Università degli Studi di Trieste

DIPARTIMENTO DI INGEGNERIA E ARCHITETTURA
Corso di Laurea in Ingegneria Elettronica e Informatica

Stabilizzazione nel piano di Gough-Stewart

Candidato:
Daniele Facco

Relatore:
Prof. Stefano Marsi

Anno Accademico 2020-2021

Indice

1 Introduzione

2 Tecnologie impiegate

2.1 Piattaforma Arduino

L'intero progetto si basa su una unità di controllo centrale, è quindi fondamentale impiegare un processore con discrete capacità elaborative. Una scelta molto comune, è quella di impiegare schede Arduino che con il vantaggio di essere OpenHardware hanno raggiunto una discreta ubiquità nel settore, anche grazie ad un ambiente di sviluppo completo e ad un sistema di librerie molto supportato dai produttori di componenti. La scheda impiegata è Arduino UNO con un processore ATmega328P, il quale permette di avere una frequenza di clock di $16MHz$, perfettamente in grado di gestire la rapidità di calcolo richiesta dal progetto. Alla scheda impiegata è stato poi montato uno "shield", un componente che rimappa i pin in modo più conveniente per connettere dispositivi quali: servomotori, display LCD, moduli bluetooth e altri, oltre che facilitare l'alimentazione esterna tramite appositi morsetti. I pin impiegati sono divisi in base alla periferica:

- Il controllo dei servomotori, svolto dai pin con capacità PWM ovvero 3, 5, 6, 9, 10 e 11.
- Il controllo del piano resistivo da 12, 13 e A0.
- La lettura del joystick, tramite l'impiego di A5, A6 e 7.

2.2 Servomotori

I servomotori sono l'elemento principale nella realizzazione della piattaforma di Stewart. Sono presenti sei attuatori, controllati in modalità PWM dalla scheda Arduino tramite la libreria `Servo.h`. L'interpretazione del segnale in ingresso è gestita da un apposito chip di controllo che lo trasforma e lo invia a un motore DC brushless, che tramite un apposito gearbox ad alta riduzione fa ruotare l'asse. La rotazione dell'asse è inoltre recepita dal controllore interno tramite un potenziometro, dando quindi la possibilità di aumentare la coppia se il setpoint non è raggiunto. L'intero sistema è riassunto in figura 1. A causa del loro elevato rapporto di riduzione questi componenti possono produrre una coppia elevata[1], questo aspetto non va sottovalutato ed è sempre necessario gestire l'evenienza di uno sforzo durante l'operazione della piattaforma. Il segnale inviato ai servomotori è periodico, di periodo $20ms$ e presenta uno stato on/off a livello logico per

[1] I servomotori impiegati possono sollevare una massa di $3,5kg$ con una manovella di lunghezza nominale $1cm$.

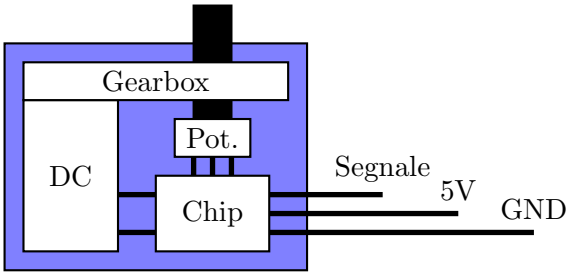


Figura 1: Componenti del servomotore.

un tempo prefissato a seconda della posizione richiesta. In figura 2 è descritto il segnale PWM al variare dell'angolo richiesto. Il controllo di questi elementi è svolto dal processore tramite uno shield che ne facilita l'installazione. Una nota particolare va fatta in merito alla potenza richiesta dai servomotori, nonostante il loro consumo in stato "idle" sia di circa $10mW$, durante il movimento o lo sforzo questo può superare i $500mW$. É quindi necessaria una alimentazione esterna che sia in grado di fornire almeno $1A$ a $5V$, in quanto la scheda alimentata tramite usb dalla presa del computer può ricevere al massimo $500mA$. Una nota particolare va fatta in merito alla scelta di impiegare servomotori rispetto agli stepper, mentre i primi presentano un posizionamento assoluto, determinato dal segnale PWM e un feedback continuo sulla posizione, i secondi si basano su un posizionamento relativo, sarebbe quindi necessaria un'azione di calibrazione ad ogni accensione del dispositivo, che è preferibile evitare.

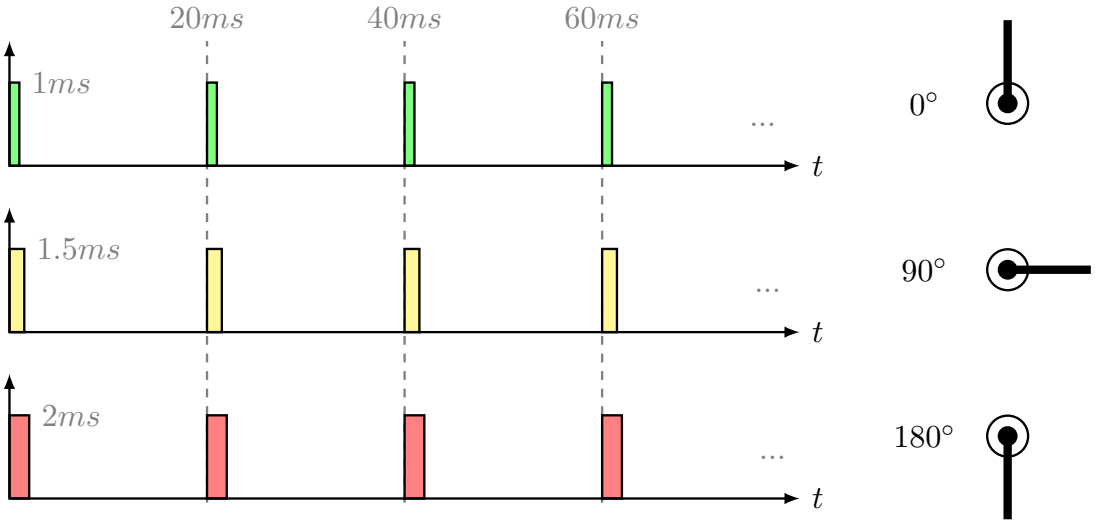


Figura 2: Angolo del servomotore al variare del segnale PWM.

2.3 Piano resistivo

Per attuare correttamente la stabilizzazione del piano è fondamentale l'impiego di un sistema resistente ai disturbi che sia in grado determinare, con la massima precisione possibile, la posizione della sfera da bilanciare. Questo può essere svolto impiegando diverse tecniche come matrici di laser, rilevazione video o per l'appunto tramite l'uso di un piano resistivo. É stato scelto di impiegare un digitalizzatore resistivo per il tracking della pallina in quanto è meno soggetto a disturbi ambientali, in particolare da effetti ottici, rispetto alla matrice laser e alla rilevazione video. Il tracking video avrebbe inoltre richiesto un notevole incremento della complessità del sistema, necessitando di un SOC completo e di software dedicato come OpenCV.

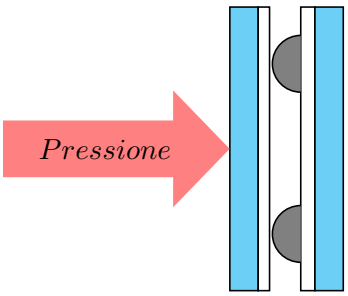


Figura 3: Sezione del digitalizzatore resistivo.

2.3.1 Tipologie e funzionamento di un piano resistivo

I piani resistivi sono generalmente identificati dal numero di connessioni in uscita, questo può generalmente essere quattro, cinque o otto. La versione a otto fili è una semplice evoluzione della variante a quattro e non sarà trattata in dettaglio. Questi dispositivi sono costituiti da due strati, di vetro o plastica, ricoperti da un sottile strato di ossido di indio-stagno[2] e separati da minuscoli spaziatori, come rappresentato in figura 3. La resistenza del conduttore è uniforme su tutta la superficie e alla pressione si forma un contatto tra i due strati che permette un passaggio di corrente. A seconda del punto di pressione si formerà un rapporto di partizione diverso, indicativo della posizione. Nella versione a quattro fili su entrambi i piani si forma un gradiente di potenziale nelle direzioni rispettivamente x e y e ognuno dei due misura il potenziale dell'altro. Nella versione a cinque invece si ha un piano su cui viene alternata la direzione del gradiente mentre l'altro effettua la misura[1]. La schematica del piano resistivo è rappresentata in figura 4, dove le resistenze rosse indicano il potenziale letto dal contatto tra i due strati

[2]Materiale scelto appositamente per le ottime proprietà di trasparenza e conduzione.

resistivi, caratteristico di una specifica posizione. Il valore di tensione letto sarà quindi inviato all'ADC della scheda Arduino, che effettuerà una quantizzazione a $10bit$, restituendo un valore da 0 a 1023, poi convertito in una posizione effettiva. Nella realizzazione del progetto è stata impiegata la variante di piano resistivo a cinque cavi.

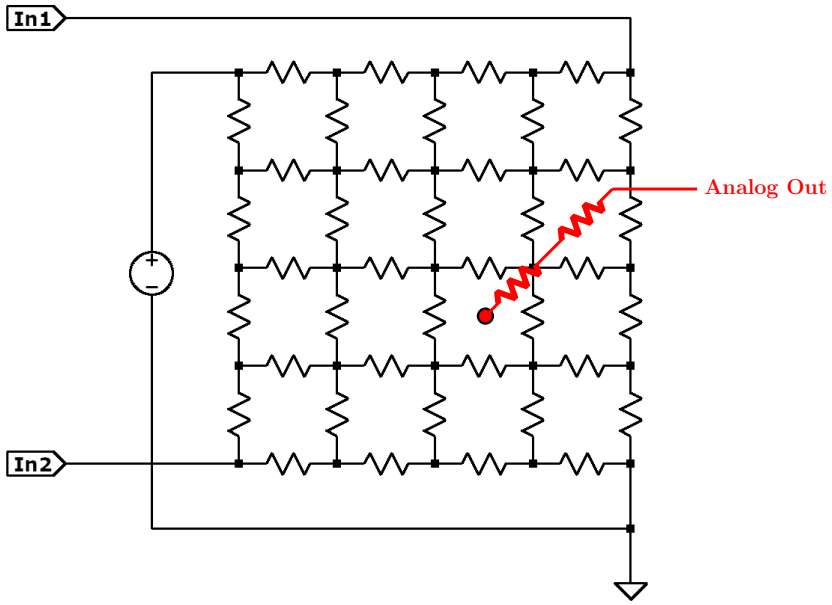


Figura 4: Schema circuitale piano resistivo a cinque fili.

2.4 Ponte ad H

La necessità di impiegare un ponte ad H come circuito di driver è emersa in seguito ad una analisi preliminare delle specifiche del digitalizzatore resistivo. Si è infatti notato, in seguito a una misura di resistenza, che questa fosse di soli 100Ω, quindi non compatibile con le periferiche dell'Arduino UNO, in quanto a $5V$ si andrebbe a richiamare una corrente di $50mA$ mentre la scheda è in grado di fornirne solo 20. L'impiego del ponte ad H ed una sorgente di alimentazione esterna stabilizzata permette di pilotare opportunamente il piano tramite l'uso dispositivi attivi come BJT o MOSFET. Non disponendo del componente finito si è scelto di realizzarlo a partire dai singoli chip in seguito ad un'analisi accurata tramite software SPICE (figura 5) e basandosi sul datasheet del ponte L298N[2]. Nella scelta della tecnologia da impiegare sono stati preferiti i MOSFET per via della caduta di tensione Drain-Source quasi nulla nello stato di conduzione dovuta a una bassissima resistenza interna. I MOSFET scelti sono degli AO3400 e A03401,

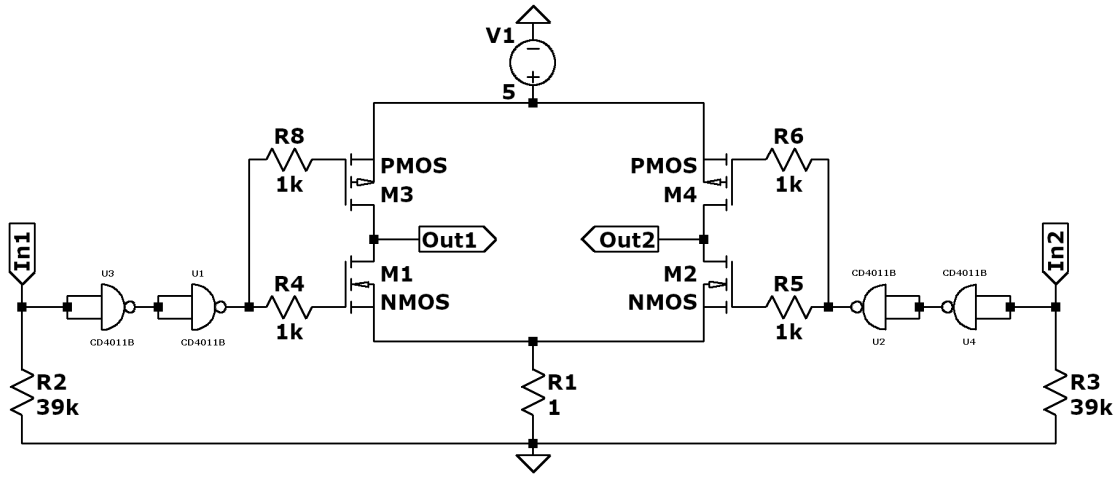


Figura 5: Schema elettrico del circuito di controllo.

rispettivamente nella configurazione n e p in quanto offrono ottime prestazioni relativamente alla I_D e alla R_{DS}[3][4]. Da un'analisi del sistema di controllo richiesto dal piano resistivo si è deciso di rinunciare alla capacità di gestire individualmente i singoli MOSFET favorendo due linee di controllo che possano commutare tra gli stati High/Low. Sono state inoltre impiegate delle porte logiche NAND con lo scopo di migliorare il pilotaggio dei MOSFET da parte di Arduino oltre che per disaccoppiare la logica di controllo dallo stadio di potenza, pratica comune anche nel caso del L298N. Come chip per le porte NAND si è scelto un TC4011BP, questo fornisce in un singolo package quattro porte logiche che sono state configurate come invertenti in cascata, principalmente per una questione di convenienza, nel caso in cui si fosse reso necessario impiegare il segnale negato[5]. Sono state poi inserite diverse resistenze, tra cui:

- Una resistenza da 1Ω verso ground per limitare la corrente di shoot-through[3] durante la commutazione.

- Due resistenze da $39k\Omega$ che dagli ingressi sono collegate verso ground, come pull down, per evitare che questi possano rimanere flottanti vista l'alta impedenza di ingresso della porta NAND.

- Quattro resistenze da $1k\Omega$ sui gate dei MOS, aggiunte in maniera preventiva nel caso si fosse verificato shoot-through per pilotare un eventuale BJT con lo scopo di interdire la formazione di un cammino diretto verso massa.

[3]Conduzione simultanea di entrambi i mosfet su un ramo durante la commutazione che porta alla creazione di un corto diretto tra alimentazione e massa.

Da una successiva analisi all'oscilloscopio si è constatato che gli impulsi di corrente dovuti al cammino diretto sono estremamente brevi, rientrando ampiamente nelle specifiche dei MOSFET impiegati. Le quattro resistenze ai gate sono state quindi lascite anche se non necessarie in quanto non peggiorano in modo sostanziale la costante dei tempi dovuta alla capacità di ingresso dei MOSFET che risulta essere come in equazione 1.

$$\tau = R \cdot C = 1 \times 10^{3} \cdot 630 \times 10^{-12} = 630ns \tag{1}$$

Quindi del tutto ininfluente rispetto alle specifiche del progetto.

2.5 Joystick analogico

Per facilitare e rendere più intuitivo il controllo del sistema è stato impiegato un joystick analogico. Questo dispositivo è costituito da due potenziometri montati ortogonalmente negli assi x e y e da un piccolo switch momentaneo. I segnali analogici generati dai due potenziometri vengono digitalizzati dall'ADC del processore mentre l'interruttore viene letto dall'input digitale. Tramite questi dati sarà possibile avere un controllo diretto del piano ed assegnale dei setpoint in maniera arbitraria.

3 Piattaforma di Gough-Stewart

3.1 Descrizione

La piattaforma di Gough-Stewart è un particolare sistema meccanico, realizzato per la prima volta nel 1954 dall'ingegnere britannico E. Gough e successivamente portata alla notorietà tramite la pubblicazione di un paper nel 1965 da parte di D. Stewart dal titolo "A Platform with Six Degrees of Freedom"[6]. Questo è descritto come un robot esapode, ovvero costituito da una base su cui sono posizionati sei attuatori di vario tipo collegati alla piattaforma superiore mediante giunti snodabili. Il progetto originale prevedeva l'uso di attuatori prismatici, ovvero pistoni lineari che presentano un solo grado di libertà, offrendo numerose proprietà relative alla precisione a alla stabilità. Nel progetto vengono impiegati servomotori che presentano il vantaggio di essere certamente più economici rispetto agli attuatori prismatici ma il sistema risultante assume un grado di complessità superiore. Tramite opportune osservazioni matematiche è possibile risalire a un modello che rende equivalente l'impiego di attuatori rotativi a quello dei lineari, garantendo un range di movimento ottimale.

3.2 Robot seriali e paralleli

È utile enunciare la differenza tra robot di tipo seriale e parallelo per comprendere le successive implicazioni a livello matematico. I robot seriali prevedono una serie di giunti snodabili controllati da attuatori, che spesso assumono le caratteristiche di un arto antropomorfo. Questi tipi di robot prendono spesso il nome di braccio meccanico e trovano ampia applicazione a livello industriale. Nei robot paralleli invece, gli attuatori agiscono tutti sullo stesso elemento tramite giunti indipendenti. La piattaforma di Stewart ne è il principale esempio di cui si trovano applicazioni nella realizzazione di simulatori avanzati di volo, nei test relativi ai veicoli da parte delle case automobilistiche e a livello industriale nella sua configurazione ridotta a tre gradi di libertà, nota come delta-robot, nelle varie catene di montaggio[7]. Entrambi i sistemi presentano sei gradi di libertà, indicati anche come $6DOF$, ma il problema matematico che li caratterizza è fondamentalmente opposto. Lo schema logico dei problemi diretti e inversi è indicato in figura 6, dove $(\vartheta_1, \vartheta_2, ..., \vartheta_n)$ indicano gli angoli dei servomotori e $(x, y, z, \varphi, \vartheta, \psi)$ la posizione dell'end-effector. Se nei robot seriali, noto l'orientamento degli attuatori, è facile risalire alla posizione dell'end effector; lo stesso non si può dire del problema inverso. Nota la posizione dell'end effector è infatti estremamente difficile risalire all'orientamento degli attuatori che permette raggiungere quella determinata locazione. Il problema della piattaforma di Stewart è esattamente l'opposto, è infatti complesso ottenere equazioni che ci permettano di descrivere le coordinate e la rotazione del piano al

Problemi:

$(x, y, z, \varphi, \vartheta, \psi) = S(\vartheta_1, \vartheta_2, ..., \vartheta_n)$ Diretto

$(\vartheta_1, \vartheta_2, ..., \vartheta_n) = S(x, y, z, \varphi, \vartheta, \psi)$ Inverso

Figura 6: Problemi diretti e inversi.

variare dell'angolo dell'attuatore rotativo mentre è più agevole risalire all'angolo dei servomotori a partire dalla posizione nota della piattaforma nello spazio impiegando semplici nozioni di trigonometria. Basti pensare che la soluzione diretta del problema prevede la risoluzione di un sistema di 30 equazioni nonlineari risolvibili in forma chiusa che possono arrivare ad avere fino a un massimo di 40 possibili soluzioni[8]. Queste nozioni di cinematica sono riassunte nella tabella 1. Nello

Problema	Seriale	Parallelo
Diretto	Facile	Difficile
Inverso	Difficile	Facile

Tabella 1: Difficoltà problemi diretti e inversi

svolgimento successivo si è preferito risalire alle equazioni della cinematica inversa della piattaforma di Stewart in quanto permettono un controllo ottimale e allo stesso tempo un minor carico del microprocessore rispetto alla soluzione diretta.

3.3 Analisi piattaforma di Stewart con attuatori rotativi

Il problema della piattaforma di Stewart controllata da attuatori rotativi è fondamentalmente costituito da due sotto-problemi. Il primo relativo alla determinazione delle lunghezze dei vettori che collegano la base alla piattaforma per ogni possibile posizione di quest'ultima, trattato nella sezione 3.4. Il secondo si occupa invece dell'implementazione stessa dei servomotori, per via della complessità di descrizione del sistema biella manovella impiegato, trattato nella sezione 3.5.

3.4 Analisi problema piattaforma

Si definiscono due sistemi di riferimento cartesiani che caratterizzano il sistema, mostrati nella figura 7, uno fisso per la base centrato in O di versori $\hat{i}, \hat{j}, \hat{k}$ e uno variabile per la piattaforma centrato in O' di versori $\hat{i'}, \hat{j'}, \hat{k'}$. Sono note le coordinate in tre dimensioni dei punti degli assi di rotazione dei servomotori (B_i) e dei giunti

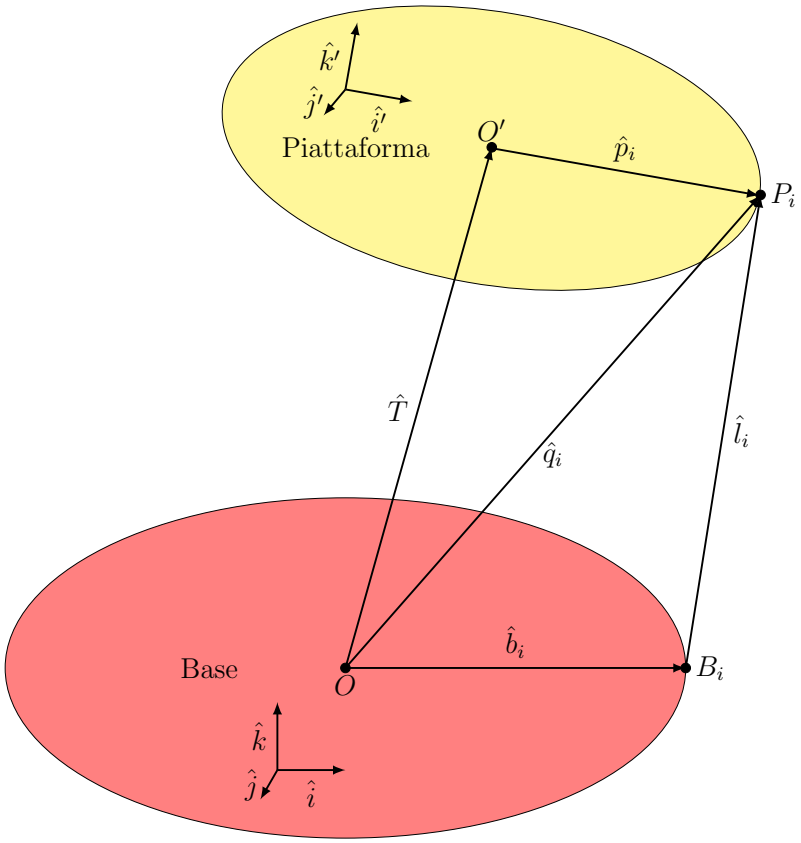


Figura 7: Sistemi di rifermento e vettori nella piattaforma di Stewart.

della piattaforma (P_i) quando questa è in posizione orizzontale a riposo rispetto al sistema di riferimento centrato nella base. Il problema consiste nel definire la posizione dei giunti della piattaforma nello spazio al variare di un set di valori $(x, y, z, \varphi, \vartheta, \psi)$ rispetto al riferimento fisso della base. I valori $(x, y, z, \varphi, \vartheta, \psi)$ precedentemente enunciati, si riferiscono alla posizione e rotazione nello spazio della piattaforma rispetto alla base; (x, y, z) sono i valori del centro della piattaforma mentre $(\varphi, \vartheta, \psi)$ sono le rotazioni, rispettivamente roll, pitch e yaw (rollio, beccheggio e imbardata). Nell'analisi, si considerano separatamente gli effetti traslativi e rotativi.

3.4.1 Analisi traslazione (x, y, z)

Le variazioni in (x, y, z) comportano una semplice traslazione dei punti della piattaforma, questa viene indicata con un vettore $\hat{T}$ che andrà poi a sommarsi alla variazione dovuta a $(\varphi, \vartheta, \psi)$.

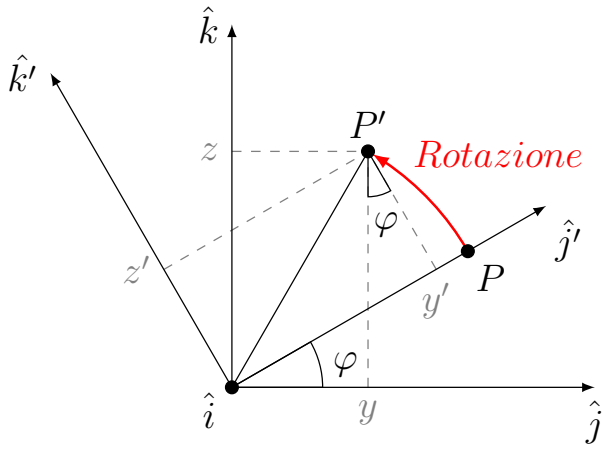


Figura 8: Punto P ruotato di φ gradi rispetto al sistema di riferimento della base.

3.4.2 Analisi rotazione $(\varphi, \vartheta, \psi)$

Per semplificare l'analisi si considerano, separatamente, gli effetti dovuti a queste tre componenti. Si studia il caso di solo rollio $(\varphi, 0, 0)$ per risalire alle coordinate di un punto $P : (x, y, z)$ rispetto alla base in seguito a una rotazione di φ gradi che lo ha traslato nel punto $P' : (x', y', z')$. Questo è rappresentato in figura 8, dove i sistemi cartesiani sovrapposti permettono di ricavare facilmente la relazione che lega le coordinate della piattaforma a quelle della base. Questo risultato è esprimibile matematicamente tramite l'applicazione lineare indicata in (2).

$$\begin{cases} x = x' \\ y = y'cos(\varphi) - z'sin(\varphi) \\ z = y'sin(\varphi) - z'cos(\varphi) \end{cases} \tag{2}$$

Per comodità di calcolo nei successivi passaggi il sistema (2) viene riscritto in forma matriciale.

$$R_x(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\varphi) & -sin(\varphi) \\ 0 & sin(\varphi) & cos(\varphi) \end{bmatrix} \quad : \quad \begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_x(\varphi) \cdot \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \tag{3}$$

Risultati analoghi si ottengono per le rotazioni indipendenti di ϑ e ψ:

$$R_y(\vartheta) = \begin{bmatrix} cos(\vartheta) & 0 & sin(\vartheta) \\ 0 & 1 & 0 \\ -sin(\vartheta) & 0 & cos(\vartheta) \end{bmatrix} \quad : \quad \begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_y(\vartheta) \cdot \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \tag{4}$$

$$R_z(\psi) = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad : \quad \begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_z(\psi) \cdot \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \tag{5}$$

Le singole matrici possono essere viste come applicazioni lineari, si procede quindi nella moltiplicazione di (3), (4) e (5) per ottenere una singola applicazione lineare composta detta matrice di rotazione globale R_g. Notare che nel prodotto tra matrici non vale la proprietà commutativa, bisogna quindi valutare attentamente l'ordine di moltiplicazione, altrimenti si otterrà una rotazione erronea. Nello svolgimento è stato scelto di svolgere le rotazioni attorno agli angoli di Eulero seguendo la convenzione ZYX[9] che prevede appunto la moltiplicazione ordinata delle matrici di rotazione di imbardata, beccheggio e rollio. I passaggi intermedi vengono omessi in quanto ripetono tre volte una semplice operazione di moltiplicazione tra matrici.

$$R_g = R_z(\psi) \cdot R_y(\vartheta) \cdot R_x(\varphi) \tag{6}$$

Il cui risultato è quindi:

$$\begin{bmatrix} cos(\vartheta)cos(\psi) & -cos(\varphi)sin(\psi) + sin(\varphi)sin(\vartheta)cos(\psi) & sin(\varphi)sin(\psi) + cos(\varphi)sin(\vartheta)cos(\psi) \\ cos(\vartheta)sin(\psi) & cos(\varphi)cos(\psi) + sin(\varphi)sin(\vartheta)sin(\psi) & -sin(\varphi)cos(\psi) + cos(\varphi)sin(\vartheta)sin(\psi) \\ -sin(\vartheta) & sin(\varphi)cos(\vartheta) & cos(\varphi)cos(\vartheta) \end{bmatrix} \tag{7}$$

Moltiplicando il vettore $\hat{p_i}$, che congiunge il centro della piattaforma al giunto, per la matrice di rotazione (7) si ottengono le coordinate del punto P_i relative al sistema di riferimento della base.

3.4.3 Conclusione problema piattaforma

Grazie ai risultati ottenuti nelle sezioni 3.4.1 e 3.4.2 si possono definire i vettori $\hat{q_i}$ che descrivono la posizione dei punti P_i rispetto al centro del sistema di riferimento della base O per ogni operazione rototraslativa.

$$\hat{q_i} = \hat{T} + R_g \cdot \hat{p_i} \tag{8}$$

Una volta ottenuto questo vettore, con una semplice operazione vettoriale, si può risalire a $\hat{l_i}$, vettore che descrive la distanza tra l'asse del servomotore e il corrispettivo giunto sulla piattaforma.

$$\hat{l_i} = \hat{T} + R_g \cdot \hat{p_i} - \hat{b_i} \tag{9}$$

Nel caso della piattaforma di Stewart, sei equazioni (9) permettono di descrivere la posizione della piattaforma al variare di $(x, y, z, \varphi, \vartheta, \psi)$. Notare come, nel caso di impiego di attuatori lineari, il problema si risolve assegnando direttamente queste lunghezze per ottenere la posizione desiderata.

3.5 Analisi problema attuatori rotativi

L'impiego di servomotori complica ulteriormente le equazioni, è infatti necessario un numero di variabili superiore per descrivere in modo appropriato il sistema. Ogni servomotore controlla un giunto biella manovella collegato alla piattaforma che regola la lunghezza del vettore $\hat{l}_i$, fondamentale è quindi trovare una relazione tra l'angolo del servomotore e il vettore $\hat{l}_i$. Anche in questo caso la soluzione si ottiene dalla scomposizione del problema in due parti: la determinazione della posizione del giunto biella manovella e la ricerca dell'angolo di rotazione α svolta rispettivamente nelle sezioni 3.5.1 e 3.5.2. A livello di notazione, nella successiva trattazione sono indicati con i pedici bm, b e q rispettivamente il punto del giunto biella-manovella, il punto di aggancio asse-manovella e il giunto sulla piattaforma, come in figura 9.

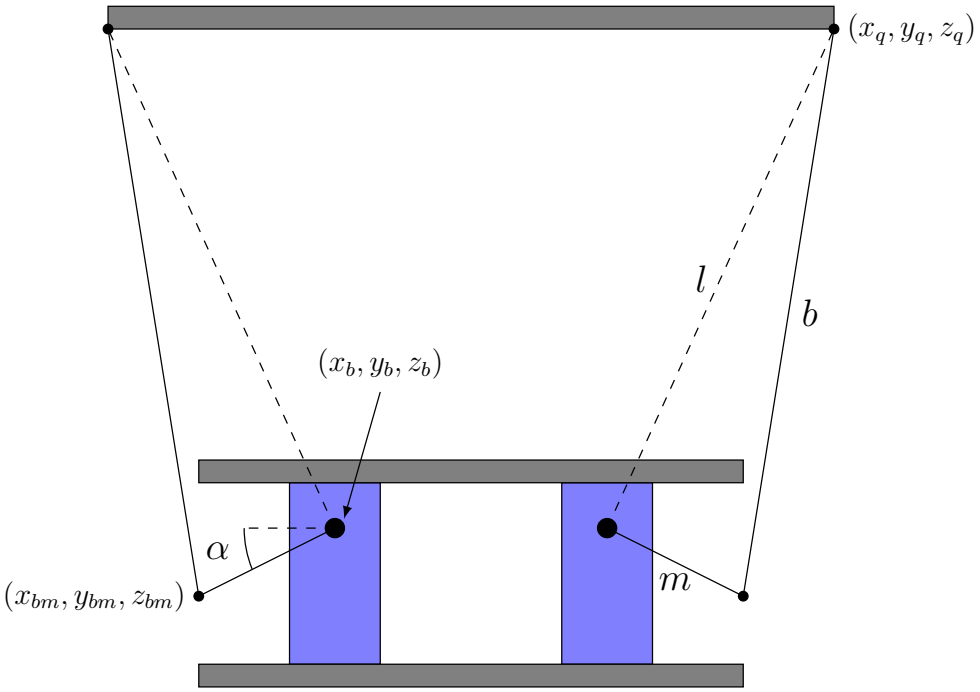


Figura 9: Sistema biella manovella con servomotori in proiezione frontale.

3.5.1 Posizione del giunto biella manovella

Note le lunghezze della biella b e della manovella m, si valutano gli angoli di inclinazione α e β dei servomotori rispetto all'orizzonte e all'asse x, come rappresentato nelle figure 9 e 10, notando come i servomotori siano a due a due specchiati. É

così possibile risalire ai sistemi (10) e (11), necessari per descrivere nello spazio tridimensionale la posizione del giunto biella manovella di ogni motore.

$$\begin{cases} x_{bm} = m \cdot cos(\alpha)cos(\beta) + x_b \\ y_{bm} = m \cdot cos(\alpha)sin(\beta) + y_b \\ z_{bm} = m \cdot sin(\alpha) + z_b \end{cases} \qquad pari \tag{10}$$

$$\begin{cases} x_{bm} = m \cdot cos(\alpha)cos(\pi + \beta) + x_b \\ y_{bm} = m \cdot cos(\alpha)sin(\pi + \beta) + y_b \\ z_{bm} = m \cdot sin(\alpha) + z_b \end{cases} \qquad dispari \tag{11}$$

Applicando le nozioni trigonometriche:

$$cos(\alpha) = -cos(\alpha) \quad cos(\pi + \beta) = -cos(\beta) \quad sin(\pi + \beta) = -sin(\beta)$$

ai sistemi (10) e (11) risulta evidente che questi due sono equivalenti.

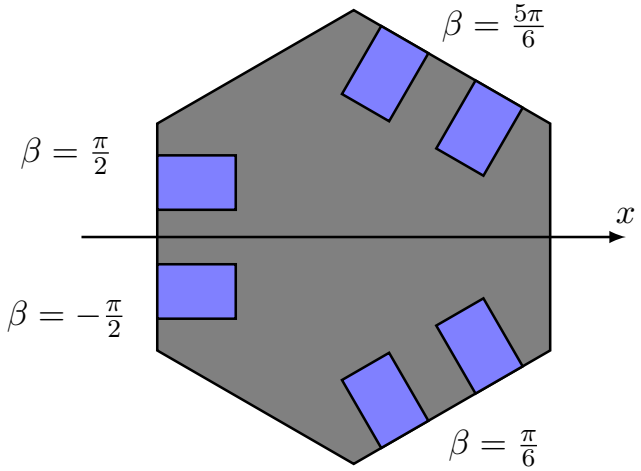


Figura 10: Proiezione verticale della base con i servomotori.

3.5.2 Determinazione angolo α

Per la determinazione dell'angolo α ottimale esistono due tecniche principali. La prima consiste in una ricerca binaria (o dicotomica) del valore che meglio soddisfa le equazioni della posizione. La seconda prevede un approccio matematico più estensivo per determinare il valore **esatto** di α in base ad una l fornita[10][11]. Il primo approccio risulta più semplice dal punto di vista realizzativo ma a suo discapito è poco efficiente, in quanto richiede un numero arbitrario di iterazioni

per raggiungere una determinata precisione di α.[4] Il secondo è invece più difficile da ottenere a causa dei calcoli non del tutto intuitivi ma garantisce di raggiungere la soluzione ottimale con una sola computazione. Per lo svolgimento del progetto è stata impiegata la seconda opzione, che prevede da ricerca di un valore di α che soddisfi le equazioni (12), (13), (14) delle lunghezze, ottenute dall'impiego del teorema di Pitagora in tre dimensioni.

$$\begin{aligned} m^2 &= (x_{bm} - x_b)^2 + (y_{bm} - y_b)^2 + (z_{bm} - z_b)^2 \\ &= (x_{bm}^2 + y_{bm}^2 + z_{bm}^2) + (x_b^2 + y_b^2 + z_b^2) - 2(x_{bm}x_b + y_{bm}y_b + z_{bm}z_b) \end{aligned} \tag{12}$$

$$\begin{aligned} l^2 &= (x_q - x_b)^2 + (y_q - y_b)^2 + (z_q - z_b)^2 \\ &= (x_q^2 + y_q^2 + z_q^2) + (x_b^2 + y_b^2 + z_b^2) - 2(x_q x_b + y_q y_b + z_q z_b) \end{aligned} \tag{13}$$

$$\begin{aligned} b^2 &= (x_q - x_{bm})^2 + (y_q - y_{bm})^2 + (z_q - z_{bm})^2 \\ &= (x_q^2 + y_q^2 + z_q^2) + (x_{bm}^2 + y_{bm}^2 + z_{bm}^2) - 2(x_q x_{bm} + y_q y_{bm} + z_q z_{bm}) \end{aligned} \tag{14}$$

I valori di m^2, l^2 e b^2 sono noti, si sostituiscono le equazioni (12) e (13) in (14), si ottiene:

$$\begin{aligned} l^2 - (b^2 - m^2) =& 2(x_b^2 + y_b^2 + z_b^2) + 2x_{bm}(x_q - x_b) + 2y_{bm}(y_q - y_b) \\ &+ 2z_{bm}(z_q - z_b) - 2(x_q x_b + y_q y_b + z_q z_b) \end{aligned} \tag{15}$$

Si sostituiscono le equazioni dei valori noti x_{bm}, y_{bm} e z_{bm} calcolate in (10) e si svolgono le dovute semplificazioni.

$$\begin{aligned} l^2 - (b^2 - m^2) =& 2(x_b^2 + y_b^2 + z_b^2) + 2[m \cdot cos(\alpha)cos(\beta) + x_b](x_q - x_b) \\ &+ 2[m \cdot cos(\alpha)sin(\beta) + y_b](y_q - y_b) \\ &+ 2[m \cdot sin(\alpha) + z_b](z_q - z_b) - 2(x_q x_b + y_q y_b + z_q z_b) \\ =& 2m \cdot cos(\alpha)cos(\beta)(x_q - x_b) \\ &+ 2m \cdot cos(\alpha)sin(\beta)(y_q - y_b) \\ &+ 2m \cdot sin(\alpha)(z_q - z_b) \\ =& 2m \cdot sin(\alpha)(z_q - z_b) \\ &+ 2m \cdot cos(\alpha)[cos(\beta)(x_q - x_b) + sin(\beta)(y_q - y_b)] \end{aligned} \tag{16}$$

L'equazione (16) è nella forma $L = Mcos(\alpha) + Nsin(\alpha)$ e può essere ulteriormente compattata considerando la formula della somma di segnali sinusoidali di diversa

[4] Mediamente un algoritmo di ricerca binaria assume una complessità *O(log n)*, dove n è direttamente correlato alla precisione del valore ottenuto.

ampiezza, secondo la quale un segnale $Acos(\alpha)+Bsin(\alpha)$ può essere riscritto come $Csin(\alpha+\nu)$, infatti:

$$Csin(\alpha+\nu) = Csin(\alpha)\cdot cos(\nu) + Ccos(\alpha)\cdot sin(\nu)$$

$$\begin{cases} A = Ccos(\nu) \\ B = Csin(\nu) \end{cases} \quad \Rightarrow \quad C = \sqrt{A^2+B^2}, \quad \nu = arctan\left(\frac{B}{A}\right) \tag{17}$$

Applicando il risultato ottenuto in (17) possiamo quindi scrivere:

$$\begin{aligned} L = \sqrt{M^2+N^2}\cdot sin\left(\alpha + arctan\left(\frac{N}{M}\right)\right) \quad &\Rightarrow \\ sin\left(\alpha + arctan\left(\frac{N}{M}\right)\right) = \frac{L}{\sqrt{M^2+N^2}} \quad &\Rightarrow \\ \alpha = arsin\left(\frac{L}{\sqrt{M^2+N^2}}\right) - arctan\left(\frac{N}{M}\right) \quad & \end{aligned} \tag{18}$$

Con rispettivamente:

$$\begin{aligned} L = l^2 - (b^2 - m^2), \quad M = 2m(z_q - z_b), \\ N = 2m[cos(\beta)(x_q - x_b) + sin(\beta)(y_q - y_b)] \end{aligned} \tag{19}$$

Si conclude così la ricerca dell'angolo α.

3.5.3 Controllo servomotori

Al fine di controllare i servomotori in modo ottimale è necessario effettuare alcuni accorgimenti. É necessario definire un'altezza h_0 e un angolo α_0 di riposo dei servomotori, questo è scelto in modo che le manovelle dei servomotori siano orizzontali, quindi $\alpha_0 = 0°$, l'altezza è ricavata sperimentalmente valutando per quale valore di h i servomotori lavorano in modo speculare in seguito all'assegnazione delle varie rotazioni. Un modo di procedere è analizzare a quale altezza, per una assegnazione di rollio di 5°, i servomotori assumono angoli opposti rispetto all'asse x. Siccome i servomotori hanno caratteristiche variabili a seconda del produttore bisogna definire una relazione tra la variazione in μs del segnale PWM e la variazione dell'angolo α in radianti. Notare inoltre come questa relazione non valga quando il servomotore si trova vicino alla massima escursione, per avere una misura accurata analizziamo la relazione tra μs e la rotazione α dalla posizione orizzontale a 45° con il seguente rapporto:

$$\begin{aligned} r &= \frac{\Delta t}{\Delta \alpha}\cdot\frac{360°}{2\pi} \\ &= \frac{375}{45°}\cdot\frac{360°}{2\pi} \quad [\mu s/rad] \end{aligned} \tag{20}$$

Il cui risultato è stato ottenuto sperimentalmente impiegando il valore $\Delta t = 375\mu s$. In questo modo moltiplicando un angolo α per la (20) otteniamo i corrispettivi μs da assegnare al servomotore. Si possono quindi scrivere le equazioni di controllo per i servomotori, distinguendo sempre tra pari e dispari.

$$\begin{cases} w_i = w_i^0 + (\alpha_i - \alpha_0) \cdot r & pari \\ w_i = w_i^0 + (\alpha_i - \alpha_0) \cdot r & dispari \end{cases} \tag{21}$$

Dove w_i^0 indicano le posizioni a riposo dei servomotori in μs. Per migliorare la sicurezza del sistema è stato introdotto un controllo degli angoli assegnati ai servomotori, in modo da evitare possibili conflitti meccanici. Il sistema di controllo analizza tutti i valori di w_i prima che questi siano assegnati ai servomotori, li confronta con un range di valori accettabili e verifica che non siano valori impossibili (`NaN`). Se le condizioni sono rispettate procede assegnando i valori, altrimenti entra in una routine di allarme e blocca il sistema. Il funzionamento globale del sistema è riassunto nello schema a blocchi presente nella sezione 3.6.

3.6 Schema a blocchi

Lo schema logico di figura 11 descrive in modo riassuntivo il controllo della piattaforma di Stewart con attuatori rotativi, nonché gli accorgimenti necessari a rendere il sistema più sicuro in caso di valori di ingresso errati. Si nota come l'assegnazione degli angoli alla piattaforma di Stewart è effettuata in modo sequenziale ma vista la velocità del processore e il tempo di risposta dovuto alla meccanica degli attuatori questa può essere assimilata ad una operazione simultanea. Questa tecnica è preferibile rispetto ad una assegnazione sequenziale in diversi istanti temporali, in quanto facendo lavorare tutti i motori all'unisono si verifica un avvicinamento lineare alla posizione desiderata, senza causare confitti meccanici. La natura parallela del robot, a questo riguardo, garantisce delle tolleranze meccaniche intrinseche ai singoli attuatori, in quanto ognuno può contribuire liberamente alla posizione dell'end effector in modo indipendentemente dagli altri, entro certi limiti, senza che si verifichino conflitti.

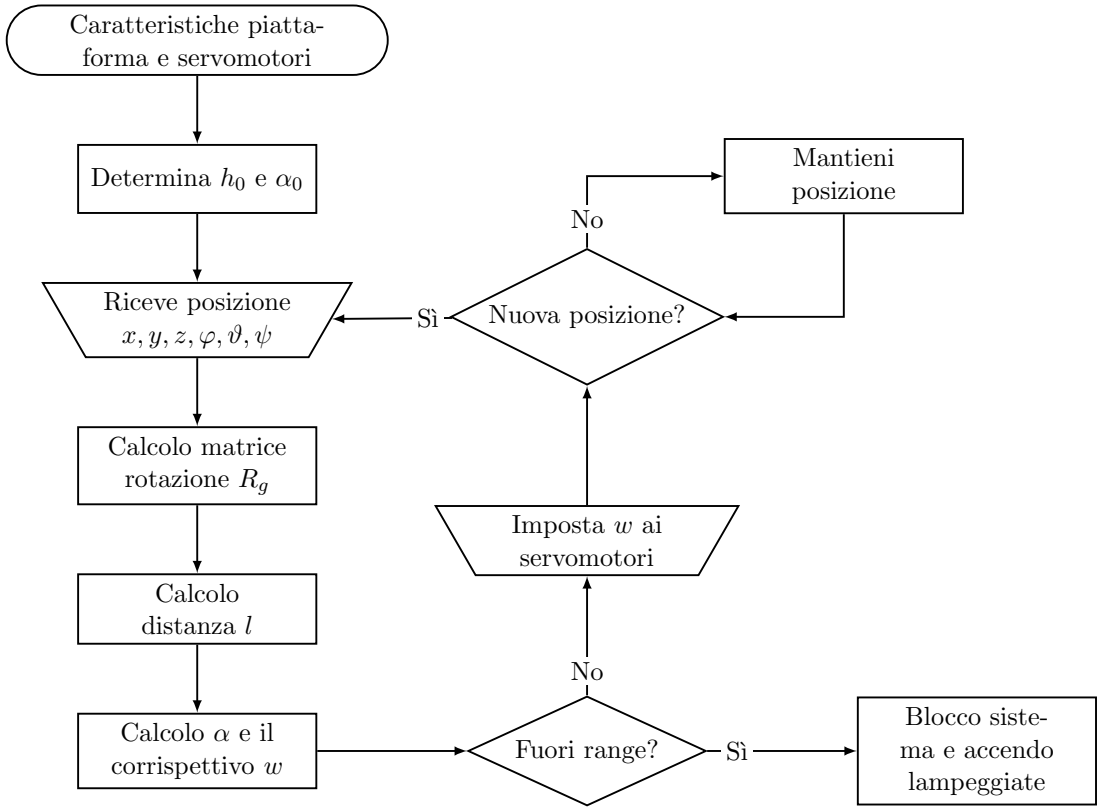


Figura 11: Schema logico controllo della piattaforma di Stewart.

4 Controllo della palla con piano stabilizzato

Il problema del controllo della palla racchiude diversi aspetti elementari che saranno analizzati nella seguente sezione, trattando in modo estensivo del controllore PID e dell'algoritmo di controllo dal punto di vista teorico, rispettivamente nelle sezioni 4.1 e 4.4. Sono inoltre trattate nella sezione 4.2 alcune migliorie pratiche del controllore PID per adattarlo al caso d'uso reale.

4.1 Controllore PID

Il controllore PID è l'elemento chiave per il corretto funzionamento del sistema di stabilizzazione, esso riceve in ingresso la posizione della palla sul piano resistivo e si occupa di fornire opportuni parametri alla piattaforma di Stewart per fare in modo che sia raggiunto un determinato setpoint. Questa azione di controllo è svolta tramite la combinazione lineare di tre contributi fondamentali, il proporzionale, l'integrale e il derivativo, da cui prende il nome. Questi effetti sono gestiti da tre variabili indipendenti K_p, K_i e K_d che ne pesano il contributo.

4.1.1 Equazione del controllore PID

Segue un'analisi delle tre equazioni che permettono il controllo del sistema:

- Il controllore proporzionale ha equazione $u(t) = K_p \cdot e(t)$ e come si può facilmente notare ha lo scopo di rispondere in modo lineare allo scostamento della palla dalla posizione desiderata. Si dice che questa componente agisce al presente.

- Il controllore integrale ha equazione $u(t) = K_i \cdot \int_0^t e(\tau)d\tau$, il suo valore aumenta in relazione al tempo e alla distanza della pallina dal setpoint. Ha quindi lo scopo di annullare l'errore a riposo del sistema. Visto il suo effetto che cosidera la memoria del sistema si dice che lavori al passato.

- Il controllore derivativo ha equazione $u(t) = K_d \cdot \frac{de(t)}{dt}$, si oppone quindi alla variazione dell'errore ed è fondamentale per la stabilità del problema. Si dice che lavori al futuro in quanto previene sistematicamente lo spostamento della pallina.

Questi tre elementi sommati restituiscono l'equazione tempo continua seguente:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau)d\tau + K_d \cdot \frac{de(t)}{dt} \tag{22}$$

Si può quindi procedere rappresentando complessivamente il sistema da controllare e il relativo controllore, che risulta essere come in figura 12, dove i segnali più importanti sono:

- x(t): il setpoint o obbiettivo da raggiungere, impostato manualmente.
- u(t): il segnale d'uscita del sistema, ottenuto dal digitalizzatore resistivo.
- e(t): l'errore, ottenuto dalla differenza tra il setpoint e il segnale in uscita.
- c(t): variabile di controllo in uscita dal controllore PID che evolve opportunamente lo stato del sistema in modo da soddisfare le richieste del setpoint.

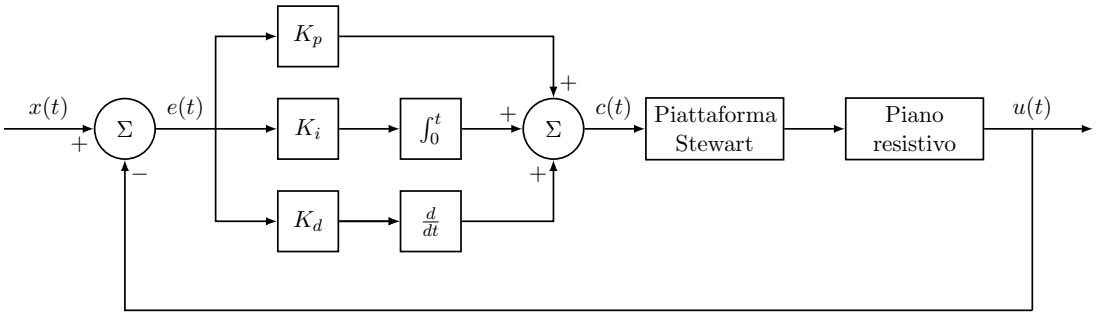


Figura 12: Schema logico controllore PID applicato al problema.

Lavorando con sistemi digitali tempo discreti non è però possibile impiegare la formula (22), sarà necessario procedere ad una discretizzazione, questa viene svolta in modo intuitivo applicando la nozione di differenze finite e considerando $\Delta t = t_k - t_{k-1}$ come periodo di campionamento. Il risultato è la seguente equazione:

$$u(t_k) = K_p \cdot e(t_k) + K_i \cdot \sum_{i=1}^{t_k} e(t_i) \cdot \Delta t + K_d \cdot \frac{e(t_k) - e(t_{k-1})}{\Delta t} \tag{23}$$

4.2 Miglioramenti e ottimizzazioni del controllore PID

Nonostante il PID sia di per sé un ottimo controllore teorico, nella realtà sono necessari degli accorgimenti per evitare comportamenti erronei che possono compromettere la stabilità e l'integrità del sistema. Segue una lista delle migliorie impiegate.

- Filtraggio del segnale: passo fondamentale per ottenere un sistema di controllo performante, dati rumorosi in ingresso rendono il sistema "nervoso" con scatti improvvisi che ne compromettono la stabilità.

- Filtraggio dell'azione derivativa: la derivata, per sua natura, si comporta come un amplificatore di segnali in alta frequenza. Va quindi previsto un filtraggio che corregga questo effetto rendendo il segnale quanto più regolare. Si nota come questo differisca dal filtraggio sul segnale in quanto corregge unicamente una proprietà interna della derivata mentre il filtraggio del segnale in uscita influenza tutte e tre le componenti del PID.

- Derivata sulla misura: consiste nell'eseguire la derivata sul segnle in uscita $u[k]$ invece che sull'errore $e[k]$, utile per evitare un brusco scatto al cambio del setpoint[5] e del tutto analoga alla derivata sull'errore con la sola differenza del segno invertito. Si nota come una derivata sulla misura di questo tipo corrisponde effettivamente alla velocità della palla, in quanto ottenuta dalla derivata prima dello spazio.

- Anti-windup dell'integrale: il termine integrale può tendere a infinito nel caso l'errore non si annulli mantenendo sempre lo stesso segno, questo fenomeno è noto come windup e comporta il fallimento del sistema di controllo se prolungato per un tempo sufficientemente grande. Per evitare questo spiacevole effetto è utile limitare la somma del termine integrale in modo che questo si annulli rapidamente al cambio di segno dell'errore.

- Limitazione dell'uscita: il controllore, non conoscendo i limiti fisici del progetto, può assegnare valori inammissibili al sistema da controllare, il che potrebbe anche risultare in una rottura. Si devono quindi stabilire dei limiti al range di valori assegnabili dal controllore.

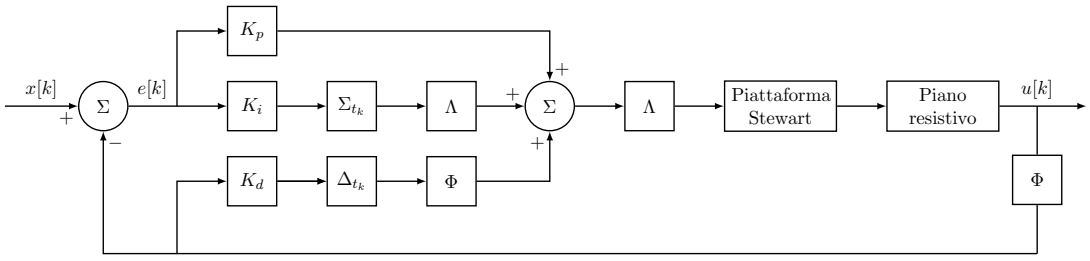


Figura 13: Schema logico controllore PID migliorato.

Questi miglioramenti operativi sono mostrati in figura 13, dove Σt_k indica l'integrale discretizzato e Δt_k la derivata alle differenze finite. I filtri e i limitatori sono poi indicati rispettivamente con le lettere greche Φ e Λ. L'equazione del

[5]Viene visto come uno spostamento molto veloce dalla componente derivativa.

PID migliorato, trascurando limitatori e filtri risulta quindi essere come indicato nell'equazione (24).

$$u(t_k) = K_p \cdot e(t_k) + K_i \cdot \sum_{i=1}^{t_k} e(t_i) \cdot \Delta t + K_d \cdot \frac{u(t_k) - u(t_{k-1})}{\Delta t} \tag{24}$$

4.3 Sistemi di riferimento

A livello di implementazione si nota come sia possibile descrivere la posizione della pallina nel piano secondo due sistemi di riferimento, il cartesiano e il polare. Questi due sono totalmente equivalenti a livello descrittivo e richiedono un controllo di due variabili tramite due PID separati rispettivamente per le variabili (x, y) e (r, φ), come descritto in figura 14. L'unica differenza sostanziale tra i due sta

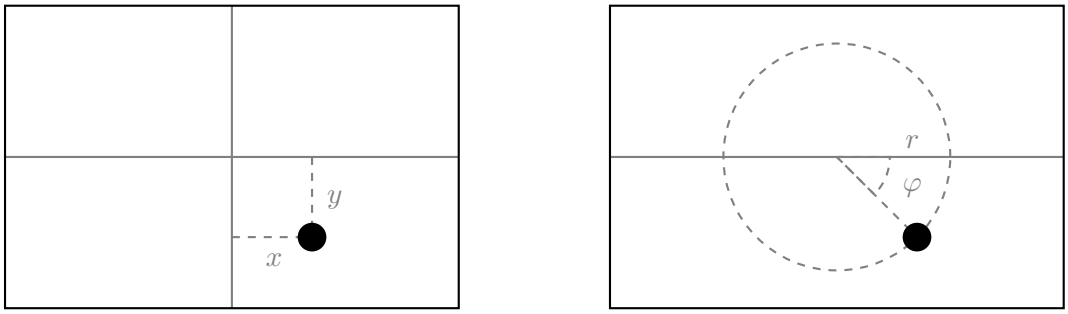


Figura 14: Sistemi di riferimeto cartesiano e polare.

nell'assegnazione dei setpoint, se ad esempio si volesse assegnare una traiettoria circolare, la sua descrizione risulterebbe essere:

- $\begin{cases} x = cos(\tau) \\ y = sin(\tau) \end{cases}$ per il sistema cartesiano.
- $\begin{cases} r = 1 \\ \varphi = \tau \end{cases}$ per il sistema polare.

Dove τ è una variabile generica che permette alla posizione di evolvere, nel caso considerato si è scelto di associarla al tempo moltiplicato per un fattore costante opportuno che ne regola la velocità. Essendo le funzioni periodiche, all'avanzare del tempo il setpoint andrà quindi a muoversi sempre sullo stesso percorso. Si è deciso di impiegare il sistema cartesiano vista la facilità di descrizione di alcune curve, come ad esempio le figure di Lissajous, che costituiscono un ottimo test per valutare le proprietà di velocità e precisione del sistema.

4.4 Algoritmo di controllo

Vista la versatilità di applicazione del controllore PID, l'intero problema di controllo si traduce nella determinazione della posizione della pallina sul piano e lo svolgimento di opportune equazioni per ottenere i parametri richiesti come la velocità e l'accelerazione. Questi sono impiegati rispettivamente nella componente derivativa del PID migliorato e nel filtraggio del segnale stesso. Il comportamento del sistema di controllo si può quindi riassumere nello schema a blocchi di figura 15, dove il blocco di assegnazione dei valori alla piattaforma richiama quello di figura 11.

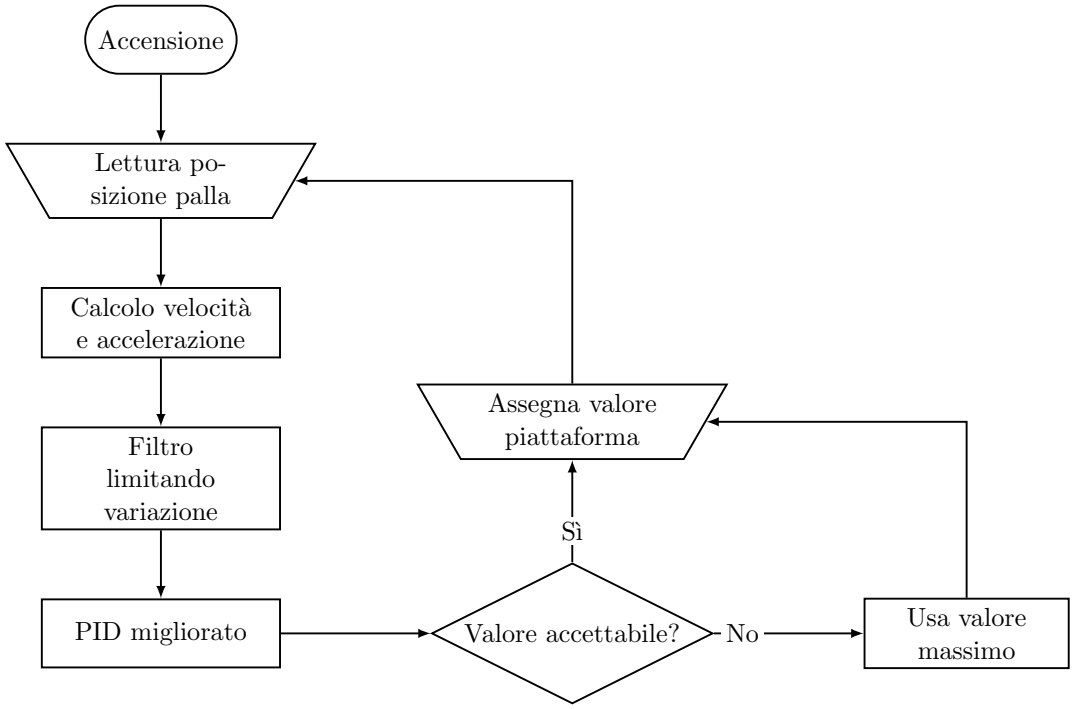


Figura 15: Schema logico sistema di stabilizzazione.

5 Realizzazione pratica

La realizzazione pratica del progetto è stata un processo discretamente complesso, svolto in modo sequenziale e analizzando costantemente le varie criticità. Questo si è svolto principalmente in 3 fasi: l'assemblaggio, la programmazione e la taratura del sistema, rispettivamente descritte in dettaglio nelle sezioni 5.1, 5.2 e 5.3.

5.1 Assemblaggio

Sono state fornite la base con i servomotori, la piattaforma ed i relativi braccetti di collegamento, il piano resistivo e la scheda Arduino UNO dotata di shield. I primi dubbi si sono verificati durante l'assemblaggio in quanto esistono quattro possibili varianti di piattaforma di Stewart con attuatori rotativi, rispettivamente con le manovelle interne o esterne e con i braccetti collegati alla piattaforma sullo stesso lato o su lati opposti. Analizzando la piattaforma si è preferito optare per una disposizione con manovelle esterne e braccetti su lati opposti. Si nota come queste configurazioni siano completamente equivalenti a livello matematico e necessitano unicamente di un ricalcolo della posizione dei punti di aggancio della piattaforma, il che mostra l'adattabilità del modello stesso. Alcuni dei pezzi necessari al completamento dell'assemblaggio, come gli agganci per la piattaforma e il piano resistivo, erano mancanti e dovevano essere realizzati. Disponendo di una stampante 3D si è optato per stampare i pezzi in modo da permettere una prototipazione rapida. Non disponendo dei file relativi alle dimensioni della base e della piattaforma, di compensato tagliato al laser, è stato fondamentale ricostruire fedelmente il modello in ambiene CAD per poi procedere alla modellazione delle parti mancanti (figura **??**).

I modelli sono stati realizzati con il software Fusion360 e stampati in materiale plastico PLA con infill a nido d'ape per avere un buon compromesso tra leggerezza e rigidità. Ultimata la costruzione si è rivolta l'attenzione alle componenti elettroniche, innanzitutto è stato realizzato un connettore apposito per permettere un collegamento stabile tra i servomotori e la scheda Arduino, evitando inoltre possibili intralci al movimento della piattaforma. Notando la fragilità delle uscite del digitalizzatore, tramite un flat-cable flessibile, si è preferito realizzare un apposito aggacio che serva innanzitutto da strain-relief e faciliti la connessione dei cavi dupont per interfacciarsi con Arduino. Come precedentemente esposto nella sezione 2.4, in seguito ad un analisi tramite multimetro della resistenza del digitalizzatore si è notato che un controllo diretto richiamerebbe solo $2V$, più che dimezzando la risoluzione del piano rispetto ai $5V$ nominali. La realizzazione pratica del driver per il piano resistivo richiede l'impiego di un montaggio apposito su basetta di vetronite per ridurre al minimo le capacità parassite e i falsi contatti rispetto ad una basetta sperimentale. Si è quindi svolta un analisi dei tempi di commutazione

con l'oscilloscopio, importantissima nella successiva parte di programmazione, con la quale si è determinato il periodo minimo di campionamento che risulta essere di soli $50\mu s$. Infine è stato implementato senza particolari accorgimenti il joystick in quanto è bastato collegare i cavi direttamente allo shield per avere delle ottime letture.

5.2 Programmazione

La programmazione è senza dubbio la parte principale del progetto e per la sua versatilità ricopre un ruolo fondamentale. Vista la complessità del sistema, lo sviluppo del software è stato diviso in diversi blocchi operativi, ognuno dei quali svolge un ruolo diverso e indipendente dagli altri, che saranno trattati nelle successive sezioni.

5.2.1 Controllo della piattaforma di Stewart

Il controllo della piattaforma di Stewart è stato trattato in modo estensivo dal punto di vista matematico nella sezione 3, la programmazione in questo caso si occupa quindi di riproporre fedelmente queste equazioni in linguaggio `C`, con alcuni accorgimenti specifici relativi all'implementazione del software stesso. Come struttura dati, visto il numero elevato di variabli, si è scelto di impiegare le matrici, vista anche la facilità di implementazione di algoritmi per eseguire le varie operazioni richieste. L'assegnazione della posizione alla piattaforma di Stewart è svolta richiamando la seguente funzione e fornendo in ingresso i sei parametri richiesti per definire univocamente la posizione del piano nello spazio.

```
setPosition(x, y, z, rol, pit, yaw);
```

Una nota particolare va fatta in merito all'algoritmo di controllo delle posizioni irraggiungibili presente nel seguente listato.

```
int emergenza = 0;
  for(int i = 0; i < 6; i++){
    if(constrain(getAmpImp(alfa[i],i),inf,sup) == sup ||
       constrain(getAmpImp(alfa[i],i),inf,sup) == inf ||
       isnan(constrain(getAmpImp(alfa[i],i),inf,sup))){
      emergenza++;
    }
  }
  if(emergenza == 0){
```

```
10      for(int i = 0; i < 6; i++){
11          servo[i].writeMicroseconds(constrain(getAmpImp(alfa[i],i),inf,sup));
12      }
13    }
14    else{
15      Serial.println("Limite raggiunto, arresto motori.");
16      fermoEmergenza();
17    }
```

Ogni computazione della posizione richiesta comporta la generazione di sei valori che saranno poi assegnati come angoli ai servomotori. Prima che questo avvenga, un ciclo `for` si occupa di controllare tutti i valori generati per verificare che questi siano compresi in un limite inferiore e superiore, ma soprattutto verifica che il valore sia effettivamente computabile. Nella formula (18) infatti, nel caso sia assegnato un valore impossibile alla funzione *arcsin*, questa diventa indefinita, espressa in `C` con il simbolo `NaN`[6], è per questo fondamentale l'impiego della funzione `isnan` per verificare se questo è il caso. Se una delle tre condizioni non è rispettata il contatore `emergenza` viene incrementato, utile per capire quanti servomotori abbiano ricevuto un valore errato. A questo punto, se non si sono verificati disguidi, si può procedere con l'assegnazione dei valori impiegando la funzione `writeMicroseconds`, che assegna l'angolo ai servomotori dopo che questo è stato opportunamente convertito in microsecondi dalla funzione `getAmpImp`. Se uno degli angoli non rispetta i vincoli l'assegnazione viene saltata e si passa direttamente alla routine di blocco `fermoEmergenza` che blocca il sistema e accende un LED lampeggiante di segnalazione.

5.2.2 Lettura e filtraggio dati ottenuti dal piano resistivo

La lettura dei segnali provenienti dal piano resistivo è svolta leggendo sequenzialmente lo stato dei pin analogici di Arduino con la seguente funzione:

```
1  void getSense(){
2    digitalWrite(in1,HIGH);
3    digitalWrite(in2,LOW);
4    delayMicroseconds(on);
5    x = (analogRead(sensePin)-100)*(0.34/820);
6
7    digitalWrite(in1,LOW);
8    digitalWrite(in2,HIGH);
```

[6]Not A Number.

```
9     delayMicroseconds(on);
10    y = (analogRead(sensePin)-100)*(0.27/820);
11    lastsense = micros();
12  }
```

Questa prevede la creazione di un gradiente lungo l'asse x, per la durata espressa dalla variabile `on` che permette la stabilizzazione del segnale, la cui durata è ricavata sperimentalmente osservando il tempo di salita all'oscilloscopio come riportato in figura 16. In seguito alla stabilizzazione si procede con il campionamento. Lo

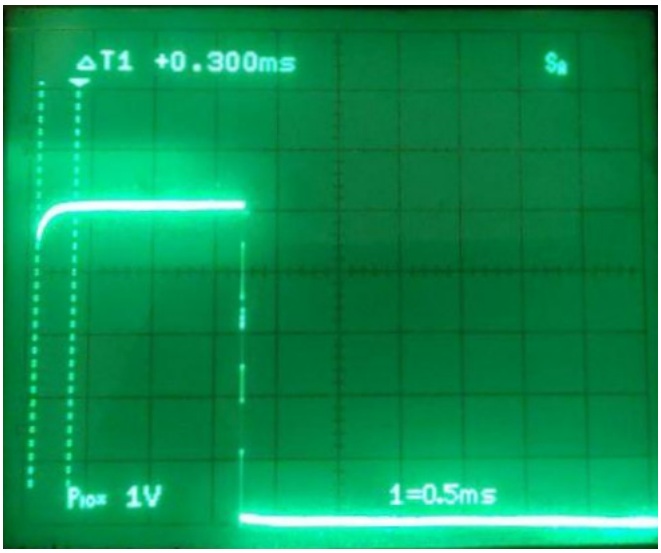


Figura 16: Misura del tempo di salita del segnale.

stesso si ripete in modo del tutto analogo per l'asse y. Viene inoltre effettuata una semplice conversione della posizione da step a metri prima di assegnarlo alla variabile. L'elemento chiave per un buon funzionamento del controllore è l'impiego di dati liberi da rumore e impulsi esterni, ciò detto, i dati provenienti dal piano resistivo si sono però subito mostrati non ideali, presentando un rumore impulsivo estremamente elevato, di ampiezza pari al 50% della massima escursione del segnale utile. Questi impulsi sono principalmente dovuti alla presenza di una lieve zigrinatura sul digitalizzatore resistivo, probabilmente dovuta ai distanziatori tra i due strati conduttivi, che dà alla pallina un moto non del tutto uniforme ma con dei piccoli sobbalzi che ne fanno perdere il contatto. Per evitare questo effetto si è provato a sovrapporre al piano uno strato di materiale liscio, che colmasse le asperità, ma questo si è rivelato di scarso successo. Visto l'obbligo di inserire un filtraggio si è preferita un implementazione software rispetto ad una hardware per via della maggiore flessibilità. Il filtraggio scelto non può comportare un ritardo

eccessivo nel tempo di risposta, si è quindi optato per un filtro costituito da due parti, una lineare e una non lineare, opportunamente pesate per migliorare il segnale. La parte lineare è costituita da una media, su un numero variabile di valori, che ha lo scopo annullare il rumore a media nulla mentre la parte non lineare opera da filtro passa basso, limitando la massima escursione del segnale tra un campionamento e il successivo con lo scopo di eliminare il rumore impulsivo. Al fine di eliminare il rumore impulsivo si sarebbe anche potuto impiegare un filtro mediano ma vista la peculiarità del disturbo, che colpisce un campione si e uno no, questo non sarebbe stato d'aiuto. Nel seguente listato è indicato il processo di filtraggio passa basso della posizione della pallina nel piano.

```
xVel = (x - xOld)/deltaTime;
yVel = (y - yOld)/deltaTime;

float maxVel = 0.4;
if(abs(xVel) > maxVel){
  if(xVel > 0){
    x = xOld + maxVel*(deltaTime);
  }
  else{
    x = xOld - maxVel*(deltaTime);
  }
}
if(abs(yVel) > maxVel){
  if(yVel > 0){
    y = yOld + maxVel*(deltaTime);
  }
  else{
    y = yOld - maxVel*(deltaTime);
  }
}
```

In questo caso la derivata della posizione è la velocità, calcolata rispetto alla posizione precedente e al tempo di ciclo. Si è quindi imposta una velocità massima, che limita la pendenza del grafico della posizione, eliminando così in modo efficace gli impulsi più sostanziali come visibile in figura 17, dove la linea verde indica il segnale originale e la rossa quello ottenuto in seguito al filtraggio. Si nota poi che svolgendo la derivata del segnale filtrato, questa tenderà ad amplificare il piccolo rumore residuo come già precedentemente discusso nella sezione 4.2, si è quindi deciso di ripetere un operazione di filtraggio analoga per la velocità limitando questa volta l'accelerazione massima.

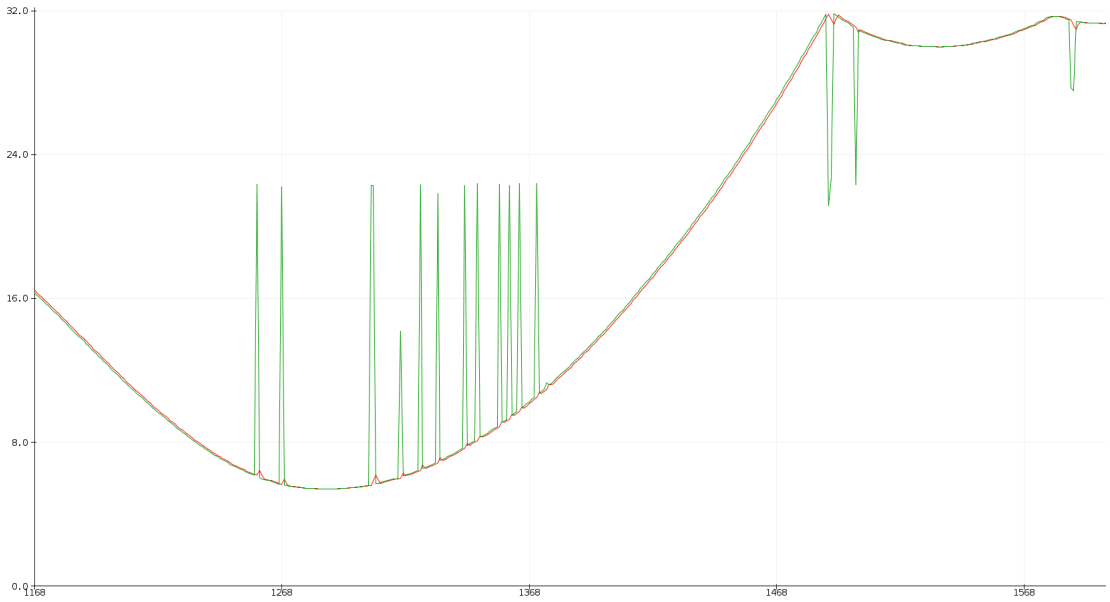


Figura 17: Confronto diretto tra segnale in ingresso e filtrato.

5.2.3 Controllore PID

Il codice del controllore PID si occupa di implementare in `C` le operazioni descritte nel capitolo 4 ed in particolare il diagramma in figura 13. Si nota in particolare come il PID a tutti gli effetti controlli solo due delle sei variabili della piattaforma, rollio e beccheggio, in quanto sufficienti a garantire la stabilizzazione della palla, mentre le altre variabili sono scelte costanti per semplicità. Questo si nota immediatamente osservando il comando di assegnazione della posizione presente nel seguente listato.

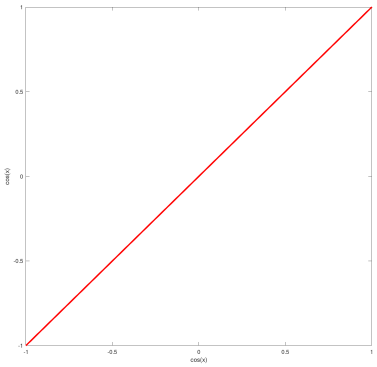
```
1 setPosition(0,0,108,radians(tiltX),radians(tiltY),radians(0));
```

In questo caso le coordinate x e y sono fissate al centro, l'altezza di riferimento è scelta a $108mm$ rispetto alla base per garantire un buon range di movimento, il rollio e il beccheggio sono impartiti dal PID con le variabili `tiltX`, `tiltY` e infine l'imbardata è bloccata a $0°$.

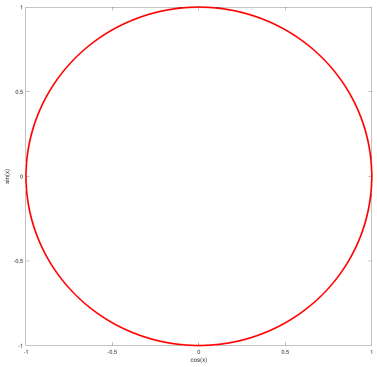
5.2.4 Assegnazione dei comandi

L'assegnazione dei comandi è svolta tramite l'ausilio di un joystick analogico, la presenza di uno switch interno ha inoltre permesso di disporre i comandi su diversi livelli. Nel livello base sono programmate delle analisi statiche, ovvero con un

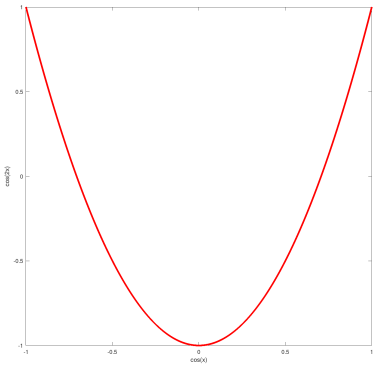
setpoint fisso, il joystick centrato imposta il setpoint al centro del piano, le altre 4 direzioni (joystick in alto, basso, sinistra, destra) hanno lo scopo di spostare il setpoint nella rispettiva direzione di $6cm$. Questi setpoint sono ottimi per valutare la risposta al gradino del sistema che sarà poi fondamentale nella sezione relativa alla taratura. Il secondo livello, raggiungibile premendo l'interruttore, offre la possibilità di eseguire delle analisi dinamiche, in queste il setpoint è in costante movimento e sono utili per mostrare la responsività e la precisione del sistema. Queste analisi dinamiche si incentrano sulla realizzazione di curve di vario tipo da parte della pallina, ottenibili come figure di Lissajous variando il rapporto tra le frequenze e la fase di due segnali sinusoidali assegnati rispettivamente agli assi x e y, come in figura 18. Si nota come sono stati mantenuti rapporti di frequenza relativamente bassi, principalmente per evitare figure troppo complesse e di difficile valutazione. Dal punto di vista della programmazione, l'implementazione di queste



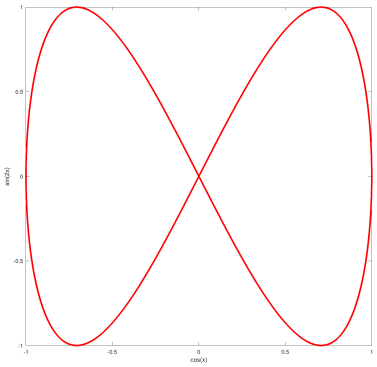
(a) Frequenza 1:1, $\Delta\phi = 0$.



(b) Frequenza 1:1, $\Delta\phi = \frac{\pi}{2}$.



(c) Frequenza 2:1, $\Delta\phi = 0$.



(d) Frequenza 2:1, $\Delta\phi = \frac{\pi}{2}$.

Figura 18: Figure di Lissajous impiegate.

figure è estremamente semplice e svolto in pochissime righe come indicato nel listato seguente.

```
// Retta
if(joyX > 300 && joyX < 700 && joyY < 300 && switchState == false){
  setX=0.173+0.05*cos(pi*millis()/(1500*2));
  setY=0.133+0.05*cos(pi*millis()/(1500*2));
}

// Cerchio
if(joyX < 300 && joyY < 700 && joyY > 300 && switchState == false){
  setX=0.173+0.05*cos(pi*millis()/(1500*2));
  setY=0.133+0.05*sin(pi*millis()/(1500*2));
}

// Parabola
if(joyX > 300 && joyX < 700 && joyY > 700 && switchState == false){
  setX=0.173+0.05*cos(pi*millis()/(1500*2));
  setY=0.133+0.05*cos(2*pi*millis()/(1500*2));
}

// Infinito
if(joyX > 700 && joyY < 700 && joyY > 300 && switchState == false){
  setX=0.173+0.05*cos(pi*millis()/(1500*2));
  setY=0.133+0.05*sin(2*pi*millis()/(1500*2));
}
```

Questa ruotine va a valutare la posizione del joystick tramite le variabili `joyX`, `joyY` e il livello di comando tramite la variabile `switchState`, dove il valore `false` indica l'analisi di tipo dinamico. Quando una di queste condizioni è verificata, il programma assegna il valore del setpoint con le variabili `setX`, `setY` in modo da far muovere la pallina secondo la traiettoria desiderata. Si nota come la realizzazione di queste figure è estremamente importante per valutare il comportamento globale del sistema in quanto forniscono tratti curvi e rettilinei, ad alta e bassa velocità.

5.3 Taratura

La taratura del controllore PID è un passo critico in quanto determina l'intensità e il tipo di risposta alla posizione della pallina nel piano. Come procedura di taratura si è inizialmente tentato di eseguire manualmente il metodo di Ziegler-Nichols[12][13], molto comune nei taratori automatici, che però prevede l'impiego

di una stima accurata del periodo di oscillazione dell'uscita da stabilizzare. Vista la difficoltà di ricavare questo dato si è quindi preferito procedere ad una taratura completamente manuale andando a valutare individualmente gli effetti delle 3 componenti del PID, questo è svolto incrementando le costanti K_p, K_i, K_d e valutando l'effetto risultante sul sistema come in tabella 2. Si è quindi deciso di

Costante	Stabilità	Errore a riposo	Velocità risposta	Overshoot	Tempo stabilizzazione
K_p	Diminuisce	Diminuisce	Aumenta	Aumenta	Non influisce
K_i	Diminuisce	Elimina	Aumenta	Aumenta	Aumenta
K_d	Aumenta	Non influisce	Diminuisce	Diminuisce	Diminuisce

Tabella 2: Effetto dell'incremento delle costanti del controllore PID sui parametri di riferimento del sistema.

procedere nel seguente modo:

1. La prima componente valutata è stata quella del controllore proporzionale, mantenendo le altre due a 0, $[K_p, 0, 0]$, che è stata aumentata fino a garantire una buona escursione al piano in modo da permettere alla palla di raggiungere ogni coordinata.

2. La componente tarata successivamente in modo indipendente, $[0, 0, K_d]$, è stata la derivata, anche vista la forte necessità di stabilizzazione, il cui parametro è stato incrementato fino a rendere la posizione della pallina stabile in ogni punto del piano, senza eccedere per evitate un comportamento "nervoso".

3. Infine è stata regolato il comportamento integrale mantenendo costante il proporzionale e al derivativo, $[\overline{K}_p, K_i, \overline{K}_d]$, che è stato aumentato fino ad annullare l'errore a riposo, permettendo alla pallina di muoversi lentamente fino a raggiungere il setpoint.

Una analisi di fondamentale importanza per valutare questi parametri è stata la risposta allo scalino, ovvero un cambio repentino di setpoint da un valore costante a un altro, che permette di analizzare tutte le componenti della risposta. La combinazione dei tre effetti dà come risultato la risposta allo scalino presente in figura 19. Come si può notare la risposta del sistema è veloce, senza un eccessivo overshoot e un discreto assesatamento del valore sul setpoint.

Figura 19: Risposta allo scalino del sistema lungo l'asse x.

6 Risultati

Vista la complessità del progetto e la diversità dei parametri valutativi i risultati saranno espressi separatamente per la piattaforma di Stewart e per il piano stabilizzato rispettivamente nelle sezioni 6.1 e 6.2.

6.1 Piattaforma di Stewart

Il controllo della piattaforma di Stewart può essere valutato in base alle capacità di movimento permesse dai sei gradi di libertà. I test eseguiti per valutare queste capacità sono stati:

- L'assegnazione di funzioni sinusoidali ai valori di x, y, z in modo da valutare sia l'escursione veritcale, sia la capacità di movimento orizzontale.

- L'assegnazione di angoli compresi tra i $\pm 5°$ alle componenti di rollio e beccheggio per testare il range di inclinazione del piano, importantissimo nella successiva fase di controllo della palla.

- L'assegnazione di una escursione di $\pm 15°$ all'imbardata per vericarne il corretto funzionamento.

Tutti questi test hanno avuto successo e la piattaforma ha dimostrato una ottima fluidità nei movimenti, è stato quindi possibile procedere all'implementazione del piano resistivo.

6.2 Piano stabilizzato

Nella sezione 5.3 è stata esposta la procedura seguita per tarare il sistema e la conseguente risposta al gradino. Questo già di per sé costituisce un ottimo risultato e una dimostrazione delle capacità di controllo ma sono necessare ulteriori test per valutarne a pieno le capacità. Nella sezione 5.2.4 sono state poi indicate le procedure per far seguire al setpoint le curve delle figure di Lissajous, queste sono certamente più interessanti e permettono di valutare la precisione del sistema con varie velocità di movimento del setpoint. Per valutare questi aspetti è stato scritto un semplice script `MATLAB` che legge la posizione della palla da un file CSV prodotto da Arduino e la riporta in un grafico a dispersione (x, y) su cui viene poi sovrapposto il tracciato esatto compiuto dal setpoint. I risultati sono rappresentati in figura **??**. Si nota in particolare come per eseguire queste figure dinamicamente sia necessario modificare sostanzialmente i parametri del PID, riducendo i parametri derivativo e integrativo per avere una maggiore fluidità e velocità di movimento a scapito della precisione.

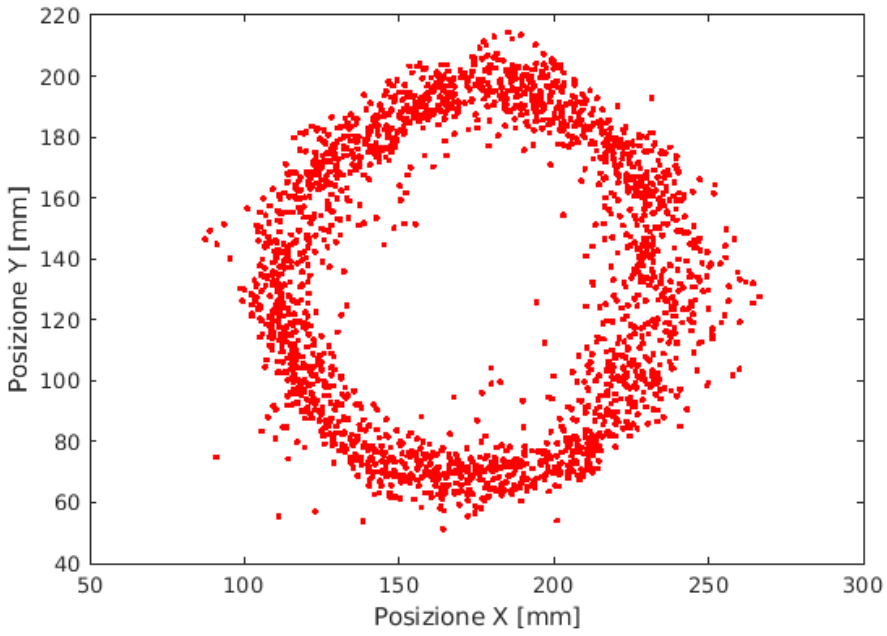


Figura 20: Tracking del setpoint sulla funzione ◯.

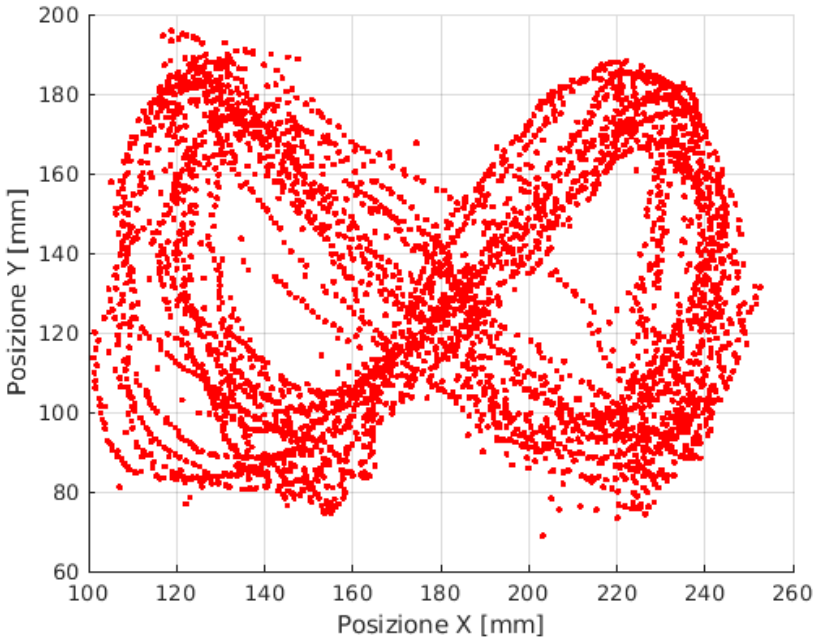


Figura 21: Tracking del setpoint sulla funzione ∞.

7 Conclusioni

Riferimenti bibliografici

[1] Rick Downs. «Using resistive touch screens for human/machine interface». In: *Analog Applications Journal Q* 3 (2005), pp. 5–10.

[2] *Dual Full-Bridge Driver*. L298N. STMicroelectronics.

[3] *N-Channel Enhancement Mode Field Effect Transistor*. AO3400. Alpha & Omega Semiconductors.

[4] *P-Channel Enhancement Mode Field Effect Transistor*. AO3401. Alpha & Omega Semiconductors.

[5] *Toshiba CMOS digital integrated circuit silicon monolithic*. TC4011BP. Toshiba Semiconductor.

[6] Doug Stewart. «A platform with six degrees of freedom». In: *Proceedings of the institution of mechanical engineers* 180.1 (1965), pp. 371–386.

[7] Ilian Bonev. «Delta parallel robot-the story of success». In: *Newsletter, available at http://www. parallelmic. org* (2001).

[8] Bhaskar Dasgupta e TS1739334 Mruthyunjaya. «The Stewart platform manipulator: a review». In: *Mechanism and machine theory* 35.1 (2000), pp. 15–40.

[9] Joan Solà. «Towards Visual Localization, Mapping and Moving Objects Tracking by a Mobile Robot: a Geometric and Probabilistic Approach». In: (feb. 2007).

[10] Lucyna Leniowska e Zbigniew Nawrat. *Postepy inżynierii biomedycznej*. 2013.

[11] Heeseung Bang e Young Sam Lee. «Implementation of a ball and plate control system using sliding mode control». In: *IEEE Access* 6 (2018), pp. 32401–32408.

[12] John G Ziegler, Nathaniel B Nichols et al. «Optimum settings for automatic controllers». In: *trans. ASME* 64.11 (1942).

[13] Narong Aphiratsakun e Nane Otaryan. «Ball on the plate Model based on PID tuning methods». In: *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE. 2016, pp. 1–4.