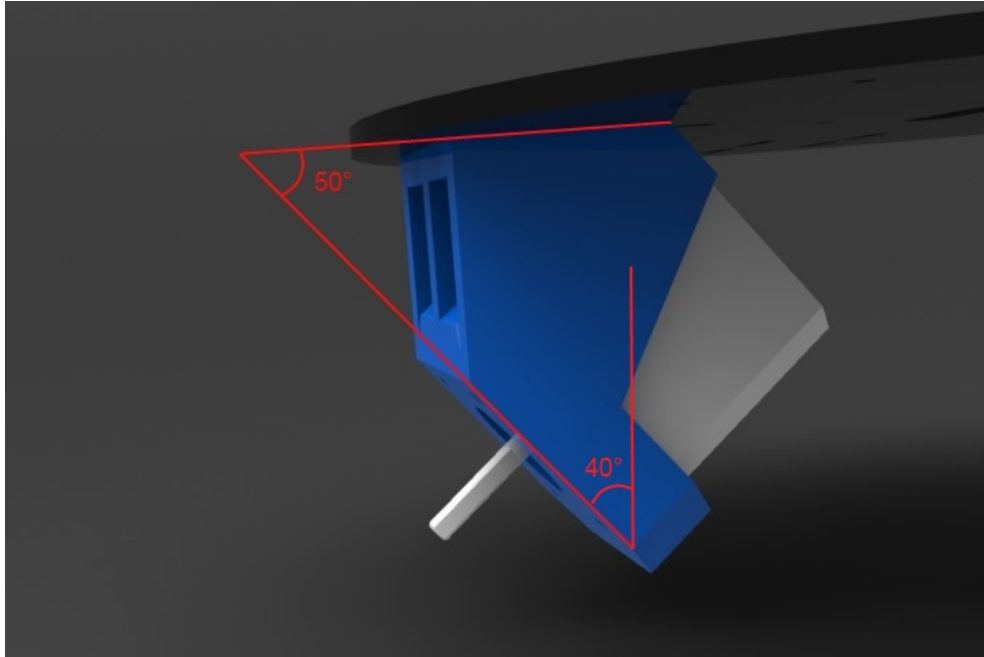


Figura 5.2: Particolare del supporto dei motori

- possibilità di bilanciare il robot anche quando questo parta inclinato di 15° rispetto alla verticale;
- possibilità di raggiungere la velocità di 0.5m/s .

La prima caratteristica impone un limite minimo sulla coppia erogata dal motore. Per il calcolo della coppia minima, C_{min} utilizziamo le due equazioni della dinamica trovate nel Capitolo 3.3:

$$\begin{aligned} & (M_{body}h_{body}^2 + J_{body} + 2\eta^2 (I_m + I_w)) \ddot{\alpha} + \\ & + (M_{body}h_{body}R_b \cos(\alpha) - 2\eta^2 (I_m + I_w)) \ddot{\beta} - M_{body}g h_{body} \sin(\alpha) = -2C \end{aligned} \quad (5.1)$$

$$\begin{aligned} & (M_{body}h_{body}R_b - 2\eta^2 (I_m + I_w)) \ddot{\alpha} + \\ & + ((M_{body} + M_b) R_b + I_b + 2\eta^2 (I_m + I_w)) \ddot{\beta} = 2C \end{aligned} \quad (5.2)$$

Risolvendo il sistema con $\alpha = 15^\circ$ e $\dot{\beta} = 0^\circ$, si ottiene che:

$$C_{min} \geq 0.2825Nm \quad (5.3)$$

La seconda caratteristica rappresenta invece un vincolo sulla velocità di rotazione minima, ω_{min} , che può compiere il motore. Per ricavare ω_{min} partiamo dalla formula ricavata nel Capitolo 3.3.1:

$$v_b = R_b \cdot \dot{\beta} \quad (5.4)$$

e sostituiamo all'interno la formula

$$\omega_m = \eta(\dot{\beta} - \dot{\alpha}) \quad (5.5)$$

sempre ricavata nello stesso capitolo. Ponendo $\dot{\alpha} = 0$ e invertendo la formula otteniamo:

$$\omega_{min} \geq \frac{v_b \eta}{R_b} \quad (5.6)$$

Sostituendo nella (5.6) i dati relativi al robot abbiamo che:

$$\omega_{min} \geq 7 \text{ giri/s} \quad (5.7)$$

Per rispettare i vincoli imposti, è stato scelto di utilizzare quattro motori 42BYGHW811 prodotti dalla Wantai e che hanno le seguenti caratteristiche: coppia pari a 0.47Nm e velocità di rotazione di 8 giri/s.

5.1.3 Ruote omnidirezionali

Per la natura del meccanismo di movimento della palla, le tradizionali ruote gommate non avrebbero permesso la rotazione della palla in ogni direzione. Per questo motivo, la scelta delle ruote è ricaduta su un tipo particolare, le ruote omnidirezionali.

Questa tipologia di ruote è formata da tanti piccoli dischi, disposti lungo la circonferenza principale e orientati perpendicolarmente alla direzione di rotazione. L'effetto che si ottiene è che la ruota può sviluppare una forza lungo la direzione di movimento, ma può anche scorrere lateralmente se viene applicata qualche forza longitudinale.

Le ruote usate nel progetto sono mostrate in Figura 5.3.



Figura 5.3: Ruote omnidirezionali utilizzate

5.2 Descrizione elettronica

L'elettronica a bordo del robot si occupa di ricevere i comandi dall'utente, acquisire i dati dai sensori inerziali e comandare i motori per far sì che il sistema rimanga in equilibrio.

I componenti posizionati a bordo del robot sono:

STM32: microcontrollore a 72MHz;

iNemo 2: unità inerziale dotata di 6 DOF;

Xbee: modulo per le comunicazioni wireless;

ST eval 6470: scheda per il controllo per motori stepper;

Wantai 42BYGHW811: motori stepper;

ST L5973: regolatore di tensione;

Batteria Piomb Sigil 12v 1.2ah: batteria al piombo da 12V.

Nelle prossime sezioni vengono descritte nel dettaglio le funzioni dei vari componenti e successivamente è mostrato come sono collegati tra di loro.

5.2.1 STM32

Gli algoritmi descritti nel Capitolo 4 sono stati implementati su un microcontrollore STM32F103. L'STM32 è un microcontrollore basato su core ARM Cortex-M3 32-bit RISC prodotto da STMicroelectronics [10]. L'architettura del microcontrollore è visibile in Figura 5.4.

L'STM32 è dotato delle seguenti caratteristiche:

- architettura 32 bit;
- frequenza di clock pari a 72MHz;
- potenza di calcolo di 1.25 DMIPS/MHz durante l'esecuzione diretta dalla memoria Flash;
- capacità memoria Flash pari a 512KB;
- capacità memoria SRAM pari a 64KB;
- 8 timers, 3 ADC a 12 bit con 7 canali ciascuno, 2 DAC;
- DMA (Direct Memory Access) a 12 canali;
- fino a 112 GPIO (General Purpose Input/Output).

L'architettura di questo microcontrollore permette di memorizzare le istruzioni e i dati sia nella memoria Flash che nella RAM; l'accesso simultaneo a queste informazioni è garantito da due bus interni indipendenti (APB1 e APB2).

L'STM32 è dotato di periferiche per l'utilizzo dei più comuni protocolli di comunicazione, tra le quali: USB (Universal Serial Bus), CAN (Controller Area Network), 3 SPI (Serial Peripheral Interface bus), 2 I2C (Inter-Integrated Circuit), 5 UART (Universal Asynchronous Receiver / Transmitter), IrDA (Infrared Data Association) e EMI (External Memory Interface).

Il microcontrollore STM32 è stato utilizzato montato su scheda STLMB, mostrata in Figura 5.5. Questa board mette a disposizione 3 stripline (due da 24 pin e una da 20 pin) per interfacciarsi direttamente con il processore: la stripline da 20 pin è dedicata alla programmazione del microcontrollore

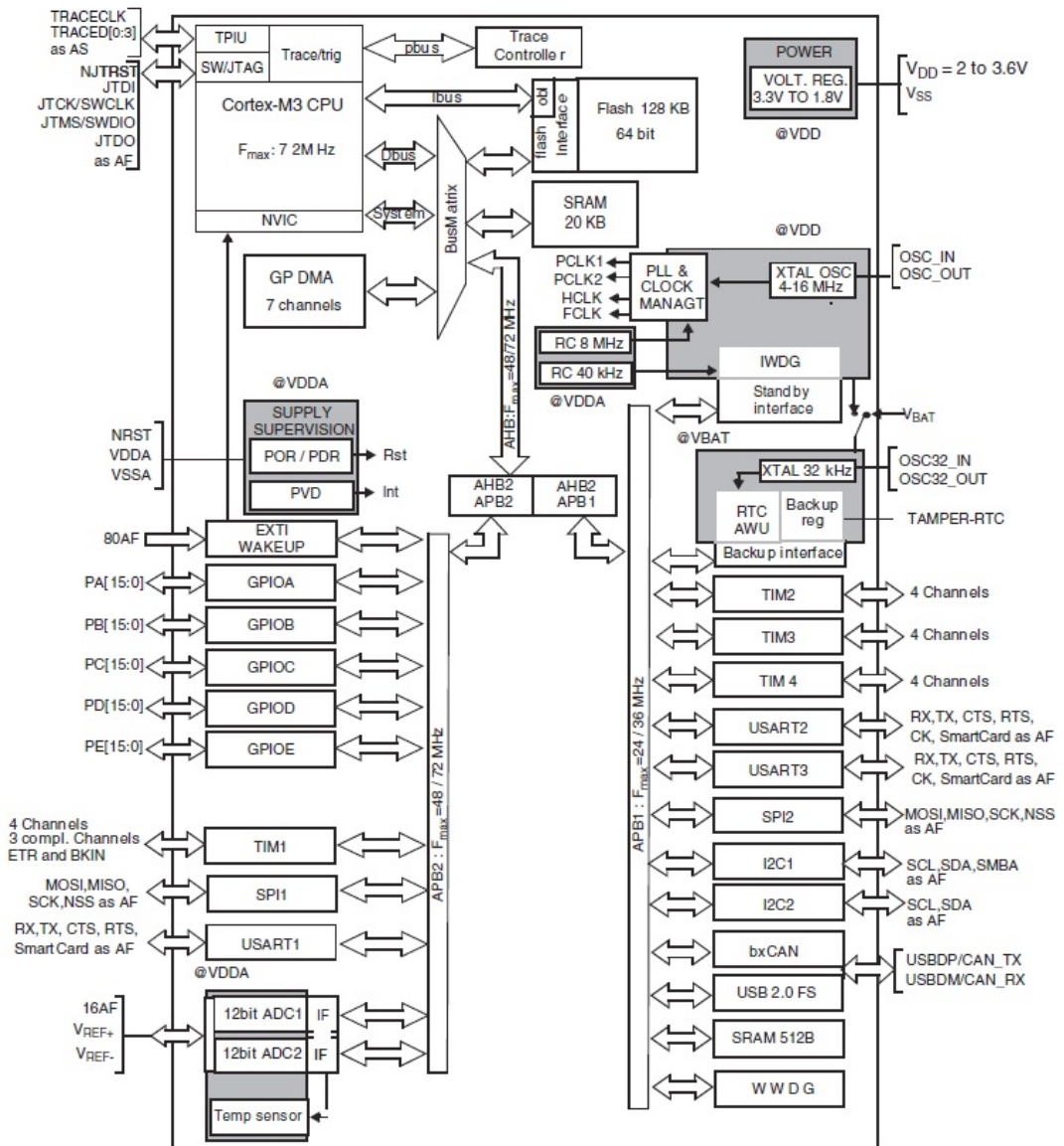


Figura 5.4: Architettura microcontrollore STM32

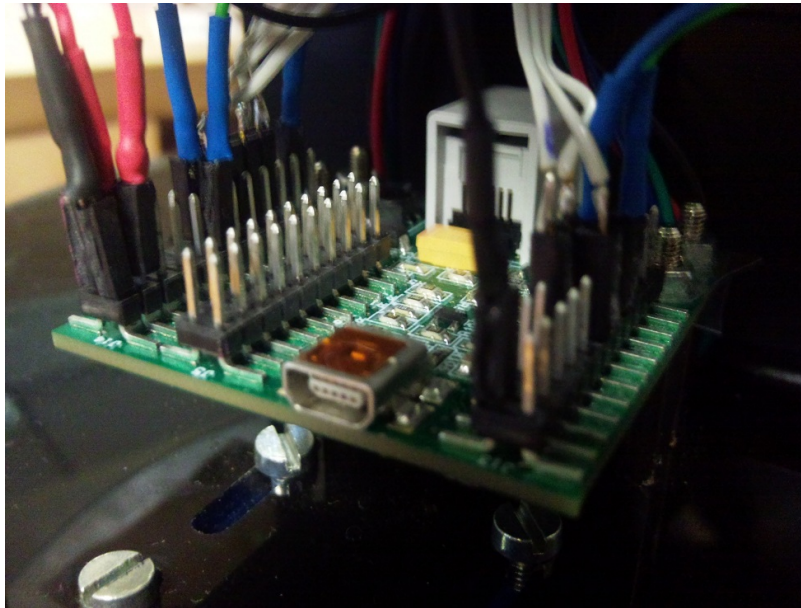


Figura 5.5: Scheda madre per l'STM32

tramite interfaccia JTAG, le altre due sono invece collegate alle varie GPIO e agli ingressi/uscite delle periferiche. L'STLMB è dotata anche di un ingresso mini USB e uno NXT per la connessione di componenti della serie Lego Mindstorm. Per osservare i vari collegamenti effettuati sulla board si rimanda all'Appendice B.

5.2.2 iNemo 2

Per lo sviluppo degli algoritmi per la stima degli angoli di inclinazione, ci si è basati su una scheda elettronica già realizzata dall'azienda italo-francese STMicroelectronics e usata come evaluation board di alcuni dispositivi inerziali.

L'iNemo2 (STEVAL-MKI062V2), Figura 5.6, è una scheda elettronica, delle dimensioni di 4 cm per 4 cm, che integra al suo interno 10 DOF composti da: modulo geomagnetico a sei assi, giroscopio a due assi, giroscopio ad un asse e sensore di pressione. Oltre questi sensori sono presenti a bordo un modulo per la misura della temperatura e un microcontrollore STM32

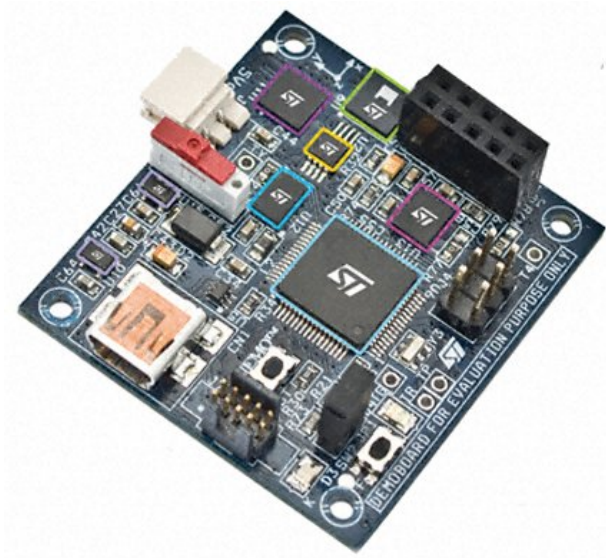


Figura 5.6: iNemo 2

(STM32F103RE).

I sensori presenti all'interno sono:

LPR430AL: giroscopio MEMS a due assi che fornisce la velocità angolare per i riferimenti di Roll e Pitch. La velocità angolare misurata è all'interno del range $+300^\circ/\text{s}$, $-300^\circ/\text{s}$ e la banda passante massima è di 140Hz. Le misure effettuate sono restituite come un segnale analogico in uscita dal dispositivo;

LY330ALH: giroscopio MEMS a singolo asse che fornisce la velocità angolare per il riferimento di Yaw. La velocità angolare misurata è all'interno del range $+300^\circ/\text{s}$, $-300^\circ/\text{s}$ e la banda passante massima è di 140Hz. Le misure effettuate sono restituite come un segnale analogico in uscita dal dispositivo;

LSM303DLH: modulo geomagnetico MEMS che integra un accelerometro e un magnetometro, entrambi triassiali. Le scale di misura dei due sensori sono selezionabili dall'utente e sono rispettivamente $\pm 2\text{g}$ / $\pm 4\text{g}$ / $\pm 8\text{g}$ per l'accelerometro e ± 1.3 / ± 1.9 / ± 2.5 / ± 4.0 / ± 4.7 / ± 5.6 / ± 8.1 Gauss per il magnetometro. Il modulo integra

un'interfaccia per la comunicazione seriale basata su I2C che supporta la frequenza di 100kHz e 400kHz e permette di generare un'interrupt quando il dispositivo è mosso;

LPS001DL: sensore di pressione MEMS capace di misurare la grandezza in un range di 300 - 1100 mbar con una risoluzione di 0.1mbar. Le comunicazioni col dispositivo avvengono tramite SI o I2C;

STLM75: sensore capace di misurare una temperatura compresa tra -55°C e $+125^{\circ}\text{C}$ e restituirne il valore tramite I2C.

La board può essere alimentata fornendo 5V sia tramite USB che tramite l'apposito connettore di alimentazione. Le comunicazioni possono avvenire tramite USB o porta seriale. Sulla board è presente anche un connettore di espansione che rende disponibile qualche GPIO proveniente dall'STM32 e una stripline che permette la programmazione del microcontrollore presente a bordo.

5.2.2.1 IMU: Realizzazione pratica

Il filtro di Kalman esteso per la stima degli angoli di inclinazione (descritto nel Capitolo 4.1.5) è stato sviluppato all'interno dell'STM32 presente a bordo dell'iNemo 2, ottenendo una IMU capace di generare risultati a 100Hz. Questo filtro si è basato solo sui dati provenienti dai due giroscopi e dall'accelerometro presenti all'interno del modulo geomagnetico.

La comunicazione avviene tramite un protocollo realizzato ad hoc su RS232. Un frame generale del protocollo di comunicazione con l'iNemo 2 è mostrato in Figura 5.7. Il pacchetto è composto da quattro parti: TipoFrame identifica se si tratta di un messaggio o di un ack, Lunghezza identifica la lunghezza totale del messaggio, TipoMessaggio contiene la tipologia di messaggio

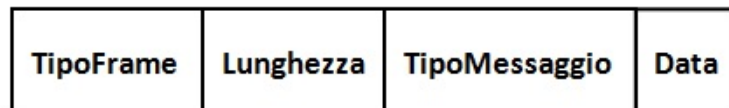


Figura 5.7: Descrizione di un frame di comunicazione con l'iNemo 2

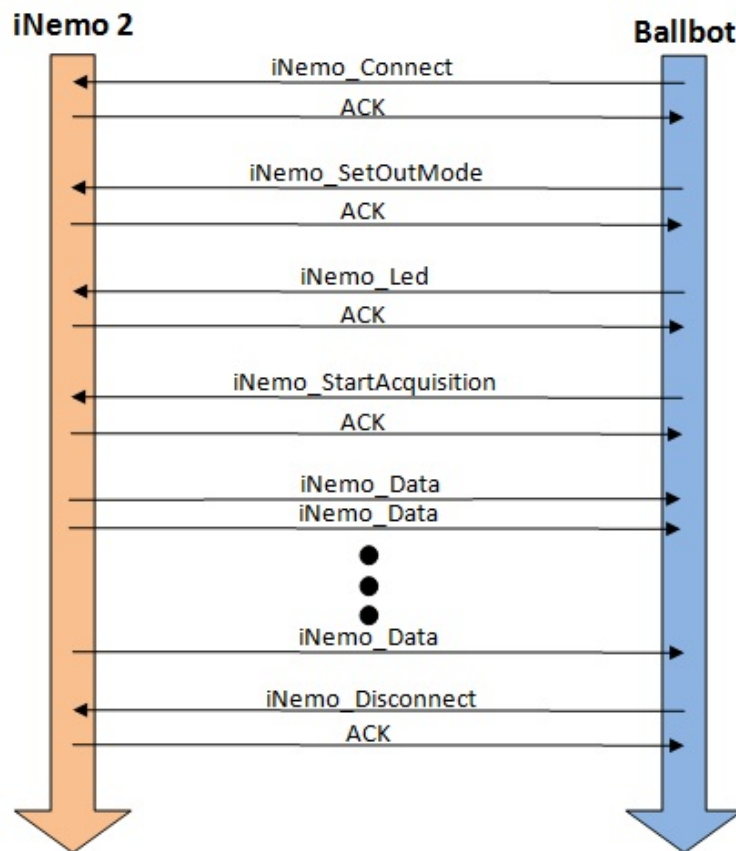


Figura 5.8: Rappresentazioni dei pacchetti scambiati tra iNemo 2 e Ballbot

inviata e Data contiene i dati da dover inviare. In particolare, TipoMessaggio può assumere i seguenti valori: iNemo_Connect, iNemo_Disconnect, iNemo_SetOutMode, iNemo_Start_Acquisition, iNemo_Led, iNemo_Data.

In Figura 5.8 è mostrato lo scambio dei pacchetti tra iNemo 2 e Ballbot. All'inizio delle comunicazioni, viene inviato all'iNemo un pacchetto di richiesta di connessione, se la richiesta è accettata tramite un ACK, si procede ad impostare i parametri di output per l'iNemo 2. Grazie a iNemo_SetOutMode è possibile specificare frequenza di invio dei dati, il numero di campioni e se inviare, oltre agli angoli filtrati, anche i valori raw letti dai sensori. Dopo di ciò, viene acceso un led sull'iNemo 2 per indicare che, fin qui, tutto è avvenuto correttamente e si invia la richiesta di acquisizione dei dati. Per come

è impostata la ricezione dei dati nei messaggi scambiati precedentemente, questi arrivano in continuazione con una frequenza di 100Hz fino a che non si invia un pacchetto di tipo iNemo_Disconnect.

5.2.3 Motori stepper

Il motore passo-passo (o stepper) è un motore elettrico sincrono in corrente continua senza spazzole, comandato in posizione, che può suddividere la propria rotazione in un grande numero di step. La posizione del motore può essere controllata accuratamente senza dover ricorrere al controllo ad anello chiuso, quindi senza l'utilizzo di sensori di posizione o velocità.

I vantaggi nell'uso dei motori passo-passo sono diversi: costo contenuto, controllo in posizione in anello aperto, precisione nel posizionamento, coppia elevata anche senza l'utilizzo di riduttori, elevata robustezza meccanica ed elettrica. D'altro canto producono minime vibrazioni soprattutto ai regimi più bassi, assorbono la stessa quantità di corrente indipendentemente dalla forza elettromotrice applicata e raggiungono velocità di rotazione non molto elevate.

5.2.3.1 Principio di funzionamento

I motori stepper sono motori che hanno come scopo quello di mantenere fermo l'albero in una posizione di equilibrio: se alimentati si limitano infatti a bloccarsi in una ben precisa posizione angolare. La possibilità di ruotare è ottenibile solo indirettamente: occorre inviare al motore una serie di impulsi di corrente, secondo un'opportuna sequenza, in modo tale da far spostare, per scatti successivi, la posizione di equilibrio. Se la rotazione degli impulsi avviene troppo velocemente è possibile arrivare allo stallo del motore.

Il motore è composto da un rotore e uno statore. Il rotore appare come una coppia di ruote dentate, affiancate e solidali con l'albero del motore, costituite da un nucleo magnetico (una ruota polarizzata come nord, l'altra come sud). Tra le due ruote è presente uno sfasamento esattamente pari ad $1/2$ del passo dei denti.

Lo statore è invece costituito da un insieme di 4 o più avvolgimenti che sono

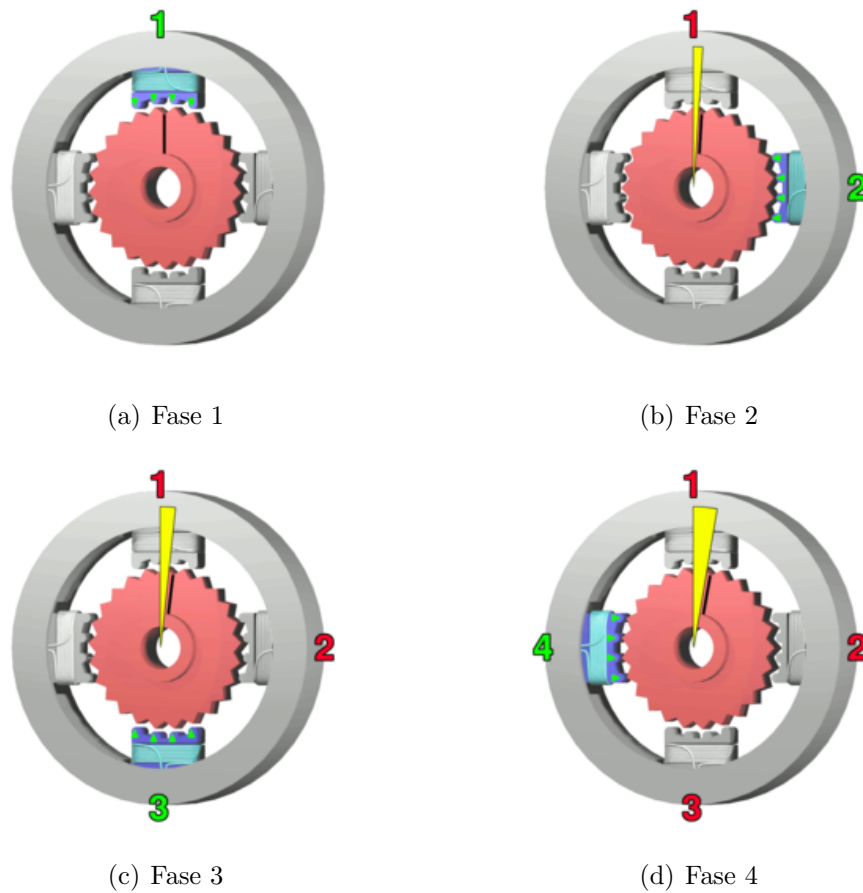


Figura 5.9: Esempio di rotazione di un motore con 100 step

arrotolati attorno a dei piccoli denti che si affiancano esattamente con quelli del rotore.

Un esempio di rotazione può essere vista in Figura 5.9. L'elettromagnete in alto (1) viene eccitato, attraendo i denti più vicini del rotore. Si inizia poi la rotazione in senso orario. L'elettromagnete (1) viene spento e si inizia ad eccitare quello a destra (2), facendo ruotare leggermente il rotore e attraendo i denti più vicini all'elettromagnete (2). Si conclude poi la rotazione di uno step, spegnendo (2) ed eccitando, in sequenza, anche gli elettromagneti (3) e (4). Il numero di gradi percorsi per step, p , può essere così calcolato:

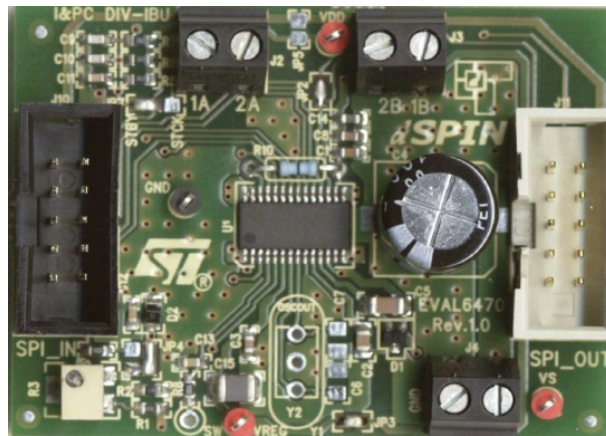


Figura 5.10: ST eval6470

$$p = 360^\circ / n_r \quad (5.8)$$

dove n_r è il numero di denti presenti nel rotore. I motori passo-passo più comuni hanno $p = 1.8^\circ/\text{s}$.

5.2.4 ST eval6470

L'ST eval6470 (Figura 5.10) è una scheda elettronica creata per il controllo di motori passo-passo. La scheda è costruita attorno al chip L6470 che si occupa sia della comunicazione che del controllo del motore collegato.

L'integrato L6470 è stato sviluppato per semplificare l'utilizzo dei motori stepper. Come si può vedere dalla Figura 5.10, che riporta il diagramma logico interno, questo chip è diviso principalmente in due parti: una per la logica di controllo, una di potenza per il comando dei motori.

La parte di logica permette la comunicazione, tramite SPI a 5Mbit/s, con un dispositivo esterno e contiene diversi registri in cui sono memorizzate le informazioni sui parametri del controllo che si sta attuando. Tramite SPI è possibile leggere / scrivere questi registri e mandare le istruzioni per comandare direttamente il motore. La parte di potenza, invece, è composta da due ponti di quattro DMOS l'uno, dotati di un circuito di protezione da sovraccarichi di corrente. L'L6470 prevede l'utilizzo del microstepping, fino

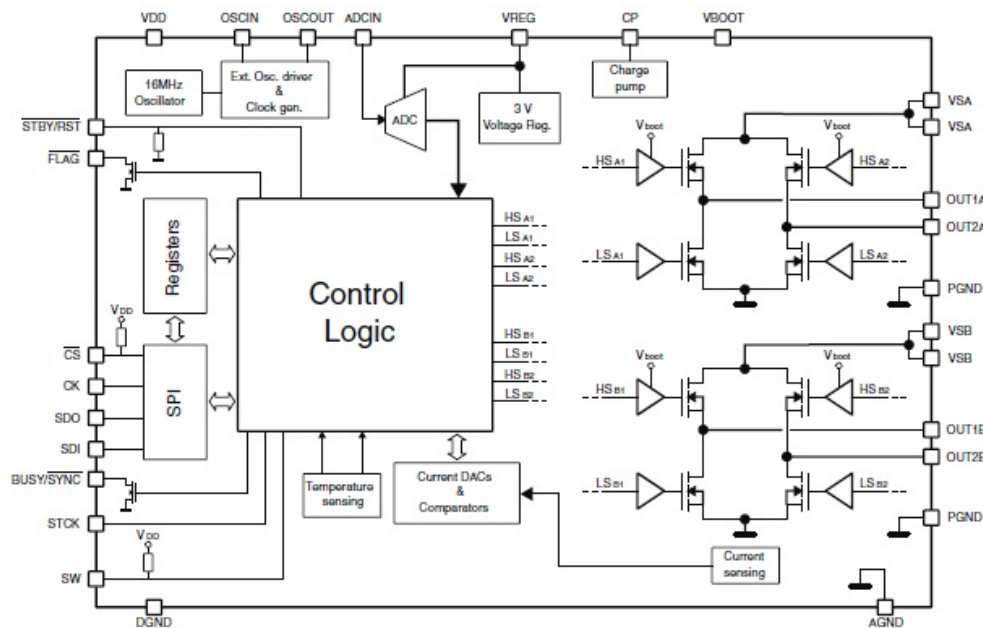


Figura 5.11: ST L6470

a 128 divisioni. Questa funzione permette di suddividere uno step in 128 microstep ottenendo un movimento più fluido e con meno vibrazioni.

5.2.4.1 SPI

L'SPI (Serial Peripheral Interface) è uno standard di comunicazione seriale tra microcontrollore e periferiche o tra più microcontrollori. La trasmissione avviene tra un dispositivo, detto master, e uno o più slave. Il master controlla il bus, emette il segnale di clock, decide quando iniziare e terminare la comunicazione. Contrariamente, il dispositivo di slave si limita a rispondere alle richieste ricevute.

Il bus SPI (Figura 5.12) è formato da tre collegamenti: SCLK, MOSI, MISO, più un altro (SS) usato per l'abilitazione della periferica di slave. Il canale SCLK (Serial Clock) è utilizzato dal master per inviare il segnale di clock per la sincronia dei singoli bit. Sulla linea MOSI (Master Output Slave Input) passano i dati provenienti dal master e diretti verso lo slave invece, su quella MISO (Master Input Slave Output) i dati passano nella direzione opposta.

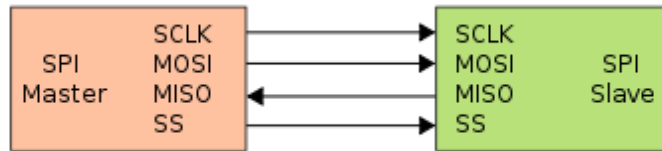


Figura 5.12: Schema del bus SPI

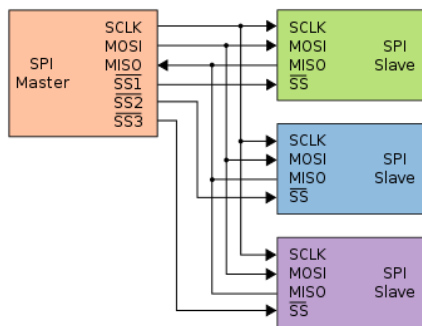


Figura 5.13: Comunicazione con più slave tramite più linee SS

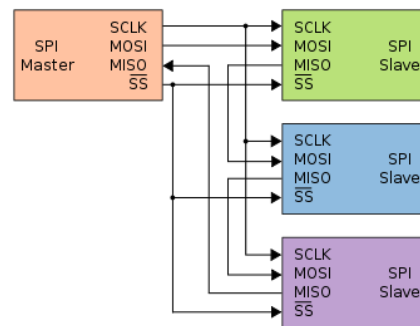


Figura 5.14: Comunicazione tra più slave tramite daisy chain

Il bus SPI ha le seguenti caratteristiche:

- è di tipo seriale;
- è sincrono, per la presenza del segnale di clock;
- è di tipo full-duplex, in quanto la comunicazione può avvenire contemporaneamente sia sulla linea MISO che sulla MOSI.

Nel caso ci sia la necessità di comunicare con due o più slave si possono utilizzare due tecniche particolari: controllare i dispositivi slave singolarmente o secondo la configurazione daisy chain.

Nel primo caso (Figura 5.13), tutti gli slave condividono le stesse linee SCLK, MISO e MOSI ma ognuno ha la propria linea SS. In questo modo, per comunicare con un determinato slave basta attivare la linea SS associata. Quando gli slave diventano molti e non si dispone di tutte le linee di abilitazione necessarie, è possibile utilizzare la configurazione daisy chain (Figura 5.14). In

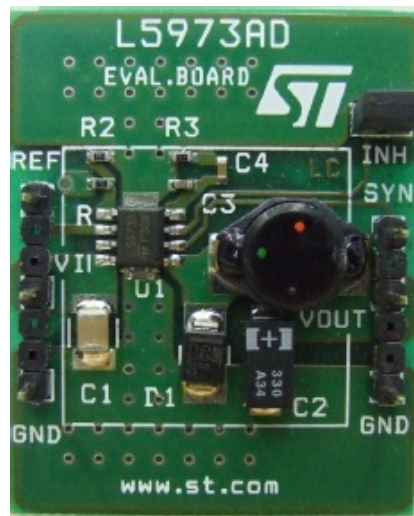


Figura 5.15: Evaluation board for L5973

questo schema, tutti gli slave condividono i segnali SCLK e SS, ma la linea MOSI è collegata alla MISO dello slave precedente. I dati, provenienti dalla linea MOSI del master, arrivano agli slave nell'ordine con cui sono stati posizionati, tramite i collegamenti MOSI-MISO, e tornano poi al master, sulla linea di MISO, il quale ha conferma che tutti li abbiano ricevuti.

5.2.5 ST L5973

Sul robot è stato alloggiato un regolatore di tensione L5973 col compito di stabilizzare la tensione e generare un potenziale adatto ad alimentare i dispositivi presenti.

L'ST L5973 è un regolatore di tensione capace di erogare 2A di corrente con una tensione di uscita che può variare da 1.235V a 35V.

Nel progetto è stato utilizzato saldato sulla board mostrata in Figura 5.15, che è stata configurata in modo da ricevere in ingresso 12V e fornire un uscita di 5V.

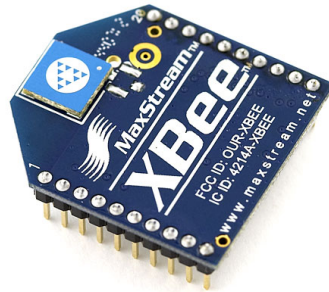


Figura 5.16: Modulo XBee

5.2.6 Modulo XBee

I moduli XBee (Figura 5.16) sono dispositivi a basso costo e a bassa potenza pensati per rendere wireless una comunicazione seriale su RS232. Questi moduli sono basati sullo standard ZigBee/IEEE 802.15.4, lavorano nella banda ISM, precisamente alla frequenza di 2.4GHz, e hanno una potenza trasmisiva che varia da 1mW per la serie standard a 100mW per la serie PRO.

Punti di forza di questi dispositivi sono: le ridotte dimensioni, l'alta affidabilità e i bassi consumi energetici. Gli XBee vengono normalmente mantenuti, infatti, in stato di "sleep" e quindi a basso consumo e vengono automaticamente attivati qualora si verifichi la necessità di una trasmissione o ricezione di dati.

La trasmissione dei dati può avvenire da un modulo verso qualsiasi altro in modo half-duplex, integrando già il controllo degli errori. Tramite l'utilizzo dei comandi AT è possibile impostare i parametri interni di configurazione.

5.2.7 Collegamenti tra i vari componenti

I collegamenti tra i vari componenti descritti nelle precedenti sezioni sono riportati in Figura 5.17.

L'STM32 ha un ruolo centrale nell'architettura, infatti, riceve e invia i dati

verso l'XBee, l'iNemo 2 e le schede dei motori. L'STM32 implementa i quattro PID necessari per il controllo del robot, Sezione 4.2.3.

L'iNemo 2 invia gli angoli RPY che identificano l'inclinazione del robot all'STM32 tramite RS232 usando il protocollo descritto nella Sezione 5.2.2.1. I dati vengono inviati con una frequenza di 100Hz.

L'STM32 comunica col PC per ricevere, tra l'altro, i comandi per muovere il robot e per inviare le informazioni di log. Avendo la necessità di controllare il robot anche a distanza, la comunicazione avviene tramite due XBee configurati per effettuare un collegamento punto-punto.

Verso le schede per il controllo dei motori (eval6470), la comunicazione avviene tramite bus SPI. L'STM32 riceve la posizione in cui si trovano i motori, che utilizza in ingresso al controllo, ed invia la velocità di attuazione dei motori, risultato del controllo. Dovendo comunicare tramite SPI con quattro schede, si è scelto di utilizzare quattro segnali di abilitazione per la scelta dello slave, uno per ogni scheda motori.

Per quanto riguarda l'alimentazione, sul robot è stata montata una batteria al piombo da 12V. L'STM32 e l'iNemo 2 prendono la corrente tramite il regolatore ST L5973, invece l'XBee prende l'alimentazione di 3.3V da un regolatore di tensione montato sulla mother board dell'STM32 (la scheda STLMB). Le schede eval6470, accettando come ingresso 12V, sono collegate direttamente alla batteria.

Nella Figura 5.18 è possibile vedere il circuito per l'accensione e la ricarica della batteria, in cui J1 è in connettore jack da 2.1/5.5 mm utilizzato per ricaricare la batteria (V1) e SW1 è il pulsante di accensione del robot. Il diodo D2 è posizionato per evitare che un'inavvertita inversione dei collegamenti per la ricarica causi danni ai dispositivi montati sul robot; nel caso, invece, di eventuali corto circuiti, l'alimentazione viene staccata dal fusibile F1. Il LED D1 (blu nel prototipo reale) indica quando il robot è acceso.

5.3 GUI

L'interfaccia grafica per il controllo del robot è stata realizzata tramite le librerie Qt di Nokia [8] e permette di controllare il robot, settarne i parametri e acquisire il log.

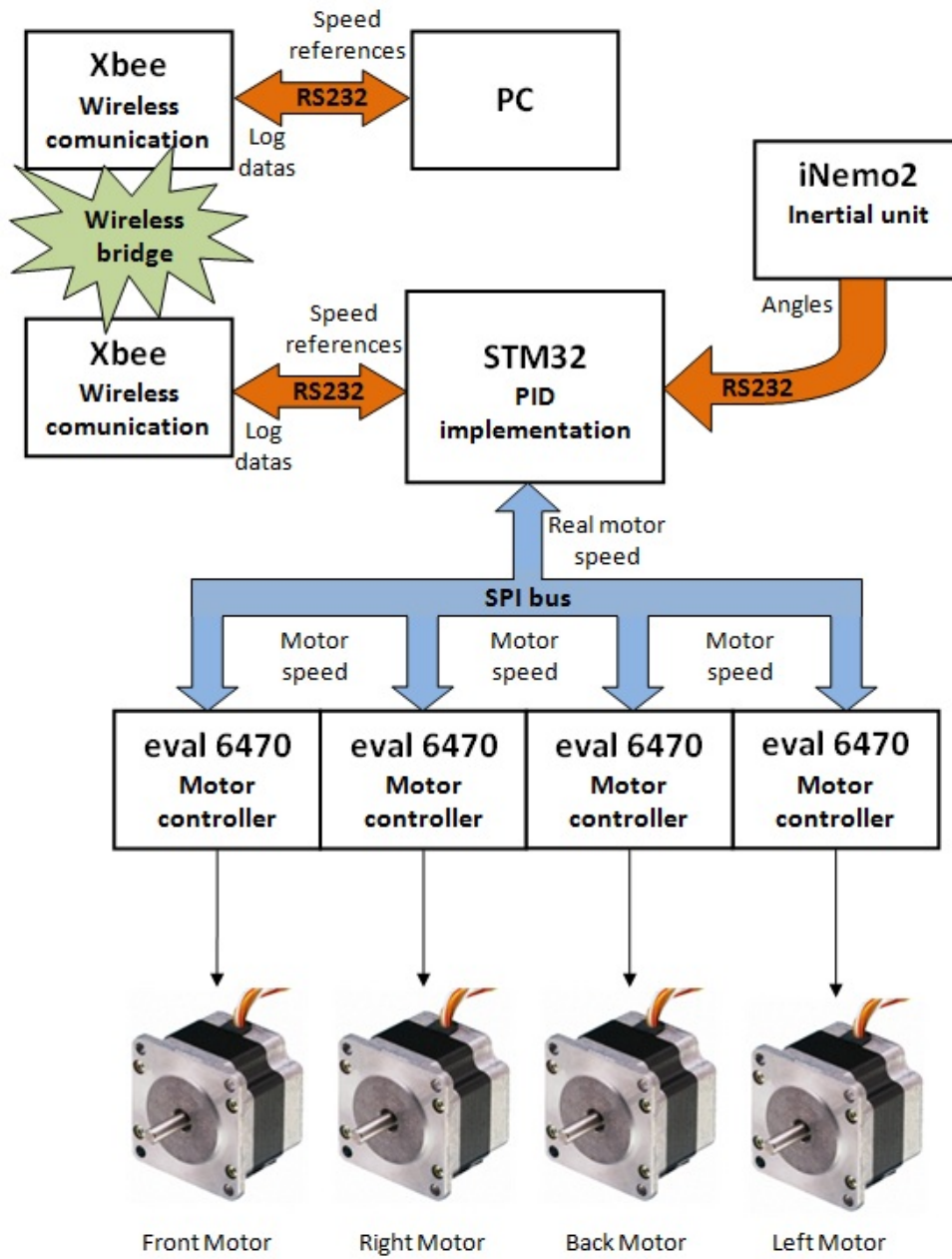


Figura 5.17: Schema dell'elettronica a bordo del robot

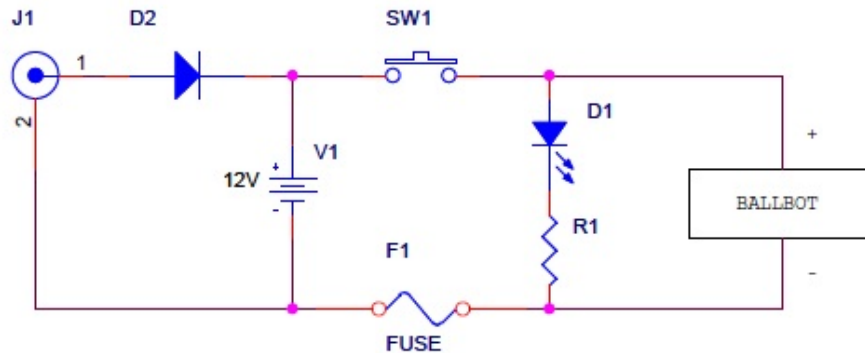


Figura 5.18: Circuito di alimentazione e carica della batteria del robot

La GUI è divisa in tre parti: “Connection”, “Control and PID parameters” e “Log”. Nella prima sezione è possibile connettersi al robot e impostare parametri di connessione quali: numero di porta seriale del PC da utilizzare, indirizzo del PC e del robot (consultate l’Appendice A per maggiori informazioni).

Nella schermata “Control and PID parameters” (Figura 5.19) è possibile impostare i parametri dei due PID di controllo (a sinistra) e comandare i movimenti del robot (a destra). Per controllare il robot è necessario utilizzare il quadrato rosso posto sulla destra della schermata: trascinando sopra il mouse con il tasto sinistro premuto è possibile impostare la velocità da far seguire al robot in direzione X e Y. Il puntino nero centrale indica la velocità nulla in entrambe le direzioni, salendo aumenta la velocità lungo la componente X, scendendo diminuisce la velocità lungo l’asse X (fino ad andare nella direzione opposta), andando verso destra aumenta la velocità lungo la componente Y e verso sinistra diminuisce la velocità lungo la Y. Sotto il quadrato rosso sono indicate le velocità effettive di movimento lungo le due direzioni.

Nella schermata di “Log” (Figura 5.20) viene catturato il file di log, inviato dal robot, e mostrato a video. Da sinistra a destra, vengono visualizzati: la velocità lungo l’asse X, la velocità lungo l’asse Y, la velocità di rotazione

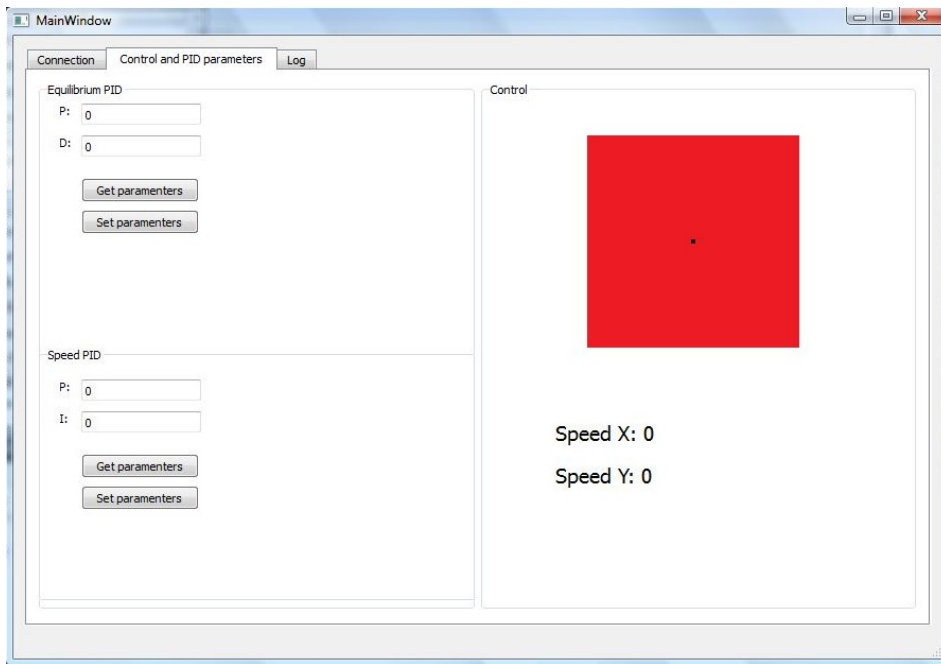


Figura 5.19: Interfaccia grafica (comandi)

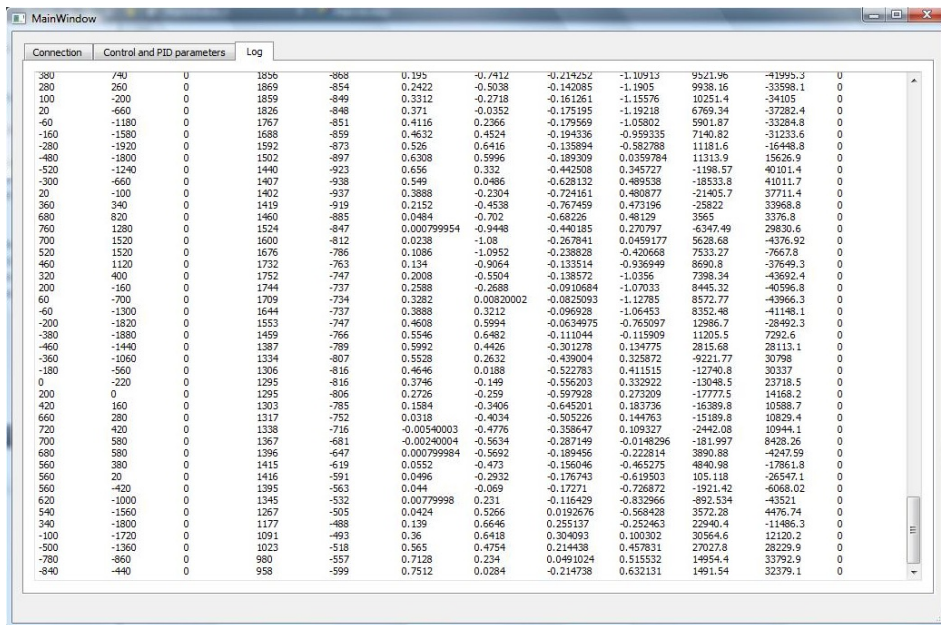


Figura 5.20: Interfaccia grafica (log)

attorno all'asse verticale, la posizione delle ruote lungo l'asse X, la posizione delle ruote lungo l'asse Y, l'offset dell'angolo ϕ misurato, l'offset dell'angolo θ misurato, l'angolo ϕ , l'angolo θ , l'accelerazione lungo l'asse X, l'accelerazione lungo l'asse Y e l'ultimo valore, sempre zero, non è utilizzato.

5.3.1 Comunicazione tra PC e Ballbot

La comunicazione tra la GUI e il robot avviene tramite RS232 passando attraverso due XBee per il collegamento wireless. Il protocollo utilizzato è di tipo affidabile e connesso, ossia i pacchetti arrivano a destinazione nell'ordine d'invio e nel qual caso un pacchetto venisse perso, viene rimandato.

I pacchetti scambiati durante la comunicazione hanno la forma mostrata in Figura 5.21. I cinque campi assumono il seguente significato: Own Addr è l'indirizzo del mittente, Dest Addr è l'indirizzo del destinatario, Tipo Messaggio descrive che tipo di dati verranno mandati, Data sono i dati da dover inviare e Checksum è un codice di controllo per verificare l'esattezza dei dati. Il campo Tipo Messaggio può assumere le seguenti tipologie: CMD_GetLog, CMD_GetSpeedPIDPar, CMD_SetSpeedPIDPar, CMD_GetStabilityPIDPar, CMD_SetStabilityPIDPar, CMD_SetSpeedRobot, CMD_SetRotSpeedRobot. Quando giunge il messaggio, il destinatario controlla il Checksum e, se risulta corretto, manda, come risposta, lo stesso messaggio ricevuto ma con indirizzi iniziali invertiti e nel campo Tipo Messaggio il valore ricevuto in OR con CMD_Reply_Add. Nel caso non giunge il messaggio di risposta, dopo 70ms il mittente rimanda il messaggio. Nell'attesa che un messaggio venga inviato correttamente non è possibile mandarne altri.

| Own Addr | Dest Addr | Tipo Messaggio | Data | Checksum |
|----------|-----------|----------------|------|----------|
|----------|-----------|----------------|------|----------|

Figura 5.21: Descrizione di un frame di comunicazione tra PC e Ballbot

5.4 Risultati ottenuti

Il risultato ottenuto ha visto la realizzazione di un prototipo funzionante di Ballbot capace di bilanciarsi autonomamente, resistere a lievi disturbi esterni senza cadere e seguire profili di velocità impostati dall'utente tramite l'utilizzo dell'interfaccia grafica descritta nel Capitolo 5.3. Una foto del prototipo funzionante mentre si bilancia sulla palla è mostrata in Figura 5.22. Il video completo della demo è possibile vederlo al seguente link: <http://www.youtube.com/watch?v=7Ib3l9KYbBM>.

Durante le prove effettuate, sono stati prodotti alcuni grafici rispetto a alcune variabili sensibili.

La prima prova effettuata è stata quella di accendere il Ballbot, attendere che si stabilizzasse sulla palla e osservare l'andamento a regime. In Figura 5.23 è mostrato l'andamento della posizione del robot nel piano orizzontale: dopo una prima fase di ricerca dell'equilibrio, si nota che riesce a bilanciarsi stando, a regime, all'interno di un quadrato di 2cm x 2cm. Nelle Figure 5.24 e 5.25 sono rappresentati gli andamenti degli angoli ϕ e θ che, a regime, subiscono oscillazioni di poco più di un grado.

Un'altra prova effettuata è stata quella di far seguire al robot un percorso diritto. Il grafico risultante è mostrato in Figura 5.26, in cui è possibile vedere la posizione del robot nel piano X Y.

L'ultimo esperimento effettuato ha visto la perturbazione del sistema tramite: variazione del baricentro (un oggetto è stato poggiato sulla sommità del robot) e spinta laterale. La posizione del robot è visibile in Figura 5.27, dove è possibile riconoscere: una prima fase (A) in cui il robot, acceso, trova l'equilibrio sulla palla, una seconda fase (B) in cui viene variato il baricentro e riesce a ribilanciarsi e una terza fase (C) in cui recupera la posizione di equilibrio dopo una spinta.

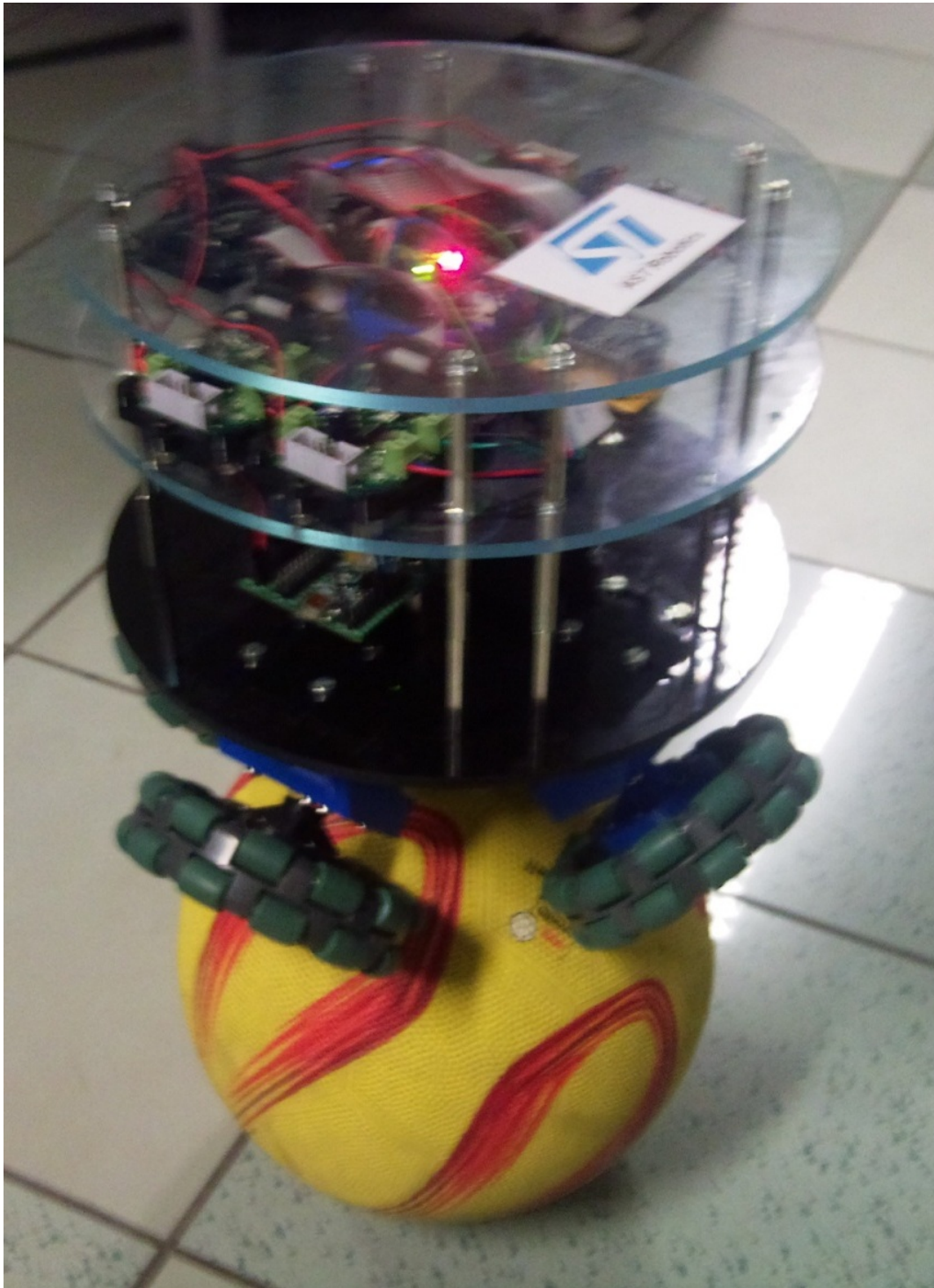


Figura 5.22: Prototipo funzionante di Ballbot

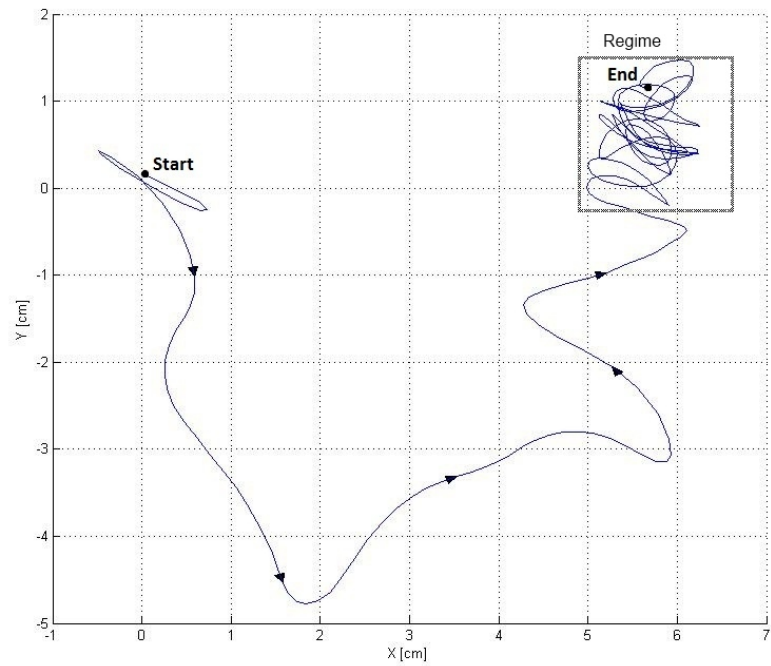


Figura 5.23: Andamento della posizione del robot nel piano 2D

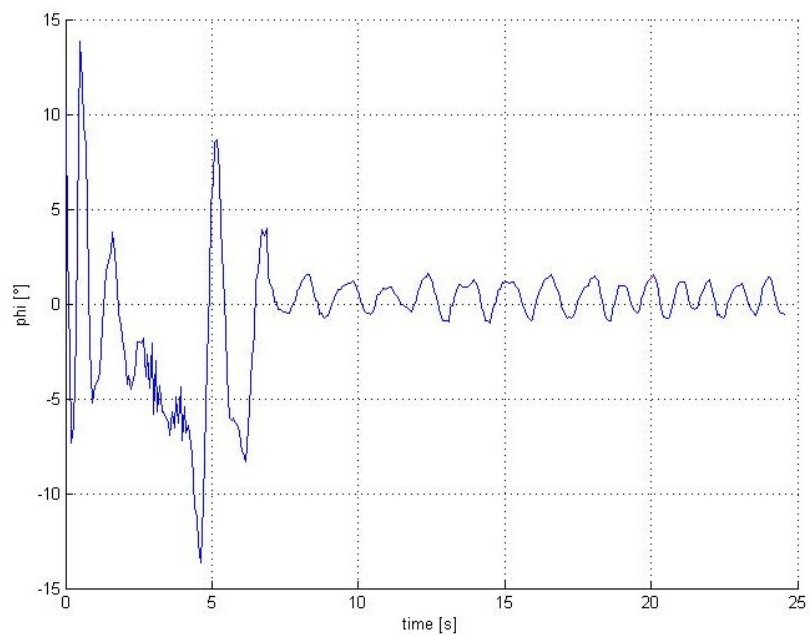


Figura 5.24: Andamento del valore di ϕ nel tempo

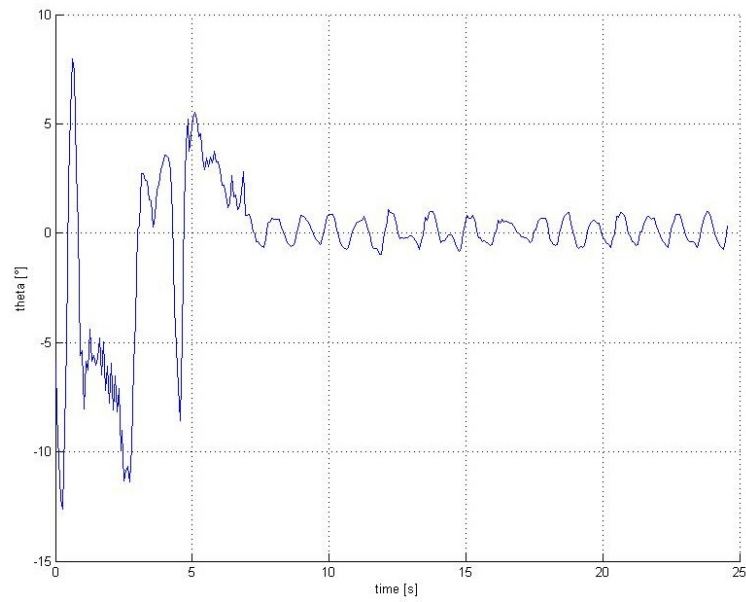


Figura 5.25: Andamento del valore di θ nel tempo

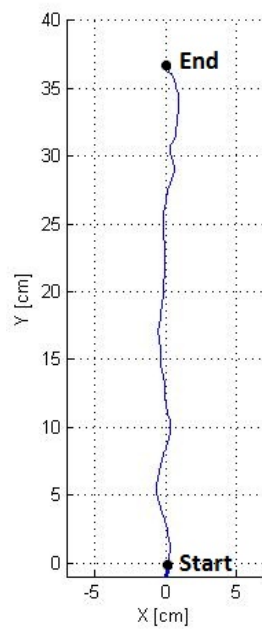


Figura 5.26: Andamento della posizione del robot nel piano 2D muovendosi lungo un percorso diritto di 36cm

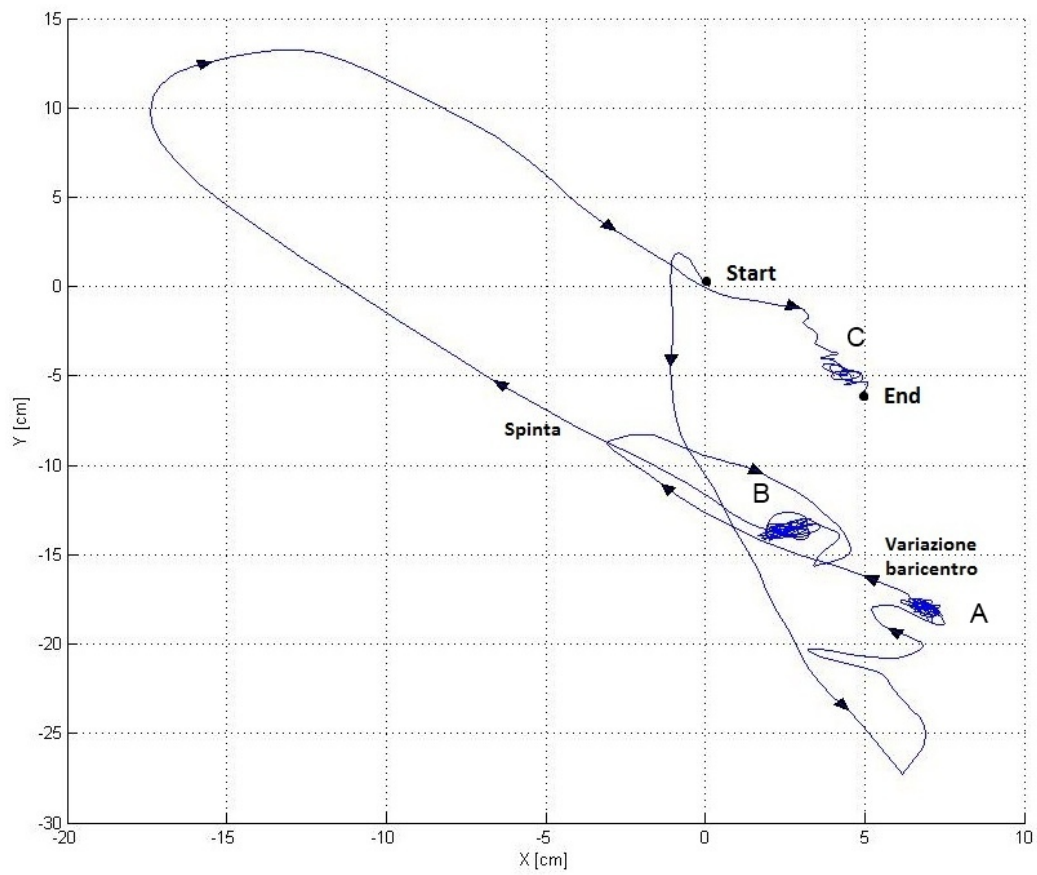


Figura 5.27: Andamento della posizione del robot nel piano 2D subendo dei disturbi

Capitolo 6

Direzioni future e conclusioni

In questo ultimo capitolo vengono riportate le considerazioni e le conclusioni riguardanti il lavoro svolto e vengono esposti i possibili sviluppi futuri.

6.1 Valutazioni conclusive

Lo sviluppo del prototipo finale ha raggiunto gli obiettivi che ci eravamo prefissati inizialmente. Il robot realizzato è, infatti, in grado di:

- bilanciarsi sulla palla e mantenere l'equilibrio stando fermo nelle vicinanze di un punto;
- resistere a piccoli disturbi esterni come spinte o oggetti appoggiati sulla parte alte del corpo;
- seguire profili di velocità desiderati e muoversi nell'ambiente comandato a distanza;
- muoversi in qualsiasi direzione senza bisogno di ruotare.

Con i test sperimentali si è osservato che nel bilanciarsi sulla palla il robot oscilla leggermente con uno scostamento dalla verticale di pochi gradi (1° - 2°) e un movimento della palla di un paio di centimetri. Questi risultati sono considerati buoni, anche se (leggermente) migliorabili.

In particolare si osserva che, l'approssimazione di considerare la dinamica del

Ballbot come ottenuta da due sistemi più semplici, disaccoppiati tra loro, è accettabile e ha portato allo sviluppo di un controllo funzionante.

6.2 Sviluppi futuri

Il prototipo di Ballbot realizzato durante l'attività di tesi può essere migliorato in modo da renderlo maggiormente stabile sulla palla e fargli ridurre l'ampiezza delle oscillazioni. Tre sono gli aspetti che possono portare le migliori volute: palla con una maggiore inerzia e più liscia, miglior tuning dei parametri dei vari algoritmi e migliori ruote omnidirezionali.

Una palla con una maggiore inerzia aiuta il robot a salirci senza che questa scappi; al contrario, se la palla avesse inerzia nulla, non sarebbe possibile arrampicarsi su di essa per mantenere l'equilibrio. Una palla del genere può essere ottenuta rivestendo di gomma una normale palla da bowling. Dedicare maggior tempo al tuning dei parametri degli algoritmi di controllo e di stima degli angoli aiuta a rendere il sistema più stabile. Un ulteriore contributo alla stabilità potrebbe essere portato dalla realizzazione di ruote omnidirezionali progettate ad hoc per il sistema in modo che tutti i rulli si trovino su un unico piano, senza spazi tra uno e l'altro e senza che possano muoversi a vuoto nei cambi di direzione prima di riprendere il rip sulla palla. Infine, un miglioramento ulteriore della stabilità può essere cercato tramite altri metodi di controllo come LQR o apprendimento per rinforzo.

Un ulteriore miglioria al sistema potrebbe essere apportata dalla sostituzione dei motori stepper con dei motori DC con riduttore che, più leggeri e dotati di una maggior coppia, possano aumentare la velocità del robot facendogli raggiungere anche angoli di inclinazione maggiore in tutta sicurezza.

Un altro aspetto per migliorare il robot potrebbe essere l'aggiunta di un controllo per gestire il movimento di Yaw, ossia la rotazione sopra la palla.

Per quanto riguarda la GUI per il controllo dell'utente, questa può essere ulteriormente migliorata permettendo, ad esempio, di preimpostare profili di velocità e percorsi senza che l'utente debba specificarlo ogni istante. Potrebbe anche essere sviluppato un sistema di controllo dei movimenti tramite smartphone. Si potrebbe infatti realizzare un'applicazione per dispositivi mobili che permette di comandare la velocità del robot tramite l'inclinazione dello

smartphone o tramite particolari gesture eseguiti sul display touch-screen. Le possibili applicazioni del Ballbot potrebbero essere diverse infatti, grazie alla sua agilità e alla possibilità di muoversi in ogni direzione senza aver bisogno di curvare, potrebbe essere facilmente impiegato in ambienti stretti e ricchi di ostacoli.

L'aggiunta di sensori come sonar o telecamere potrebbe rendere possibile l'interazione in ambienti dove vive l'uomo per fargli svolgere semplici compiti, come può essere il trasporto automatico di oggetti.

Bibliografia

- [1] David P. Anderson. <http://www.geology.smu.edu/dpa-www/robo/nbot/>. 2004.
- [2] Yi Yu Li Cheng-Kai Chan Ching-Wen Liao, Ching-Chih Tsai. *Dynamic Modeling and Sliding-Mode Control of a Ball Robot with Inverse Mouse-Ball Drive*. 2009.
- [3] Katherine Mullens Aaron Burmeister Susan Miles Nathan Farrington Kari Thomas Hoa G. Nguyen, John Morrel and Douglas W. Gage. *Segway Robotic Mobility Platform*. 2004.
- [4] M.J.D. Hayes J.B. Holland and R.G. Langlois. *A Slip Model for the Spherical Actuation of the Atlas MotionPlatform*. 2005.
- [5] R. E. Kalman. *A New Approach to Linear Filtering and Prediction Problems*. 1960.
- [6] Dean Kamen. <http://www.segway.com/>. 2001.
- [7] Masaaki Kumaga and Takaya Ochiai. *Development of a Robot Balanced on a Ball*. 2009.
- [8] Qt. <http://qt.nokia.com/>.
- [9] S. Yutam R. Nakajima, T. Tsubouchi and Koyanagi E. *A developement of a new mechanism of an autonomous unicycle*. 1997.
- [10] STMicroelectronics. <http://www.st.com/>.

- [11] George Kantor Umashankar Nagarajan and Ralph L. Hollis. *Trajectory Planning and Control of an Underactuated Dynamically Stable Single Spherical Wheeled Mobile Robot*. 2009.

Appendice A

Manuale di utilizzo

In questa sezione viene spiegato come avviare il robot e controllarlo per fargli compiere i movimenti desiderati.

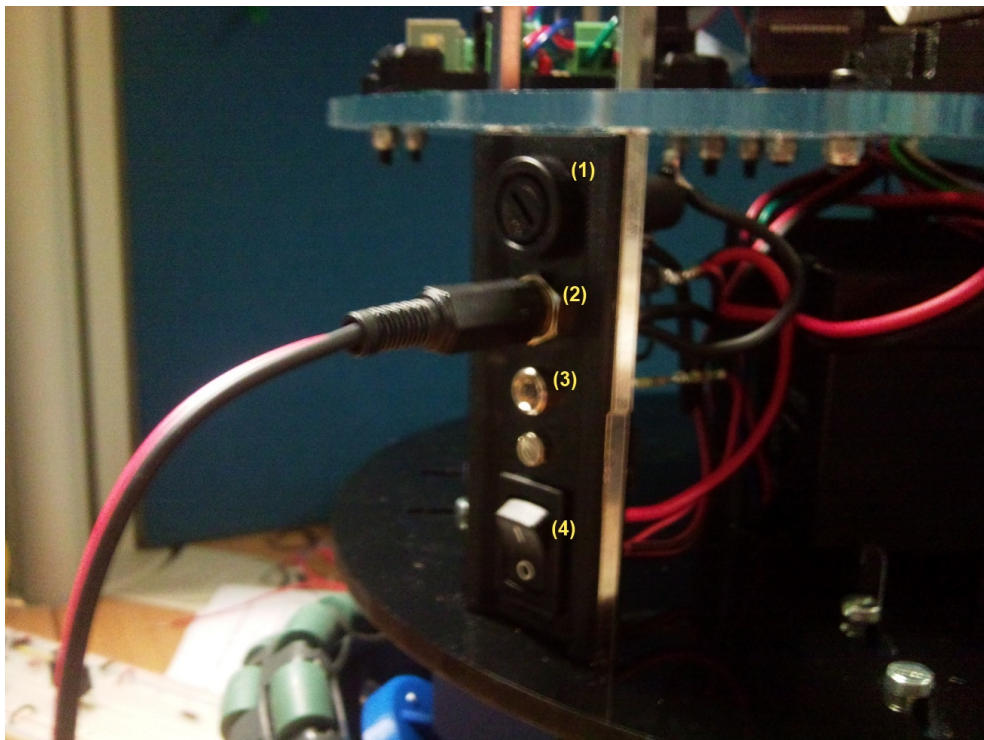


Figura A.1: Pannello accensione robot

Il pulsante di accensione (Figura A.1 (4)) è posto sul pannellino nero sul

retro del robot; prima di accenderlo è necessario poggiare il robot sulla palla in modo che le ruote non inizino a girare tentando di bilanciarsi. Sullo stesso pannello sono posizionati anche il fusibile per evitare sovracorrenti (Figura A.1 (1)), il connettore per l'alimentazione (Figura A.1 (2)) e il LED che mostra quando il robot è acceso (Figura A.1 (3)). Il robot accetta un'alimentazione di 14.4V tramite jack da 2.1/5.5 mm; con l'alimentazione collegata è comunque possibile avviare il robot.

Acceso e posto sulla palla, il robot inizierà a bilanciarsi cercando di rimanere in equilibrio sulla palla.

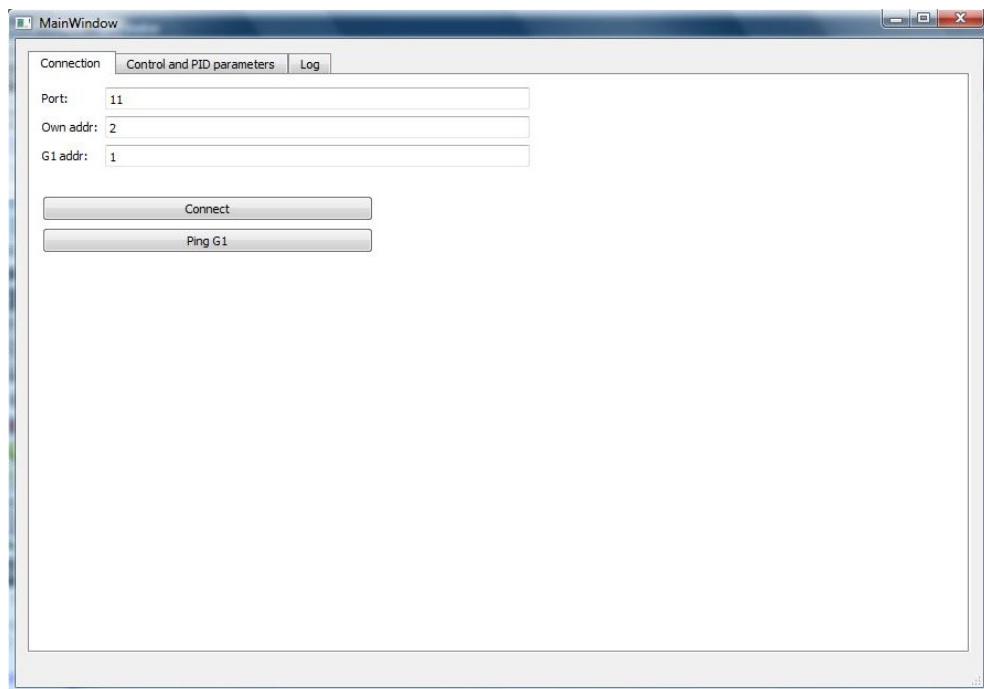


Figura A.2: GUI: schermata di connessione

Una volta che si è stabilizzato, è possibile connettersi al robot per farlo muovere a piacimento dell'utente. Dopo aver collegato l'XBee al PC, avviare la GUI di controllo; questa si presenterà come in Figura A.2. In questa schermata è possibile specificare il numero di porta seriale del PC da utilizzare per la connessione, il proprio indirizzo e l'indirizzo del Ballbot. Nel caso fosse necessario cambiare questi ultimi due parametri sarà obbligatorio modificarli

anche all'interno del programma in esecuzione sul robot. Stabilita la connessione, è possibile accertarsi che tutto funzioni mandando un messaggio di ping tramite il pulsante "Ping G1".

Per controllare il robot, spostarsi nella schermata "Control and PID parameters" in cui è possibile modificare i parametri di configurazione dell'algoritmo di controllo e inviare comandi di velocità, come descritto nella Sezione 5.3.

Appendice B

Collegamenti tra le board

In Figura B.1 è possibile vedere uno schematico riportante tutte le board e i collegamenti effettuati a bordo del robot. In particolare, è facile riconoscere tutte le schede che sono state precedentemente descritte. Oltre queste, è possibile vederne una chiamata "Motor connection" che si occupa di dividere il bus SPI e assegnare, ad ogni scheda di controllo dei motori, in corretto segnale di enable.

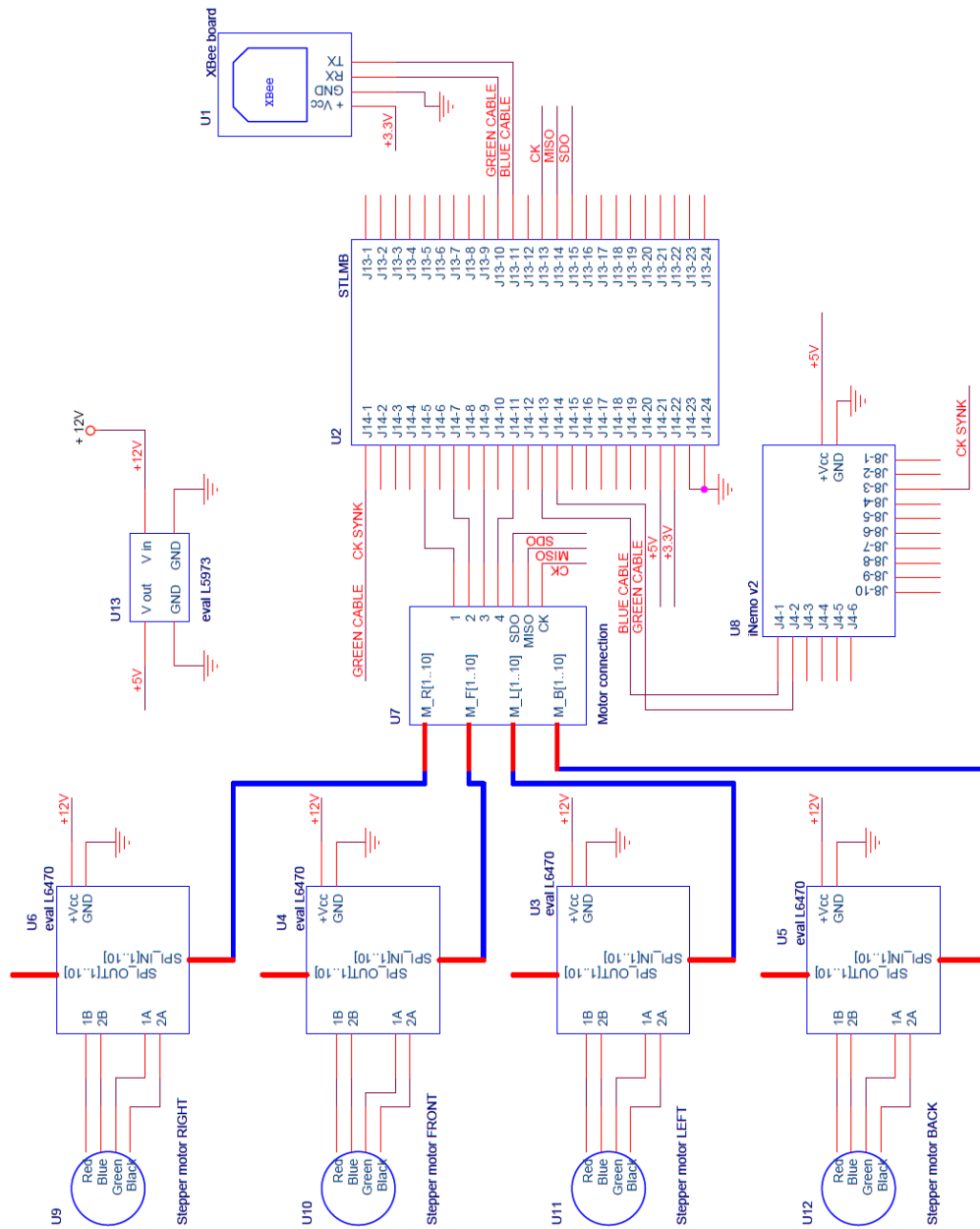


Figura B.1: Collegamenti tra le varie board montate sul robot

Appendice C

Elenco delle abbreviazioni

Abbreviazioni usate nel testo:

ABS: Acrilnitrile Butadiene Stirene;

ACK: ACKnowledge;

ADC: Analog to Digital Converter;

DC: Direct Current;

DMPIS: Dhrystone Million Instruction Per Second;

DMOS: Double-Diffused Metal Oxide Semiconductor;

DOF: Degree Of Freedom;

EKF: Extended Kalman Filter;

FPGA: Field Programmable Gate Array;

GPIO: General Pourpose Input Output;

GUI: Graphical User Interface;

I2C: Inter Integrated Circuit;

ISM: Industrial Scientific and Medical;

IMU: Inertial Measurement Unit;

JTAG: Joint Test Action Group;

LED: Light Emitting Diode;

LQR: Linear Quadratic Regulator;

MEMS: Micro Electro-Mechanical System;

MISO: Master Input Slave Output;

MOSI: Master Output Slave Input;

PC: Personal Computer;

PID: Proportional Integral Derivative;

RAM: Random Access Memory;

RISC: Reduced Instruction Set Computer;

RPY: Roll Pitch Yaw;

RS232: Recommended Standard 232;

SCLK: Serial Clock;

SPI: Serial Peripheral Interface;

SS: Slave Select;

USB: Universal Serial Bus.