UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTA' DI INGEGNERIA

Corso di Laurea triennale in Ingegneria Meccanica

ANALISI DI UN ALGORITMO LINEARE DI CALIBRAZIONE NELLA ROBOTICA PARALLELA

ANALYSIS OF A LINEAR CALIBRATION ALGORITHM IN PARALLEL ROBOTICS

Relatore:

Prof. Ing. **Matteo Claudio Palpacelli**

Tesi di Laurea di:

Vincenzo Orlando

A.A. 2019/2020

INDICE

PREMESSA

La seguente tesi è frutto di un lavoro di tirocinio svolto con l'obiettivo di verificare l'efficienza di un algoritmo di calibrazione di tipo lineare, che approssima al meglio la matrice pseudo-inversa a partire dalla matrice rettangolare di calibrazione. Esso è applicato ad un pentalatero planare ed il processo analitico-applicativo è coadiuvato da Matlab come software di calcolo.

Il punto di partenza è stato lo studio della cinematica del robot: si è definito un modello matematico dello sesso attraverso il quale è stato possibile effettuare l'analisi che ha permesso di ottenere un sistema di equazioni che consentono, a partire da dati iniziali definiti, di risalire alla posizione del terminale del robot nello spazio. Si è ricorso a Matlab, che oltre a fornire il risultato numerico, permette di mostrarlo visivamente utilizzando un comando che ha la funzione di mostrare la configurazione su un grafico, permettendo la chiara visione della posizione del terminale. Una volta messe in memoria diverse configurazione del robot, nella seconda fase è stata determinata la matrice jacobiana del sistema, attraverso una serie di passaggi matematici svolti con l'utilizzo del software.

Successivamente, per simulare la misura delle configurazioni di un robot reale affetto da errori geometrici, si è ipotizzato un errore infinitesimo associato alla lunghezza delle aste del robot e sono state salvate diverse configurazioni, (ossia diverse posizioni del terminale nel piano), utilizzando come dati di ingresso gli angoli utilizzati nel trovare le configurazioni del robot nel caso ideale e le aste di lunghezza variata. In questo modo, le equazioni hanno restituito risultati che discostano da quelli ottenuti in precedenza; in particolare la differenza rappresenta l'errore infinitesimo della posizione del terminale per il robot reale simulato.

Una volta terminata questa operazione applicata ad ogni configurazione, è stato possibile definire un sistema di questo tipo:

$$\partial \boldsymbol{x} = \boldsymbol{A} \cdot \partial \boldsymbol{l}$$

Dove $\partial \boldsymbol{x}$ è il vettore che racchiude gli errori infinitesimi delle coordinate del terminale nelle varie configurazioni, $\boldsymbol{A}$ è la matrice che racchiude tutte le matrici jacobiane relative alle varie configurazioni considerate, mentre $\partial \boldsymbol{l}$ è il vettore che racchiude gli errori infinitesimi ipotizzati delle aste.

Lo scopo principale è quello di risalire all'errore delle aste noto l'errore infinitesimo della posizione del terminale e la matrice $\boldsymbol{A}$ per ogni configurazione, e confrontarlo con quello ipotizzato verificando quanto questi si avvicinino; ossia risolvere il sistema:

$$\partial \boldsymbol{l} = \boldsymbol{B} \cdot \partial \boldsymbol{x}.$$

In questa fase ha un ruolo fondamentale l'algoritmo di calibrazione utilizzato per determinare la matrice $\boldsymbol{B}$, ossia l'inversa che meglio approssima la matrice $\boldsymbol{A}$. In base all'efficienza dell'algoritmo si ottengono risultati prossimi a quelli ipotizzati in partenza.

Una volta messi a confronto i risultati ottenuti con quelli iniziali si è fatta una sorta di stima percentuale dell'errore commesso dall'algoritmo nella calibrazione, in modo da evidenziare i casi in cui conviene o non conviene utilizzarlo.

CAPITOLO 1 - INTRODUZIONE

Il tema della calibrazione di robot manipolatori inizia alla fine degli anni '70 e inizio anni '80 con l'avvento di nuove tecnologie di controllo automatico dei robot nel settore manufatturiero. Un robot manipolatore è caratterizzato da una serie di aste collegate attraverso giunti con una estremità collegata a telaio e con l'altra estremità collegata ad un terminale chiamato "end-effector". L'obiettivo principale è quello di controllare il robot in modo da garantire che il terminale assuma posizioni ben precise nello spazio. In particolar modo il controllo del robot avviene attraverso dei software che sfruttano un modello matematico "ideale" che lo rappresenta.
La realizzazione fisica del robot comporta l'introduzione di errori. Le macchine adottate per realizzarlo posso essere anche molto precise, ma di fatto, non si riuscirà mai ad ottenere un risultato ideale. Questo comporta una variazione tra le posizioni teoricamente assunte dal terminale secondo il modello matematico e quelle reali, questa variazione è la fonte dell' inaccuratezza del robot stesso. La calibrazione interviene, proprio, per accrescere l'accuratezza del robot modificando il modello matematico in modo da simulare il comportamento del robot reale. Per raggiungere questo scopo si utilizzano "algoritmi" matematici che sono proprio il fulcro di questa operazione.

CAPITOLO 2 - PENTALATERO PLANARE

Per poter descrivere in modo corretto i passaggi svolti durante il percorso e giungere alle conclusioni finali, è necessario partire dall'analisi cinematica del robot. In questi anni con il rapido sviluppo dei robot paralleli il "pentalatero planare" è ampiamente utilizzato nell'industria meccanica. Il robot è caratterizzato da cinque aste e cinque cerniere ed il suo movimento avviene sul piano xy.

La seguente immagine è una rappresentazione semplificata del robot:

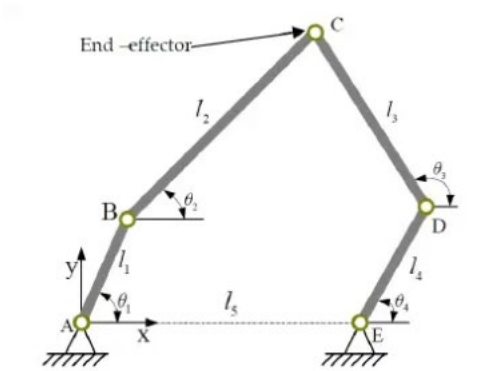

Fig.1 - Schema cinematico del pentalatero planare.

Sono presenti parti attive e parti passive: l'asta 5 è fissa, rappresenta il supporto, mentre 1 e 4 sono le aste guida. Dalla figura si nota che gli angoli caratteristici sono:

θ_1 , θ_2, θ_3, θ_4, θ_5 e vengono scelti θ_1 e θ_4 come gradi di libertà che permetto al robot di essere controllato.

ANALISI CINEMATICA

Dopo aver descritto il robot, per poter procedere con la calibrazione è necessario definire un valido modello matematico del robot. L'obiettivo del modello è quello di tradurre lo spostamento delle cerniere nella posizione del terminale e giungere a questo obiettivo è possibile attraverso due modi: cinematica diretta in cui a partire dagli angoli θ_1 e θ_4 si ottiene la posizione del terminale; cinematica inversa in cui gli angoli θ_1 e θ_4 sono determinati delle coordinate della posizione del terminale.

In riferimento alla figura 1, considerando le relazioni geometriche del maccanismo si deducono le equazioni cinematiche scritte in forma da evidenziare le coordinare del punto C che rappresenta il terminale, e che nello studio è stato denominato punto P:

$$x_p = l_1 cos\theta_1 + l_2 cos\theta_2 = l_5 + l_4 cos\theta_4 + l_3 cos\theta_3 \qquad (3)$$

$$y_p = l_1 sin\theta_1 + l_2 sin\theta_2 = l_3 sin\theta_3 + l_4 sin\theta_4 \qquad (4)$$

Trasponendo il sistema di riferimento dal centro della cerniera attuata di sinistra a metà dell'asta 5 l'equazione (3) varia in questo modo:

$$x_p = l_1 cos\theta_1 + l_2 cos\theta_2 - \frac{l_5}{2} = l_4 cos\theta_4 + l_3 cos\theta_3 + \frac{l_5}{2} \qquad (3')$$

Dalla (3') e dalla (4) è possibile trovare che θ_1 e θ_4 sono (gradi di libertà) indipendenti nel sistema, e θ_2 e θ_3 possono essere determinate da θ_1 e θ_4 come segue :

$$\boldsymbol{\theta_3} = \Bigg(2 \cdot \tan^{-1}((-l_1^2 + 2\cos(\theta_1 - \theta_4)l_1 l_4 + 2cos\theta_1\, l_1 l_5 + l_2^2 + 2l_2\, l_3 + l_3^2 - l_4^2 - 2cos\theta_4 l_4 l_5 - l_5^2) \cdot (l_1^2 - 2cos(\theta_1 - \theta_4)l_1 l_4 - 2cos\theta_1\, l_1 l_5 - l_2^2 + 2l_2 l_3 - l_3^2 + l_4^2 + 2cos\theta_4 l_4 l_5 + l_5^2))^{\frac{1}{2}} + - \left(\frac{2 l_3 l_4 sin\,\theta_4}{l_1^2 + 2cos\theta_1\, l_1 l_3 - 2\cos(\theta_1 - \theta_4)l_1 l_4 - 2cos\theta_1\, l_1 l_5 - l_2^2 + l_3^2 - 2cos\theta_4 l_4 l_3 - 2l_3 l_5 + l_4^2 + 2cos\theta_4 l_4 l_5 + l_5^2}\right)\Bigg).$$

(5)

θ_3 è indicato come vettore in quanto il seguente risultato è sia positivo che negativo.

$$\theta_2 = atan2(l_3 sin\theta_3 + l_4 sin\theta_4 - l_1 22 sin\theta_1, l_5 + l_4\, cos\theta_4 + l_3 sin\theta_3 - l_1 cos\theta_1).$$

(6)

Le equazioni (5) e (6) dipendono dagli angoli di attuazione θ_1 e θ_4 e dalla lunghezza delle aste del sistema, e, sostituendo le (5) e (6) nelle (3') e (4), si determina, attraverso la "cinematica diretta del robot", la posizione del robot nello spazio a partire proprio da θ_1 e θ_4 e dalla lunghezza delle aste.

Matlab permette di simulare, come il robot si muove nello spazio, impostando una funzione chiamata "Kindir_rig" la quale ha come input due vettori:

1) $\boldsymbol{geom} = \begin{pmatrix} geom_1 \\ geom_2 \\ geom_3 \\ geom_4 \\ geom_5 \end{pmatrix} = \begin{pmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \end{pmatrix}$

2) $\boldsymbol{q} = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} \theta_1 \\ \theta_4 \end{pmatrix}$

Il primo è il vettore che contiene le lunghezze delle aste, mentre il secondo è il vettore che contiene gli angoli θ_1 e θ_4. In output si ottiene il vettore delle coordinate del terminale:

$\boldsymbol{P} = \begin{pmatrix} x_p \\ y_p \end{pmatrix}$

A questo punto sfruttando le equazioni della cinematica del robot e utilizzando il comando "plot", Matlab riesce a fornire una rappresentazione grafica del pentalatero nelle diverse configurazioni scelte.

```
function x = kindir_rig(q,geom)

% caso mostrato nell'articolo sul pentalatero con tutte aste differenti e
% sistema di riferimento centrato al l5/2.
l1 = geom(1);
l2 = geom(2);
l3 = geom(3);
l4 = geom(4);
l5 = geom(5);

theta1 = q(1);
theta4 = q(2);

theta3_vett = [ 2*atan((((- l1^2 + 2*cos(theta1 - theta4)*l1*l4 + 2*cos(theta1)*l1*l5 + l2^2 + 2*l2*l3 + l3^2 - l4^2 - 2*cos(theta4)*l4*l5 - l5^2)*
                -2*atan((((- l1^2 + 2*cos(theta1 - theta4)*l1*l4 + 2*cos(theta1)*l1*l5 + l2^2 + 2*l2*l3 + l3^2 - l4^2 - 2*cos(theta4)*l4*l5 - l5^2)*

theta3 = theta3_vett(2);
theta2 = atan2(l3*sin(theta3)+l4*sin(theta4)-l1*sin(theta1),l5+l4*cos(theta4)+l3*cos(theta3)-l1*cos(theta1));

x(1) = l1*cos(theta1)+l2*cos(theta2)-l5/2;
x(2) = l1*sin(theta1)+l2*sin(theta2);

plot([-l5/2,-l5/2+l1*cos(theta1),-l5/2+l1*cos(theta1)+l2*cos(theta2)],[0,l1*sin(theta1),l1*sin(theta1)+l2*sin(theta2)],'r'); hold on;
plot([l5/2,l5/2+l4*cos(theta4),l5/2+l4*cos(theta4)+l3*cos(theta3)],[0,l4*sin(theta4),l4*sin(theta4)+l3*sin(theta3)],'b');
axis equal;
format long
```

:

Fig.2 - Command window di Matlab con dati e operazioni per il calcolo delle coordinate del robot.

Il comando plot(X,Y) ha come input i vettori delle coordinate X e Y del singolo ramo del pentalatero, e, poiché il robot è caratterizzato da due rami, con il primo plot si va a rappresentare il ramo di sinistra. Infatti all'interno della parentesi sono state inserite tutte le coordinate X e Y dei punti del suddetto ramo. Il programma unisce i punti andando ad ottenere così il ramo voluto. Procedimento analogo per il ramo di destra.

I dati iniziali considerati per l'analisi del robot sono i seguenti:

$l_1 = 0.130\,m$
$l_2 = 0.300\,m$
$l_3 = 0.300\,m$
$l_4 = 0.130\,m$
$l_5 = 0.260\,m$

Una volta sostituiti questi dati all'interno della funzione è possibile ottenere diverse configurazioni del robot al variare degli angoli θ_1 e θ_4.

Il seguente è un esempio di ciò che il comando "plot" restituisce una volta sostituiti i dati relativi alla lunghezza delle aste, scelto $\theta_1 = \frac{\pi}{2}$, $\theta_4 = \frac{\pi}{2}$ e avviando il programma:

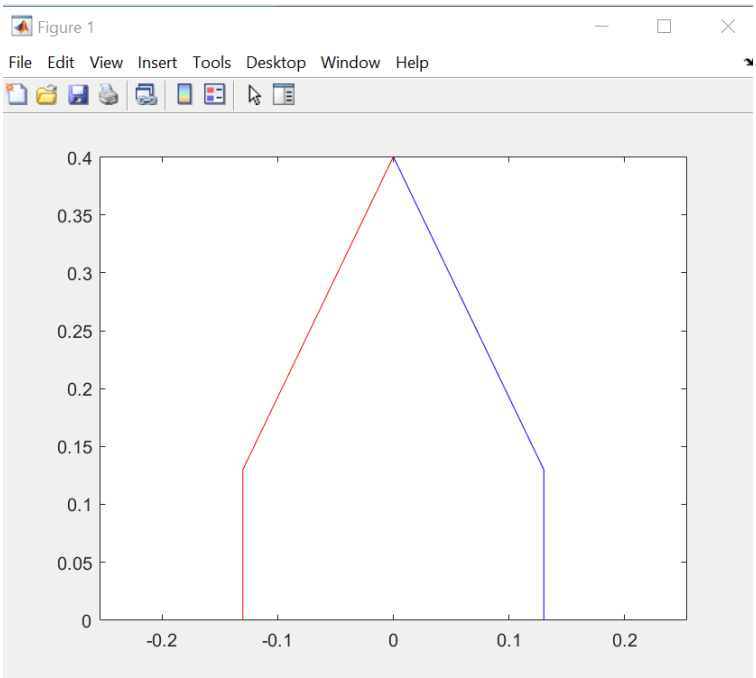

Fig.3 - Grafico di rappresentazione schematica del robot nella configurazione riportata.

Le coordinate del terminale risultanti risultanti sono:

$$x_p = 0$$

$$y_p = 0.4004$$

Ciò che ricaviamo dall'analisi cinematica del pentalatero planare è quello di poter ottenere per qualsiasi angolo in input (grado di libertà) la posizione del terminale e questo è un risultato importante che servirà in seguito.

CAPITOLO 3 - JACOBIANA DEL SISTEMA

Un passaggio fondamentale nello studio del robot è stato quello di determinale la jacobiana del sistema.
Sia $\boldsymbol{f}: U \subset \mathbb{R}^n \longrightarrow \mathbb{R}^m$ una funzione definita su un insieme aperto U dello spazio euclideo $\mathbb{R}^n$ la matrice jacobiana della funzione $\boldsymbol{f}$ in $\boldsymbol{x} = (x_1, \dots, x_n)$ è la matrice delle derivate parziali prime della funzione calcolate in $\boldsymbol{x}$:

$$J\,\boldsymbol{f} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix},$$

scritta anche:

$$(J\,\boldsymbol{f})_{ij} = \frac{\partial f_i(x)}{\partial x_j}; \qquad (7)$$

In Matlab il comando che permette di ricavare la matrice Jacobiana è: "jacobian($\boldsymbol{f}$,$\boldsymbol{x}$)"
- $\boldsymbol{f}$ rappresenta la funzione di input che può essere scalare o vettoriale.
- $\boldsymbol{x}$ è il vettore delle variabili rispetto alle quali viene calcolata la matrice.

Nel caso del pentalatero planare si ha che:

$$\boldsymbol{f} = \begin{pmatrix} x_p \\ y_p \end{pmatrix} = \begin{pmatrix} l_1 cos\theta_1 + l_2 cos\theta_2 \\ l_1 sin\theta_1 + l_2 sin\theta_2 \end{pmatrix};$$

La funzione è un vettore contenente le equazioni che indicano la posizione del terminale.

$$\boldsymbol{x} = \begin{pmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \end{pmatrix};$$

Il vettore contiene le variabili di derivazione.

Partendo dalla funzione $\boldsymbol{f}$ e derivando entrambe le equazioni rispetto a $\boldsymbol{x}$ si ottiene l'equazione della jacobiana del sistema:

$$\boldsymbol{J} = \begin{pmatrix} \frac{\partial x_p}{\partial l_1} \cdots \frac{\partial x_p}{\partial l_n} \\ \frac{\partial y_p}{\partial l_1} \cdots \frac{\partial y_p}{\partial l_n} \end{pmatrix}.$$

n=1,…,5 e varia a seconda delle aste che consideriamo come variabili.

Il procedimento descritto svolto in Matlab, si può riassumere attraverso la seguente immagine che rappresenta proprio la command window del programma:

```
syms l1 l2 l3 l4 l5 theta1 theta4 theta5 real
diary j.txt
diary on

theta3 = 2*atan((((- l1^2 + 2*cos(theta1 - theta4)*l1*l4 + 2*cos(theta1)*l1*l5 + l2^2 + 2*l2*l3 + l3^2 - l4

theta2 = atan((l3*sin(theta3)+l4*sin(theta4)-l1*sin(theta1))/(l5+l4*cos(theta4)+l3*cos(theta3)-l1*cos(theta

x_c = l1*cos(theta1)+l2*cos(theta2)-l5/2;
y_c = l1*sin(theta1)+l2*sin(theta2);

M = jacobian([x_c;y_c],[l1;l2;l3;l4;l5]);
J= simplify(M)
```

Fig.4 - Command window di Matlab con dati e operazioni per il calcolo della jacobiana.

Il risultato che restituisce il programma è una matrice di dimensione 2·5 con termini molto lunghi e complessi; attraverso il comando "simplify" è stato possibile semplificare la matrice "$\boldsymbol{M}$" per renderla più maneggevole ottenendo in questo modo la matrice $\boldsymbol{J}$.

Partendo dalle equazioni cinematiche è possibile scriverle sotto forma differenziale, sfruttando la matrice jacobiana ottenuta.

Richiamando la relazione (7), e applicandola al pentlatero si può scrivere:

$$\boldsymbol{J} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{l}} \tag{8}$$

o anche:

$$\partial \boldsymbol{x} = \boldsymbol{J} \cdot \partial \boldsymbol{l} \tag{9}$$

Il termine $\partial \boldsymbol{x}$, dal punto di vista pratico, rappresenta il vettore di dimensione 2·1 degli errori infinitesimi delle coordinate del terminale:

$\boldsymbol{J}$ è la matrice jacobiana che dimensione 2·l dove l è il numero delle variabili alle quali sono riferite le derivate parziali.

$\partial \boldsymbol{l}$ è il vettore di dimensione n·1 che contiene le lunghezze delle aste, intese come variabili si vedrà in seguito che non in tutti i casi le aste sono state considerate tutte variabili e quindi il vettore dl non è da considerare sempre di dimensione 5·1. Da un punto di vista teorico con questo risultato è stato possibile correlare gli errori della posizione del terminale con gli errori delle lunghezze delle aste ed è fondamentale nel processo di calibrazione.

Questa è l'equazione di partenza per l'applicazione dell'algoritmo.

È necessario, però, raccogliere dei dati che rappresentano i termini dell'equazione, e che sono necessari per poter ottenere la matrice pseudo-inversa.

CAPITOLO 4 - RACCOLTA DELLE MISURAZIONI

La fase successiva dello studio è caratterizzata dall'eseguire una serie di misurazioni, cioè, ipotizzare una serie di valori degli angoli θ_1 e θ_4 e registrare le coordinate del terminale sfruttando le equazioni della cinematica diretta.

Ciò che si può verificare, infatti, è che le coordinate fornite dal modello possono differire dalle coordinate che il robot assume nella realtà e questo è spiegato dal fatto che durante la realizzazione fisica del robot la lunghezza delle aste non sarà mai precisa e identica a quella ideale a cui fa riferimento il computer.

Questo fenomeno è dovuto al fatto che le macchine che andranno a realizzare il robot hanno comunque un range di tolleranza che andrà ad inficiare sulla precisione del robot.

É importante quindi determinare un set di misurazioni, nella calibrazione, per poter confrontare i risultati reali con quelli ideali. Nel caso in studio, è stato ipotizzato un errore delle aste in modo da simulare il comportamento reale del robot all'interno del software e si sono registrati le posizioni del terminale risultanti utilizzando come dati di input gli stessi angoli usati nel caso ideale. I risultati ottenuti dalle misurazioni fatte nel caso ideale sono state, poi , confrontate con questi risultati evidenziando quanto differiscono.

Utilizzando la funzione "kindir_rig" e inserendo i valori delle lunghezze delle aste e degli angoli ipotizzati è stato possibile ricavare i valori delle coordinate del terminale e visualizzare graficamente il risultato in modo da poter individuare eventuali configurazioni critiche.

Tabella 1

θ1	θ2	xc	yc
π/6	2π/3	-0.124309521372560	0.345310406881844
3π/2	π/2	-0.167630546142402	0.167630546142402
5π/8	5π/4	0.109573011465880	0.199431904879566
3π/5	5π/3	0.124664816681780	0.179055107165586
7π/4	9π/8	-0.211046683186257	0.153190752307846
11π/8	π/12	-0.025901105672631	0.137443467466440

Gli angoli riportati nella tabella 3 sono quelli ipotizzati, e per ogni configurazione è riportato anche la posizione del terminale.

Oltre al risultato numerico il programma attraverso il comando "plot" già visto in precedenza fornisce anche un risultato grafico, considerano come esempio la prima configurazione:

$\theta_1 = \frac{\pi}{6}$ e $\theta_4 = \frac{2\pi}{3}$

$$\overline{P_1} = \begin{pmatrix} x_p \\ y_p \end{pmatrix} = \begin{pmatrix} -0.124309521372560 \\ 0.345310406881844 \end{pmatrix}$$

Graficamente si ha:

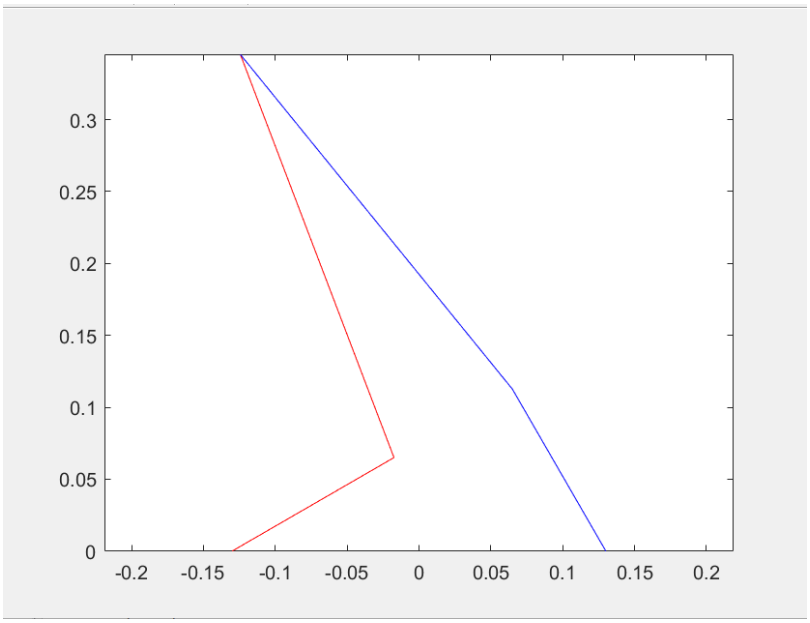

Fig.5 - Rappresentazione grafica del robot nella configurazione riportata.

Una volta memorizzate le configurazioni del robot nel caso ideale si passa a memorizzare le configurazioni che il robot assume introducendo piccoli scostamenti dalla lunghezza iniziale.

Si considera un errore infinitesimo poiché rispecchia l'errore reale delle aste, nel senso che, è vero che la realizzazione del robot comporta la formazione di errori, ma è anche vero che si vuole che questi siano più piccoli possibili in modo da ottenere risultati sempre più precisi.

Sono state affrontate tre situazioni differenti, ossia sono stati considerati dei dati iniziali differenti per osservare ed anlizzare i diversi risultati.

Nel primo caso il vettore $\partial \boldsymbol{l}$ degli errori infinitesimi ipotizzati, è uguale a:

$$\partial \boldsymbol{l} = \begin{pmatrix} 0.3 \\ 0.5 \\ 0.4 \\ 0.5 \\ 0.6 \end{pmatrix}$$

Gli errori ipotizzati sono tutti dell'errore del centesimo di millimetro.

Come dato di ingresso alla funzione quindi la lunghezza delle aste sarà pari a $\boldsymbol{l} + \boldsymbol{dl}$ ossia il vettore delle lunghezze iniziali più gli errori. Applicando l'identico processo sviluppato attraverso la funzione "kindir_ring" in Matlab, come nel caso "ideale", sono stati registrati i seguenti risultati:

Tabella 2

θ1	θ2	xc	yc
π/6	2π/3	-0.124357543445946	0.345362137662089
3π/2	π/2	-0.167644346610803	0.167652991501097
5π/8	5π/4	0.109590940501112	0.199432022113322
3π/5	5π/3	0.124683817491602	0.179044269436324
7π/4	9π/8	-0.211076557154329	0.153215854148847
11π/8	π/12	-0.025874053642013	0.137433049525516

Nel secondo caso di studio il vettore degli errori considerati è uguale a:

$$\partial \boldsymbol{l} = \begin{pmatrix} 0.4 \\ 0.1 \\ 8.4 \\ 0.5 \end{pmatrix}$$

Nel secondo e nel terzo caso le misurazioni fatte sono superiori a quelle fatte nel primo, in modo da fornire all'algoritmo più dati in input così che esso possa trovare un valore che approssima meglio i casi che si possono manifestare.

In aggiunta alle misurazioni riportate nella tabella 3 si hanno anche le seguenti misurazioni:

θ1	θ2	xc	yc
π/8	11π/6	0.226416285221403	0.234564062518475
7π/6	3π/2	-0.016266213508123	0.131927842708629
π/10	9π/5	0.231058958823069	0.223559717827923

I risultati ottenuti dopo l'aggiunta degli errori sono i seguenti:

Tabella 3

θ1	θ2	xc	yc
π/6	2π/3	-0.124570772869802	0.345212388346241
3π/2	π/2	-0.167717617352330	0.167616532460822
5π/8	5π/4	0.109549424025079	0.199519779227871
3π/5	5π/3	0.124650632376017	0.179133161596668
7π/4	9π/8	-0.211388280680957	0.152945734390619
11π/8	π/12	-0.025971951542734	0.137482329754876
π/8	11π/6	0.226361107951706	0.234642468176043
7π/6	3π/2	-0.016322245467989	0.131987762375615
π/10	9π/5	0.231002982171188	0.223639957996585

Considerando invece l'ultimo caso:

$$\partial \boldsymbol{l} = \begin{pmatrix} 0.1 \\ -0.3 \\ 0.1 \\ -0.5 \end{pmatrix} (mm)$$

Tabella 4

θ1	θ2	xc	yc
π/6	2π/3	-0.123838570213244	0.345185743347502
3π/2	π/2	-0.167720431956863	0.167616532460822
5π/8	5π/4	0.109065491502751	0.200098303986003
3π/5	5π/3	0.124252797148851	0.179551877480170
7π/4	9π/8	-0.211880812982364	0.152110781451644
11π/8	π/12	-0.026485671909934	0.137327558440610
π/8	11π/6	0.225887149278569	0.234909006640386
7π/6	3π/2	-0.016981180618453	0.132142012192243
π/10	9π/5	0.230495324837312	0.223951396430760

Utilizzando il comando "format_long", in Matlab, è stato possibile mettere in evidenza le cifre decimali più piccole dei risultati per vedere meglio la differenza tra questi, cosa che è meno evidente se si considerano solo le prime cifre decimali.

Andando, poi, a calcolare la differenza tra i risultati ottenuti nel caso in cui le aste vengano considerate ideali, e quelli in cui si è ipotizzato l'errore per ogni configurazione, si ottiene il vettore degli errori infinitesimi che richiamando la relazione:

$$\partial \boldsymbol{x} = \boldsymbol{J} \cdot \partial \boldsymbol{l}$$

rappresenta il vettore $\partial \boldsymbol{x}$.

Questa operazione è applicata a tutti i casi, considerando il primo caso:

Tabella 5

θ1	θ2	dx	dy
π/6	2π/3	-0.480220733860004	0.517307802450073
3π/2	π/2	-0.138004684009929	0.224453586950002
5π/8	5π/4	0.179290352319977	0.001172337560085
3π/5	5π/3	0.190008098219924	-0.108377292619977
7π/4	9π/8	-0.298739680719973	0.251018410009907
11π/8	π/12	0.270520306180008	-0.104179409240179

Tutti i valori riportati nella tabella sono moltiplicati di un fattore 10^{-4} tenendo conto dell'unità di misura in metri.

-Considerando il secondo caso:

Tabella 6

θ1	θ2	dx	dy
π/6	2π/3	0.261251497242002	0.098018535602984
3π/2	π/2	-0.870712099280135	-0.140136815799941
5π/8	5π/4	-0.235874408009984	0.878743483050115
3π/5	5π/3	-0.141843057629987	0.780544310819975
7π/4	9π/8	-0.341597494700002	-0.245017917227003
11π/8	π/12	-0.708458701030000	0.388622884359979
π/8	11π/6	-0.551772696970043	0.784056575680037
7π/6	3π/2	-0.560319598660007	0.599196669859858
π/10	9π/5	-0.559766518810068	0.802401686620102

Anche in questo caso tutti i valori sono moltiplicati per 10^{-4} eccetto per la configurazione numero 5 e 10.

-Considerando il terzo caso:

Tabella 7

θ1	θ2	dx	dy
π/6	2π/3	0.470951159316002	-0.124663534341984
3π/2	π/2	-0.089885814461010	-0.413780700243005
5π/8	5π/4	-0.507519963129005	0.666399106437010
3π/5	5π/3	-0.412019532929006	0.496770314584005
7π/4	9π/8	-0.000834129796107	-0.001079970856202
11π/8	π/12	-0.584566237302998	-0.115909025830002
π/8	11π/6	-0.529135942833991	0.344944121910995
7π/6	3π/2	-0.714967110329999	0.214169483613996
π/10	9π/5	-0.563633985757012	0.391678602837009

Tutti i risultati sono moltiplicati per 10^{-3}, tranne i risultati relativi alla quinta configurazione.

I successivi dati registrati sono quelli relativi alle jacobiane per ogni configurazione. Utilizzando l'equazione della jacobiana, implementandola in Matlab , e sostituendo gli angoli θ_1 e θ_4 per ogni singola configurazione ipotizzata. Anche questa operazione è stata applicata a tutti i casi considerati.

Considerando ad esempio la prima configurazione nel primo caso si ha:

$$\boldsymbol{J_1} = \begin{pmatrix} 0.473173847859758 & 0.180438854694034 \\ 2.983234599613208 & 1.137618735423782 \\ -2.476824024504726 & -2.014747581001846 \\ 2.445470282659623 & 1.989243195178390 \\ -1.382515713982796 & -0.717873527691779 \end{pmatrix}$$

La matrice ha dimensione 5·2 tenendo conto del fatto tutte e cinque le lunghezze delle aste in gioco sono state considerate come variabili e che le equazioni di riferimento sono due.

Mentre se prendiamo in considerazione il secondo o terzo caso la matrice jacobiana non avrà dimensione 5·2 ma 4·2 poiché le variabili diventano 4 e non più 5.

Un esempio è il seguente, nel quale sono riportati i risultati ottenuti dalla configurazione 1 del secondo caso:

$$\boldsymbol{J_1} = \begin{pmatrix} 0.473173847859758 & 0.180438854694034 \\ 2.983234599613208 & 1.137618735423782 \\ -2.476824024504726 & -2.014747581001846 \\ 2.445470282659623 & 1.989243195178390 \\ & \end{pmatrix}$$

I termini sono identici alla matrice analizzata nel primo caso, infatti i dati in ingresso sono li stessi, l'unica differenza che, in accordo con quanto detto in precedenza relativo alle riduzioni delle variabili, la dimensione della matrice risulta differente.

Riassumendo, il procedimento svolto ha consentito, partendo dagli errori ipotizzati delle aste attraverso la cinematica diretta di ricavare gli errori delle coordinate del

terminale per ogni configurazione.
A rigor di logica invertendo il processo, ossia partendo dall'errore della posizione del terminale dovrebbe essere possibile risalire all'errore delle aste.

Partendo dalla seguente relazione:

$$\partial \boldsymbol{x} = \boldsymbol{J} \cdot \partial \boldsymbol{l}$$

si vuole, quindi, ricavare il vettore $\partial \boldsymbol{l}$, conoscendo il vettore $\partial \boldsymbol{x}$. In termini matematici questo si traduce nell'invertire l'equazione :

$$\partial \boldsymbol{l} = \boldsymbol{J}^{-1} \cdot \partial \boldsymbol{x} \qquad (8)$$

L'equazione precedente, in realtà, non ha senso matematico, in quanto la matrice $\boldsymbol{J}$ non è quadrata ma di dimensione 2·l, e quindi non può essere invertita.
Allora per bypassare questa problematica si costruisce un sistema del tipo:

$$\partial \boldsymbol{x_1} = \boldsymbol{A} \cdot \partial \boldsymbol{l} \qquad (9)$$

$\partial \boldsymbol{x_1}$ è un vettore di dimensione nx1 dove n è uguale al numero di configurazioni analizzate nel caso di studio moltiplicato per due. Questo si riferisce al fatto che si considera per ogni configurazione sia la differenza tra le coordinate x del risultato

ideale con quello reale della posizione del terminale sia la differenza delle coordinate y.

$$\partial \boldsymbol{x_1} = \begin{pmatrix} x_{i1} - x_{r1} \\ y_{i1} - y_{r1} \\ \vdots \\ x_{in} - x_{rn} \\ y_{in} - y_{rn} \end{pmatrix} \tag{10}$$

$\boldsymbol{A}$ è una matrice contenente tutte tutte le matrici jacobiane relative alle configurazioni di un determinato caso.

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{J_1} \\ \vdots \\ \boldsymbol{J_n} \end{pmatrix} \tag{11}$$

$\partial \boldsymbol{l}$ è il vettore di dimensione l·1, dove l indica il numero delle aste di cui si è ipotizzato l'errore.

Anche la matrice A dell'equazione (11), però, non è quadrata ed è proprio a questo punto che entra in gioco l'algoritmo di calibrazione .

CAPITOLO 5 - ALGORITMO DI CALIBRAZIONE

Il ruolo centrale dello studio svolto è ricoperto dall'algoritmo di calibrazione.

In un processo di calibrazione standard, l'algoritmo ha il compito di ridurre al minimo l'errore che intercorre tra il modello matematico e il robot reale, andando ad aumentare l'accuratezza e la precisone dello stesso.

Anche l'algoritmo, però, non è un elemento ideale, non elimina l'errore al 100%.

Ci sono diversi algoritmi, di diversa complessità, utilizzati in questo campo, e ognuno ha una propria efficacia.

Verificare l'efficienza dell'algoritmo di calibrazione, ossia andare a valutare quanto questo sia efficace e se conviene o meno applicarlo è l'obiettivo di questo studio.

Il processo di verifica coincide con il processo inverso dello studio del robot, ossia partire dall'errore della posizione del terminale e ricavare l'errore sulla lunghezza delle aste.

Richiamando le equazioni (6) e (9) , come detto in precedenza entrambe le matrici $\boldsymbol{J}$ e $\boldsymbol{A}$ non sono invertibili, perché non quadrate, quindi non è possibile ottenere un risultato diretto, ed è questo il motivo per cui si parla di verifica.

Quello che si può fare, infatti, è quello di cercare una matrice inversa che meglio approssima la matrice $\boldsymbol{A}$. Si considera la matrice $\boldsymbol{A}$ invece che la matrice $\boldsymbol{J}$ poiché come prima cosa per l'applicazione dell'algoritmo è necessario che la matrice abbia rango massimo e questo non può essere se viene fatta una sola misurazione in quanto il rango massimo sarebbe uguale a 2, cioè la massima matrice quadrata contenuta nella matrice 2·5, e come seconda cosa l'approssimazione è tanto più efficace tanto più è elevato il numero di misurazioni effettuate, e questo spiega anche il perché delle diverse configurazioni effettuate precedentemente.

Data una matrice $\boldsymbol{A}$ di dimensioni n·m, una matrice $\boldsymbol{B}$ m·n è detta pseudo -inversa di $\boldsymbol{A}$ se verifica le seguenti quattro proprietà:

- $\boldsymbol{ABA} = \boldsymbol{A}$
- $\boldsymbol{BAB} = \boldsymbol{B}$
- $(\boldsymbol{AB})^T = \boldsymbol{AB}$
- $(\boldsymbol{BA})^T = \boldsymbol{BA}$

Data una matrice $\boldsymbol{A}$, esiste un'unica matrice pseudo-inversa che soddisfa queste proprietà.

Se la matrice $\boldsymbol{A}$ ha rango massimo esiste una semplice espressione algebrica per determinare la pseudo-inversa. In particolare, data la matrice $\boldsymbol{A}$ di dimensioni n·m e rango m, la matrice pseudo inversa di $\boldsymbol{A}$ è la matrice:

$$\boldsymbol{B} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T \tag{12}$$

L'espressione utilizzata è intesa come algoritmo in quanto è, appunto, un sistema sistematico di calcolo.

In Matlab il programma che restituisce come risultato la pseudo inversa è il comando "pinv".

Andando a considerare il primo caso di studio, la command window risulta essere:

```
clear all; clc;
%-Configurazione theta1=pi/6; theta4=(2*pi)/3
dx1=  [-0.124357543445946 ;  0.345362137662089] -[-0.124309521372560 ;  0.345310406881844];
%-Configurazione theta1=(3*pi)/2; theta4=pi/2
dx2=  [ -0.167644346610803 ;  0.167652991501097]-  [-0.167630546142402 ;  0.167630546142402];
%-Configurazione theta1=(5*pi)/8; theta4=(5*pi)/4
dx3= [0.109590940501112  ; 0.199432022113322]-[0.109573011465880  ; 0.199431904879566] ;
%-Configurazione theta1=(3*pi)/5; theta2=(5*pi)/3
dx4= [0.124683817491602  ; 0.179044269436324]-[0.124664816681780  ; 0.179055107165586];
%-Configurazione theta1=(7*pi)/4 theta4=(9*pi)/8
dx5= [ -0.211076557154329 ;  0.153215854148847]-[ -0.211046683186257  ; 0.153190752307846];
%-Configurazione theta1=(11*pi)/8; theta4=pi/12
dx6= [-0.025874053642013  ; 0.137433049525516]-[-0.025901105672631  ; 0.137443467466440];


 dx  =[dx1;dx2;dx3;dx4;dx5;dx6];
J1 =[0.473173847859758  , 2.983234599613208,  -2.476824024504726 ,  2.445470282659623 , -1.382515713982796; 0.180438854694034  , 1.137618735423782
J2 =[0.128487689350869 ,  1.024335566626997 , -0.129510587219157 ,  0.128487689350869 , -0.516245180427600; 0.016245180427600  , 0.129510587219156
J3 =[-0.521385758693502 , -0.302684842197962,   1.111706624476991  , 0.965978587972051 , -1.572136752604810;0.957916218408371   ,1.103945899679277
J4 =[-0.184409768520147 ,  0.184965047864143 , -0.973386732978281 ,  0.322597640772775 , 0.456634831503373;0.981108360575758 , -0.984062592399924 ,
J5 =[3.855862600199276 ,  3.859338630002672 , -3.195291198109372  , 0.702992342778286 , -2.342304410157813; 2.720974757849411 ,  2.723427695190104,
J6 =[ -0.034408739564185 ,  0.873425443892829,  -0.352008175219244 ,  0.252582038495254 , -0.319481125309102; 0.020554268757230 , -0.521745973277
 A = [J1;J2;J3;J4;J5;J6];
 B = pinv(A);
 dl = B*dx
```

Fig.6 - Command window di Matlab che racchiude operazioni per il calcolo della matrice $\boldsymbol{B}$.

Da cui si evince come l'input del comando è la matrice $\boldsymbol{A}$.

La diversificazione in casi differenti è servita per verificare se cambiando le condizioni operative l'algoritmo risultava essere più o meno efficiente e i risultati sono riportati nelle seguenti tabelle:

Tabella 8

dl	Caso 1	Caso 2	Caso 3
dl1	-0.2042	-0.3778	0.1553
dl2	-0.1713	-0.3921	-0.1367
dl3	-0.2530	0.0863	0.2499
dl4	-0.1701	0.8475	0.2205
dl5	-0.2240		

Questi dati sono tutti moltiplicati per 10^{-4} poiché l'unità di misura è il metro.

CAPITOLO 6 - DISCUSSIONE DEI RISULTATI

Utilizzando i risultati del capitolo precedente, è possibile andare a ricavare degli indici statistici che evidenziano in modo più chiaro le prestazioni dell'algoritmo nei diversi casi, per poter poi trarne delle conclusioni, ossia se conviene o meno utilizzarlo, oppure se ci sono delle determinate condizioni in cui l'algoritmo è più efficiente rispetto ad altre.

Tabella 9

Errore 1 (%)	Calibrazione1	Calibrazione 2	Calibrazione 3
p1	168.1%	1044.5%	55.3%
p2	134.3%	4021%	54.43%
p3	163.3%	89.73%	149.9%
p4	134.02%	69.5%	144.1%
p5	137.3%	0%	0%

Tabella 10

Errore 2 (%)	Calibrazione 1	Calibrazione 2	Calibrazione 3
p12	50.42%	41.78%	5.53%
p22	67.13%	40.21%	16.33%
p32	65.3%	75.37%	14.99%
p42	67.01%	79.75%	72.05%
p52	82.4%	0%	0%

Le tabelle evidenziano in modo chiaro l'errore commesso dall'algoritmo, espresso in termini percentuali. Dividendo ogni colonna in righe si è attribuito ad ogni singola asta del robot un valore distinto dell'errore, ogni riga infatti fa riferimento ad una specifica asta.

Il concetto di errore è fondamentale perché è il parametro caratteristico che permette di valutare positivamente o negativamente l'efficienza dell' algoritmo ed è quindi importante descriverne accuratamente il significato.

Partendo dalla lunghezza "ideale" di una generica asta del robot, si va ad ipotizzare un errore infinitesimo: chiamando *L* una generica lunghezza di una generica asta nelle condizioni ideali, si ipotizza un errore infinitesimo dl_r, ottenendo che la nuova lunghezza dell'asta sarà pari a $L+ dl_r$.

L'applicazione dell'algoritmo restituirà un errore infinitesimo dl_A, il quale va confrontato con l'errore ipotizzato, dl_R, e in base a quanto questi risultati sono prossimi tra loro si determina quanto l'algoritmo sia funzionante.

L'errore considerato nella prima tabella, rapporta la differenza tra l'errore reale e quello risultante dall'applicazione dell'algoritmo, all' errore ipotizzato ossia:

$$px = \left(\frac{dl_R - dl_A}{dl_R}\right) \cdot 100. \tag{13}$$

In questo modo è possibile vedere più chiaramente che nel caso in cui il numeratore sia maggiore di 1 il risultato è maggiore di 100% il che vuol dire che ci si sta allontanando dal valore ipotizzato. Questo porta alla conclusione che l'algoritmo non sta funzionando correttamente e che non è conveniente utilizzarlo.

L'errore considerato nella seconda tabella, invece, rappresenta proprio la differenza tra dl_R e dl_A in valore assoluto ossia:

$$px2 = (|dl_R - dl_A|) \cdot 100. \tag{14}$$

Si può notare dalla divisione delle tabelle che sono stati analizzati tre casi distinti:

I° CASO

Partendo dalla lunghezza ideale assegnata alle aste si è aggiunto un errore infinitesimo ad ogni asta dell'ordine di decimo di millimetro in particolare definendo:

$$dl_R = \begin{pmatrix} dlr_1 \\ dlr_2 \\ dlr_3 \\ dlr_4 \\ dlr_5 \end{pmatrix}$$

il vettore degli errori delle lunghezze ipotizzate, nel caso in questione si ha:

$$dl_R = 10^{-4} \cdot \begin{pmatrix} 0.3 \\ 0.5 \\ 0.4 \\ 0.5 \\ 0.6 \end{pmatrix} (m);$$

l'applicazione dell'algoritmo restituisce come risultato il seguente vettore:

$$dl_A = 10^{-4} \cdot \begin{pmatrix} -0.2042 \\ -0.1713 \\ -0.2530 \\ -0.1701 \\ -0.2240 \end{pmatrix} (m).$$

Applicando le formule (13) e (14) per ogni asta è stato possibile giungere ai risultati riportati nelle tabelle, i quali, mostrano che l'algoritmo non è stato efficace in quanto tutti i valori della prima tabella superano il 100%, il che vuol dire che l'algoritmo restituisce un valore che non solo non si avvicina, ma addirittura si allontana dalla soluzione voluta, ciò si evince anche dal fatto che il segno dei due vettori a confronto è discorde. Passando ai risultati riportati nella seconda tabella questi sono tutti superiori al 50% il che conferma il fatto che i risultati ottenuti sono molti discostanti tra di loro, poiché anche la differenza tra i valori numerici a prescindere dal segno è elevata.

II° CASO

Nel caso in questione il vettore degli errori ipotizzati è:

$$dl_R = 10^{-5} \cdot \begin{pmatrix} 0.4 \\ 0.1 \\ 8.4 \\ 0.5 \end{pmatrix} (m);$$

mentre il vettore restituito dopo l'applicazione dell'algoritmo è:

$$dl_A = 10^{-4} \cdot \begin{pmatrix} -0.3778 \\ -0.3921 \\ 0.0863 \\ 0.8475 \end{pmatrix} (m).$$

In questo secondo caso l'asta 5 è stata considerata invariata infatti dalle tabelle si evince che l'errore dell'asta 5 è pari a zero e si è considerato un errore del centesimo di millimetro tranne per l'asta 3. I risultati relativi alle aste 1 e 2 della seconda tabella potrebbero trarre in inganno poiché essendo circa il 41.78% e il 40.21% rispettivamente potrebbero indurre a pensare che l'algoritmo abbia dato un risultato ragionevole in quanto non eccessivamente elevato. In realtà entrambi questi risultati sono bassi perché è basso il termine più grande della sottrazione a cui viene aggiunto (poiché si hanno segni discordi) un termine di ordine inferiore, quindi quasi ininfluente. Andando a vedere ciò che è riportato nella tabella 1 questo aspetto viene messo ancora più in risalto. I valori ottenuti, infatti, sono altissimi e questo è appunto il risultato del fatto che non solo l'algoritmo ha fornito valori di segno opposto ma anche di un ordine di grandezza maggiore a quello ipotizzato, questo si riflette sull'ordine di grandezza dei dati percentuali. Per quanto riguarda le restanti aste nonostante i segni concordi e quindi valori per lo meno più vicini a quelli ipotizzati, infatti siamo al di sotto del 100%, la differenza dei risultati è comunque di un ordine di grandezza, infatti, dalla (13) si vede come, anche in questo caso, il valore percentuale ottenuto si avvicini molto a quello del membro più grande. L'algoritmo anche in questo caso è risultato inefficace.

III° CASO

Nel caso in questione il vettore degli errori ipotizzati è:

$$dl_R = 10^{-3} \cdot \begin{pmatrix} 0.1 \\ -0.3 \\ 0.1 \\ -0.5 \end{pmatrix} (m);$$

mentre il vettore restituito dopo l'applicazione dell'algoritmo è:

$$dl_A = 10^{-4} \cdot \begin{pmatrix} 0.1553 \\ -0.1367 \\ 0.2499 \\ 0.2205 \end{pmatrix} (m).$$

Anche per questo caso l'asta 5 è stata considerata invariata, mentre l'errore considerato è dell'ordine del millimetro. Iniziando dalle prime quattro aste si vede come i risultati ottenuti siano di segno concorde ai valori ipotizzati, infatti, i valori riportati sulla tabella 2 sono relativamente bassi. Per quanto riguarda, invece, i dati della prima tabella per le prime due aste otteniamo un errore inferiore al 50% e quindi l'algoritmo si è avvicinato alla condizione ipotizzata, mentre le restati due il valore supera il 100% quindi l'algoritmo non sta svolgendo correttamente il suo lavoro. I dati associati all'asta 4 sono spiegati dal fatto che nonostante la omogeneità dimensionale, i segni risultano comunque discordi, mentre quelli dell'asta 3 dal fatto che i valori numerici sono molto differenti.

CONCLUSIONI

In questo lavoro di tesi si è analizzato un algoritmo di calibrazione di tipo lineare, e la sua applicazione al pentalatero planare. L'algoritmo basa il suo funzionamento sulla minimizzazione dell'errore della matrice pseudo-inversa, ottenuta dalla misurazione di vare pose del robot. Partendo dall'analisi cinematica del robot, si è giunti a dei risultati finali dai quali sono stati ricavati degli indici statistici che mostrano le prestazioni dell'algoritmo applicato ai vari casi considerati.

L'algoritmo non ha, in linea generale, restituito valori prossimi a quelli ipotizzati, dai dati emerge, infatti, come a volte sia stato addirittura deviatorio, fornendo risultati che si allontanano molto dai dati iniziali. Dovendo confrontare i casi considerati, le prestazioni migliori sono state ottenute nel terzo caso, dove i valori sono più accettabili, anche se solo in parte, dei restanti due casi.

COMMENTI FINALI

Il lavoro proposto, derivante da un percorso di collaborazione, tra lo studente ed il professore, ha consentito un iniziale approccio tra esso e il mondo lavorativo. Al giorno d'oggi infatti il mercato dei robot è in veloce espansione e la calibrazione è un aspetto direttamente collegato ad essa. Essendo questo un campo di conoscenze troppo ampio, lo studente che veniva a contatto per la prima volta con questa area correlata al suo percorso di studi, ha focalizzato la sua attenzione su un determinato robot e un determinato algoritmo. Un risultato importante del tirocinio è quello di dare una nuova prospettiva allo studente che volendo può continuare su questa scia, ampliando le sue conoscenze relative a quel determinato settore. L'attività si è terminata con la stesura della tesi, un documento che racchiude tutti i passaggi logici e pratici dello studente, che hanno portato a partire dalla raccolta iniziale delle informazioni al raggiungimento delle conclusioni derivanti dai risultati ottenuti.

L'elemento di partenza per la scrittura del seguente documento è stata la raccolta dati, ricavati dall'attività svolta. Essendo considerati, infatti, tre casi separati, sono stati affrontati in maniera distinta in modo da avere un quadro più organizzato. Ogni documento è riferito ad un singolo caso e diviso nelle diverse configurazioni. Per ogni configurazione sono state registrate le coordinate del terminale nel caso ideale e reale, la differenza tra le due, la jacobiana caratteristica, ed infine nella parte finale di ogni foglio i risultati statistici riferiti al caso specifico.

In questo modo è stato anche più facile recuperare i dati da implementare in Matlab. Per essere inserite nel documento, poi, questi dati sono stati raggruppati in delle tabelle schematiche e compatte, da permettere una immediata visione e per poter essere discusse in modo opportuno.

BIBLIOGRAFIA

- Louis Vathan.B, Hoshila Kumar and Brighton Issac John.H. Kinematic. *Analysis of Five-Bar Mechanism inIndustrial Robotics*.

- Mooring, Benjamin W. *Fundamentals of manipulator calibration*. Publicazione a cura dwlla Wiley-lnterscience. Anno 1991.