



UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTA' DI INGEGNERIA

Corso di Laurea triennale in Ingegneria Meccanica

**ANALISI DI SENSIBILITA' DEI PARAMETRI DI UN MODELLO DI CALIBRAZIONE
CINEMATICA NELLA ROBOTICA PARALLELA**

**SENSITIVITY ANALYSIS OF THE PARAMETERS OF A KINEMATIC CALIBRATION
MODEL IN PARALLEL ROBOTICS**

Relatore:

Prof. Matteo Claudio Palpacelli

Tesi di Laurea di:

Antonio Leccese

A.A. 2019 / 2020

INDICE

PREMESSA	2
CAPITOLO 1 – PROFILI INTRODUTTIVI	4
CAPITOLO 2 – PENTALATERO PLANARE	5
CAPITOLO 3 – JACOBIANA DEL SISTEMA	11
CAPITOLO 4 – RACCOLTA DELLE MISURAZIONI SIMULATE	14
CAPITOLO 5 – ALGORITMO DI CALIBRAZIONE	27
CAPITOLO 6 – DISCUSSIONE DEI RISULTATI	31
CAPITOLO 7 – RISOLUZIONE DEI PROBLEMI	34
CONCLUSIONI	35
BIBLIOGRAFIA	36

PREMESSA

Il presente lavoro è frutto di un tirocinio curriculare svolto con l'obiettivo di valutare l'efficienza di un algoritmo di calibrazione di tipo lineare, applicato ad un pentalatero articolato planare con l'ausilio del software di calcolo Matlab, e confrontare i risultati ottenuti con quelli di un lavoro analogo, svolto in precedenza da un altro studente sullo stesso tipo di meccanismo. Quest'ultimo studio, infatti, ha presentato risultati che evidenziano una scarsa efficacia dell'algoritmo di calibrazione, nonostante la correttezza delle analisi condotte. Lo scopo è, quindi, quello di valutare quanto vari l'accuratezza dell'algoritmo al variare della scala dei valori di studio presi come input.

La fase iniziale dello studio ha riguardato l'analisi della cinematica del robot, partendo dalla definizione del corrispondente modello matematico, da cui è stato estrapolato un sistema di equazioni che, partendo da dati iniziali sulle dimensioni delle aste del robot definiti, ha permesso di risalire alla posizione del suo terminale (*end-effector*) nel piano. L'utilizzo di Matlab si è rivelato utile a fornire i risultati numerici cercati, nonché a mostrarli graficamente attraverso un comando che ha la funzione di plottare la particolare configurazione su un grafico, permettendo la chiara visione della posizione del terminale, delle aste e dei giunti.

Nella seconda fase, una volta messe in memoria diverse configurazioni del robot, è stata determinata la matrice Jacobiana del sistema, attraverso una serie di passaggi matematici svolti sempre attraverso il software.

Successivamente, per simulare la misura delle configurazioni (ovvero le diverse posizioni del terminale nel piano) di un robot reale affetto da errori geometrici, sono state ipotizzate diverse combinazioni di errori infinitesimi associati alla lunghezza delle aste; sono state, quindi, ricavate diverse configurazioni per ognuno di questi casi, utilizzando come dati di input gli angoli di inclinazione delle aste collegate ai motori del robot, utilizzati precedentemente per determinare le configurazioni nel caso ideale, e le lunghezze variate delle aste, ottenendo risultati che si discostano da quelli precedentemente ottenuti. Tale differenza rappresenta l'errore infinitesimo della posizione del terminale per il robot reale simulato.

Applicata tale operazione a ogni configurazione, è stato possibile definire un sistema del tipo:

$$\partial x = A \cdot \partial l$$

dove ∂x è il vettore che racchiude gli errori infinitesimi delle coordinate del terminale nelle varie configurazioni, A è la matrice che racchiude tutte le matrici Jacobiane relative alle varie configurazioni, mentre ∂l è il vettore che racchiude gli errori infinitesimi ipotizzati delle aste.

Lo scopo principale è quello di risalire all'errore delle aste ∂l , noto l'errore infinitesimo ∂x e la matrice A per ogni configurazione, e confrontarlo con quello ipotizzato, verificando quanto questi si avvicinino. Ciò si traduce nella risoluzione del sistema:

$$\partial l = B \cdot \partial x.$$

Nella determinazione della matrice B , chiamata pseudo-inversa, ovvero l'inversa che meglio approssima la matrice A , assume un ruolo fondamentale l'algoritmo di calibrazione. In base alla sua efficienza, infatti, si possono ottenere risultati più o meno prossimi a quelli ipotizzati.

Per valutare in quali casi l'utilizzo dell'algoritmo di calibrazione fosse utile, e in quali risultasse invece sconveniente, sono infine stati messi a confronto i risultati ottenuti ed è stata operata una stima percentuale dell'errore commesso dall'algoritmo nella calibrazione.

CAPITOLO 1 – PROFILI INTRODUTTIVI

Verso la fine degli anni '70 e l'inizio degli anni '80, crebbe l'interesse, in ambito industriale, circa l'utilizzo di robot manipolatori per la produzione automatizzata. Essi consistono in una serie di aste rigide collegate da giunti. Un'estremità del manipolatore è collegata a una superficie rigida detta "telaio", e costituisce la base del robot, mentre l'altra è equipaggiata con una piattaforma, detta "*end-effector*", che permette il montaggio di un utensile.

L'impiego dei robot manipolatori in ambito industriale ebbe inizialmente poca fortuna, in quanto vennero riscontrati numerosi problemi tecnici, riguardanti per lo più:

- l'impossibilità di programmarli *off-line*, cioè fuori dal loro ambiente di lavoro, in modo da eliminare i tempi di fermo macchina derivanti dalla programmazione in auto-apprendimento e studiare i molteplici scenari di una cella di lavoro prima di organizzare la produzione;
- l'impossibilità di condividere i programmi con altri robot.

Nel tentativo di risolvere queste problematiche, è stata posta molta attenzione verso il tema della calibrazione dei robot manipolatori.

L'obiettivo principale del robot è quello di spostare l'*end-effector* verso determinate posizioni o lungo specifiche traiettorie; risulta essere di primaria importanza, pertanto, che il manipolatore abbia un elevato grado di accuratezza. Il controllo del robot è affidato a un software che sfrutta un modello matematico ideale che lo rappresenta. La realizzazione fisica del robot, tuttavia, non è esente da errori, in quanto le macchine adottate per realizzarlo, per quanto precise, non potranno mai dare dimensioni perfettamente uguali a quelle considerate nel caso ideale. Ciò comporta una differenza tra le posizioni teoricamente assunte dal terminale, secondo il modello matematico ideale impostato, e quelle realmente occupate, e questo dà origine all'inaccuratezza del manipolatore. La calibrazione, quindi, è un processo che serve per accrescere l'accuratezza del robot, mediante la modifica del modello matematico utilizzato, al fine di simularne il comportamento reale. Per fare ciò, vengono utilizzati algoritmi di vario tipo e con diversi principi di funzionamento.

CAPITOLO 2 – PENTALATERO PLANARE

Il tipo di manipolatore studiato nell'ambito del lavoro di tirocinio di cui questa tesi è frutto è un robot parallelo, in particolare un pentalatero planare; per darne una definizione più precisa, si ricorda che un robot viene detto parallelo se la sua cinematica è basata esclusivamente su circuiti chiusi. Il robot è composto da 5 aste e 5 cerniere e il suo movimento avviene sul piano x-y. Il meccanismo ha due gradi di libertà ed è molto utilizzato negli ultimi anni nell'ambito dell'industria meccanica.

La seguente immagine è una rappresentazione semplificata del robot:

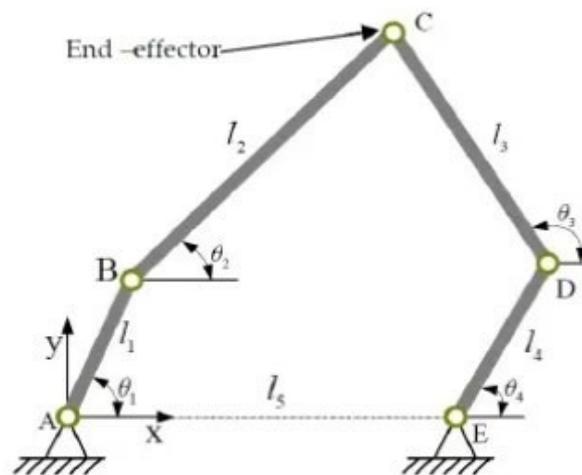


Fig.1 Schema cinematico del pentalatero planare.

Il pentalatero ha una parte fissa, ovvero l'asta 5, che funge da telaio, alla quale sono collegate, tramite due giunti motori posti alle sue estremità, altrettante aste, a loro volta collegate tramite giunti semplici ad altre due aste, che si incontrano, chiudendo la catena cinematica, nel giunto ove sarà collegata la piattaforma (*end-effector*) che permetterà il montaggio dell'utensile. Le aste 1-4 sono, quindi, mobili e sono dette "aste guida". Dalla figura si notano gli angoli caratteristici $\theta_1, \theta_2, \theta_3, \theta_4$; di questi, θ_1 e θ_4 , relativi ai due giunti motori, vengono scelti come gradi di libertà che permettono il movimento e il controllo del manipolatore.

Il primo step dello studio condotto riguarda l'esecuzione dell'analisi cinematica del pentalatero. Infatti, per poter eseguire la calibrazione del robot, risulta necessario ottenere un valido modello matematico del manipolatore, che traduca le informazioni sulle posizioni dei giunti in quelle sulla relativa posizione del terminale. Esistono due diverse modalità di esecuzione dell'analisi cinematica: una che sfrutta la cinematica diretta, permettendo di risalire alla posizione del terminale partendo dalle informazioni sugli angoli θ_1 e θ_4 , e un'altra che utilizza la cinematica inversa, dove sono i due angoli ad essere ricavati partendo dalle informazioni sulle coordinate della posizione del terminale.

Dalle relazioni geometriche applicabili al meccanismo, utilizzando la cinematica diretta, è possibile ricavare le equazioni delle coordinate di posizione del terminale, identificato dal punto P:

$$x_p = l_1 \cos\theta_1 + l_2 \cos\theta_2 = l_5 + l_4 \cos\theta_4 + l_3 \cos\theta_3 \quad (1)$$

$$y_p = l_1 \sin\theta_1 + l_2 \sin\theta_2 = l_3 \sin\theta_3 + l_4 \sin\theta_4 \quad (2)$$

Spostando il sistema di riferimento dalla cerniera attuata di sinistra al centro dell'asta 5, l'equazione (1) varia in questo modo:

$$x_p = l_1 \cos\theta_1 + l_2 \cos\theta_2 - \frac{l_5}{2} = l_4 \cos\theta_4 + l_3 \cos\theta_3 + \frac{l_5}{2} \quad (1')$$

Dalla (1') e dalla (2) si ha che θ_1 e θ_4 sono indipendenti nel sistema, e θ_2 e θ_3 possono essere determinate da θ_1 e θ_4 come segue:

$$\theta_3 = \left(2 \cdot \operatorname{atan} \left(\left(\left((-l_1^2 + 2 \cos(\theta_1 - \theta_4) l_1 l_4 + 2 \cos \theta_1 l_1 l_5 + l_2^2 + 2 l_2 l_3 + l_3^2 - l_4^2 - 2 \cos \theta_4 l_4 l_5 - l_5^2) \cdot (l_1^2 - 2 \cos(\theta_1 - \theta_4) l_1 l_4 - 2 \cos \theta_1 l_1 l_5 - l_2^2 + 2 l_2 l_3 - l_3^2 + l_4^2 + 2 \cos \theta_4 l_4 l_5 + l_5^2) \right)^{\frac{1}{2}} + 2 l_1 l_3 \sin \theta_1 - \left(\frac{2 l_3 l_4 \sin \theta_4}{l_1^2 + 2 \cos \theta_1 l_1 l_3 - 2 \cos(\theta_1 - \theta_4) l_1 l_4 - 2 \cos \theta_1 l_1 l_5 - l_2^2 + l_3^2 - 2 \cos \theta_4 l_4 l_3 - 2 l_3 l_5 + l_4^2 + 2 \cos \theta_4 l_4 l_5 + l_5^2} \right) \right) \right) \right) \quad (3)$$

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

$$\theta_2 = \operatorname{atan2}(l_3 \sin \theta_3 + l_4 \sin \theta_4 - l_1 \sin \theta_1, l_5 + l_4 \cos \theta_4 + l_3 \cos \theta_3 - l_1 \cos \theta_1) \quad (4)$$

Le equazioni (3) e (4) dipendono dagli angoli di attuazione θ_1 e θ_4 e dalla lunghezza delle aste del sistema.

Sostituendo le (3) e (4) nelle (1') e (2) si determinano le coordinate del terminale in funzione di θ_1 e θ_4 e della lunghezza delle aste, trovando diverse configurazioni del manipolatore a seconda dei casi.

La rappresentazione grafica delle varie configurazioni viene ottenuta mediante l'utilizzo del software Matlab, che permette, impostando la funzione "kindir_rig", di simulare come il robot si può muovere nel piano. Tale funzione ha come input due vettori:

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

$$\mathbf{q} = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} \theta_1 \\ \theta_4 \end{pmatrix}$$

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

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

A questo punto, utilizzando il comando “plot”, partendo dalle equazioni della cinematica descritte, Matlab fornisce una rappresentazione grafica del pentalatero, nelle varie configurazioni scelte. La *Command Window* del programma che racchiude le operazioni descritte è la seguente:

```

1  kindir_rig.m  +
2  function x = kindir_rig(q,geom)
3
4  %caso mostrato nell'articolosul pentalatero con tutte le aste differenti e
5  %sistema di riferimento centrato al 15/2
6
7  l1 = geom(1);
8  l2 = geom(2);
9  l3 = geom(3);
10 l4 = geom(4);
11 l5 = geom(5);
12
13 theta1 = q(1);
14 theta4 = q(2);
15
16 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)*(l1^2 - 2*cos(theta1)
17
18 theta3 = theta3_vett(2);
19
20 theta2 = atan2(l3*sin(theta3)+l4*sin(theta4)-l1*sin(theta1),l5+l4*cos(theta4)+l3*cos(theta3)-l1*cos(theta1));
21
22 x(1) = l1*cos(theta1)+l2*cos(theta2)-l5/2;
23 x(2) = l1*sin(theta1)+l2*sin(theta2);
24
25 plot([-15/2,-15/2+l1*cos(theta1),-15/2+l1*cos(theta1)+l2*cos(theta2)], [0,l1*sin(theta1),l1*sin(theta1)+l2*sin(theta2)],'r'); hold on;
26 plot([15/2,15/2+l4*cos(theta4),15/2+l4*cos(theta4)+l3*cos(theta3)], [0,l4*sin(theta4),l4*sin(theta4)+l3*sin(theta3)],'b');
27
28 axis equal;
29 format long

```

Fig. 2 Command Window di Matlab per le funzioni kindir_rig e plot.

Il pentalatero viene considerato come composto da due diversi rami, uno a destra e uno a sinistra, uniti a formare il meccanismo. Il comando “plot” ha come input i vettori delle coordinate x e y del singolo ramo, quindi verranno eseguiti due plot per rappresentare entrambi i rami, il primo per la parte sinistra, il secondo per quella destra. All’interno della parentesi sono riportate tutte le coordinate dei punti del relativo ramo, che vengono uniti per ottenere il risultato voluto.

I dati iniziali considerati per l’analisi del robot sono:

$$l_1 = 0.130 \text{ m}$$

$$l_2 = 0.300 \text{ m}$$

$$l_3 = 0.300 \text{ m}$$

$$l_4 = 0.130 \text{ m}$$

$$l_5 = 0.260 \text{ m}$$

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

Ad esempio, scegliendo come angoli $\theta_1 = \frac{\pi}{2}$ e $\theta_4 = \frac{\pi}{2}$ e mandando in run il programma, si ottiene:

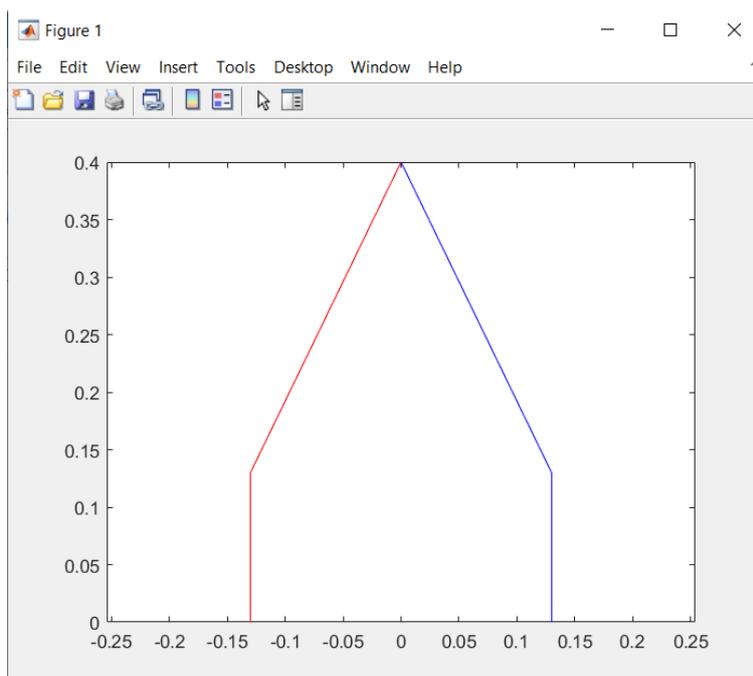


Fig. 3 Grafico di rappresentazione schematica del robot nella configurazione scelta.

Le coordinate del terminale risultanti per tale configurazione sono:

$$x_p = 0$$

$$y_p = 0.4004$$

In conclusione, l'analisi cinematica e l'utilizzo di Matlab permettono di ricavare - partendo dalla geometria del robot e dagli angoli di input, che sono i gradi di libertà del meccanismo - la posizione del terminale per qualsiasi delle configurazioni che il manipolatore può assumere. Questo risultato è molto importante e si rivela assai utile ai fini della calibrazione.

CAPITOLO 3 – JACOBIANA DEL SISTEMA

Un passaggio fondamentale nello studio del robot è la determinazione della matrice Jacobiana del sistema.

Sia $\mathbf{f}: U \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ una funzione definita su un insieme aperto U dello spazio euclideo \mathbb{R}^n . Allora la matrice Jacobiana della funzione \mathbf{f} in $\mathbf{x} = (x_1, \dots, x_n)$ è la matrice delle derivate parziali prime della funzione calcolate in \mathbf{x} :

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

scritta anche come:

$$(J\mathbf{f})_{ij} = \frac{\partial f_i(\mathbf{x})}{\partial x_j} \quad (5)$$

In Matlab il comando che permette di ricavare la matrice Jacobiana è “jacobian (\mathbf{f}, \mathbf{x})”, dove:

- \mathbf{f} rappresenta la funzione di input, che può essere scalare o vettoriale;
- \mathbf{x} è il vettore delle variabili rispetto alle quali viene calcolata la matrice.

Nel caso del pentalatero planare si ha che:

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

quindi, la funzione è un vettore contenente le equazioni che indicano la posizione del terminale;

$$\mathbf{x} = \begin{pmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \end{pmatrix},$$

che è il vettore contenente le variabili di derivazione.

L'equazione della Jacobiana del sistema si ottiene, quindi, partendo dalla funzione f e derivando entrambe le equazioni rispetto a \mathbf{x} :

$$J = \begin{pmatrix} \frac{\partial x_p}{\partial l_1} & \dots & \frac{\partial x_p}{\partial l_n} \\ \frac{\partial y_p}{\partial l_1} & \dots & \frac{\partial y_p}{\partial l_n} \end{pmatrix},$$

$n = 1, \dots, 5$ e varia a seconda del numero di aste considerate come variabili.

Tale procedimento, in Matlab, è eseguito mediante i seguenti passaggi:

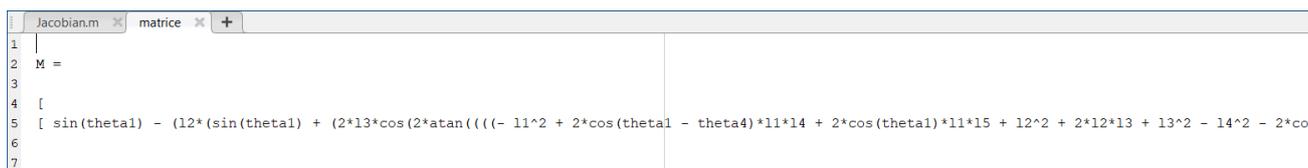


```

1 - | syms l1 l2 l3 l4 l5 theta1 theta4 theta5 real
2 - |
3 - | theta3 = 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)*(l1^2 -
4 - |
5 - | theta2 = atan((l3*sin(theta3)+l4*sin(theta4)-l1*sin(theta1))/(l5+l4*cos(theta4)+l3*cos(theta3)-l1*cos(theta1)));
6 - |
7 - | x_c = l1*cos(theta1)+l2*cos(theta2)-l5/2;
8 - | y_c = l1*sin(theta1)+l2*sin(theta2);
9 - |
10 - | M = jacobian([x_c;y_c],[l1;l2;l3;l4;l5]);
11 - |
12 - | diary matrice
13 - |
14 - | M
15 - |
16 - | diary off

```

Fig. 4.a Command Window di Matlab per il calcolo della Jacobiana.



```

1 - |
2 - | M =
3 - |
4 - | [
5 - | sin(theta1) - (l2*(sin(theta1) + (2*l3*cos(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
6 - |
7 - |

```

Fig. 4.b Matrice Jacobiana.

Mandando in *run* la funzione rappresentata si ottiene M , ovvero la matrice 2·5 che corrisponde alla Jacobiana del sistema, in funzione delle lunghezze delle aste e degli angoli θ_1 e θ_4 . Sostituendo, poi, i valori geometrici e di configurazione desiderati, si ottengono le relative matrici in forma numerica.

La matrice Jacobiana ottenuta risulta essere molto utile per ricavare la forma differenziale delle equazioni cinematiche.

Infatti, applicando al pentalatero la (5), si può scrivere:

$$\mathbf{J} = \frac{\partial \mathbf{x}}{\partial \mathbf{l}} \quad (6)$$

o anche

$$\partial \mathbf{x} = \mathbf{J} \cdot \partial \mathbf{l} \quad (7)$$

Dal punto di vista pratico:

- $\partial \mathbf{x}$ rappresenta il vettore di dimensione 2·1 degli errori infinitesimi delle coordinate del terminale;
- \mathbf{J} è la matrice Jacobiana di dimensione 2· l , dove l è il numero delle variabili alle quali sono riferite le derivate parziali;
- $\partial \mathbf{l}$ è il vettore di dimensione n ·1 che contiene le lunghezze delle aste intese come variabili. Tale vettore non è da considerarsi sempre di dimensione 5·1 perché non in tutti i casi le aste vengono considerate tutte variabili in lunghezza.

Questo risultato è di fondamentale importanza nel processo di calibrazione, in quanto permette di correlare gli errori della posizione del terminale con quelli della lunghezza delle aste. Risulta, però, necessario raccogliere dati, tramite misurazioni, che rappresentino i termini dell'equazione, che sono necessari per ottenere la matrice pseudo-inversa.

CAPITOLO 4 – RACCOLTA DELLE MISURAZIONI SIMULATE

La fase successiva dello studio è caratterizzata dall'esecuzione di una serie di simulazioni di misurazioni volte a determinare la posizione del terminale a seconda della configurazione assunta dal robot, sfruttando le equazioni della cinematica diretta.

Tuttavia, le coordinate fornite dal modello matematico possono differire da quelle assunte dal robot nel caso reale, perché la realizzazione fisica delle aste del manipolatore non darà risultati identici a quelli considerati nel caso ideale cui si fa riferimento. Questo è dovuto al fatto che le macchine utilizzate per la realizzazione delle componenti del manipolatore non possono eseguire lavorazioni con una precisione tale da realizzare pezzi di dimensioni esattamente uguali a quelle nominali.

Risulta, quindi, importante realizzare un set di misurazioni simulate (e non sperimentali) abbastanza ampio, in modo da poter confrontare i numerosi risultati reali ottenuti con quelli ideali corrispondenti. Nella pratica, vengono innanzitutto ipotizzati degli errori sulla lunghezza delle aste, in modo da simulare il comportamento reale del robot, dopodiché si procede con l'esecuzione delle varie misurazioni, utilizzando come dati di input gli stessi angoli usati nel caso ideale, in modo da ottenere le relative informazioni sulle posizioni assunte dal terminale. Tali dati vengono, infine, confrontati con i corrispondenti risultati del caso ideale e viene evidenziato quanto essi differiscano tra loro.

Anche in questo caso, l'utilizzo, su Matlab, della funzione "kindir_rig" si rivela essere di particolare aiuto. Infatti, dopo aver inserito i dati di input sulle lunghezze delle aste nel caso reale e sugli angoli specifici per ogni configurazione, essa permette non solo di ricavare le informazioni sulla posizione del terminale e la rappresentazione grafica del robot tramite il comando "plot", ma anche di individuare eventuali configurazioni critiche.

CASO IDEALE

Considerando le lunghezze delle aste del robot pari a

$$l_1 = 0.130 \text{ m}$$

$$l_2 = 0.300 \text{ m}$$

$$l_3 = 0.300 \text{ m}$$

$$l_4 = 0.130 \text{ m}$$

$$l_5 = 0.260 \text{ m}$$

e scelte diverse configurazioni, le relative posizioni del terminale P sono:

Tabella 1

θ_1	θ_4	x_p (m)	y_p (m)
$\pi/6$	$2\pi/3$	-0.124310	0.345310
$3\pi/2$	$\pi/2$	-0.167631	0.167631
$5\pi/8$	$5\pi/4$	0.109573	0.199432
$3\pi/5$	$5\pi/3$	0.124665	0.179055
$7\pi/4$	$9\pi/8$	-0.211047	0.153191
$11\pi/8$	$\pi/12$	-0.025901	0.137443
$\pi/8$	$11\pi/6$	0.226416	0.234564
$7\pi/6$	$3\pi/2$	-0.016266	0.131928
$\pi/10$	$9\pi/5$	0.231059	0.223560

La rappresentazione grafica della prima configurazione riportata è:

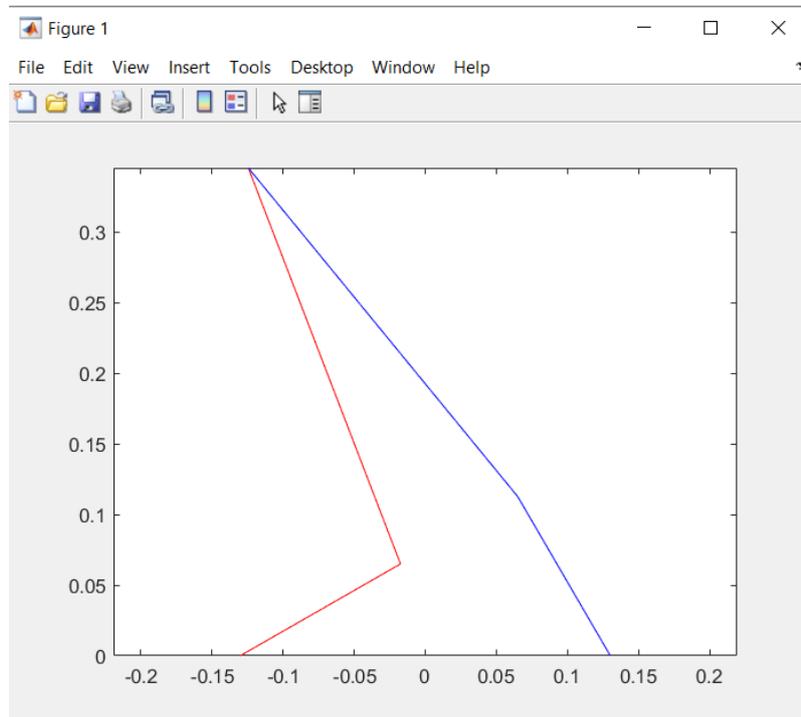


Fig. 5 *Rappresentazione grafica del robot nella prima configurazione.*

Memorizzate tutte le configurazioni nel caso ideale, si fa lo stesso per il caso reale, considerando i dati delle lunghezze delle aste soggetti a errori infinitesimi, dovuti alle piccole imprecisioni inevitabilmente presenti nella realizzazione dei pezzi. Difatti, seppure la realizzazione fisica del robot comporti la formazione di errori, essi saranno di entità molto piccola per ottenere risultati di alta precisione.

Per effettuare uno studio approfondito e accurato sono necessari gli studi di diverse condizioni iniziali delle dimensioni delle aste, e quindi diversi vettori $\partial \mathbf{l}$ degli errori infinitesimi ipotizzati.

Nello studio che ha portato alla realizzazione di questa tesi, sono state analizzate tre diverse casistiche:

$$1. \partial \mathbf{l} = 10^{-3} \cdot \begin{pmatrix} 0.2 \\ -0.5 \\ 0.3 \\ -0.1 \end{pmatrix} (m)$$

Il dato di ingresso della funzione è quindi $\mathbf{l} + \partial \mathbf{l}$, ovvero il vettore delle lunghezze nominali, più l'errore infinitesimo ipotizzato. Ripetendo gli stessi procedimenti fatti per il caso ideale attraverso la funzione "kindir_rig", si ottengono i seguenti risultati:

Tabella 2

θ_1	θ_4	x_p (m)	y_p (m)
$\pi/6$	$2\pi/3$	-0.126069	0.344129
$3\pi/2$	$\pi/2$	-0.168015	0.166878
$5\pi/8$	$5\pi/4$	0.108873	0.199999
$3\pi/5$	$5\pi/3$	0.124080	0.179321
$7\pi/4$	$9\pi/8$	-0.214241	0.150043
$11\pi/8$	$\pi/12$	-0.026480	0.136976
$\pi/8$	$11\pi/6$	0.225775	0.234884
$7\pi/6$	$3\pi/2$	-0.017147	0.131878
$\pi/10$	$9\pi/5$	0.230393	0.223910

$$2. \partial l = 10^{-3} \cdot \begin{pmatrix} 0.1 \\ 0.3 \\ 0.2 \\ 0.3 \\ -0.5 \end{pmatrix} (m)$$

Tabella 3

θ_1	θ_4	x_p (m)	y_p (m)
$\pi/6$	$2\pi/3$	-0.125705	0.345016
$3\pi/2$	$\pi/2$	-0.168106	0.167740
$5\pi/8$	$5\pi/4$	0.110196	0.199157
$3\pi/5$	$5\pi/3$	0.125189	0.179151
$7\pi/4$	$9\pi/8$	-0.212847	0.151979
$11\pi/8$	$\pi/12$	-0.025951	0.137856
$\pi/8$	$11\pi/6$	0.227102	0.2346500
$7\pi/6$	$3\pi/2$	-0.015958	0.132168
$\pi/10$	$9\pi/5$	0.231782	0.223593

$$3. \partial l = 10^{-4} \cdot \begin{pmatrix} 0.5 \\ 0.6 \\ -0.2 \\ 0.3 \\ -0.3 \end{pmatrix} (m)$$

Tabella 4

θ_1	θ_4	x_p (m)	y_p (m)
$\pi/6$	$2\pi/3$	-0.124211	0.345415
$3\pi/2$	$\pi/2$	-0.167628	0.167639
$5\pi/8$	$5\pi/4$	0.109664	0.199359
$3\pi/5$	$5\pi/3$	0.124740	0.179027
$7\pi/4$	$9\pi/8$	-0.211054	0.153188
$11\pi/8$	$\pi/12$	-0.025868	0.137445
$\pi/8$	$11\pi/6$	0.226589	0.234538
$7\pi/6$	$3\pi/2$	-0.016220	0.131909
$\pi/10$	$9\pi/5$	0.231237	0.223524

Utilizzando, in Matlab, il comando “format_long”, è possibile mostrare le cifre decimali più piccole di ogni risultato, in modo da evidenziare in maniera più precisa le differenze tra di essi.

Calcolando le differenze, per ciascuna configurazione, tra i risultati ottenuti nel caso ideale e quelli dei vari casi reali ipotizzati, si ottiene il vettore degli errori infinitesimi ∂x , espresso dalla relazione:

$$\partial x = J \cdot \partial l$$

Questa operazione è applicata a tutti i tre casi reali ipotizzati:

1.

Tabella 5

θ_1	θ_4	∂x	∂y
$\pi/6$	$2\pi/3$	0.176	0.118
$3\pi/2$	$\pi/2$	0.384	0.753
$5\pi/8$	$5\pi/4$	0.700	-0.568
$3\pi/5$	$5\pi/3$	0.585	-0.266
$7\pi/4$	$9\pi/8$	0.319	0.315
$11\pi/8$	$\pi/12$	0.579	0.467
$\pi/8$	$11\pi/6$	0.641	-0.320
$7\pi/6$	$3\pi/2$	0.881	0.049
$\pi/10$	$9\pi/5$	0.666	-0.350

Tutti i valori riportati nella tabella sono moltiplicati per 10^{-3} per tener conto dell'unità di misura in metri, ad eccezione delle configurazioni 1 e 5, dove il moltiplicatore è 10^{-2} .

2.

Tabella 6

θ_1	θ_4	∂x	∂y
$\pi/6$	$2\pi/3$	0.139	0.029
$3\pi/2$	$\pi/2$	0.476	-0.110
$5\pi/8$	$5\pi/4$	-0.623	0.274
$3\pi/5$	$5\pi/3$	-0.524	-0.096
$7\pi/4$	$9\pi/8$	0.180	0.121
$11\pi/8$	$\pi/12$	0.050	-0.413
$\pi/8$	$11\pi/6$	-0.686	-0.086
$7\pi/6$	$3\pi/2$	-0.309	-0.240
$\pi/10$	$9\pi/5$	-0.723	-0.033

Tutti i valori riportati nella tabella sono moltiplicati per 10^{-3} per tener conto dell'unità di misura in metri, ad eccezione delle configurazioni 1 e 5, dove il moltiplicatore è 10^{-2} .

3.

Tabella 7

θ_1	θ_4	∂x	∂y
$\pi/6$	$2\pi/3$	-0.098	-0.104
$3\pi/2$	$\pi/2$	-0.002	-0.009
$5\pi/8$	$5\pi/4$	-0.091	0.073
$3\pi/5$	$5\pi/3$	-0.075	0.029
$7\pi/4$	$9\pi/8$	0.008	0.003
$11\pi/8$	$\pi/12$	-0.033	-0.002
$\pi/8$	$11\pi/6$	-0.173	0.026
$7\pi/6$	$3\pi/2$	-0.046	0.019
$\pi/10$	$9\pi/5$	-0.178	0.035

Tutti i valori riportati nella tabella sono moltiplicati per 10^{-3} per tener conto dell'unità di misura in metri.

Come illustrato nell'introduzione, questa tesi nasce con l'obiettivo di evidenziare le differenze che emergono dal confronto tra i risultati ottenuti dal lavoro di tirocinio e quelli ricavati da un lavoro precedentemente svolto da un altro studente, al fine di valutare quanto il variare delle condizioni iniziali considerate, ovvero gli errori ipotizzati sulle aste del robot, influisca sull'efficienza dell'algoritmo di calibrazione studiato.

Le casistiche considerate per tale lavoro sono:

$$1. \partial \mathbf{l} = 10^{-4} \cdot \begin{pmatrix} 0.3 \\ 0.5 \\ 0.4 \\ 0.5 \\ 0.6 \end{pmatrix} (m)$$

Tabella 8

θ_1	θ_4	$x_p (m)$	$y_p (m)$
$\pi/6$	$2\pi/3$	-0.124357	0.345362
$3\pi/2$	$\pi/2$	-0.167644	0.167652
$5\pi/8$	$5\pi/4$	0.109590	0.199432
$3\pi/5$	$5\pi/3$	0.124683	0.179044
$7\pi/4$	$9\pi/8$	-0.211076	0.153215
$11\pi/8$	$\pi/12$	-0.025874	0.137433

$$2. \partial \mathbf{l} = 10^{-5} \cdot \begin{pmatrix} 0.4 \\ 0.1 \\ 8.4 \\ 0.5 \end{pmatrix} (m)$$

Tabella 9

θ_1	θ_4	$x_p (m)$	$y_p (m)$
$\pi/6$	$2\pi/3$	-0.124570	0.345212
$3\pi/2$	$\pi/2$	-0.167717	0.167616
$5\pi/8$	$5\pi/4$	0.109549	0.199519
$3\pi/5$	$5\pi/3$	0.124650	0.179133
$7\pi/4$	$9\pi/8$	-0.211388	0.152945
$11\pi/8$	$\pi/12$	-0.025971	0.137482
$\pi/8$	$11\pi/6$	0.226361	0.234642
$7\pi/6$	$3\pi/2$	-0.016322	0.131987
$\pi/10$	$9\pi/5$	0.231002	0.223639

$$3. \partial \mathbf{l} = 10^{-3} \cdot \begin{pmatrix} 0.1 \\ -0.3 \\ 0.1 \\ -0.5 \end{pmatrix} (m)$$

Tabella 10

θ_1	θ_4	$x_p (m)$	$y_p (m)$
$\pi/6$	$2\pi/3$	-0.123838	0.345185
$3\pi/2$	$\pi/2$	-0.167720	0.167616
$5\pi/8$	$5\pi/4$	0.109065	0.200098
$3\pi/5$	$5\pi/3$	0.124252	0.179551
$7\pi/4$	$9\pi/8$	-0.211880	0.152110
$11\pi/8$	$\pi/12$	-0.026485	0.137327
$\pi/8$	$11\pi/6$	0.225887	0.234909
$7\pi/6$	$3\pi/2$	-0.016981	0.132142
$\pi/10$	$9\pi/5$	0.230495	0.223951

da cui:

1.

Tabella 11

θ_1	θ_4	∂x	∂y
$\pi/6$	$2\pi/3$	-0.48	0.52
$3\pi/2$	$\pi/2$	-0.14	0.22
$5\pi/8$	$5\pi/4$	0.18	0.001
$3\pi/5$	$5\pi/3$	0.19	-0.11
$7\pi/4$	$9\pi/8$	-0.30	0.25
$11\pi/8$	$\pi/12$	0.27	-0.10

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

2.

Tabella 12

θ_1	θ_4	∂x	∂y
$\pi/6$	$2\pi/3$	0.26	0.10
$3\pi/2$	$\pi/2$	-0.87	-0.14
$5\pi/8$	$5\pi/4$	-0.24	0.88
$3\pi/5$	$5\pi/3$	-0.14	0.78
$7\pi/4$	$9\pi/8$	-0.34	-0.25
$11\pi/8$	$\pi/12$	-0.71	0.39
$\pi/8$	$11\pi/6$	-0.55	0.78
$7\pi/6$	$3\pi/2$	-0.56	0.60
$\pi/10$	$9\pi/5$	-0.56	0.80

Tutti i valori riportati nella tabella sono moltiplicati per 10^{-4} per tener conto dell'unità di misura in metri, ad eccezione delle configurazioni 1 e 5, nelle quali il moltiplicatore è 10^{-3} .

3.

Tabella 13

θ_1	θ_4	∂x	∂y
$\pi/6$	$2\pi/3$	0.471	-0.125
$3\pi/2$	$\pi/2$	-0.090	-0.015
$5\pi/8$	$5\pi/4$	-0.508	0.666
$3\pi/5$	$5\pi/3$	-0.412	0.497
$7\pi/4$	$9\pi/8$	-0.083	-0.108
$11\pi/8$	$\pi/12$	-0.585	-0.116
$\pi/8$	$11\pi/6$	-0.529	0.345
$7\pi/6$	$3\pi/2$	-0.715	0.214
$\pi/10$	$9\pi/5$	-0.564	0.392

Tutti i valori riportati nella tabella sono moltiplicati per 10^{-3} per tener conto dell'unità di misura in metri, ad eccezione della configurazione 5, per la quale il moltiplicatore è 10^{-2} .

Dal confronto si evince che, per errori ipotizzati dello stesso ordine, i risultati ottenuti nei due studi sono simili. Ciò fornisce un primo importante indizio sul risultato della comparazione dei due lavori condotti.

Andando avanti con lo studio, prendendo in considerazione i risultati ottenuti in precedenza, si può vedere che, come illustrato nel capitolo 3, utilizzando l'equazione della Jacobiana, implementandola in Matlab e sostituendo, per ogni caso di dimensioni delle aste ipotizzato, gli angoli relativi a ciascuna configurazione, è possibile calcolare tutte le relative matrici.

Nel caso della prima configurazione del primo caso si ha:

$$J_1 = \begin{pmatrix} 0.4470 & 2.9987 & -2.5006 & 2.4652 \\ 0.1658 & 1.1124 & -1.9942 & 1.9660 \end{pmatrix}$$

La matrice ha dimensione 2·4 perché in tal caso sono state considerate come variabili solo quattro delle cinque lunghezze delle aste, mentre le equazioni di riferimento sono due.

Nel secondo o terzo caso, invece, sono variabili tutte e cinque le lunghezze delle aste, quindi si ottiene una matrice 2·5 come, ad esempio, per la prima configurazione del secondo caso:

$$J_1 = \begin{pmatrix} 0.4919 & 2.9929 & -2.4805 & 2.4522 & -1.4021 \\ 0.1920 & 1.1684 & -2.0419 & 2.0185 & -0.7426 \end{pmatrix}$$

In conclusione, tramite i procedimenti illustrati, è possibile, partendo dagli errori ipotizzati sulle aste, e con l'utilizzo della cinematica diretta, risalire agli errori delle coordinate del terminale per ciascun caso e configurazione considerati.

A rigor di logica, dovrebbe essere possibile eseguire il processo anche nel verso opposto; questo significa partire dagli errori delle coordinate del terminale e risalire agli errori delle aste.

In altri termini, ciò significa, data la relazione:

$$\partial x = J \cdot \partial l,$$

ricavare il vettore ∂l conoscendo ∂x . In termini matematici ciò si traduce nell'invertire l'equazione:

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

L'equazione precedente, in realtà, non ha senso matematico, in quanto la matrice J non è quadrata, ma di dimensione $2 \cdot l$, e quindi non è invertibile.

Per ovviare a questa problematica si costruisce un sistema del tipo:

$$\partial \mathbf{x}_1 = \mathbf{A} \cdot \partial \mathbf{l} \quad (9)$$

$\partial \mathbf{x}_1$ è un vettore di dimensione $n \cdot 1$ dove n è uguale al numero di configurazioni analizzate moltiplicato per due, poiché per ognuna di esse si considera la differenza tra le coordinate di posizione del terminale ideali e quelle del caso reale sia per x che per y :

$$\partial \mathbf{x}_1 = \begin{pmatrix} x_{i1} - x_{r1} \\ y_{i1} - y_{r1} \\ \vdots \\ x_{in} - x_{rn} \\ y_{in} - y_{rn} \end{pmatrix} \quad (10)$$

\mathbf{A} è una matrice contenente tutte le matrici Jacobiane relative alle configurazioni di un determinato caso:

$$\mathbf{A} = \begin{pmatrix} J_1 \\ \vdots \\ J_n \end{pmatrix} \quad (11)$$

$\partial \mathbf{l}$ è il vettore di dimensione $l \cdot 1$, dove l è il numero di aste di cui si è ipotizzato l'errore.

Tuttavia, essendo anche \mathbf{A} una matrice non quadrata, essa non è invertibile. A questo punto entra in gioco l'algoritmo di calibrazione.

CAPITOLO 5 – ALGORITMO DI CALIBRAZIONE

L'algoritmo di calibrazione assume, nello studio svolto, un punto centrale. Infatti, il processo di calibrazione permette di ridurre al minimo l'errore che intercorre tra il robot reale e il relativo modello matematico ideale. Ciò influisce, di conseguenza, sulla precisione e l'accuratezza del manipolatore.

Tuttavia, anche la calibrazione non è un processo ideale, e quindi non è in grado di eliminare completamente l'errore.

Esistono diverse tipologie di algoritmi di calibrazione che differiscono tra loro per efficacia, nonché per tipologia di modello matematico utilizzato. Si può, infatti, fare una distinzione tra gli algoritmi di tipo lineare e non lineare:

Per i primi si utilizza un modello matematico lineare e si fa ricorso alla matrice Jacobiana; esistono diverse tecniche di identificazione lineare (ovvero i procedimenti utilizzati per risalire agli errori presenti nel modello reale del robot), tra le quali si distingue la stima dei minimi quadrati lineari. Nello studio effettuato per la realizzazione di questa tesi, è stato utilizzato un particolare metodo di identificazione che rientra tra quelli contemplati nell'insieme delle tecniche sopracitate, ovvero il **metodo della pseudo-inversa**; esso prevede l'utilizzo della matrice pseudo-inversa, che è l'inversa che meglio approssima la matrice contenente le Jacobiane di tutte le configurazioni del robot per un determinato caso.

I secondi, invece, sfruttano modelli matematici non lineari, che non necessariamente implicano l'utilizzo della matrice Jacobiana. Esistono anche per questa tipologia di algoritmi vari metodi, ad esempio:

- **Metodo di ricerca diretta**

Utilizzato per risolvere problemi di ottimizzazione, non richiede l'utilizzo della matrice Jacobiana, nonché informazioni sul gradiente o su derivati superiori della funzione obiettivo; si preferisce quando la funzione obiettivo non è differenziabile o non è neanche continua.

- **Metodi del gradiente**

Utilizzati in molte situazioni pratiche di calibrazione, quando i vincoli sugli errori dei parametri cinematici vengono trascurati; richiede l'utilizzo della matrice Jacobiana.

Come illustrato nel capitolo 4, avendo a che fare con un modello lineare, il processo di calibrazione da utilizzare deve permettere di ricavare gli errori delle lunghezze delle aste partendo da quelli di posizione del terminale, ricercando la matrice inversa che meglio approssima la matrice \mathbf{A} . Questo perché \mathbf{A} , non essendo quadrata, non è invertibile. La scelta di ricercare la pseudo-inversa della \mathbf{A} , piuttosto che della \mathbf{J} , sta nel fatto che, per prima cosa, la matrice deve avere rango massimo, e quindi non sarebbe possibile effettuare una calibrazione facendo una sola misurazione, in quanto il rango massimo sarebbe uguale a 2, cioè la massima matrice quadrata contenuta nella $2 \cdot 5$; inoltre, \mathbf{A} racchiude le \mathbf{J} di più configurazioni, quindi rappresenta un elevato numero di misurazioni, che è proprio la condizione necessaria affinché l'approssimazione sia efficace.

Il metodo di ottenimento della matrice pseudo-inversa è il seguente:

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

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

Data una matrice \mathbf{A} , esiste un'unica matrice pseudo-inversa che verifica le precedenti proprietà.

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

$$\mathbf{B} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \quad (12)$$

In Matlab, il comando che fornisce la pseudo-inversa è “pinv”. Andando a considerare il primo caso studiato si ha:

```

1  %-Configurazione theta1=pi/6; theta4= pi*(2/3)
2  dx1 = [-0.126069160115904 ; 0.344129073847372] - [-0.124309521372560 ; 0.345310406881844];
3  %-Configurazione theta1=pi*(3/2); theta4=pi/2
4  dx2 = [-0.168014668765464 ; 0.166877658127723] - [-0.167630546142402 ; 0.167630546142402];
5  %-Configurazione theta1=pi*(5/8); theta4=pi*(5/4)
6  dx3 = [0.108872574688449 ; 0.199999461310617] - [0.109573011465880 ; 0.199431904879566];
7  %-Configurazione theta1=pi*(3/5); theta4=pi*(5/3)
8  dx4 = [0.124079994975986 ; 0.179320935546834] - [0.124664816681780 ; 0.179055107165586];
9  %-Configurazione theta1=pi*(7/4); theta4=pi*(9/8)
10 dx5 = [-0.214241053258387 ; 0.150042769599401] - [-0.211046683186257 ; 0.153190752307846];
11 %-Configurazione theta1=pi*(11/8); theta4=pi/12
12 dx6 = [-0.026479683786182 ; 0.136976013800500] - [-0.025901105672631 ; 0.137443467466440];
13 %-Configurazione theta1=pi/8; theta4=pi*(11/6)
14 dx7 = [0.225774851782682 ; 0.234884070435325] - [0.226416285221403 ; 0.234564062518475];
15 %-Configurazione theta1=pi*(7/6); theta4=pi*(3/2)
16 dx8 = [-0.017146746588780 ; 0.131878389039923] - [-0.016266213508123 ; 0.131927842708629];
17 %-Configurazione theta1=pi/10; theta4=pi*(9/5)
18 dx9 = [0.230393408332785 ; 0.223909946463563] - [0.231058958823069 ; 0.223559717827923];
19
20 dx = [dx1;dx2;dx3;dx4;dx5;dx6;dx7;dx8;dx9];
21 J1 = [0.4470 , 2.9987 , -2.5006 , 2.4652; 0.1658 , 1.1124 , -1.9942 , 1.9660];
22 J2 = [0.1296 , 1.0244 , -0.1306 , 0.1296; 0.0161 , 0.1275 , -1.0240 , 1.0161];
23 J3 = [-0.5243 , -0.2995 , 1.1106 , 0.9635; 0.9589 , 1.1041 , -0.2745 , -0.2381];
24 J4 = [-0.1825 , 0.1830 , -0.9737 , 0.3245; 0.9814 , -0.9845 , -0.2339 , 0.0779];
25 J5 = [3.8783 , 3.8804 , -3.2238 , 0.6778; 2.6792 , 2.6807 , -3.4425 , 0.7238];
26 J6 = [-0.0341 , 0.8729 , -0.3523 , 0.2531; 0.0205 , -0.5229 , -0.9546 , 0.6858];
27 J7 = [-0.2513 , 0.7491 , -1.2188 , 0.4591; 0.3232 , -0.9634 , -0.0617 , 0.0232];
28 J8 = [0.0099 , 0.6693 , -0.8921 , -0.5845; -0.0114 , -0.7717 , -0.4976 , -0.3261];
29 J9 = [-0.2280 , 0.7619 , -1.2509 , 0.3548; 0.2969 , -0.9922 , -0.0129 , 0.0037];
30 A = [J1;J2;J3;J4;J5;J6;J7;J8;J9];
31 B = pinv(A);
32 dl = B*dx;

```

Fig. 6 Command Window di Matlab che racchiude le operazioni per il calcolo della matrice A , della pseudo-inversa e dell'errore infinitesimo sulle aste ∂l .

Le diverse casistiche analizzate sul robot hanno lo scopo di verificare se, cambiando le condizioni operative, l'algoritmo risulti essere più o meno efficiente.

Dallo studio dei tre casi considerati si ha:

Tabella 14

∂l	Caso 1	Caso 2	Caso 3
∂l_1	-0.177	-0.426	-0.76
∂l_2	-0.242	-0.283	0.12
∂l_3	0.384	-0.280	-0.48
∂l_4	-0.198	-0.190	0.003
∂l_5		0.006	-0.18

I valori riportati nella tabella sono moltiplicati per 10^{-3} per quanto riguarda i primi due casi, e per 10^{-4} per il caso 3, sempre per tener conto dell'unità di misura in metri.

Anche qui è stato effettuato un confronto con i risultati ottenuti nel corso dello stesso lavoro di tirocinio e tesi già citato in precedenza.

In particolare, si ha:

Tabella 15

∂l	Caso1	Caso 2	Caso 3
∂l_1	-0.20	-0.38	0.16
∂l_2	-0.17	-0.39	-0.14
∂l_3	-0.25	0.09	0.25
∂l_4	-0.17	0.85	0.22
∂l_5	-0.22		

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

CAPITOLO 6 – DISCUSSIONE DEI RISULTATI

I risultati del capitolo precedente sono utili per stimare l'effettiva efficienza dell'algoritmo di calibrazione utilizzato. Difatti, attraverso la stima dell'errore percentuale commesso dall'algoritmo, è possibile determinare le condizioni in cui esso risulta funzionare correttamente e quelle dove risulta inadeguato. Ciò permette di valutare quanto e se sia conveniente utilizzarlo.

Indicando con ∂l_R l'errore infinitesimo ipotizzato per ogni asta, e con ∂l_A il corrispondente errore infinitesimo ricavato a seguito dell'applicazione dell'algoritmo, si calcola l'errore percentuale di quest'ultimo, per ogni asta del robot soggetta a errore in ognuno dei casi considerati:

$$p_x = \frac{|\partial l_A - \partial l_R|}{\partial l_R} \cdot 100 \quad (13)$$

Si ottiene quindi:

Tabella 16

Errore (%)	Caso 1	Caso 2	Caso 3
p ₁	188.715%	526%	252.216%
p ₂	-51.662%	194.19%	79.32%
p ₃	27.85333%	240.175%	-137.805%
p ₄	-98.49%	163.4667%	98.91%
p ₅	-	-101.236%	-40.8567%

Dai risultati ottenuti si possono trarre le seguenti conclusioni:

- CASO 1:
Dei quattro valori riportati, quelli relativi alle aste 2 e 3 sono considerevolmente inferiori al 100%; questi valori, sebbene non propriamente piccoli, permetterebbero di considerare in maniera abbastanza positiva il funzionamento dell'algoritmo. Tuttavia, nel complesso, essendo gli altri due valori molto elevati, in particolare quello relativo all'asta 1, che supera ampiamente il 100%, fornendo valori ben lontani da quelli desiderati, non è possibile considerare ben funzionante l'algoritmo. Per l'asta 5 non è stato calcolato l'errore percentuale, in quanto non vi è stato ipotizzato, inizialmente, alcun errore infinitesimo.
- CASO 2:
I cinque valori riportati sono tutti superiori al 100%. Ciò significa che in questo caso l'algoritmo non funziona bene ed è addirittura fuorviante.
- CASO 3:
In questo caso, di cinque valori calcolati, solo uno può essere considerato relativamente piccolo, mentre gli altri confermano che l'algoritmo risulta inefficace.

Come già fatto prima, i dati sono stati confrontati con quelli derivanti dal lavoro di tirocinio e tesi citato in precedenza, che sono:

Tabella 17

Errore (%)	Caso 1	Caso 2	Caso 3
p ₁	168.07%	1044.5%	84.47%
p ₂	134.26%	4021%	-95.44%
p ₃	163.25%	89.73%	75.01%
p ₄	134.02%	1595%	-104.41%
p ₅	137.33%	-	-

Tali risultati confermano che l'algoritmo, anche al variare della scala di valori presa come input, risulta essere inefficace. Ciò conferma che, nonostante i procedimenti utilizzati per lo studio condotto fossero corretti, l'algoritmo di calibrazione studiato non funziona bene.

CAPITOLO 7 – RISOLUZIONE DEI PROBLEMI

Come verificato dallo studio condotto, e dal confronto con l'altro lavoro considerato, l'algoritmo di calibrazione lineare utilizzato risulta essere inadeguato e poco efficiente. I motivi alla base di tale problematica possono essere ricercati nell'impostazione della cinematica del robot, che potrebbe essere non sufficientemente complessa da garantire un'analisi precisa e quindi una corretta calibrazione.

In particolare, ciò che potrebbe influenzare in maniera positiva l'efficacia dell'algoritmo è riscrivere le equazioni della cinematica diretta aggiungendo le variabili passive θ_2 e θ_3 , inizialmente considerate dipendenti dalle variabili attuate θ_1 e θ_4 e dalle lunghezze delle aste.

L'aggiunta di altre due variabili indipendenti (o anche una sola di esse), che si sommano alle quattro già considerate, ovvero le già citate variabili attuate e le coordinate di posizione del terminale, permettono di ottenere una matrice Jacobiana più grande; ciò significa che la matrice pseudo-inversa sarà a sua volta più grande e permetterà di ottenere una calibrazione più accurata ed efficiente.

Tuttavia, in alcuni casi, l'aggiunta delle variabili passive potrebbe non bastare per ottenere il grado di accuratezza richiesto. In questi casi, risulta necessario cambiare la tipologia di algoritmo utilizzato.

Come già illustrato nel capitolo 5, esistono diversi tipi di algoritmo, che differiscono tra loro per efficacia e modello matematico utilizzato. Se, come nel caso studiato, un algoritmo di tipo lineare non dovesse essere efficace, uno di tipo non lineare potrebbe risultare maggiormente accurato. Ciò significa che, per risolvere il problema di una calibrazione poco efficiente, si potrebbe cambiare il modello matematico considerato per il robot oggetto di studio, passando da uno di tipo lineare a uno non lineare, che permette di identificare gli errori dei parametri cinematici secondo diversi metodi, alcuni non richiedenti l'impiego della matrice Jacobiana.

CONCLUSIONI

Lo scopo di questo lavoro è stato quello di confrontare l'analisi effettuata sul robot esaminato con uno studio analogo, svolto in precedenza da un altro studente nell'ambito di un similare percorso di tirocinio curriculare, con l'obiettivo di valutare il comportamento dell'algoritmo di calibrazione lineare utilizzato al variare della scala dei valori presi come input.

L'implementazione del particolare modello matematico lineare relativo al pentalatero studiato e l'utilizzo del software Matlab hanno permesso di illustrare le numerose configurazioni che il manipolatore assume, in funzione di determinati valori delle variabili attuate, per tre distinti casi di studio, ognuno dei quali relativo ad un determinato set di errori ipotizzati sulla lunghezza delle aste; hanno, inoltre, consentito di calcolare le corrispondenti matrici Jacobiane, indispensabili per l'utilizzo dell'algoritmo.

Per il calcolo degli errori infinitesimi di posizione del terminale e degli errori percentuali commessi dall'algoritmo, entrambi tabellati, è stato fatto ricorso al programma Microsoft Excel, che ha permesso di trovare i risultati in maniera immediata, dopo aver correttamente impostato le relative funzioni.

I risultati ottenuti da ognuno dei numerosi procedimenti svolti sono stati confrontati con quelli ricavati dallo studio precedentemente citato; dalla comparazione dei valori ottenuti, è stato possibile dimostrare che la variazione della scala dei valori di input non ha influito considerevolmente sui prodotti finali delle procedure svolte.

Ne è risultata una conferma sulla poca efficienza dell'algoritmo che, in linea generale, non ha restituito valori prossimi a quelli ipotizzati in nessuno dei due studi, e si è dimostrato, in alcuni casi, addirittura fuorviante.

L'analisi condotta suggerisce, pertanto, che per ottenere una buona calibrazione del robot analizzato, sarebbe necessario aumentare la complessità del modello studiato, oppure cambiare la tipologia di algoritmo utilizzato.

BIBLIOGRAFIA

- [1] Callegari, M., *“Considerazioni sulla progettazione delle macchine a cinematica parallela”*, *Progettare* (VNU), N° 247, Luglio-Agosto 2001.
- [2] Callegari, M., Fanghella, P., Pellicano, F., *“Meccanica applicata alle macchine”*, Città Studi Edizioni, 2017.
- [3] Louis Vathan B., Hoshila Kumar and Brighton Isaac John H. *“Kinematic Analysis of Five-Bar Mechanism in Industrial Robotics”*, Proceedings of International Conference on Advances in Design and Manufacturing, Vol. 1, N° 3, 2014.
- [4] Mooring, Benjamin W., Roth, Zvi S., Driels, Morris R. *“Fundamentals of manipulator calibration”*, Calhoun, 1991.