Robotica con Laboratorio

Marco Caggiano Alessandro Nicolosi [1]

26 febbraio 2010

[1]Università di Roma Tor Vergata - Corso di Laurea Magistrale in Ingegneria dell'Automazione

Indice

Elenco delle tabelle

Elenco delle figure

1 Introduzione

L'obiettivo del progetto è quello di creare un programma che permetta di impartire comandi al robot *SCORBOT-ER* per realizzare un compito specifico. Inizialmente viene fatta un'analisi dell'architettura del sistema, formata dal robot e da una unità di potenza comandata da un *Personal Computer*, seguita poi dall'analisi della struttura del robot e del suo funzionamento. Per la realizzazione del progetto è stata necessaria un'analisi completa della cinematica diretta e della cinematica inversa. Assegnati i valori degli angoli dei giunti, la cinematica diretta realizza un trasformazione dallo *spazio dei giunti* allo *spazio operativo*. L'analisi è stata effettuata secondo la convenzione di *Denavit-Hartenberg*. Sono state poi rilevate le misure effettive del robot e per verificarne la correttezza è stata eseguita una procedura di taratura.

La cinematica inversa realizza la trasformazione opposta, cioè dallo spazio operativo allo spazio dei giunti. Quindi specificando una posa dell'effettore si calcolano i valori degli angoli di giunto corrispondenti, se possibile. Infatti, affinchè la posizione richiesta sia ammissibile è necessario che appartenga allo spazio di lavoro del robot. Sono state poi aggiunte delle condizioni allo spazio di lavoro per evitare che l'effettore si trovi in zone proibite.

Completata l'analisi si è passati all'implementazione delle varie funzioni che realizzano la cinematica e la movimentazione del robot. Il programma è stato scritto in linguaggio C.

Per poter impartire comandi al robot è stato implementato uno *pseudolinguaggio* che contiene una serie di macroistruzioni ad alto livello. La lista di macroistruzioni viene memorizzata in un file di testo che viene interpretato dal programma durante la sua esecuzione.

Tramite l'uso di tutti questi elementi è stato così possibile realizzare il progetto, non prima però di aver creato la struttura fisica che permette al robot di eseguire il lavoro richiesto.

Il progetto riguarda la realizzazione di un temperamatite automatico: il robot prende una matita posta su un supporto, in base allo spessore la tempera nel temperino adatto e poi svuota il cestello del temperino nel cestino.

Per migliorare il funzionamento del robot sono state apportate alcune ottimizzazioni che riguardano il controllo e la precisione. Sono stati individuati poi dei casi particolari che riguardano situazioni in cui la punta della pinza potrebbe sbattere con il piano di lavoro. Per questo sono stati presi alcuni accorgimenti per ovviare a tale problematica.

2 Analisi strutturale

2.1 Architettura del sistema

Il sitema è costituito da un *robot* collegato ad una *unità di potenza* che si occupa di gestire la comunicazione tra il *Personal Computer* e il robot.

L'unità di potenza si interfaccia al PC tramite una scheda commerciale che gestisce gli input/output analogici e digitali. Dall'altro lato invece è collegata direttamente con i motori e gli encoders del robot.

2.2 Descrizione del robot

Lo *SCORBOT-ER* è un robot manipolatore a base fissa formato da un insieme di corpi rigidi (bracci) interconnessi tra di loro per mezzo di articolazioni (giunti).

Un giunto è un'articolazione tra due bracci consecutivi e può essere di tipo *prismatico* o *rotoidale*. Ogni giunto conferisce un singolo grado di libertà alla struttura.

Un *giunto prismatico* realizza un moto relativo di traslazione tra i due bracci, mentre un *giunto rotoidale* realizza un moto relativo di rotazione.

Il robot in questione ha 5 gradi di libertà e sono presenti solo giunti rotoidali.

L'insieme dei bracci e dei giunti costituiscono una *catena cinematica aperta* dove un estremo è collegato alla base e l'altro all'organo terminale, la pinza. Nel robot si individuano una struttura portante, che ne assicura la mobilità, e un organo terminale costituito da una pinza. La *struttura portante* è costituita da 5 giunti denominati come segue: base, shoulder, elbow, pitch, roll.

Il giunto di *base* permette al robot di muoversi sul piano parallelo al piano di lavoro di un angolo di quasi 360 gradi.

I movimenti di *shoulder, elbow, pitch* permettono di muovere i bracci su un piano perpendicolare al piano di lavoro.

Il movimento di *roll* consente una rotazione della pinza intorno all'asse uscente dalla pinza stessa.

Infine la *pinza* posta sull'estremità del robot può essere comandata in apertura e chiusura.

La movemimentazione di ogni giunto è realizzata tramite un motore in corrente continua collegato a un riduttore. Sull'albero dei motori sono montati degli encoders che rilevano la posizione angolare. Inoltre sono presenti dei microswitch che segnano la posizione iniziale del giunto.

2.3 Analisi secondo la rappresentazione di *Denavit-Hartenberg*

Il moto complessivo della struttura è realizzato mediante la composizione di moti elementari di ogni braccio rispetto al precedente, per poter manipolare un oggetto nello spazio, pertanto, è richiesta la descrizione di *posizione* e *orientamento* (posa) dell'organo terminale.

Un corpo rigido è completamente descritto nello spazio in termini della sua posizione e del suo orientamento rispetto a una terna di riferimento.

Per descrivere l'analisi cinematica del robot si utilizza la convenzione di *Denavit-Hartenberg*. Tale convenzione delinea un metodo generale e sistematico per definire posizione e orientamento relativi di due bracci consecutivi; il problema si riconduce

all'individuazione di terne solidali a ciascun braccio e alla determinazione della trasformazione di coordinate che lega le due terne.

Seguendo le regole dettate dalla procedura di *Denavit-Hartenberg* è stata effettuata la seguente analisi della struttura del robot:

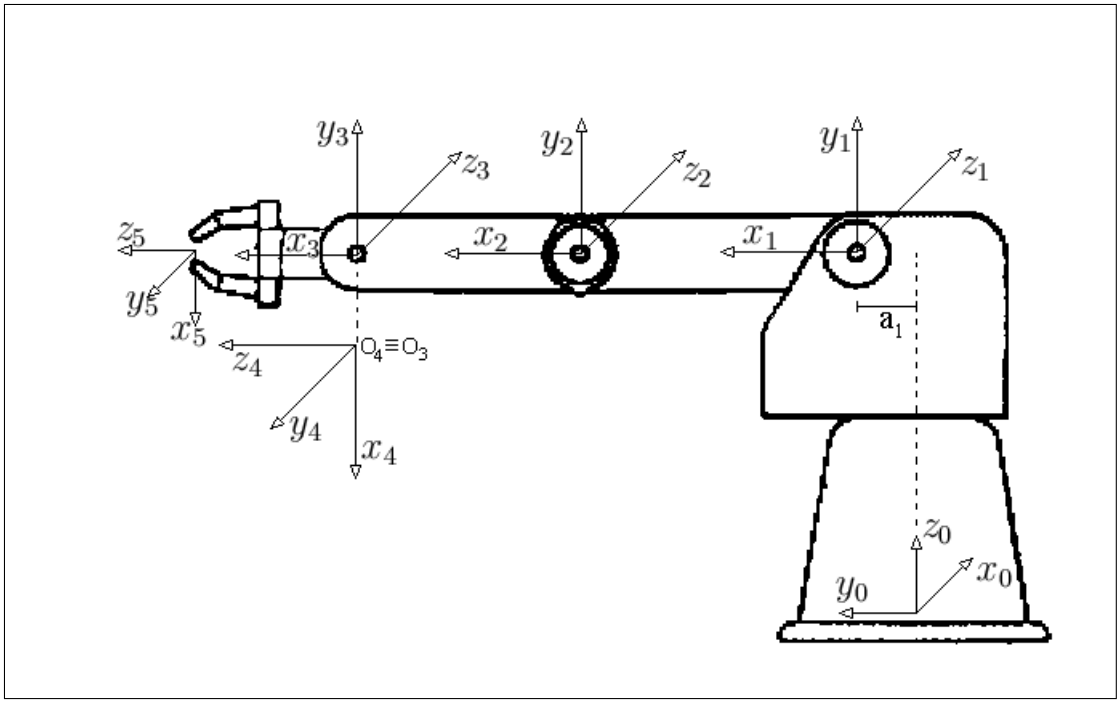

Figura 1: Sistemi di riferimento di Denavit Hartenmberg

- **Scelta degli assi** z

  L'asse z_0, relativo al giunto di *base*, è stato scelto verso l'alto.

  Gli assi z_1, z_2, z_3, che si riferiscono rispettivamente a *shoulder, elbow e pitch*, sono considerati con verso entrante, mentre gli assi z_4, del *roll*, e z_5, dell'*effettore*, sono uscenti dal centro della pinza.

- **Scelta dei centri** O

  Il centro O_0 è stato individuato nell'intersezione tra l'asse del giunto di base e il piano di lavoro.

  I centri O_1, O_2, O_3 sono stati fissati nei centri di rotazione dei rispettivi giunti.

  Il centro O_4 è corrispondente al centro O_3 e il centro O_e è stato posto nell'estremità dell'effettore.

- **Scelta degli assi** x

  Scelto x_0 lungo un lato del piano di lavoro, segue che y_0 è lungo l'altro lato del piano. Gli altri assi x_1,x_2,x_3,x_4,x_5 sono stati scelti come da convenzione lungo la normale comune agli assi z_{i-1} e z_i con verso positivo dal giunto i al giunto $i+1$.

  Gli assi y seguono di conseguenza avendo preso in considerazione terne destrorse ortonormali.

Fissati i riferimenti è stato possibile costruire la tabella riassuntiva dei parametri di giunto (ϑ e d) e dei parametri di link (α e a) come definiti nella convenzione di *Denavit-Hartenberg*.

Si nota la presenza di un parametro a_1 in quanto il centro di rotazione del giunto di *shoulder* non è perfettamente allineato con il centro della rotazione di *base*.

link/giunto	θ	d	a	α
base	θ_1	ℓ_1	a_1	$\pi/2$
shoulder	θ_2	0	a_2	0
elbow	θ_3	0	a_3	0
pitch	θ_4	0	0	$\pi/2$
roll	θ_5	ℓ_5	0	0

Tabella 1: Tabella di *Denavit-Hartenberg*

Di seguito vengono indicati i significati dei simboli utilizzati nella tabella:

- a_i: distanza di O_i dal *punto croce*;
- d_i: distanza da O_{i-1} al *punto croce*;
- α_i: angolo intorno all'asse x_i tra l'asse z_{i-1} e l'asse z_i valutato positivo in senso antiorario;
- ϑ_i: angolo intorno all'asse z_{i-1} tra l'asse x_{i-1} e l'asse x_i valutato positivo in senso antiorario

3 Misure e taratura

Per sviluppare un'analisi della cinematica diretta e inversa coerente è importante conoscere in modo preciso i parametri fisici della struttura del robot.

Per le misure sono stati utilizzati comuni attrezzi di misura: un *metro*, una *squadra* con angolo di 90 gradi e una *livella*.

3.1 Misura delle lunghezze del robot

Determinati e numerati progressivamente i centri di ogni giunto è stata misurata la distanza tra ognuno di essi e quello successivo.

Le misure hanno dato i seguenti risultati:

link/giunto	θ	d	a	α
base	θ_1°	*35 cm*	*2.6 cm*	90°
shoulder	θ_2	*0 cm*	*22.1 cm*	0°
elbow	θ_3	*0 cm*	*22.1 cm*	0°
pitch	θ_4°	*0 cm*	*0 cm*	90°
roll	θ_5	*14.3 cm*	*0 cm*	0°

Tabella 2: Misure dei parametri

3.2 Calcolo dei rapporti di riduzione

Note le lunghezze dei bracci del robot si è passati all'analisi degli angoli di giunto.

Sull'albero di ogni motore è presente un *encoder incrementale*, che, durante il movimento del robot, rileva il numero di passi encoder corrispondenti alla variazione della posizione angolare dell'asse del motore. E' importante precisare che ogni motore è collegato a un *riduttore*, quindi la posizione angolare rilevata è quella a valle del riduttore.

Su ogni giunto è inoltre presente un *microswitch* che segnala una posizione angolare nota. Considerata questa posizione come la posizione iniziale, cioè la posizione in cui per convenzione il valore dell'angolo è pari a zero, dal valore dell'encoder è possibile conoscere il numero di passi encoder percorsi a partire dalla posizione iniziale. Tale valore è proporzionale alla posizione angolare del giunto espressa in radianti (o gradi).

Quindi per conoscere il valore dell'angolo è necessario calcolare questa *costante di proporzionalità*. Per fare ciò abbiamo fatto girare ogni giunto di 90 gradi a partire da una posizione geometrica nota.

Tali misure sono state effettuate facendo lavorare il robot nell'area corrispondente al piano di lavoro, in modo da avere tutto il gioco meccanico da un solo lato, che sarà poi quello in cui andrà a lavorare effettivamente il robot.

Per ottenere un arco di circonferenza di 90 gradi sono state utilizzate diverse tecniche: per l'angolo di base sono state tracciate sul piano di lavoro due rette perpendicolari tra loro, mentre per gli angoli di *elbow*, *shoulder* e *pitch* sono stati posizionati i rispettivi bracci prima in una posizione parallela al piano di lavoro e poi perpendicolare al piano stesso. Tali posizioni sono state individuate mediante l'uso di una *livella*.

Procedimento analogo è stato fatto per l'angolo di *roll* variando la posizione della pinza. Calcolando la differenza tra i due valori forniti dall'encoder, uno all'inizio e uno alla fine dell'arco di circonferenza, è stato possibile ricavare il numero di passi encoder corrispondenti a un angolo di 90 gradi o $\pi/2$ radianti. Il procedimento è stato ripetuto per ogni giunto e sono stati ricavati i seguenti valori:

$$K_1 = -1960, \quad K_2 = 1494, \quad K_3 = -1454, \quad K_4 = 400, \quad K_5 = -370$$

3.3 Taratura

La verifica della precisione di tali misura è stata possibile solo dopo aver sviluppato la cinematica diretta e inversa. Per una corretta taratura del robot è stata utilizzata una tecnica empirica: posto un foglio sul piano di lavoro sono stati fissati dei riferimenti e alcuni punti noti:

punto	x	y	z	φ	θ_5
p1	*15 cm*	*35 cm*	*2 cm*	-90°	0°
p2	*15 cm*	*30 cm*	*2 cm*	-90°	0°
p3	*15 cm*	*25 cm*	*2 cm*	-90°	0°
p4	*20 cm*	*35 cm*	*2 cm*	-90°	0°
p5	*20 cm*	*30 cm*	*2 cm*	-90°	0°
p5	*20 cm*	*25 cm*	*2 cm*	-90°	0°
p4	*25 cm*	*35 cm*	*2 cm*	-90°	0°
p5	*25 cm*	*30 cm*	*2 cm*	-90°	0°
p5	*25 cm*	*25 cm*	*2 cm*	-90°	0°

Tabella 3: Punti di taratura

Assegnati i valori di posizione (x, y, z) dei punti e l'orientamento sono stati ricavati i valori degli angoli dei giunti relativi a ogni punto in modo da portare l'estremità dell'effettore nella posizione desiderata.

Da una prima misura effettuata è stato constatato un errore di circa *1 cm* tra la posizione raggiunta e la posizione desiderata, errore perlopiù relativo all'angolo di *base*. Tale errore dipende soprattutto dal gioco meccanico presente in ogni giunto e dalla conversione da passi encoder al valore dell'angolo.

Apportata una correzione alla costante di proporzionalità di ogni encoder è stato possibile ridurre l'errore a meno di un centimetro.

4 Cinematica diretta

4.1 Calcolo della cinematica diretta

Dalla tabella di Denavit-Hartenberg è possibile ricavare i parametri per il calcolo della cinematica diretta.

A questo punto si è in grado di esprimere la trasformazione di coordinate che lega ogni terna a quella successiva. Questa trasformazione è data da una traslazione di d_i lungo z, una rotazione di ϑ intorno all'asse z originale, una traslazione di a lungo x corrente e una rotazione di α intorno all'asse x corrente.

Ne segue la matrice generica composta dal prodotto di 4 matrici

$$ {}^{i-1}T_i = T_{z,\theta_i} T_{tr,[00d_i]} T_{tr,[a_i00]} T_{x,\alpha_i} \tag{4.1} $$

Quindi è possibile ottenere seguenti le matrici di trasformazione omogenee:

$$ {}^{0}T_1 = \left(\begin{array}{ccc|c} \cos(\theta_1^*) & 0 & \sin(\theta_1^*) & a_1\cos(\theta_1^*) \\ \sin(\theta_1^*) & 0 & -\cos(\theta_1^*) & a_1\sin(\theta_1^*) \\ 0 & 1 & 0 & \ell_1 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) $$

$$ {}^{1}T_2 = \left(\begin{array}{ccc|c} \cos(\theta_2) & -\sin(\theta_2) & 0 & a_2\cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & a_2\sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) $$

$$ {}^{2}T_3 = \left(\begin{array}{ccc|c} \cos(\theta_3) & -\sin(\theta_3) & 0 & a_3\cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & a_3\sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \tag{4.2} $$

$$ {}^{3}T_4 = \left(\begin{array}{ccc|c} \cos(\theta_4^*) & 0 & \sin(\theta_4^*) & 0 \\ \sin(\theta_4^*) & 0 & -\cos(\theta_4^*) & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) $$

$$ {}^{4}T_e = \left(\begin{array}{ccc|c} \cos(\theta_5) & -\sin(\theta_5) & 0 & 0 \\ \sin(\theta_5) & \cos(\theta_5) & 0 & 0 \\ 0 & 0 & 1 & \ell_5 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) $$

dove:

$$ \theta_1^* = \theta_1 + 90^\circ \tag{4.3a} $$

$$ \theta_4^* = \theta_4 + 90^\circ \tag{4.3b} $$

Svolgendo in ordine i prodotti tra le matrici si ottiene la matrice di trasformazione omogenea finale:

$$ {}^{0}T_e = {}^{0}T_1\,{}^{1}T_2\,{}^{2}T_3\,{}^{3}T_4\,{}^{4}T_e \tag{4.4} $$

L'ultima colonna di questa matrice esprime la posizione dell'effettore rispetto al sistema di riferimento fisso:

$$x_e = -\sin\theta_1 * (d_4\cos(\theta_2+\theta_3+\theta_4) + d_3\cos(\theta_2+\theta_3) + d_2\cos\theta_1 + a_1) \tag{4.5a}$$

$$y_e = \cos\theta_1 * (d_4\cos(\theta_2+\theta_3+\theta_4) + d_3\cos(\theta_2+\theta_3) + d_2\cos\theta_1 + a_1) \tag{4.5b}$$

$$z_e = d_4\sin(\theta_2+\theta_3+\theta_4) + d_3\sin(\theta_2+\theta_3) + d_2\sin\theta_1 + d_1 \tag{4.5c}$$

5 Cinematica inversa

L'equazione *cinematica diretta* consente di definire le relazioni funzionali esistenti tra le variabili di giunto e la posa dell'organo terminale.

La *cinematica inversa*, invece, permette di determinare le variabili di giunto una volta assegnata la posizione della pinza e l'orientamento.

La risoluzione di tale problema è molto importante perchè consente di tradurre le specifiche di moto, assegnate nello spazio operativo, nei moti corrispondenti nello spazio dei giunti. In questo modo con opportune trasformazioni, è possibile comandare i motori del robot in modo tale che l'estremità della pinza raggiunga la posizione e l'orientamento desiderati.

E' doveroso però considerare alcuni inconvenienti: mentre con l'equazione cinematica diretta è possibile determinare la posa della pinza in maniera univoca, questo non vale nel calcolo della cinematica inversa. Infatti, assegnati posizione e orientamento, non è sempre possibile trovare una soluzione analitica in forma chiusa e si possono avere soluzioni multiple, infinite o non ammissibili.

5.1 Analisi della cinematica inversa

Assegnata una posa per la pinza in termini di posizione (x_e, y_e, z_e) e orientamento (ϕ, ψ) si vogliono determinare gli angoli $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$.

La postura del manipolatore può essere espressa in forma compatta mediante il vettore (5 x 1)

$$\mathsf{E} = \begin{bmatrix} \mathbf{p_e} \\ \phi \\ \psi \end{bmatrix}, \qquad \mathbf{p_e} = \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} \tag{5.1}$$

La determinazione di una soluzione in forma chiusa richiede intuizione algebrica per individuare le equazioni significative e considerazioni geometriche per individuare i punti significativi della struttura del robot.

L'angolo di *base* può essere immediatamente calcolato dalla conoscenza di x_e e y_e tramite la funzione ATAN2:

$$\theta_1 = \mathrm{ATAN2}(y_e, x_e) \tag{5.2}$$

in particolare:

$$\sin\theta_1 = \frac{\sqrt{{x_e}^2 + {y_e}^2}}{y_e} \tag{5.3a}$$

$$\cos\theta_1 = \frac{\sqrt{{x_e}^2 + {y_e}^2}}{x_e} \tag{5.3b}$$

Noto l'angolo di *base* è possibile passare al calcolo dei restanti angoli.

Individuato il punto P_W come il centro del pitch, dalla conoscenza della posizione dell'effettore è possibile ricavare univocamente le coordinate di questo punto:

$$P_W := \begin{pmatrix} p_{Wx} \\ p_{Wy} \\ p_{Wz} \end{pmatrix}$$

$$p_{Wx} = x_e - d_5 \cos\phi \sin\theta_1 \tag{5.4a}$$
$$p_{Wy} = y_e - d_5 \cos\phi \cos\theta_1 \tag{5.4b}$$
$$p_{Wz} = z_e + d_5 \sin\phi \tag{5.4c}$$

Le coordinate del punto P_W sono inoltre date dalla quarta colonna della matrice di trasformazione ${}^0T_4$, ma sono ricavabili anche geometricamente:

$$p_{Wx} = \sin\theta_1(a_1 + a_2\cos\theta_2 + a_3\cos(\theta_2 + \theta_3)) \tag{5.5a}$$
$$p_{Wy} = \cos\theta_1(a_1 + a_2\cos\theta_2 + a_3\cos(\theta_2 + \theta_3)) \tag{5.5b}$$
$$p_{Wz} = d_1 + a_2\sin\theta_2 + a_3\sin(\theta_2 + \theta_3) \tag{5.5c}$$

Eguagliando le espressioni (5.4) e (5.5) si ottiene:

$$x_e - d_5\cos\phi\sin\theta_1 = \sin\theta_1(a_1 + a_2\cos\theta_2 + a_3\cos(\theta_2 + \theta_3)) \tag{5.6a}$$
$$y_e - d_5\cos\phi\cos\theta_1 = \cos\theta_1(a_1 + a_2\cos\theta_2 + a_3\cos(\theta_2 + \theta_3)) \tag{5.6b}$$
$$z_e + d_5\sin\phi = d_1 + a_2\sin\theta_2 + a_3\sin(\theta_2 + \theta_3) \tag{5.6c}$$

Definiamo X, Y, Z come segue:

$$X = x_e - d_5\cos\phi\sin\theta_1 - a_1\sin\theta_1 \tag{5.7a}$$
$$Y = y_e - d_5\cos\phi\cos\theta_1 - a_1\cos\theta_1 \tag{5.7b}$$
$$Z = z_e + d_5\sin\phi - d_1 \tag{5.7c}$$

in questo modo, quadrando primo membro e secondo membro delle equazioni (5.6) e sommandole, dopo opportune semplificazioni, si ottiene:

$$X^2 + Y^2 + Z^2 = {a_2}^2 + {a_3}^2 + 2a_2a_3\cos\theta_3 \tag{5.8}$$

da cui è possibile ricavare il coseno dell'angolo θ_3

$$\cos\theta_3 = \frac{X^2 + Y^2 + Z^2 - {a_2}^2 + {a_3}^2}{2a_2a_3} \tag{5.9}$$

e quindi anche il seno mediante la formula:

$$\sin\theta_3 = \pm\sqrt{1 - \cos{\theta_3}^2} \tag{5.10}$$

Scegliendo opportunamente il segno + o il segno - si fa lavorare il robot a gomito alto o a gomito basso.

A questo punto il valore dell'angolo di *elbow* è calcolabile con l'utilizzo della funzione ATAN2 in questo modo:

$$\theta_3 = \text{ATAN2}(\sin\theta_3, \cos\theta_3) \tag{5.11}$$

Noti θ_1 e θ_3 si può procedere al calcolo dell'angolo di *shoulder*.

Riprendendo le equazioni (5.6) e avendo definito X, Y, Z come nelle (5.7), risulta:

$$X = \sin\theta_1(a_2\cos\theta_2 + a_3\cos(\theta_2+\theta_3)) \tag{5.12a}$$
$$Y = \cos\theta_1(a_2\cos\theta_2 + a_3\cos(\theta_2+\theta_3)) \tag{5.12b}$$
$$Z = a_2\sin\theta_2 + a_3\sin(\theta_2+\theta_3) \tag{5.12c}$$

Quadrando e sommando il primo e il secondo membro delle (5.12a) e (5.12b) si ottiene il seguente sistema:

$$\begin{cases} \sqrt{X^2+Y^2} &= a_2\cos\theta_2 + a_3\cos(\theta_2+\theta_3) \\ &= a_2\cos\theta_2 - a_3\sin\theta_2\sin\theta_3 + a_3\cos\theta_2\cos\theta_3 \\ Z &= a_2\sin\theta_2 + a_3\sin(\theta_2+\theta_3) \\ &= a_2\sin\theta_2 + a_3\sin\theta_2\cos\theta_3 + a_3\cos\theta_2\sin\theta_3 \end{cases} \tag{5.13}$$

che può essere espresso in forma matriciale come segue:

$$\begin{bmatrix} \sqrt{X^2+Y^2} \\ Z \end{bmatrix} = \begin{bmatrix} a_2 + a_3\cos\theta_3 & -a_3\sin\theta_3 \\ a_3\sin\theta_3 & a_2 + a_3\cos\theta_3 \end{bmatrix} \begin{bmatrix} \cos\theta_2 \\ \sin\theta_2 \end{bmatrix} \tag{5.14}$$

Il sistema (5.14) è costituito da 2 equazioni in 2 incognite che può essere immediatamente risolto in modo da ottenere i seguenti risultati:

$$\begin{cases} \cos\theta_2 = \frac{Y^2+Z^2-{d_2}^2-{d_3}^2}{2d_2d_3} \\ \sin\theta_2 = \pm\sqrt{1-\cos{\theta_2}^2} \end{cases} \tag{5.15}$$

Dalle equazioni (5.15) è possibile così ricavare il valore dell'angolo θ_2:

$$\theta_2 = \text{ATAN2}(\sin\theta_2, \cos\theta_2) \tag{5.16}$$

Calcolati gli angoli di *base*, *shoulder* e *elbow*, l'angolo di *pitch* può essere ricavato direttamente dall'orientamento in quanto:

$$\theta_4 = \phi - \theta_2 - \theta_3 \tag{5.17}$$

Infine l'angolo di *roll* è espresso direttamente dall'orientamento ψ

$$\theta_5 = \psi \tag{5.18}$$

Sono così state ottenute tutte le 5 variabili di giunto in modo che assegnati questi valori agli angoli dei giunti del robot la posa dell'effettore abbia la posizione e orientamento desiderati.

5.2 Spazio di lavoro

La geometria del robot e i fine-corsa meccanici imposti sui giunti delimitano una regione detta *spazio di lavoro*.

Si definisce *spazio di lavoro* di un manipolatore la regione descritta dall'origine della terna utensile quando ai giunti del manipolatore si fanno eseguire tutti i moti possibili. Sovente si usa distinguere tra spazio di lavoro *raggiungibile* e spazio di *destrezza* intendendo con quest'ultimo la regione che l'effettore può raggiungere con almeno un orientamento.

Nel caso del robot in questione lo spazio di lavoro *effettivo* è un sotto insieme dello spazio operativo *teorico*, in quanto vanno presi in considerazione altri limiti geometrici dovuti alla presenza del cilindro di base del robot.

Inoltre è stata delimitata un'area di lavoro in cui si vuole far lavorare il robot e sono state fatte alcune considerazioni sui giunti in modo da ricavare degli insiemi di ammissibilità per ogni angolo.

5.2.1 Spazio di lavoro teorico

Lo spazio di lavoro teorico del robot può essere calcolato prendendo in considerazione l'angolo di *elbow*. Infatti prendendo il coseno di θ_3, espresso come nella formula (5.9), e sapendo che deve essere compreso tra -1 e 1, possiamo ricavare la seguente disuguaglianza:

$$(a_2 - a_3)^2 \leq X^2 + Y^2 + Z^2 \leq (a_2 + a_3)^2 \tag{5.19}$$

che, tenendo in considerazione le espressioni di X, Y, Z della (5.7), può essere riscritta in questo modo:

$$C + (a_2 - a_3)^2 \leq |W|^2 \leq (a_2 + a_3)^2 + C \tag{5.20}$$

dove

$$C = 2a_1(p_{Wx} \sin\theta_1 + p_{Wy} \cos\theta_1) + 2p_{Wz}d_1, \qquad |W|^2 = {p_{Wx}}^2 + {p_{Wy}}^2 + {p_{Wz}}^2 \tag{5.21}$$

Quindi la regione che costituisce lo spazio di lavoro del robot ha una forma toroidale con una cava interna, che può essere immaginata come una corona circolare centrata nell'origine del sistema di riferimento fisso e traslata di C, che viene fatta poi ruotare seguendo l'angolo θ_1.

5.2.2 Spazio di lavoro effettivo

Il robot presenta nella sua struttura un cilindro di altezza H e raggio R. Per evitare che l'effettore possa collidere con tale cilindro è necessario sottrarre questa regione dallo spazio di lavoro precedentemente calcolato. Per fare ciò è stata imposta la seguente condizione:

$${x_e}^2 + {y_e}^2 > R^2, \qquad \forall z_e \leq H \tag{5.22}$$

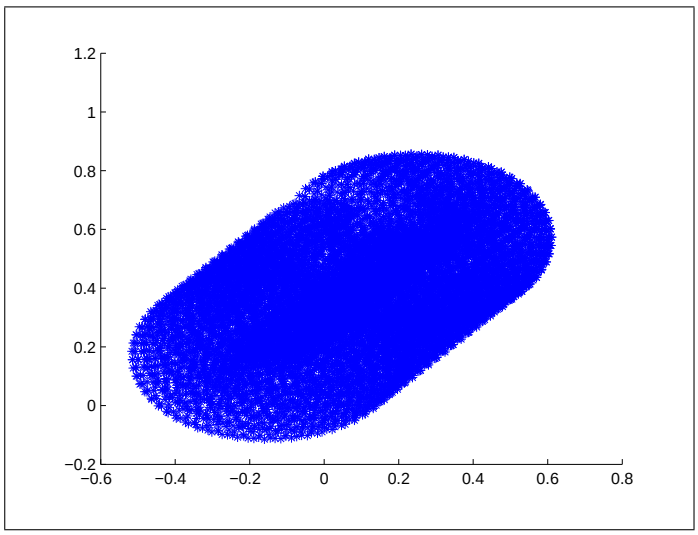

Figura 2: Sezione dello spazio di lavoro effettivo sul piano (y,z)

5.2.3 Area di lavoro

Per evitare che il robot si trovi a lavorare in una zona non consentita, viene definita un'area di lavoro corrispondente a un parallelepipedo che ha come base il piano di lavoro:

$$\begin{cases} -10 \leq x_e \leq 30 \\ -10 \leq y_e \leq 45 \\ z_e > 0 \end{cases} \tag{5.23}$$

5.2.4 Angoli critici

Durante i movimenti del robot sono stati individuati degli angoli critici, che sono i valori minimi e massimi che gli angoli di ogni giunto possono assumere. Questi valori sono però relativi alla posizione del robot e quindi sono stati considerati soltanto i casi migliori di tali posizioni supponendo che l'effettore sia situato nell'area di lavoro e nello spazio di lavoro. Questo non vuole essere un modo per evitare tutte le posizioni critiche del robot ma soltanto una limitazione di esse. Dall'analisi effettuata sono emersi i seguenti valori espressi direttamente in passi encoder:

angolo	min *passi enc*	max *passi enc*
base	-3621	2957
shoulder	-2385	227
elbow	-2500	2350
pitch	-550	900

Tabella 4: Angoli critici

L'angolo di *roll* non è stato limitato in quanto può effettuare sempre una rotazione completa senza alcun vincolo.

6 Implementazione

Dopo aver effettuato l'analisi della cinematica diretta e della cinematica inversa, si è passati all'implementazione delle rispettive funzioni. Dati i valori dei cinque angoli di giunto, la funzione `cin_diretta` restituisce la posizione dell'effettore. Invece, la funzione `cin_inversa` calcola i i valori degli angoli corrispondenti alla posa assegnata. L'appartenenza della posa allo spazio di lavoro viene controllata dalla funzione `verifica_range`. Infine l'effettiva movimentazione del robot viene realizzata mediante la funzione `ROBOTMoveAll`. Queste quattro funzioni vengono analizzate nel dettaglio di seguito.

6.1 Cinematica diretta

Il calcolo della cinematica diretta è stato implementato con la funzione `cin_diretta` che prende in ingresso due vettori: `th[5]` e `pos[3]`.

La funzione legge i valori degli angoli di giunto, espressi in radianti, dal vettore `th[5]` e memorizza nel vettore `pos[3]` le coordinate, espresse in metri, della posizione dell'effettore in riferimento alla terna di base.

Il calcolo viene eseguito in base alle equazioni (4.5) ottenute dalla matrice di trasformazione omogenea ${}^0T_e$.

Nelle formule implementate sono stati utilizzati i seguenti parametri del robot:

- L[0] : lunghezza del link 1 (piano base - shoulder)
- L[1] : lunghezza del link 2 (shoulder - elbow)
- L[2] : lunghezza del link 3 (elbow - pitch)
- L[3] : lunghezza del link 4 (pitch - punta della pinza)
- a1 : distanza minima tra l'asse z_0 e il giunto shoulder

Di seguito viene riportato il codice della funzione `cin_diretta`:

```
void cin_diretta(double th[5], double pos[3]) {

    double k;
    k = L[3]*cos(th[1]+th[2]+th[3])+L[2]*
                        *cos(th[1]+th[2])+L[1]*cos(th[1])+a1;
    pos[0] = -sin(th[0])*k;
    pos[1] = cos(th[0])*k;
    pos[2] = L[3]*sin(th[1]+th[2]+th[3])+L[2]*
                        *sin(th[1]+th[2])+L[1]*sin(th[1])+L[0];
}
```

6.2 Cinematica inversa

Nella funzione del calcolo della cinematica inversa vengono passati in ingresso il vettore `pos[3]` e la variabile `phi` relativi a posizione e orientamento desiderati e inoltre viene specificato, tramite la variabile `gomito`, il funzionamento a gomito alto (+1) o a gomito basso (-1). I valori ottenuti vengono memorizzati, in radianti, nel vettore degli angoli di giunto `th[5]`.

I parametri del robot sono gli stessi che sono stati già elencati nel paragrafo precedente.

L'orientamento ϕ è pari alla somma degli angoli di *shoulder*, *elbow* e *pitch*.

Durante il calcolo la funzione verifica se la posa è all'interno dello spazio di lavoro; in caso non lo fosse viene generato un messaggio di errore e si interrompe il calcolo restituendo il valore -1.

Se il calcolo va a buon fine viene restituito il valore 1;

```
int cin_inversa(double th[5], double pos[3], double phi, int gomito)
{
    // variabili temporanee per il calcolo di th[2]
        double costh2, y_segnato, z_segnato, modulo;
    // variabili temporanee per il calcolo di th[1]
        double alpha, beta;
    // variabili temporanee per il calcolo di th[0]

    // Inizio calcolo di th[2] <=> elbow
        modulo = sqrt(pos[0]*pos[0]+pos[1]*pos[1]);
        y_segnato = -a1 + modulo - L[3]*cos(phi);
        z_segnato = pos[2] - L[0] - L[3]*sin(phi);

        costh2 = (y_segnato*y_segnato+z_segnato*z_segnato-
                                -L[1]*L[1]-L[2]*L[2])/(2*L[1]*L[2]);

        if (costh2 >= 1 || costh2 <= -1)
        {
            printf("Errore: la posizionoe desiderata
                    non e' all'interno del suo spazio operativo\n");
            return -1;
        }

        th[2] = atan2( gomito*sqrt(1-costh2*costh2), costh2 );
    // fine calcolo di th[2]

    // Inizio calcolo di th[1] <=> shoulder
        alpha = L[2]*cos(th[2])+L[1];
        beta = L[2]*sin(th[2]);
        th[1] = atan2(alpha*z_segnato-beta*y_segnato,
                                alpha*y_segnato+beta*z_segnato);
    // fine calcolo di th[1]
```

```
    // Inizio calcolo di th[3] <=> Pitch
        th[3] = phi-th[1]-th[2];
    // fine calcolo di th[3]

    // Inizio calcolo di th[0] <=> Base
        th[0] = atan2(-pos[0],pos[1]);
    // fine calcolo di th[0]
        return 1;
}
```

6.3 Spazio di lavoro

Prima di comandare la movimentazione del robot è necessario controllare che la posizione desiderata sia effettivamente raggiungibile e che non violi le condizioni specificate nel paragrafo 5.2.

La verifica viene eseguita dalla funzione `verifica_range` che prende in input il vettore contenente i valori in passi encoder dei cinque angoli di giunto.

Inizialmente viene verificato che il valore di ogni angolo sia all'interno del range di ammissibilità del rispettivo giunto.

Poi tramite il calcolo della cinematica diretta si controlla che la posizione desiderata non coincida con il cilindro della base del robot.

Infine si verifica che la posizione sia all'interno dell'area di lavoro, definita nel paragrafo 5.2.3, e in particolare che non vada a collidere con il piano di lavoro.

La funzione restituisce valore 1 se la verifica è andata a buon fine, 0 altrimenti.

```
int verifica_range(int encrif[5])
{
    double pos[3], th[5];

    // verifico se i passi encoder sono dentro un range accettabile
    if(encrif[0] < -3621 || encrif[0] > 2957) return 0;
    if(encrif[1] < -2385 || encrif[1] > 227) return 0;
    if(encrif[2] < -2500 || encrif[2] > 2350) return 0;
    if(encrif[3] < -550 || encrif[3] > 900) return 0;

    enc2thAll(encrif, th);
    cin_diretta(th, pos);
    // Verifico se sto all'interno del cilindro della base
    if(pos[0]*pos[0]+pos[1]*pos[1] < 0.015 && pos[2] < 0.40)
    {
    cprintf("\r posizioni: %f %f %f \n\r",pos[0],pos[1],pos[2]);
    cprintf("\r condizione 2: %f \n\r",pos[0]*pos[0]+pos[1]*pos[1]);
    return 0;
    }
```

```
    // Verifico se sto andando sotto il piano di lavoro
    if(pos[2] < 0.001) return 0;

    return 1;
}
```

6.4 Movimentazione del robot

Il movimento del robot può essere comandato imponendo una posa e calcolando gli angoli corrispondenti tramite la cinematica inversa, oppure assegnando direttamente i cinque angoli di giunto.

In entrambi i casi viene generato un vettore dei riferimenti contenente i valori desiderati per ogni giunto espressi in passi encoder.

Poi tramite la funzione `ROBOTMoveAll` si comanda il robot in modo tale che gli encoder di ogni giunto segnino valori corrispondenti a quelli presenti nel vettore dei riferimenti degli angoli di giunto. In questo modo il robot raggiungerà la posizione o la configurazione desiderata.

La funzione `ROBOTMoveAll` riceve in ingresso il vettore dei riferimenti `encrif[5]` e come passo iniziale richiama la funzione `verifica_range` per accertarsi che la posizione desiderata sia raggiungibile.

Se la verifica da esito positivo procede.

Prima di eseguire il movimento del robot sono state introdotte alcune ottimizzazioni per ottenere un funzionamento migliore del robot: il guadagno del controllore viene calcolato dinamicamente a seconda della distanza dal punto desiderato; viene corretto l'errore di *pitch* che si ha durante il movimento di *roll*; inoltre vengono presi alcuni accorgimenti per evitare che la pinza tocchi il piano di lavoro quando il robot si trova ad un'altezza critica.

Tutte queste ottimizzazioni vengono trattate nel dettaglio nel paragrafo 9.

Infine la funzione memorizza nel vettore `ROBOTDefCtrlPosReq` i riferimenti desiderati. Così l'*ISR* si occuperà di comandare il movimento del robot affinchè si raggiungano i riferimenti assegnati.

7 Pseudocodice

Per poter impartire dei comandi al robot è stato introdotto uno pseudolinguaggio composto da macroistruzioni che possono essere impartite direttamente dall'utente.

La lista delle macroistruzioni viene memorizzata in un file seguendo una certa sintassi. Indicando il nome del file il programma tramite la funzione `leggiFile` scorre il file riga per riga e richiama la funzione `AddInstruction` che interpreta i comandi e li memorizza nell'array `commands`.

Questo array è costituito da una struttura che contiene tutte le informazioni sulle operazioni da eseguire.

La funzione `muoviRobotFile` in base ai comandi e ai valori memorizzati esegue una dopo l'altra le macroistruzioni elencate nel file.

7.1 Descrizione delle macroistruzioni

Sono state definite le seguenti macroistruzioni:

- **Home**:

  Riporta il robot nella posizione di *Home* tramite la funzione `ROBOTHome`.

- **Pausa**:

  Esegue una pausa di una durata pari al numero dei secondi specificati e memorizzati nella variabile `sec`.

- **Vaigiunti**:

  Consente di specificare gli angoli relativi desiderati per ogni giunto (in gradi). Effettua la conversione da gradi a radianti e poi da radianti a passi encoder che vengono poi imposti al robot.

- **Vaieff**:

  Specificando posizione (in millimetri) e orientamento (in gradi) il robot raggiunge la posa desiderata. Calcola la cinematica inversa e ottiene i riferimenti per ogni giunto.

- **Apripinza**:

  Comanda l'apertura della pinza tramite la funzione `ROBOTGripperOpen`.

- **Chiudipinza**:

  Comanda l'apertura della pinza tramite la funzione `ROBOTGripperClose`.

- **Sommaeff**:

  Rileva la presenza di un oggetto all'estremità della pinza in modo da considerare come nuovo centro dell'effettore la punta dell'oggetto.

- **PreHome**:

  Porta il robot in una posizione preimpostata vicino alla *Home* in modo da semplificare la ricerca della *Home* alla successiva esecuzione del programma.

- **Roll**:

  Esegue una rotazione della pinza per un numero di giri specificato accertandosi che gli altri giunti del robot restino immobili. La rotazione viene eseguita con incrementi successivi in modo da consentire la correzione del pitch (si veda il paragrafo 9.3).

- **Impostamatita**:

  Non fa altro che salvare la posizione del temperino per poi raggiungerla con Vaiatemperino.

- **Vaiatemperino**:

  Raggiunge la posizione del temperino definita con il comando *Impostamatita* a una altezza specificata.

- **Sali**:

  Se il tratto specificato (in millimetri) è positivo il robot si muove in modo che l'effettore segua una traiettoria verticale dal basso verso l'alto. Se il segno è negativo il movimento è dall'alto verso il basso.

8 Progetto: *Il temperamatite*

8.1 Obiettivo

Si vuole realizzare un temperamatite automatico che sia in grado di temperare una matita e svuotare successivamente il cestino.

L'operazione si divide in due fasi: il robot prende una matita posta in una posizione nota, la pone sul temperamatite corrispondente e la tempera; la matita viene poi riportata nella sua posizione iniziale e quindi posata.

Nella seconda fase il robot prende il cestello contenente i residui della matita e con un opportuno movimento lo svuota nel cestino posto sul piano di lavoro. Infine il cestello viene riposizionato sotto il temperino e il robot torna in posizione di riposo pronto ad eseguire una nuova operazione.

8.2 Realizzazione della struttura

Per permettere al robot di eseguire il lavoro richiesto è stato necessario realizzare alcune strutture e disporle insieme agli altri oggetti in determinate posizioni sul piano di lavoro.

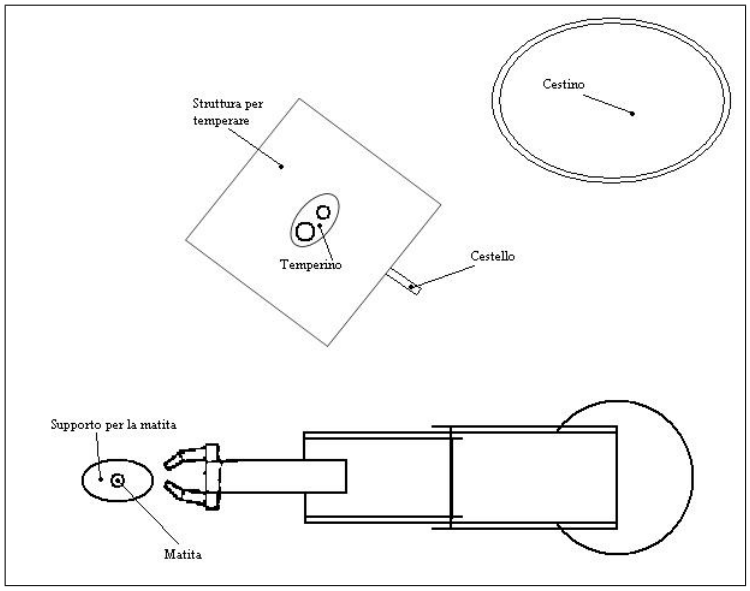

Figura 3: Schema del piano di lavoro

La struttura più importante è la base di supporto per il temperino. La struttura è stata realizzata con una base in legno su cui sono stati posti altri due pezzi di legno laterali, che fanno da sostegno per un cartoncino semi-rigido. Al centro del cartoncino è stato fatto un foro per inserire il temperino formato da due lame e due fori di diverso diametro, per poter temperare matite con diversi spessori. La scelta di questo tipo di cartoncino è stata fatta per dare elasticità alla struttura e consentire di temperare la matita in modo migliore. Sotto il temperino viene posto un cestello, in cui cadono i residui della matita, dotato di un manico lungo in modo da facilitarne la presa.

Come base di appoggio delle matite sono stati utilizzati altri due pezzi di legno di dimensioni uguali disposti a una certa distanza tra di loro e paralleli. Sul legno sono

stati realizzati degli incavi per fare in modo che le matite restino ferme in un punto d'equilibrio.

Sul piano viene inoltre posto un semplice cestino in cui sarà svuotato il cestello del temperino.

8.3 Implementazione

Il lavoro del robot consiste in una serie di macroistruzioni successive, opportunamente implementate. Per il significato di ogni macroistruzione si veda il capitolo 7.1. La lista delle macroistruzioni viene salvata in un file di testo e caricato dal programma al momento dell'esecuzione del lavoro richiesto.

Come primo movimento il robot raggiunge la sua posizione di *Home*. Il robot può così iniziare ad eseguire la sequenza di operazioni: si sposta verso la prima matita, la prende e ne calcola lo spessore; a questo punto si considera la lunghezza della matita, nota a priori, in modo che il nuovo centro dell'effettore coincida con la punta della matita stessa; poi la pinza si sposta all'altezza del foro del temperino con la dimensione della matita presa, la lascia cadere e la riprende; inizia così la rotazione che permette di temperare la matita. Finita questa operazione il robot si alza in modo da far uscire la matita dal temperino e si sposta nel punto in cui era la matita inizialmente per posarla. Se ci sono altre matite da temperare il robot ripete le operazioni eseguite finora per ogni matita terminando così la prima fase.

Nella seconda fase si sposta la pinza, dopo averla aperta, nella posizione del manico cestello, si chiude la pinza e si muove in modo che il cestello esca dalla base del temperino. Il cestello viene portato all'altezza del cestino e con un movimento prima di *roll* e poi di *pitch* rovescia il cestello e lo si svuota scuotendolo con piccoli movimenti di *roll*. Il robot riporta il cestello sotto il temperino e torna nella posizione di riposo terminando così le operazioni.

Di seguito viene riportata la lista delle macroistruzioni necessarie ad eseguire le operazioni sopra descritte:

```
VAIHOME
PRINT ----------_Prendi_matita_-----------
PRINT Vai_sopra_la_matita
VAIEFF 0 400 200 -90 0
APRIPINZA
PRINT Scendi_per_prendere_la_matita
VAIEFF 0 400 120 -90 0
PAUSA 500
CHIUDIPINZA
PAUSA 500
PRINT Ho_preso_la_matita_e_la_imposto
IMPOSTAMATITA
PRINT Estrai_la_matita
VAIEFF 0 400 200 -90 0
SOMMAEFF 135
PAUSA 500
PRINT =======_Fine_Prendi_Matita
```

```
PRINT -----------_Vai_al_temperino_-------------
PRINT Metiti_sopra_la_bocca_del_temperino_a_2_cm_di_altezza
VAIATEMPERINO 30
PAUSA 500
PRINT Lascia_la_matita
APRIPINZA
PAUSA 500
PRINT Afferra_la_matita_per_temperarla
VAIATEMPERINO -50
PAUSA 500
CHIUDIPINZA
PAUSA 500
PRINT Premi_sul_temperino
VAIATEMPERINO -58
PAUSA 500
PRINT Tempera
ROLL -1
PAUSA 500
PRINT =======_Fine_Vai_al_temperino
PRINT -------------_Sali_e_posa_la_matita_------------
VAIATEMPERINO 20
PAUSA 500
PRINT Vai_a_posare_la_matita_in_2_passaggi_intermedi
VAIEFF 10 400 135 -90 0
VAIEFF 10 400 80 -90 0
PAUSA 500
APRIPINZA
PAUSA 500
SOMMAEFF -135
PRINT ----------_Prendi_spazzatura_-------------
PRINT Avvicinati_al_manico
VAIEFF 100 250 130 -90 0
PRINT ci_sono
PAUSA 2000
VAIEFF 170 170 40 -90 -90
PAUSA 500
PRINT Vai_al_manico
VAIEFF 220 230 10 -90 0
PAUSA 500
CHIUDIPINZA
PAUSA 500
PRINT ---------_Esci_spazzatura_---------
PRINT Esci_un_po
VAIEFF 175 205 40 -90 0
PAUSA 500
PRINT pitch_e_sali
VAIEFF 130 130 150 -135 0
```

```
PAUSA 500
PRINT sali_ancora
VAIEFF 130 130 300 -135 0
%PAUSA 500
PRINT Sali_per_andare_al_cestino
VAIEFF 250 250 400 -90 0
%PAUSA 500
PRINT Vai_sopra_il_cestino
VAIEFF 270 0 500 -90 0
PAUSA 500
PRINT Roll
VAIEFF 270 0 500 -90 -180
PAUSA 500
PRINT Rovescia
VAIEFF 300 0 600 70 0
PAUSA 500
PRINT Svuota
VAIEFF 300 0 600 70 -40
VAIEFF 300 0 600 70 40
VAIEFF 300 0 600 70 -40
VAIEFF 300 0 600 70 40
PAUSA 500
PRINT ---------_Posa_la_pipa_----------
PRINT Rovescia_al_contrario
VAIEFF 270 0 500 -50 0
PAUSA 500
PRINT Roll_al_contrario
VAIEFF 270 0 500 -50 180
PAUSA 500
PRINT Scendi_per_levarti_dal_cestino
VAIEFF 250 250 400 -90 0
%PAUSA 500
PRINT scendi_ancora1
VAIEFF 130 130 300 -135 0
PAUSA 500
PRINT scendi_ancora2
VAIEFF 130 130 150 -135 0
PAUSA 500
PRINT Entra_un_po
VAIEFF 175 205 40 -90 0
PAUSA 500
PRINT Posa_la_pipa
VAIEFF 210 240 10 -90 0
PAUSA 500
APRIPINZA
PAUSA 500
PRINT Allontanati_dal_manico
```

```
VAIEFF 170 170 40 -90 0
CHIUDIPINZA
VAIEFF 230 210 150 -90 0
PAUSA 500
PREHOME
PAUSA 500
```

9 Ottimizzazioni

9.1 Controllo

Per migliorare la legge di controllo del robot è stata introdotta un'azione proporzionale-integrativa che interviene quando l'errore è minore di una certa soglia.

La modifica è stata effettuata nella funzione `ROBOTDefCtrlFn` presente all'interno del file `SCORB.C`

Viene prima calcolato l'integrale dell'errore per ogni giunto e poi viene ricavata la tensione da applicare a ogni motore mediante la seguente formula:

$$mVolt = K_P * e(t) + K_I * \int_0^t e(\tau)\, d\tau \tag{9.1}$$

Da notare una piccola differenza nel caso del giunto di *elbow* per il quale si è preferito utilizzare un K_P fisso.

La porzione di codice che realizza quanto detto è la seguente:

```
for (nLink = 0; nLink < ROBOT_MAX_LINK; nLink++)
    if(labs(ROBOTDefCtrlPosError[nLink]) < 40)
        integrale[nLink] += ROBOTDefCtrlPosError[nLink]*0.05;

for (nLink = 0; nLink < ROBOT_MAX_LINK; nLink++)
    {

    mVolt = ROBOTDefCtrlPosError[nLink] * ROBOTDefCtrlKP + 300*integrale[nLink];
       // Per l'elbow ho bisogno di un kp pari a 100
       if(nLink == 2) {
           mVolt = ROBOTDefCtrlPosError[nLink] * 100 + 300*integrale[nLink];
       }

       if(mVolt > 4000) mVolt = 4000;
     ROBOTSetMotorOutput(nLink, mVolt);
    }
```

9.2 Precisione

Durante la taratura del robot è stato constatato un notevole errore di precisione.

Per ridurre tale errore, oltre alla modifica nella funzione di controllo descritta precedentemente, viene introdotto un guadagno dinamico. Nella funzione `ROBOTMoveAll` viene calcolato l'errore di posizione e si modifica dinamicamente il guadagno del controllore K_p che diventa più grande quanto più si avvicina alla posizione desiderata.

Questo viene fatto per avere una maggiore precisione ed evitare allo stesso tempo che il controllo diventi troppo *aggressivo* quando l'errore è molto grande. Il valore di K_p viene riportato al valore di *default* alla fine della funzione.

9.3 Casi particolari

Prima di comandare il movimento al robot nella funzione `ROBOTMoveAll` viene controllata l'altezza del punto O_3 inteso come il centro del *pitch*. Per evitare che la pinza del robot tocchi il piano di lavoro durante un movimento di *pitch* (caso molto comune), quando il punto O_3 si trova ad un'altezza pari o minore della lunghezza dell'ultimo braccio (d_5), un eventuale movimento di *pitch* viene scomposto in due movimenti: prima si alza il robot e poi lo si riabbassa per raggiungere la posizione desiderata.

Inoltre, è stato notato che ogni movimento di *roll* causa un piccolo movimento di *pitch* che diventa però consistente all'aumentare del *roll*. Per ovviare a tale problema si esegue una correzione del *pitch* alla fine di ogni movimento di *roll* in modo da riportare il pitch nella posizione in cui era prima di eseguire il *roll*.

10 Considerazioni finali

Dal lavoro svolto sono state tratte alcune considerazioni interessanti. Ad esempio è stato possibile osservare come da un compito in teoria semplice, come quello del temperamatite automatico, sono emerse numerose problematiche. Innanzitutto è stata necessaria un'elevata precisione nel raggiungere le posizioni desiderate. Ottenere ciò non è stato affatto semplice perchè nella struttura del robot sono presenti molti giochi meccanici e inoltre è difficile effettuare misure esatte sia dei bracci del robot che delle costanti di riduzione. Senza contare i limiti fisici che il robot presenta. E' stato necessario quindi capire a fondo molti aspetti legati al controllo e alla meccanica del robot, oltre che ad un'accurata analisi della cinematica diretta e inversa e a come limitare lo spazio di lavoro del robot per evitare situazioni impreviste.

Infine l'intero lavoro è stato utile per entrare a contatto in modo pratico con la robotica ed è stata una soddisfazione riuscire a comandare un braccio robotico e realizzare una serie di operazioni completamente automatiche.

Si ringrazia per questo il prof. Zaccarian.