

UNIVERSITÀ DEGLI STUDI DI BERGAMO

Scuola di Ingegneria

Dipartimento di Ingegneria Gestionale, dell'Informazione
e della Produzione

Corso di Laurea Magistrale in Ingegneria Informatica

Indagine teorica e sperimentale sul controllo centralizzato per un manipolatore a cinematica parallela

Tesi magistrale

Daniele Ravasio, mat. 1045934

Simone Cortinovis, mat.

Anno Accademico 2020-2021

Indice

1	Introduzione	5
1.1	Obiettivi e descrizione	5
1.2	Software modellazione teorica	6
1.2.1	Matlab	6
1.2.2	Adams	7
2	Cinematica Manipolatore	8
2.1	Cinematica Diretta	9
2.1.1	Posizione	9
2.1.2	Velocità	11
2.1.3	Accelerazione	12
2.2	Cinematica inversa	13
2.2.1	Posizione	13
2.2.2	Velocità	14
2.2.3	Accelerazione	15
3	Dinamica Manipolatore	16
3.1	Prerequisiti per il calcolo della dinamica	16
3.1.1	Jacobiana J_{34} e calcolo $\dot{\theta}_3, \dot{\theta}_4$	16
3.1.2	Jacobiana \dot{J}_{34} e calcolo di $\ddot{\theta}_3, \ddot{\theta}_4$	17
3.1.3	Matrici di inerzia	18
3.2	Principio dei lavori virtuali	19
3.3	Dinamica inversa	19
3.4	Dinamica diretta	20
	Modellazione su Adams	22
	Validazione e confronto	23
3.5	Punti di singolarità	24
	Primo e secondo caso	25
	Terzo caso	26
	Terzo e quarto caso	27

3.6	Manipolabilità	28
3.7	Workspace	29
4	Modellazione end-effector	32
4.1	Cinematica end-effector	33
4.2	Dinamica della vite	34
5	Tecnologie implementate	35
5.1	Simulink Real time	35
5.2	EtherCAT	35
5.2.1	Proprietà	36
5.2.2	Gestione della rete	36
5.2.3	Implementazione interfacce	37
5.3	CME2	37
5.4	EC-Engineer	39
6	Sistema reale	40
6.1	Struttura del robot	40
6.1.1	Azionamenti	41
6.1.2	Beckhoff EK1814	41
6.1.3	Configurazione della rete	42
6.2	Implementazione nel sistema reale	45
	Inizializzazione	45
	Input	46
	Stateflow	47
	Controllo vite	48
	Controllo braccia	48
	Coppie uscita	49
6.3	Stateflow	51
6.3.1	Fase di Zero	51
6.3.2	Fase di Homing	53
6.3.3	Fase di traiettoria	54

6.4	Struttura del manipolatore	55
6.4.1	Sistema vite	55
6.4.2	Sistema braccia	55
6.4.3	Gestione della traiettoria	55
6.5	Controllo	55
6.5.1	Controllo Vite	55
6.5.2	Controllo PD	55
6.5.3	Controllo in dinamica inversa	55
6.6	Traiettorie eseguite	56
7	Conclusioni	58
7.1	Confronto modellazione teorica-pratica	58
7.2	Sviluppi futuri	58

1 Introduzione

1.1 Obiettivi e descrizione

Alla base di questa tesi vi è il manipolatore PKM (*parallel kinematic manipulator*). Un manipolatore parallelo è un sistema meccanico che utilizza "catene" seriali per supportare un *end-effector*, ogni catena solitamente è corta, semplice e di conseguenza può essere rigida rispetto a movimenti non voluti rispetto ad un manipolatore seriali. La movimentazione e la flessibilità di un *joint* è vincolata dall'effetto delle altre catene, questo rende il manipolatore rigido rispetto alle sue componenti.

In particolare, nel nostro caso il manipolatore è composto da quattro link, due distali ovvero motorizzati e due prossimali, all'estremità è connessa una vite che consente traslazione e rotazione nell'asse Z, in questa configurazione il robot arriva ad avere 5 gradi di libertà.

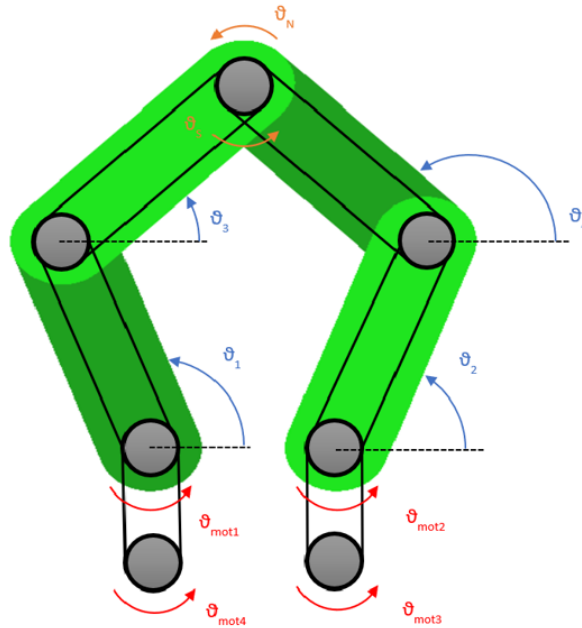


Figura 1: Robot PKM

L'obiettivo di questa tesi riguarda lo studio teorico e pratico di vari approcci di controllo, in modo da andare a trovare il migliore, in primis verrà eseguita una modellazione teorica del manipolatore, successivamente verrà svolta un'attività sperimentale che comprende l'implementazione degli schemi di controllo e la generazione

di traiettorie bidimensionali e tridimensionali,

Escludendo l'introduzione, la tesi è articolata sette capitoli: nel secondo capitolo viene presentata l'analisi cinematica del manipolatore, in particolare verranno analizzate sia cinematica diretta che inversa, nel terzo capitolo si parla di dinamica, punti di singolarità e manipolabilità del robot; il quarto capitolo mostra la modellazione dell'*end-effector*, nel nostro caso la vite; il quinto capitolo va a presentare tutte le tecnologie implementate a livello pratico, e per concludere nel sesto capitolo viene presentato il sistema reale, includendo la struttura, l'implementazione, il sistema di controllo ed i problemi riscontrati con relative soluzioni; nel settimo capitolo, infine, verranno esposte le conclusioni, grazie ad un confronto ottenuto dai risultati teorici e quelli pratici.

1.2 Software modellazione teorica

Per la modellazione teorica del manipolatore, abbiamo bisogno di strumenti software, in particolare utilizzeremo Matlab e Adams, il primo ci consentirà di realizzare un modello software del manipolatore, il secondo, sempre a partire da un modello ci servirà a validare i dati ottenuti dal primo in modo da verificare la loro correttezza.

1.2.1 Matlab

Matlab, abbreviazione di *Matrix Laboratory*, è una piattaforma di calcolo ottimizzata nella risoluzione di problemi tecnici. Matlab è un linguaggio ad alte prestazioni per la computazione tecnica, comprende: computazione, visualizzazione e programmazione in un ambiente di facile utilizzo dove i problemi e le soluzioni vengono espressi mediante una notazione matematica, gli utilizzi tipici riguardano: matematica, sviluppo di algoritmi, modellazione, simulazione, prototipazione, analisi dei dati, intelligenza artificiale, verifiche di computazione. La base di matlab è un vettore che non ha bisogno di dimensioni, in questo modo permette la risoluzione di molti problemi di computazione, specialmente quelli in formulazione vettoriale e matriciale velocemente, senza dover ricorrere all'utilizzo di linguaggi come il C. Matlab inoltre possiede delle *toolbox* ovvero moduli aggiuntivi che permettono di specializzarsi in un campo, in particolare sono insieme di funzioni MATLAB che estendono l'am-

biente, permettendogli di risolvere particolari classi di problemi, esempi di moduli sono reti neurali, processamento di segnali, sistemi di controllo.

Nel nostro caso abbiamo definito il modello teorico del robot, abbiamo poi calcolato cinematica, dinamica e punti di singolarità, nei capitoli successivi verranno mostrate le operazioni fatte.

1.2.2 Adams

Adams è un software utilizzato nel campo della dinamica *multibody*, in particolare nell'analisi dei modelli, infatti dopo che è stato progettato un modello può essere importato in adams ed è possibile fare analisi, simulazioni e validazioni, andando quindi a simulare la fisica del mondo reale. Adams è anche ottimizzato per problemi di grandi dimensioni.

Il software ha una GUI completa, infatti consente anche di disegnare direttamente il modello nello spazio tridimensionale o di importare file come STEP e IGS. I *joint* possono essere aggiunti tra due corpi per vincolare il loro movimento, inoltre al sistema possono essere passati input come velocità, forze e condizioni iniziali. Adams simula il comportamento del sistema al variare del tempo, consente anche l'animazione e la computazione di proprietà come le forze, le inerzie e le accelerazioni, è anche possibile nel sistema includere elementi complessi dinamicamente come per esempio molle, corpo flessibili, contatto tra corpi. É inoltre possibile esportare tutti i dati in formato tabellare per fare analisi successive.

Per quanto riguarda il nostro caso, abbiamo utilizzato Adams per modellare il robot, assegnargli le coppie e confrontare i valori della cinematica e dinamica con quelli ottenuti da Matlab, nei capitoli successivi verrà presentato un confronto tra questi dati.

2 Cinematica Manipolatore

L'obiettivo di questa sezione è quello di andare ad illustrare le metodologie che ci hanno consentito di ottenere sia la cinematica diretta che quella inversa, entrambe per posizione, velocità ed accelerazione.

Prima di proseguire nei paragrafi seguenti andiamo a definire una tabella con i principali parametri del robot: Tutti e quattro i link hanno lunghezza, massa, po-

Nome	Descrizione	Valore
$l[m]$	lunghezza link	0.25
$m[kg]$	massa link	2.9
$c_m[m]$	posizione centro di massa	1.25
$J_r[kg \cdot m^2]$	momento d'inerzia baricentrico	$5.22 \cdot 10^{-2}$
$d[m]$	lunghezza semitelaio	0.09

Tabella 1: Parametri manipolatore

sizione del centro di massa e momento di inerzia uguali, per questo si è deciso di rappresentare i dati una sola volta.

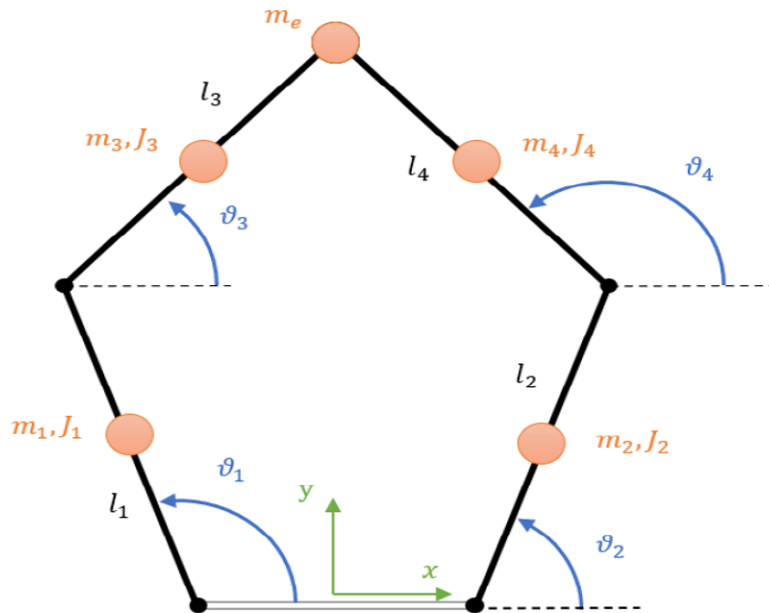


Figura 2: Rappresentazione fisica del robot

2.1 Cinematica Diretta

La cinematica diretta si occupa di trovare il legame tra i parametri interni del robot e la posa che esso assume, per posa si intende la posizione e l'orientamento. In questa sezione andremo ad analizzare la cinematica diretta di posizione, velocità ed accelerazione.

2.1.1 Posizione

Nella cinematica diretta di posizione, a partire dal robot e da θ_1 e θ_2 , riusciamo a ricavare la posizione dei link non motorizzati, i loro angoli, che sono rispettivamente θ_3 e θ_4 e la posizione $[x, y]$ dell'*end-effector*.

L'approccio utilizzato per il calcolo della cinematica diretta è stato quello delle equazioni alle circonferenze, in particolare vengono definite due equazioni

- Circonferenza centrata in E1 che passa per l'end-effector e la base del primo link
- Ciconferenza centrata in E2 che passa per l'end-effector e la base del secondo link

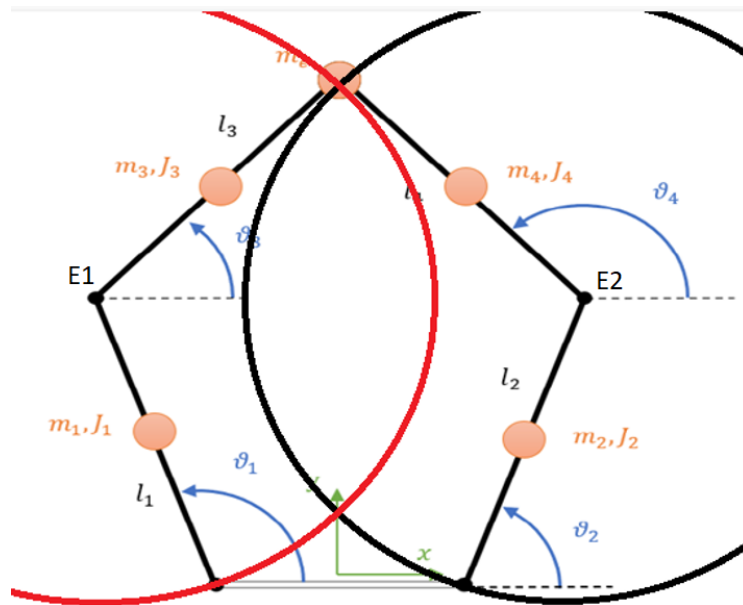


Figura 3: Equazioni alle circonferenze

Dalla combinazione di queste due equazioni otteniamo il seguente sistema:

$$\begin{cases} (x - \frac{l}{2} - l \cos \theta_1)^2 + (y - l \sin \theta_1)^2 = l^2 \\ (x + \frac{l}{2} - l \cos \theta_2)^2 + (y - l \sin \theta_2)^2 = l^2 \end{cases} \quad (1)$$

Da queste, andando a sviluppare i calcoli possiamo definire i parametri:

$$A = l^2(\sin \theta_2 - \sin \theta_1)^2 + (-2d - l(\cos \theta_2 - \cos \theta_1))^2$$

$$B = -2l^2d(\sin \theta_2 - \sin \theta_1)(\cos(\theta_2 + \theta_1) + l(\sin \theta_2 - \sin \theta_1)(-2d - l(\cos \theta_2 - \cos \theta_1)) \\ (-2d - 2l \cos \theta_2) - 2l \sin \theta_2(-2d - l(\cos \theta_2 - \cos \theta_1))^2$$

$$C = l^2d^2(\cos \theta_2 + \cos \theta_1)^2 - ld(\cos \theta_2 + \cos \theta_1)(-2d - l(\cos \theta_2 - \cos \theta_1)) \\ (-2d - 2l \cos \theta_2) + (d^2 + 2dl \cos \theta_2)(-2dl(\cos \theta_2 - \cos \theta_1))^2$$

Il passo successivo è quello di ricavare la posizione $P = [x, y]$ dell'end-effector; si può procedere grazie alla formula risolutiva delle equazioni di secondo grado

$$s_{1,2}^1 = \frac{-b \pm \Delta}{2a}$$

Si può notare facilmente che la formula fornisce due risultati diversi, in una caso quando si utilizza $+\Delta$ e l'altro quando si usa $-\Delta$, è normale perché secondo il teorema fondamentale dell'algebra possono esserci fino a due soluzioni, in questo caso una coincidente con la posizione dell'end-effector e l'altra proiettata sulla base del semi-telaio.

$$x = \frac{y \cdot l(\sin \theta_2 - \sin \theta_1) - l \cdot d(\cos \theta_2 + \cos \theta_1)}{-2d - l(\cos \theta_2 - \cos \theta_1)}, y = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \quad (2)$$

Possiamo poi andare a trovare le posizioni dei link distali mediante relazioni geometriche nel seguente modo:

$$E_{1X} = -d + l \cdot \cos \theta_1 \quad E_{1Y} = l \cdot \sin \theta_1$$

¹ $\Delta = \sqrt{b^2 - 4ac}$

e

$$E_{2X} = d + l \cdot \cos \theta_2 \quad E_{2Y} = l \cdot \sin \theta_2$$

Adesso che abbiamo E_1 ed E_2 possiamo andare a trovare gli angoli θ_3, θ_4 in funzione di θ_1 e θ_2

$$\theta_3 =$$

$$2 \cdot tg^{-1} \frac{\sqrt{(\sin \theta_2 - \sin \theta_1) + \frac{1}{2}((\cos \theta_2 - \cos \theta_1 + \frac{18}{25})^2 + (\frac{(\sin \theta_1 - \sin \theta_2)^2}{2})^2 + (\sin \theta_1 - \sin \theta_2)^2)}}{(\cos \theta_2 - \cos \theta_1) + \frac{(\cos \theta_2 - \cos \theta_1 + \frac{18}{25})^2}{2} + \frac{(\sin \theta_1 - \sin \theta_2)^2}{2} + \frac{18}{25}}$$

$$\theta_4 =$$

$$-2 \cdot tg^{-1} \frac{\sqrt{(\sin \theta_2 - \sin \theta_1) + (\frac{1}{2}(\cos \theta_2 - \cos \theta_1 + \frac{18}{25})^2 + (\frac{(\sin \theta_1 - \sin \theta_2)^2}{2})^2 + (\sin \theta_1 - \sin \theta_2)^2)}}{(\cos \theta_1 - \cos \theta_2) + \frac{(\cos \theta_2 - \cos \theta_1 + \frac{18}{25})^2}{2} + \frac{(\sin \theta_1 - \sin \theta_2)^2}{2} + \frac{18}{25}}$$

Di conseguenza, alla fine della cinematica diretta siamo riusciti ad ottenere i parametri:

$$[x, y, E1, E2, \theta_3, \theta_4]$$

tutti espressi in funzione di θ_1 e θ_2 .

2.1.2 Velocità

Una volta ottenute le posizioni possiamo passare alle velocità, mediante la cinematica diretta di velocità possiamo ricavare le velocità sulle coordinate x e y dell'*end-effector*. Possiamo infine definire una jacobiana che ci permette di trovare il rapporto appena espresso.

Per semplicità di calcoli, andiamo a definire:

$$N_{21} = -l \sin \theta_1 (x + d - l \cdot \cos \theta_1 + l \cdot \cos \theta_1 (y - l \sin \theta_1))$$

$$N_{22} = -l \cos \theta_2 \cdot \frac{y - l \sin \theta_2 (x + d - l \cos \theta_1)}{x - d - l \cos \theta_2} + l \sin \theta_2 (x + d - l \cos \theta_1)$$

$$D_2 = \frac{y - l \sin \theta_2 (x + d - l \cos \theta_1)}{x - d - l \cos \theta_2}$$

Una volta definiti e calcolati questi valori possiamo andare a costruire i termini della jacobiana:

$$\begin{aligned}
J_{11} &= -\frac{y - l \sin \theta_2}{x - d - l \cos \theta_2} \cdot J_{21} \\
J_{12} &= -\frac{y - l \sin \theta_2}{x - d - l \cos \theta_2} \cdot J_{22} - l \sin \theta_2 + \frac{y - l \sin \theta_2}{x - d - l \cos \theta_2} \cdot l \cos \theta_2 \\
J_{21} &= \frac{N_{21}}{D_2} \\
J_{22} &= \frac{N_{22}}{D_2}
\end{aligned}$$

Posizionando i termini della matrice possiamo quindi definire J come:

$$J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \quad (3)$$

Avendo quindi definito la jacobiana possiamo ricavare la velocità dell'*end-effector* $\dot{P} = [\dot{x}, \dot{y}]$, nel seguente modo:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = J \cdot \dot{\Theta} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} J_{11}\dot{\theta}_1 + J_{12}\dot{\theta}_2 \\ J_{21}\dot{\theta}_1 + J_{22}\dot{\theta}_2 \end{bmatrix} \quad (4)$$

2.1.3 Accelerazione

Anche per quanto riguarda l'accelerazione il processo simile a quanto visto nei paragrafi precedenti, in questo caso a partire da tutti i parametri precedentemente ricavati e da $\ddot{\Theta}$ composto da $\ddot{\theta}_1$ e $\ddot{\theta}_2$ ricaviamo le accelerazioni all'*end-effector*. L'idea base è quella di risolvere la seguente equazione:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \dot{J}\dot{\Theta} + J\ddot{\Theta} = \dot{J} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + J \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} \quad (5)$$

Andando a svolgere le derivate ed i calcoli prima di poter trovare le accelerazioni definiamo:

$$\begin{aligned}
A_{acc} = & \dot{x}^2 + 2l \sin \theta_2 \dot{x} \dot{\theta}_2 + (l \sin \theta_2 \dot{\theta}_2)^2 + (x - d - l \cos \theta_2)(l \cos \theta_2 \dot{\theta}_2 + l \sin \theta_2 \ddot{\theta}_2) + \\
& \dot{y}^2 - 2l \cos \theta_2 \dot{y} \dot{\theta}_2 + (l \cos \theta_2 \dot{\theta}_2)^2 + (y - l \sin \theta_2)(l \sin \theta_2 \dot{\theta}_2^2 - l \cos \theta_2 \ddot{\theta}_2)
\end{aligned}$$

$$B_{acc} = \dot{x}^2 + 2l \sin \theta_1 \dot{x} \dot{\theta}_1 + (l \sin \theta_1 \dot{\theta}_1)^2 + (x + dl \cos \theta_1)(l \cos \theta_1 \dot{\theta}_1^2 + l \sin \theta_1 \ddot{\theta}_1) + \\ \dot{y}^2 - 2l \cos \theta_1 \dot{y} \dot{\theta}_1 + (l \cos \theta_1 \dot{\theta}_1)^2 + (y - l \sin \theta_1)(l \sin \theta_1 \dot{\theta}_1^2 - l \cos \theta_1 \ddot{\theta}_1)$$

Infine, andiamo a trovare $\ddot{P} = [\ddot{x}, \ddot{y}]$, nel seguente modo:

$$\ddot{x} = -\frac{\ddot{y}(y - l \sin \theta_1)}{x + d - l \cos \theta_1} \quad (6)$$

$$\ddot{y} = \frac{\frac{B_{acc} \cdot (x - d - l \cos \theta_2)}{x + dl \cos \theta_1 - A_{acc}}}{y - l \sin \theta_2 - \frac{x - d - l \cos \theta_2}{(x + d - l \cos \theta_1) \cdot (y - l \sin \theta_1)}} \quad (7)$$

Alla fine della cinematica diretta, a partire dai vettori delle posizioni, velocità ed accelerazioni $(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2, \ddot{\theta}_1, \ddot{\theta}_2)$ siamo riusciti ad ottenere le posizioni, velocità ed accelerazioni riferite all'*end-effector*.

2.2 Cinematica inversa

Il problema della cinematica inversa consiste nel ricavare i valori degli angoli da assegnare ai parametri del robot per riuscire a seguire una determinata legge di moto o traiettoria a partire dalla posizione alle estremità, in questo caso l'*end-effector*. Anche l'analisi della cinematica inversa è stata fatta per posizione, velocità ed accelerazione.

2.2.1 Posizione

Nella cinematica inversa di posizione, a partire dalla posizione dell'*end-effector* $P = [x, y]$ andiamo a ricavare θ_1 e θ_2 . Definiamo i seguenti parametri:

$$p = 2dl + 2xl$$

$$e = 2yl$$

$$f = x^2 + d^2 + y^2 + 2px$$

che serviranno per il calcolo di θ_1 , e:

$$a = -2dl + 2xl$$

$$b = 2yl$$

$$c = x^2 + d^2 + y^2 - 2xd$$

che serviranno per il calcolo di θ_2 .

Procediamo quindi con i calcoli, andando a trovare le soluzioni:

$$\theta_1 = 2 \arctan \frac{e + \sqrt{p^2 + e^2 - f^2}}{p + f} \quad (8)$$

e

$$\theta_2 = 2 \arctan \frac{b - \sqrt{a^2 + b^2 - c^2}}{a + c} \quad (9)$$

Si può notare che sia per θ_1 che per θ_2 la somma di termini sotto la radice quadrata può dare un risultato reale o un risultato complesso, in caso che esca un risultato reale non c'è alcun problema, ma nel caso in cui $p^2 + e^2 - f^2 \leq 0$ oppure $a^2 + b^2 - c^2 \leq 0$ potrebbe verificarsi un caso di singolarità².

2.2.2 Velocità

Per quanto riguarda il calcolo della cinematica inversa in velocità abbiamo bisogno delle velocità $\dot{P} = [\dot{x}, \dot{y}]^T$ e della Jacobiana che lega θ_1 e θ_2 . Prendiamo l'equazione 3 andiamo poi ad invertirla e ricaviamo le velocità $\dot{\theta}_1$ e $\dot{\theta}_2$ nel seguente modo:

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = J^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \quad (10)$$

Con

$$J^{-1} = \frac{1}{J_{11}J_{22} - J_{12}J_{21}} \begin{bmatrix} J_{22} & -J_{12} \\ J_{21} & J_{11} \end{bmatrix} \quad (11)$$

²I punti di singolarità sono punti nei quali il manipolatore non si comporta in modo standard, potrebbero causarsi anche rotture, verranno descritti in modo approfondito nel capitolo riguardante la dinamica

Siamo quindi riusciti ad ottenere le velocità degli angoli a partire dalle velocità all'end-effector.

2.2.3 Accelerazione

Anche per le accelerazioni la logica di funzionamento è la medesima, volendo trovare $\ddot{\theta}_1$ e $\ddot{\theta}_2$ a partire da $\ddot{P} = [\ddot{x}, \ddot{y}]^T$ implica che dobbiamo trovare la soluzione a:

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = J^{-1} \dot{P} + J^{-1} \ddot{P} = J^{-1}(\ddot{P} - \dot{J}\dot{\Theta}) \quad (12)$$

3 Dinamica Manipolatore

Il modello dinamico del manipolatore ci fornisce una descrizione matematica della relazione che è instaurata tra le forze agenti sul robot (generalizzate) ed il movimento prodotto dalla sua struttura, cioè le configurazioni che assume nel tempo. Inizialmente, nel calcolo della dinamica sono stati usati tre metodi diversi, il metodo delle azioni vincolari, il metodo di Lagrange e quello dei lavori virtuali. Si è poi deciso di proseguire esclusivamente con il PLV. L'obiettivo di questa sezione è quello di mostrare l'approccio e i risultati ottenuti per il calcolo della dinamica diretta ed inversa.

3.1 Prerequisiti per il calcolo della dinamica

Prima di andare ad analizzare i metodi utilizzati, è importante andare a ricavare tutte le matrici delle quali avremo bisogno, in particolare è necessario andare a definire delle matrici che ci permettano di ottenere θ_3 e θ_4 in funzione di θ_1 e θ_2 .

3.1.1 Jacobiana J_{34} e calcolo $\dot{\theta}_3, \dot{\theta}_4$

A partire da θ_1, θ_2 possiamo, mediante la cinematica diretta ottenere θ_3 e θ_4 , unendo queste quattro componenti e considerando anche le velocità $\dot{\theta}_1$ e $\dot{\theta}_2$ possiamo andare a ricavare la matrice J_{34} che esprime θ_3 e θ_4 in funzione di θ_1, θ_2 e possiamo ricavare anche $\dot{\theta}_3, \dot{\theta}_4$, per far questo andiamo a definire le seguenti quantità:

$$N_{13} = \frac{\cos \theta_4}{\sin \theta_4} \sin \theta_1 - \cos \theta_1$$

$$N_{23} = \cos \theta_2 - \frac{\cos \theta_4}{\sin \theta_4} \sin \theta_2$$

$$D_{13} = \cos \theta_3 - \frac{\cos \theta_4}{\sin \theta_4} \sin \theta_3$$

Da queste tre formule possiamo andare a ricavare $\dot{\theta}_3$ nel seguente modo:

$$\dot{\theta}_3 = \frac{\dot{\theta}_1 N_{13}}{D_{13}} + \frac{\dot{\theta}_2 N_{23}}{D_{13}} \quad (13)$$

Proseguiamo ora definendo le equazioni che ci serviranno per calcolare $\dot{\theta}_4$:

$$\begin{aligned} N_{14} &= \frac{\sin \theta_1 \cos \theta_3}{\sin \theta_3} - \frac{\cos \theta_4}{\sin \theta_4} + \frac{\cos \theta_4}{\sin \theta_4} \sin \theta_1 - \cos \theta_1 \\ N_{24} &= -\frac{\sin \theta_2 \cos \theta_3}{\sin \theta_3} - \frac{\cos \theta_4}{\sin \theta_4} - \frac{\cos \theta_4}{\sin \theta_4} \sin \theta_2 + \cos \theta_2 \\ D_{14} &= \frac{\sin \theta_4 \cos \theta_3}{\sin \theta_3} - \frac{\cos \theta_4}{\sin \theta_4} \end{aligned}$$

Otteniamo quindi:

$$\dot{\theta}_4 = \dot{\theta}_1 \frac{N_{14}}{D_{14}} + \dot{\theta}_2 \frac{N_{24}}{D_{14}} \quad (14)$$

Il passo finale è quello di andare a rappresentare la matrice jacobiana che lega le velocità $\dot{\theta}_3$ e $\dot{\theta}_4$ con le velocità in ingresso al manipolatore:

$$J_{34} = \begin{bmatrix} \frac{N_{13}}{D_{13}} & \frac{N_{23}}{D_{13}} \\ \frac{N_{14}}{D_{14}} & \frac{N_{24}}{D_{14}} \end{bmatrix} \quad (15)$$

3.1.2 Jacobiana J_{34} e calcolo di $\ddot{\theta}_3, \ddot{\theta}_4$

Per andar ad ottenere la jacobiana finale, ed i valori delle accelerazioni sui link distali occorre derivare tutti gli elementi visti in precedenza, in particolare:

$$\begin{aligned} \dot{N}_{13} &= \frac{-1}{\sin^2 \theta_4 \cdot \dot{\theta}_4 \sin \theta_1} + \frac{\cos \theta_4}{\sin \theta_4} \cos \theta_1 \dot{\theta}_1 + \sin \theta_1 \dot{\theta}_1 \\ \dot{N}_{23} &= \frac{1}{\sin^2 \theta_4 \cdot \dot{\theta}_4 \sin \theta_2} - \frac{\cos \theta_4}{\sin \theta_4} \cos \theta_2 \dot{\theta}_2 - \sin \theta_2 \dot{\theta}_2 \\ \dot{D}_{13} &= -\sin \theta_3 \dot{\theta}_3 + \frac{1}{\sin^2 \theta_4} \dot{\theta}_4 \sin \theta_3 - \frac{\cos \theta_4}{\sin \theta_4} \cos \theta_3 \dot{\theta}_3 \\ \dot{N}_{14} &= \cos \theta_1 \dot{\theta}_1 \left(\frac{\cos \theta_3}{\sin \theta_3} - \frac{\cos \theta_4}{\sin \theta_4} \right) + \sin \theta_1 \left(\frac{1}{\sin^2 \theta_3} \dot{\theta}_3 + \frac{1}{\sin^2 \theta_4} \dot{\theta}_4 \right) + \\ &\quad + \frac{-1}{\sin^2 \theta_4} \dot{\theta}_4 \sin \theta_1 + \cot \theta_4 \cos \theta_1 \dot{\theta}_1 + \sin \theta_1 \dot{\theta}_1 \\ \dot{N}_{24} &= -\cos \theta_2 \dot{\theta}_2 \left(\frac{\cos \theta_3}{\sin \theta_3} - \frac{\cos \theta_4}{\sin \theta_4} \right) - \sin \theta_2 \left(\frac{-1}{\sin^2 \theta_3} \dot{\theta}_3 + \frac{1}{\sin^2 \theta_4} \dot{\theta}_4 \right) - \\ &\quad - \frac{-1}{\sin^2 \theta_4} \dot{\theta}_4 \sin \theta_2 - \cot \theta_4 \cos \theta_2 \dot{\theta}_2 - \sin \theta_2 \dot{\theta}_2 \end{aligned}$$

$$\dot{D}_{14} = \cos \theta_4 \dot{\theta}_4 (\cot \theta_3 - \cot \theta_4) + \sin \theta_4 \left(\frac{-1}{\sin^2 \theta_3} \dot{\theta}_3 + \frac{1}{\sin^2 \theta_4} \dot{\theta}_4 \right)$$

Esprimendo la matrice jacobiana \dot{J}_{34} in funzione dei parametri appena trovati scriviamo:

$$\dot{J}_{34} = \begin{bmatrix} \frac{\dot{N}_{13}D_{13} - N_{13}\dot{D}_{13}}{D_{13}^2} & \frac{\dot{N}_{23}D_{13} - N_{23}\dot{D}_{13}}{D_{13}^2} \\ \frac{\dot{N}_{14}D_{14} - N_{14}\dot{D}_{14}}{D_{14}^2} & \frac{\dot{N}_{24}D_{14} - N_{24}\dot{D}_{14}}{D_{14}^2} \end{bmatrix} \quad (16)$$

Per concludere andiamo a trovare:

$$\begin{bmatrix} \ddot{\theta}_3 \\ \ddot{\theta}_4 \end{bmatrix} = \dot{J}_{34} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + J_{34} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} \quad (17)$$

3.1.3 Matrici di inerzia

Per trovare la soluzione all'equazione del PLV introduciamo le matrici che hanno avuto un ruolo fondamentale nel calcolo:

$$\begin{aligned} J_1 &= \begin{bmatrix} -0.5l \sin \theta_1 & 0 \\ 0.5l \cos \theta_1 & 0 \end{bmatrix} \Rightarrow \dot{J}_1 = \begin{bmatrix} -0.5l \cos \theta_1 \dot{\theta}_1 & 0 \\ -0.5l \sin \theta_1 \dot{\theta}_1 & 0 \end{bmatrix} \\ J_2 &= \begin{bmatrix} 0 & -0.5 * l * \sin \theta_2 \\ 0 & 0.5 * l * \cos \theta_2 \end{bmatrix} \Rightarrow \dot{J}_2 = \begin{bmatrix} 0 & -0.5l \cos \theta_2 \dot{\theta}_2 \\ 0 & -0.5l \sin \theta_2 \dot{\theta}_2 \end{bmatrix} \\ J_3 &= \begin{bmatrix} -l \sin \theta_1 + 0.5 \sin \theta_3 \cdot J_{34}(1, 1) & -0.5l \sin \theta_3 \cdot J_{34}(1, 2) \\ l \cos \theta_1 + 0.5 \cos \theta_3 \cdot J_{34}(1, 1) & 0.5l \cos \theta_3 \cdot J_{34}(1, 2) \end{bmatrix} \\ J_4 &= \begin{bmatrix} -0.5l \sin \theta_4 \cdot J_{34}(2, 1) & -l \sin \theta_2 + 0.5 \sin \theta_4 \cdot J_{34}(2, 2) \\ 0.5l \cos \theta_4 \cdot J_{34}(2, 1) & l \cos \theta_2 + 0.5 \cos \theta_4 \cdot J_{34}(2, 2) \end{bmatrix} \\ J_E &= \begin{bmatrix} -l(\sin \theta_1 + \sin \theta_3 \cdot J_{34}(1, 1)) & -l \sin \theta_3 \cdot J_{34}(1, 2) \\ l(\cos \theta_1 + \cos \theta_3 \cdot J_{34}(1, 1)) & l \cos \theta_3 \cdot J_{34}(1, 2) \end{bmatrix} \end{aligned}$$

Importanti nel calcolo della dinamica saranno anche le derivate delle matrici che abbiamo appena visto, ovvero $\dot{J}_3, \dot{J}_4, \dot{J}_E$.

3.2 Principio dei lavori virtuali

Il lavoro virtuale è il lavoro svolto da una forza reale che agisce attraverso uno spostamento virtuale o da una forza virtuale che agisce attraverso uno spostamento reale. Uno spostamento virtuale è uno spostamento coerente con i vincoli della struttura, cioè che soddisfano le condizioni al contorno in corrispondenza degli appoggi. Una forza virtuale è un qualsiasi sistema di forze in equilibrio.

Per problemi nei quali i corpo sono composti da membri interconnessi che si possono muovere relativamente gli uni rispetto agli altri, originando diverse configurazioni di equilibrio un buon metodo di analisi è quello del "principio dei lavori virtuali" conosciuto anche come PLV ci permette di ottenere una relazione relativamente semplice, è basato sul concetto di Lavoro sviluppato da una forza, ed inoltre ci consente di analizzare la stabilità di un sistema in equilibrio.

$$\sum_{i=1}^m F_j \delta q_j \quad (18)$$

3.3 Dinamica inversa

Il problema della dinamica inversa consiste nel determinare le coppie ai giunti necessarie per generare il movimento a partire da posizione, velocità ed accelerazione. Andando a sviluppare l'equazione dei principi virtuali troviamo le coppie dei link motorizzati nel seguente modo:

$$\begin{aligned} \delta \theta^T C = & \delta \theta^T I_2 \ddot{\theta} + \delta \theta^T J_{34}^T I_2 (J_{34} \ddot{\theta} + \dot{J}_{34} \dot{\theta}) + \delta \theta^T \frac{25}{4} m l^2 \\ & \left(\begin{bmatrix} -\cos \theta_1 & -\sin \theta_1 \\ -\cos \theta_2 & -\sin \theta_2 \end{bmatrix} \dot{\theta}^2 + \begin{bmatrix} -\sin \theta_1 & \cos \theta_1 \\ -\sin \theta_2 & \cos \theta_2 \end{bmatrix} \ddot{\theta} \right) + \delta \theta^T J_{34}^T \frac{9}{4} l^2 (m + m_v) \\ & \left(\begin{bmatrix} -\cos \theta_3 & -\sin \theta_3 \\ -\cos \theta_4 & -\sin \theta_4 \end{bmatrix} J_{34} J_{34}^T \dot{\theta}^2 + \begin{bmatrix} -\sin \theta_3 & \cos \theta_3 \\ -\sin \theta_4 & \cos \theta_4 \end{bmatrix} (\dot{J}_{34} \dot{\theta} + J_{34} \ddot{\theta}) \right) \end{aligned}$$

Semplificando e raccogliendo possiamo esprimere l'equazione come:

$$\tau = M \ddot{\theta} + K \dot{\theta} \quad (19)$$

Dove:

$$M = J_r I_2 + m(J_1^T J_1 + J_2^T J_2 + J_3^T J_3 + J_4^T J_4) + J_r J_{34}^T J_{34} + m_v J_E^T J_E \quad (20)$$

$$K = m(J_1^T \dot{J}_1 + J_2^T \dot{J}_2 + J_3^T \dot{J}_3 + J_4^T \dot{J}_4) + J_r J_{34}^T \dot{J}_{34} + m_v J_E^T \dot{J}_E \quad (21)$$

Sostituendo, possiamo andare ad esprimere l'equazione 19 nel seguente modo:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = M \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + K \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \quad (22)$$

Per tutti i calcoli svolti fino ad adesso le coppie devono considerarsi prese all'end-effector

3.4 Dinamica diretta

Il problema della dinamica diretta invece consiste nel determinare le accelerazioni ai giunti a partire dalle coppie, dalla posizione e velocità iniziali di entrambi i link. Identifichiamo quindi Θ come vettore delle condizioni iniziali, in particolare possiamo definirlo come segue:

$$\Theta = \begin{bmatrix} \theta_1(t_0) \\ \theta_2(t_0) \\ \dot{\theta}_1(t_0) \\ \dot{\theta}_2(t_0) \end{bmatrix}$$

Possiamo andare a calcolare θ_3 e θ_4 come abbiamo visto in precedenza nella sezione 2.1.1, e di conseguenza anche tutte le matrici viste nella sezione 3.1. Con tutti questi dati possiamo andare a ricalcolare le equazioni 20 e 21.

Andiamo ora a definire l'equazione della dinamica diretta andando ad invertire l'equazione 19 in questo modo:

$$\ddot{\theta} = M^{-1}(-K\dot{\theta} + \tau) \quad (23)$$

Infine, da questa possiamo andare anche a calcolare velocità e posizioni integrando l'equazione.

Avendo sia il grafico della legge di moto iniziale relativa a posizione, velocità ed accelerazione assegnata agli angoli θ_1 e θ_2 , che le coppie, possiamo andare a calcolare la dinamica con le coppie e le condizioni iniziali e *plottare* il confronto tra queste due curve. In particolare andiamo a vedere il confronto su accelerazioni, velocità e posizioni sia per θ_1 che per θ_2 : Una volta calcolati tutti i parametri di nostro

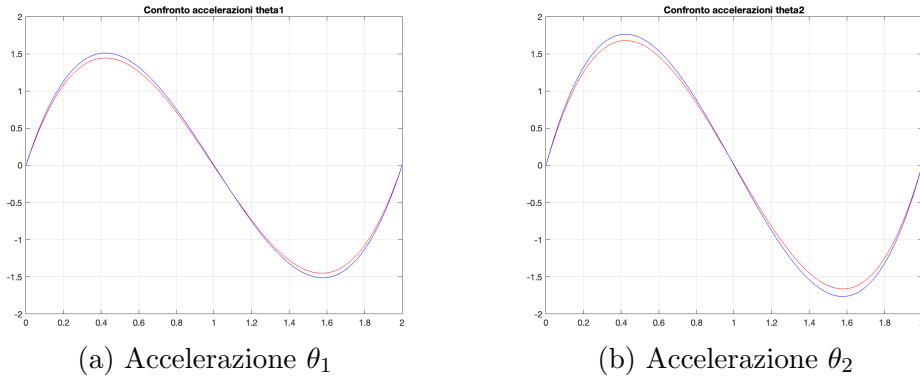


Figura 4: Confronto dinamica leggi di moto su accelerazioni

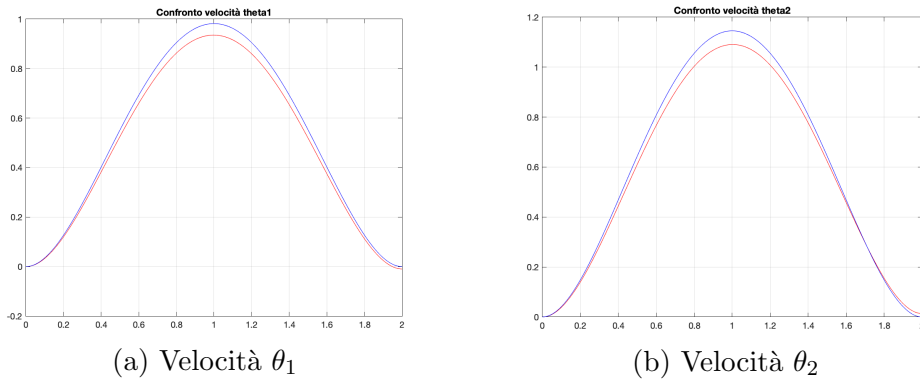


Figura 5: Confronto dinamica leggi di moto su velocità

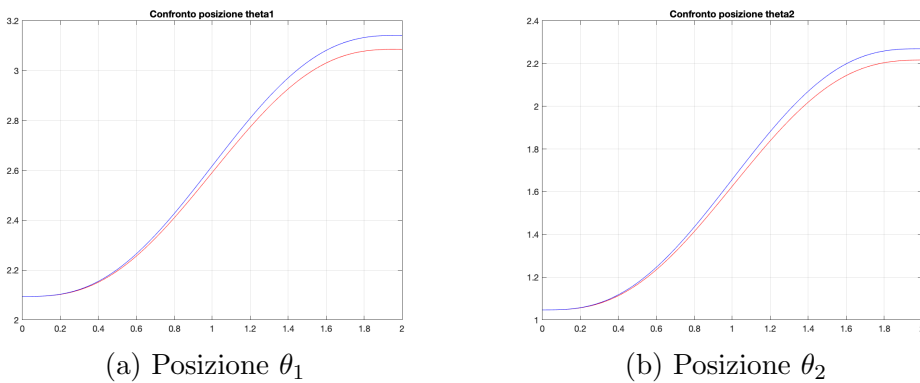


Figura 6: Confronto dinamica leggi di moto su posizioni

interesse andiamo a costruire il modello simulink che sarà utile per una visione d'insieme:

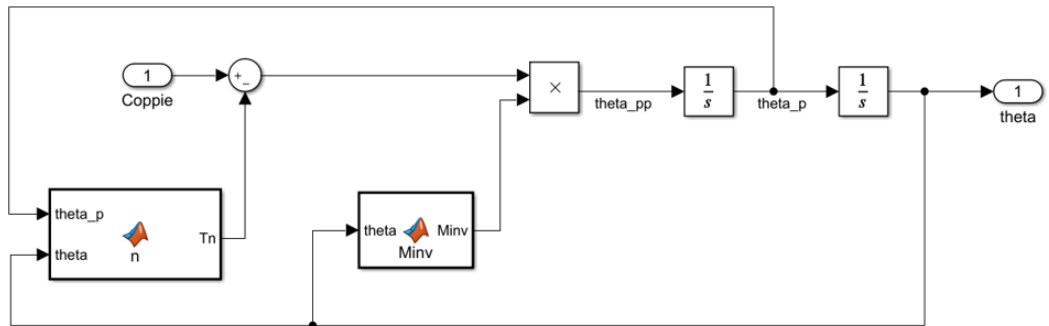
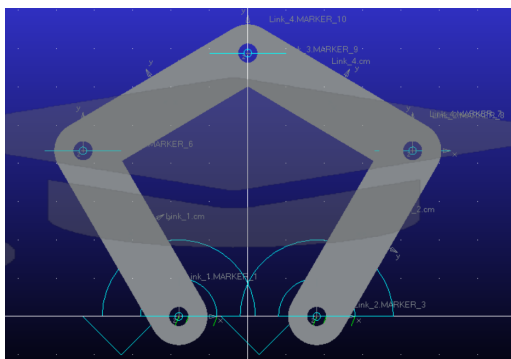


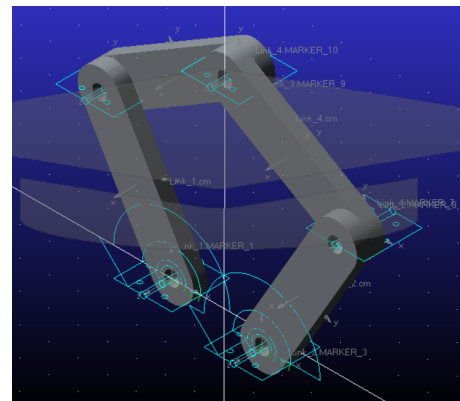
Figura 7: Modello simulink manipolatore

Modellazione su Adams

Per quanto riguarda la modellazione su adams, è stato realizzato un prototipo del manipolatore composto da aste rigide, i due link motorizzati sono fissati mediante delle cerniere ed abbiamo anche la presenza dell'end-effector. Una volta definiti i



(a) Vista dall'alto



(b) Vista in diagonale

Figura 8: Modello Adams manipolatore 5R

vincoli e le modalità di movimento del manipolatore si è passati alla fase successiva ovvero quella dell'analisi del modello e della simulazione, andando ad assegnare leggi di moto al modello adams e verificare il suo comportamento rispetto al modello creato su simulink.

Validazione e confronto

Le leggi di moto assegnate al modello adams sono state anche assegnate nella stessa maniera al modello simulink, in particolare sono state utilizzate: Una volta asse-

Motore	Legge di moto
Motore 1 (θ_1)	Sinusoidale <i>sin</i>
Motore 2 (θ_2)	Sinusoidale 2 sin

Tabella 2: Leggi di moto validazione

gnata la legge ed eseguita la simulazione si è fatto un confronto: Andando poi a

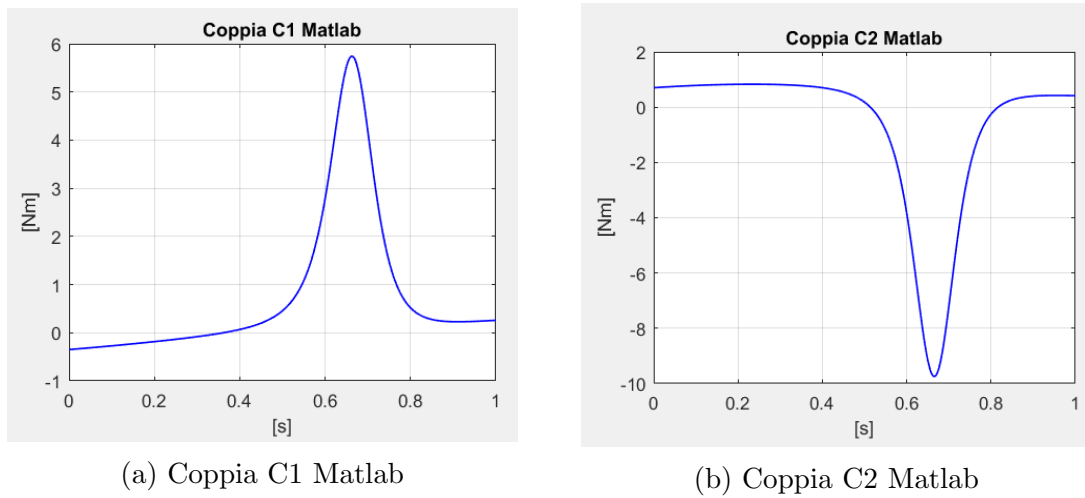


Figura 9: Coppie in uscita Simulink

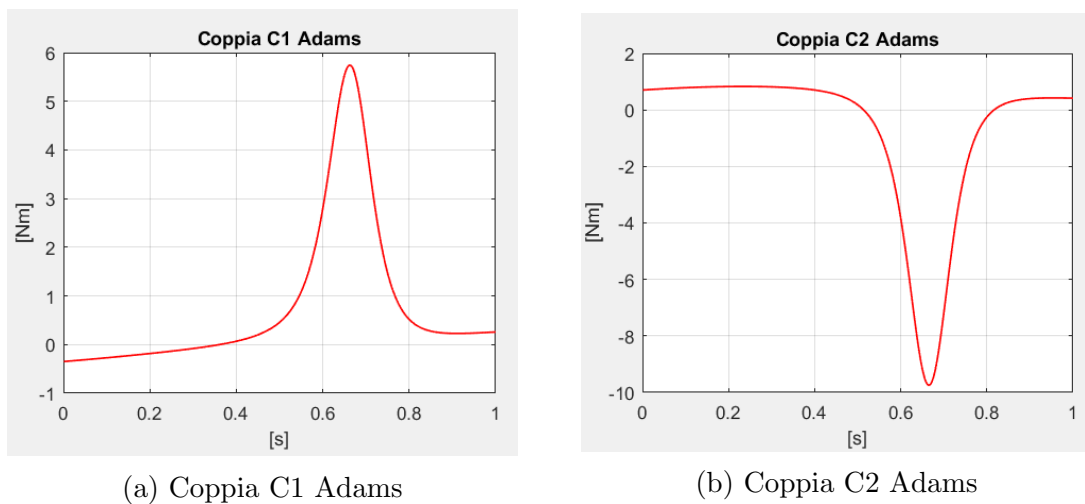


Figura 10: Coppie in uscita Adams

fare una differenza tra questi due grafici riusciamo a trovare l'andamento dell'er-

rore sulle coppie: Analizzando il grafico vediamo che l'errore è nell'ordine di 10^{-7}

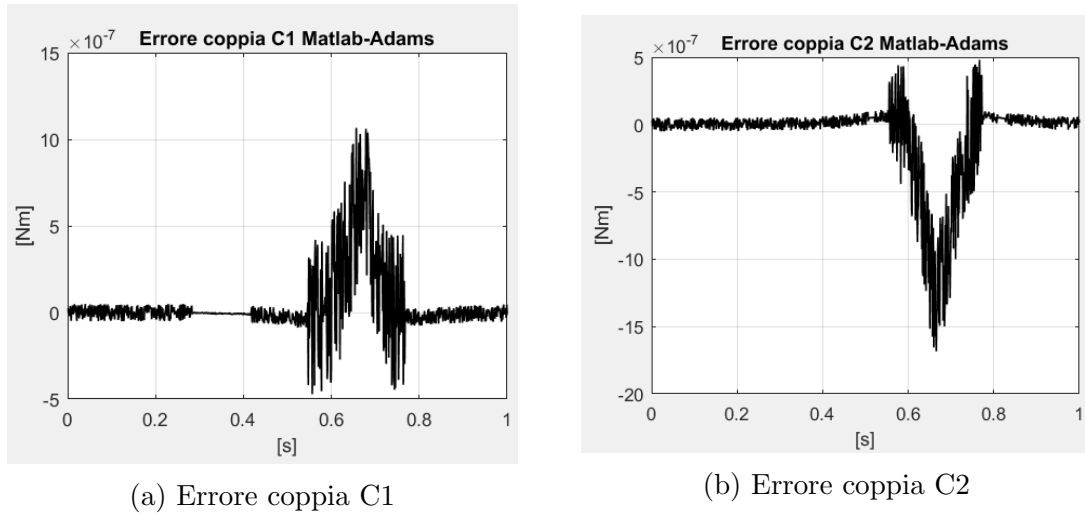


Figura 11: Errore Simulink-Adams

di conseguenza è possibile vedere che la validazione ha portato un risultato positivo in quanto le coppie del modello Simulink e le coppie del modello Adams sono praticamente identiche a parte un fattore d'errore dato dalle diverse modalità di calcolo/integrazione.

3.5 Punti di singolarità

Nell'ambito matematico, una singolarità è un punto nel quale un oggetto non è definito, oppure un punto nel quale l'oggetto non ha un comportamento normale, nel nostro caso i punti di singolarità saranno punti che andranno a delimitare lo spazio di lavoro del robot. Definiamo spazio di lavoro del robot tutto un insieme di punti nei quali il robot ha un funzionamento normale e non presenta problematiche.³ Andando a considerare la foto vista nell'introduzione, possiamo trovare sei casi di punti di singolarità, in particolare però non sono punti ma sono traiettorie. Di conseguenza il robot avrà come spazio di lavoro, tutto lo spazio che è interno (delimitato) da queste traiettorie.

³Passando per un punto di singolarità il robot potrebbe aver problemi che potrebbero causare anche la rottura di parti meccaniche

Primo e secondo caso

In questo primo caso abbiamo \overrightarrow{CD} che è parallelo a \overrightarrow{DE} , schematicamente possiamo andarlo a rappresentare nel seguente modo

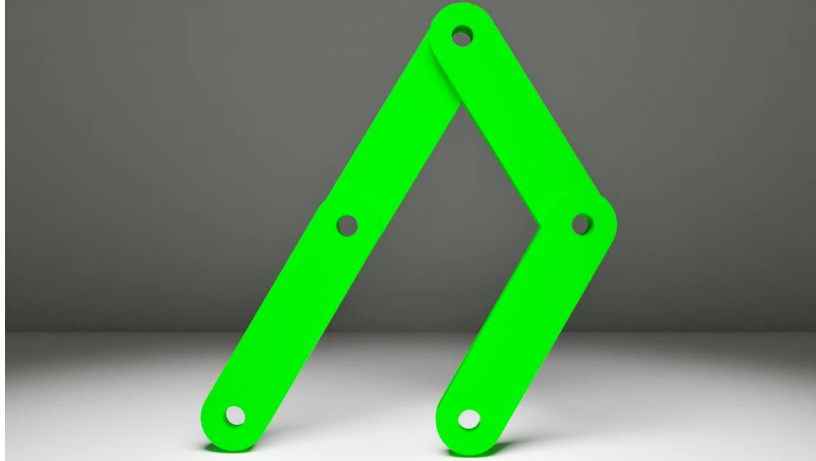


Figura 12: Caso 1 singolarità

Lasciando la x libera possiamo ricavare la y come:

$$y_1 = \sqrt{4l^2 - (x - d)^2} \quad (24)$$

Per quanto riguarda il secondo caso è molto simile al primo, la differenza sta nel fatto che abbiamo la catena \overrightarrow{AB} parallela a \overrightarrow{BC} . Il procedimento è simile a prima,

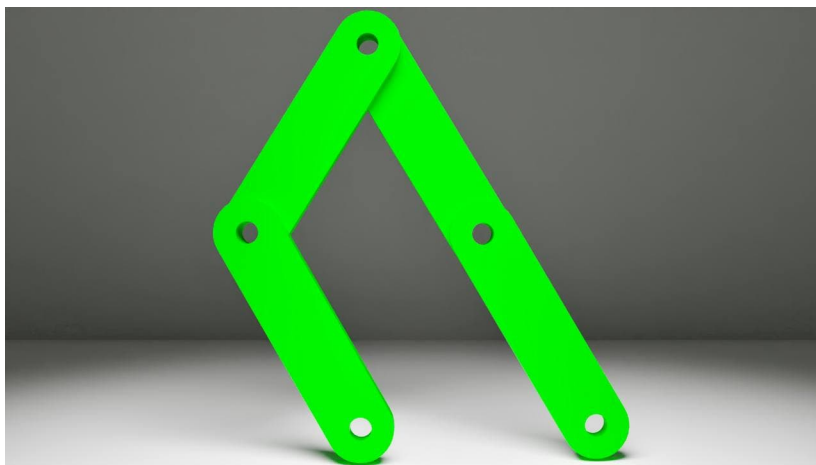


Figura 13: Caso 2 singolarità

lasciando sempre la x libera possiamo trovare la y come:

$$y_2 = \sqrt{4l^2 - (x + d)^2} \quad (25)$$

Entrambi i casi producono come risultato una circonferenza.

Terzo caso

Il terzo caso di singolarità avviene quando i due link non motorizzati sono paralleli, questa configurazione aggiunge un grado di libertà al sistema, in quanto l'end-effector per muoversi ha necessità di una maggior coppia per riuscire a superare la situazione di stallo

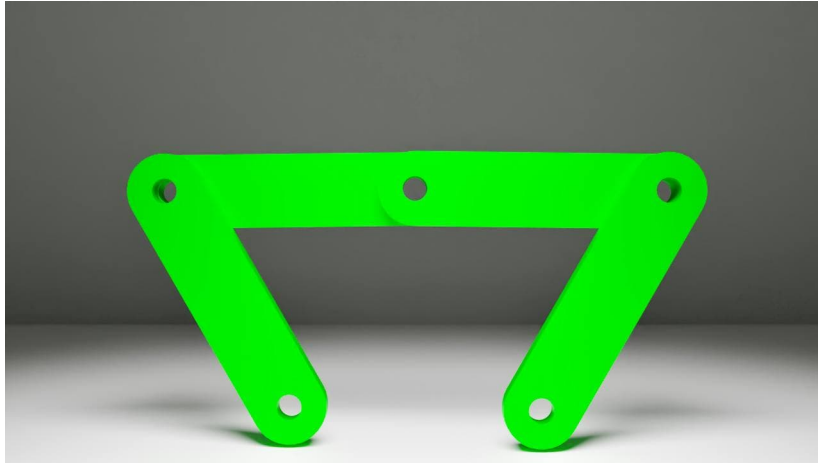


Figura 14: Caso 3 singolarità

Per quanto riguarda la soluzione, andiamo a considerare θ_1 che varia da 0 a 360° e andiamo a cercare le coppie di valori $[x, y]$ relative alla singolarità. Uscirà un'equazione di secondo grado, con i seguenti coefficienti:

$$a = l^2 \sin^2 \theta_1 + 4d^2 - 4dl \cos \theta_1 + l^2 \cos^2 \theta_1$$

$$b = 2l^3 \sin \theta_1 + 2dl^2 \sin \theta_1 \cos \theta_1 - 2dl \sin \theta_1 (2d - 2l \cos \theta_1)$$

$$c = l^2(l^2 + d^2 \cos^2 \theta_1 + 2ld \cos \theta_1) - 2dl(l + d \cos \theta_1)(2d - l \cos \theta_1) + d^2 - l^2(2d - l \cos \theta_1)^2$$

Risolviemo l'equazione di secondo grado:

$$\Delta = b^2 - 4ac$$

Andiamo a trovare le radici y di quest'equazione, definendo poi: $sx = -b + l \cos \theta_1$ ed $sy = l \sin \theta_1$ possiamo andare a trovare le soluzioni dell'equazione come:

$$x_3 = \left| \frac{x_{3p} + sx}{2} \right|, y_3 = \left| \frac{y_{3p} + sy}{2} \right| \quad (26)$$

Con

$$x_{3p} = \frac{l^2 + y_{3p}l \sin \theta_1 + ld \cos \theta_1}{2d - l \cos \theta_1}$$

Quarto e quinto caso

Il quarto e quinto caso sono casi di singolarità non realizzabili nella pratica, ma sono di interesse teorico. Il primo caso prevede che la posizione dell'end-effector coincida con la posizione del primo giunto motorizzato e nell'altro caso coinciderà con il secondo giunto motorizzato. Come soluzioni avremo semplicemente due punti e possiamo andare a calcolarli nei seguenti modi: Le soluzioni le possiamo trovare

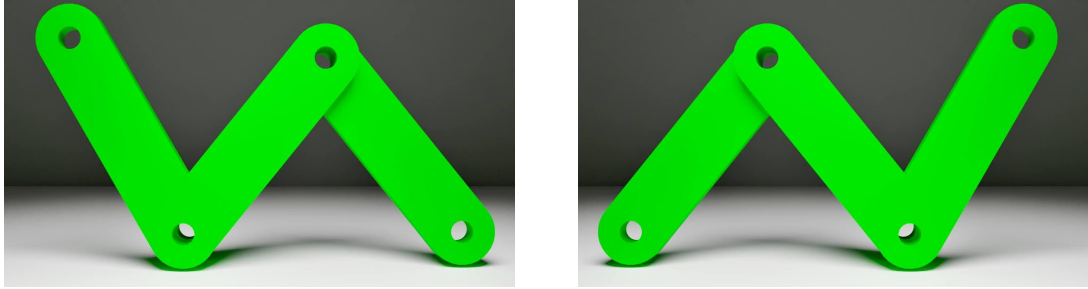


Figura 15: Caso 4 e 5 singolarità

impostando che la lunghezza della x vale nel quarto caso d e nel quinto $-d$, andando quindi a trovare le soluzioni come:

$$y_4 = \sqrt{-(x - d)^2} \quad (27)$$

e

$$y_5 = \sqrt{-(x + d)^2} \quad (28)$$

Andando ad unire tutti casi visti fino ad ora possiamo vederli visualmente nell'asse $[x, y]$

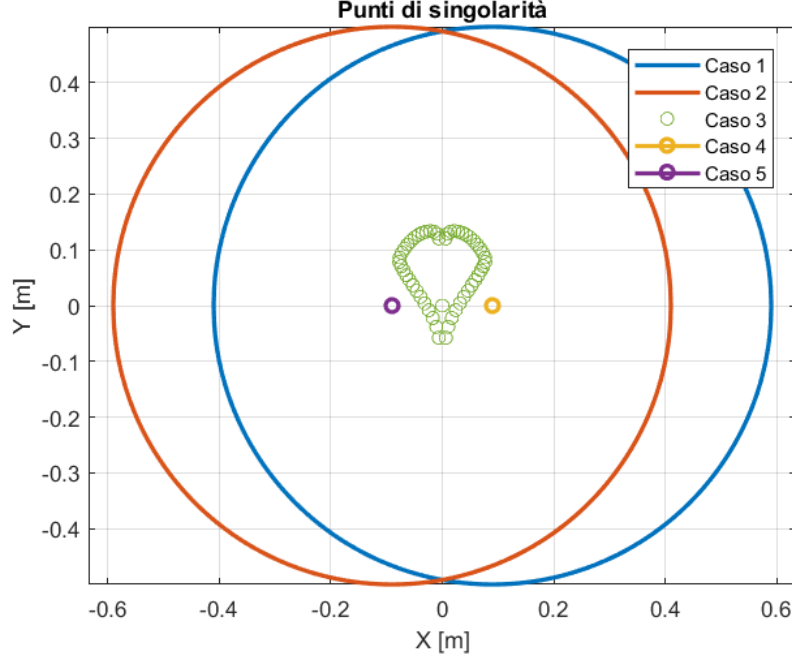


Figura 16: Punti di singolarità

3.6 Manipolabilità

La manipolabilità ci permette di avere una rappresentazione geometrica delle capacità che ha un punto del nostro sistema. Per andare a calcolarla abbiamo bisogno dell'equazione 3, vista nella sezione 2.1.2.

Andiamo a definire la matrice

$$J_{man} = JJ^T \quad (29)$$

Da questa possiamo ricavare gli autovalori Λ . Definiamo poi l'indice di manipolabilità \mathbf{r} come:

$$r = \frac{\max \lambda}{\min \lambda} \quad (30)$$

Questo numero può variare tra 1 e $+\infty$, più è piccolo e meno si rischia di andare in singolarità. Possiamo andare a rappresentare il numero di condizionamento tramite i grafici seguenti:

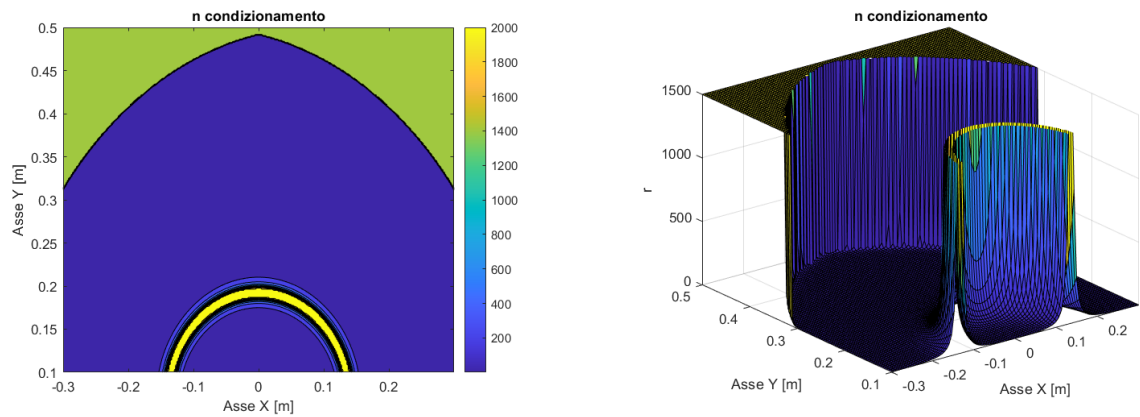


Figura 17: Numero di condizionamento

Nel primo grafico viene mostrato il piano x,y ed il numero di condizionamento è definito come una colormap, i punti di color blu sono quelli con un r piccolo ed in questi non siamo in condizioni di singolarità, invece quelli tendenti al verde/giallo sono i casi di singolarità che abbiamo visto prima. Nella seconda figura abbiamo la stessa rappresentazione però in tre dimensioni, utilizziamo l'asse z per rappresentare il numero di condizionamento; anche qua le zone di singolarità sono chiaramente visibili. In questo secondo grafico invece andiamo a concentrarci sulla zona di movimentazione ideale del manipolatore, più ci si avvicina allo zero, più il valore di r aumenta, questo per il fatto che stiamo raggiungendo una zona di singolarità.

3.7 Workspace

Dalle analisi appena effettuate sui punti di singolarità e sul numero di condizionamento è stato possibile descrivere lo spazio di lavoro del manipolatore. Unendo poi

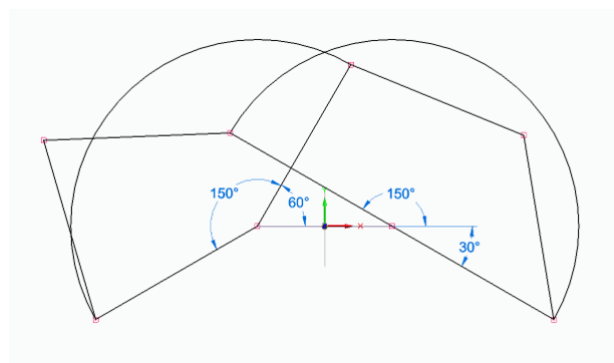


Figura 18: Angoli di movimentazione dei giunti motorizzati

gli angoli di movimentazione del manipolatore, è stato possibile descrivere lo spazio di lavoro mediante un rettangolo inscritto che non viola i punti di singolarità



Figura 19: *Workspace* 5R

In particolare è stata anche fatta un'analisi sul numero di condizionamento del *workspace*: Su tutto il piano x,y si nota come il numero di condizionamento assume valori

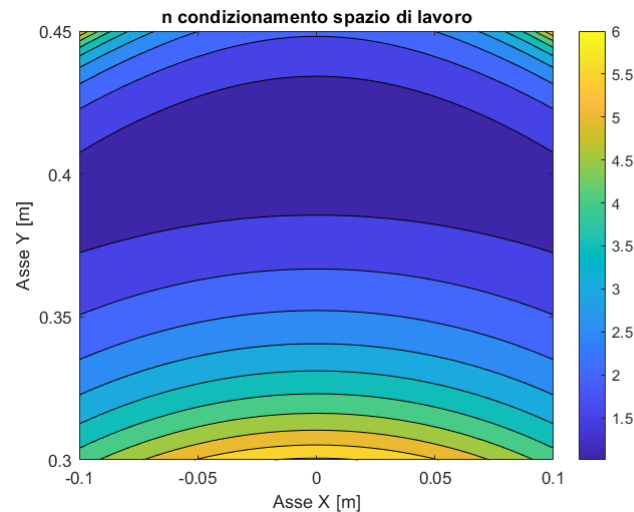


Figura 20: Numero di condizionamento workspace

bassi, questo implica che non vi è singolarità, gli unici punti critici sono quelli esterni, nei quali il valore è vicino a 6, (comunque minore di ∞), questi punti vanno a

rappresentare casi nei quali il manipolatore avrà più difficoltà a muoversi, ma non sono veri e propri casi di singolarità .

4 Modellazione end-effector

L'obiettivo di questo capitolo consiste nel presentare la modellazione della cinematica e dinamica dell'end-effector.

L'end-effector, identificato anche come utensile, è composto da due parti: La prima

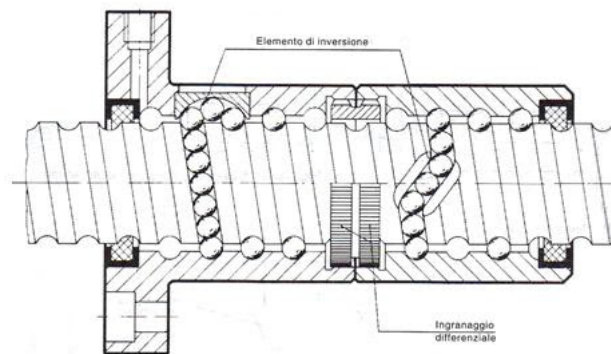


Figura 21: Struttura vite

è una vite a ricircolo di sfere che permette un movimento rotativo, mentre la seconda parte è una guida lineare che permette una traslazione positiva o negativa nell'asse z. Entrambi i componenti sono collegati a motori che permettono la movimentazione interfacciandosi con degli azionamenti. All'estremità della vite è possibile collegare un utensile, come è possibile vedere dalla foto è stato collegato un utensile utilizzato per disegnare le traiettorie su di un foglio.



Figura 22: Utensile da disegno

Per proseguire con la trattazione della parte cinematica e dinamica andiamo ad introdurre i parametri della vite:

Nome	Descrizione	Valore
$m_v[kg]$	massa vite	0.36
$p_v[m]$	passo vite	0.02
$I_v[kg \cdot m^2]$	momento inerzia vite	$6.40 \cdot 10^{-6}$

Tabella 3: Parametri end-effector

A livello teorico si è partito definendo una legge di modo per entrambi i componenti della vite, si è selezionata una legge polinomiale, per la guida è stata fatta una legge sulla posizione, mentre per la vite a ricircolo di sfere sull'orientamento.

4.1 Cinematica end-effector

Come anticipato, abbiamo quindi la presenza di due leggi di moto che identifichiamo con z_{ee} e φ_v . Il risultato della cinematica di posizione è il parametro V , per velocità ed accelerazione saranno le sue derivate, \dot{V} e \ddot{V} definite come:

$$V = \begin{bmatrix} Z \\ \theta_Z \end{bmatrix}, \dot{V} = \begin{bmatrix} \dot{Z} \\ \dot{\theta}_Z \end{bmatrix}, \ddot{V} = \begin{bmatrix} \ddot{Z} \\ \ddot{\theta}_Z \end{bmatrix}$$

V è composto da una parte di traslazione (Z) e da una parte di rotazione θ_Z ; Per andar a ricavare la cinematica abbiamo quindi bisogno della legge di moto e della jacobiana della vite, che possiamo definire come:

$$J_e = \begin{bmatrix} \frac{p_v}{2\pi} & \frac{p_v}{2\pi} \\ 0 & 1 \end{bmatrix} \quad (31)$$

Andando ora a combinare i parametri e sostituendoli nell'equazione precedente otteniamo:

$$V = J_e \begin{bmatrix} z_{ee} \\ \varphi_v \end{bmatrix}, \dot{V} = J_e \begin{bmatrix} \dot{z}_{ee} \\ \dot{\varphi}_v \end{bmatrix}, \ddot{V} = J_e \begin{bmatrix} \ddot{z}_{ee} \\ \ddot{\varphi}_v \end{bmatrix} \quad (32)$$

4.2 Dinamica della vite

Per la definizione della dinamica della vite è stata usata, come per la dinamica dei link la tecnica del PLV. Per questa soluzione avremo bisogno delle accelerazioni dei due elementi quindi \ddot{Z}_{ee} e $\ddot{\varphi}_v$. Andiamo poi ad introdurre la matrice di massa della vite, composta dalla massa e dall'inerzia e definita come:

$$M_v \begin{bmatrix} m_e & 0 \\ 0 & I_v \end{bmatrix} \quad (33)$$

La soluzione della dinamica la troviamo quindi come:

$$C_{ee} = M_v J_e \begin{bmatrix} \ddot{z}_{ee} \\ \ddot{\varphi}_v \end{bmatrix} \quad (34)$$

Sono poi stati eseguiti vari test per la movimentazione della vite a livello teorico, nella seguente immagine è possibile vedere un risultato di questi test, in particolare nella figura sono mostrate le coppie fornite ai motori della vite e la posizione iniziale e finale della vite, tutto questo in funzione del tempo.

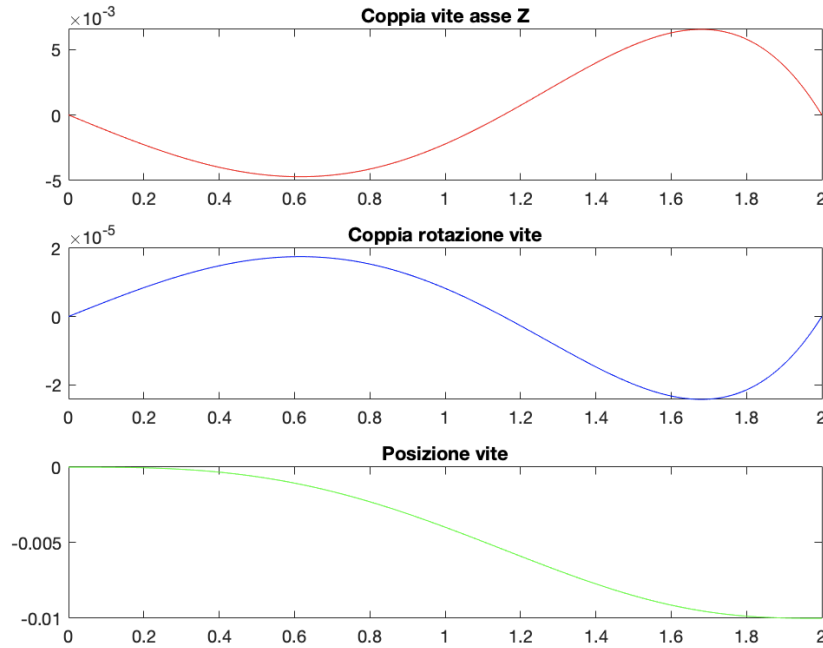


Figura 23: Coppie e posizione end-effector

5 Tecnologie implementate

Conclusa la spiegazione della modellazione di bracci e vite è il momento di iniziare l'attività sperimentale. Lo scopo di questa sezione è l'introduzione alle tecnologie software utilizzate per la modellazione fisica del manipolatore.

5.1 Simulink Real time

Simulink real-time è un plugin di matlab, consente di creare applicazioni real-time ed eseguirle su un hardware target, come per esempio un computer. Simulink-real time gira su un computer di "sviluppo" che è il PC utente, ed è diverso dal sistema che permette il movimento in real-time e l'esecuzione del modello reale ovvero il dispositivo target. Di conseguenza simulink real-time può raggiungere tempi di campionamento molto più veloci per uno stesso modello rispetto ad kernel real-time che condivide le risorse hardware con il computer, vi è la possibilità di raggiungere anche i 20 kHz. Un'altra caratteristica fondamentale è il fatto che su SLRT è possibile far funzionare il dispositivo target in modalità *stand-alone* cioè può riuscire ad avviarsi senza il bisogno di una procedura di installazione.

5.2 EtherCAT

L'obiettivo di questa sottosezione è quello di presentare il protocollo di comunicazione utilizzato per interfacciarsi col robot PKM, evidenziandone le sue caratteristiche principali.

EtherCAT è una tecnologia ethernet sviluppata in origine da Beckhoff automation, il protocollo è stato pubblicato nello standard IEC61158⁴, soddisfa requisiti *hard* e *soft* real time in particolare nell'ambito dell'automazione. Una caratteristica particolare sono i tempi di ciclo che sono molto veloci infatti una durata media di un tempo di ciclo è inferiore a $100\mu s$. Il principio base di funzionamento si basa sul concetto di *Master/Slave*.

Venne introdotto nell'aprile 2003, e nei mesi successivi è nata una società chia-

⁴<https://webstore.iec.ch/publication/59890>

mata EtherCAT Technology Group (ETG) che è diventata una delle più grandi organizzazioni ethernet al mondo.

5.2.1 Proprietà

Il master EtherCAT invia un pacchetto, chiamato telegramma che va ad attraversare tutti i nodi, ogni singolo slave collegato legge i dati che riguardano lui e scrive i dati prodotti intanto che il telegramma si propaga sulla rete verso i nodi successivi. Non appena il pacchetto arriva all'ultimo nodo, è quest'ultimo che si occupa di reinviarlo al master grazie alla comunicazione full-duplex presa da Ethernet, facendo questo, il flusso di dati teorico riesce a superare i 100Mbit/s . Il fatto che il master sia l'unico nodo che può inviare frame in maniera attiva garantisce prestazioni deterministiche. Il master utilizza un Media Access Controller (MAC) standard, senza alcun processore dedicato alla comunicazione. Questo consente di implementare un dispositivo master su qualunque piattaforma hardware dotata di una porta di rete, indipendentemente dal Sistema Operativo o software applicativo utilizzato. I dispositivi EtherCAT slave integrano un cosiddetto EtherCAT Slave Controller (ESC) in grado di processare i frame on-the-fly e in modo puramente hardware, il che rende le prestazioni della rete predicibili e indipendenti dalla particolare implementazione dei dispositivi slave.

5.2.2 Gestione della rete

La rete EtherCAT è sicura, viene implementata una tecnologia denominata *safety over EtherCAT* che consente la realizzazione di architetture di sicurezza più semplici e flessibili di una logica standard a relè. Vi è la possibilità di trovare questa tecnologia standardizzata nella specifica IEC 61784-3, il sistema di comunicazione è parte del *black channel*, ovvero una parte considerata non rilevante ai fini della sicurezza; questo fa uso di un solo canale per trasferire sia i dati standard che quelli di sicurezza, i frame di sicurezza vengono identificati come *safety container*, e contengono i dati critici del processo e l'informazione necessaria per garantirne l'integrità. Un *safety container* viene mappato dentro i dati di processo ciclici di comunicazione, possono viaggiare tramite cavi di rame, fibre ottiche e connessioni *wireless*, questo

introduce flessibilità, e rende più semplice e sicuro connettere parti della macchina anche lontane fra di loro. In una macchina completamente connessa, andare ad implementare una funzione d'arresto di emergenza totale risulterà semplice anche se altre parti della macchina sono connesse con tecnologie diverse.

5.2.3 Implementazione interfacce

Come abbiamo accennato precedentemente, il principio di funzionamento di *EtherCAT* è il concetto di *master/slave*. Per come si è evoluta questa tecnologia, e considerando che l'interfaccia non richiede una CPU ad elevate prestazioni, è possibile andare ad aggiungere un dispositivo di I/O ad un controllore senza andare ad aumentare significativamente i costi complessivi. Per un cliente, è importante l'interoperabilità tra i dispositivi di più fornitori, per questo prima di poter introdurre un dispositivo sul mercato vengono fatti tutti i test, che verificano che l'implementazione rispetti la specifica EtherCAT.

L'interfaccia master ha l'unico requisito di avere una porta ethernet, per l'implementazione viene utilizzato o l'ethernet controller integrato oppure una scheda di rete base; nella maggior parte dei casi l'ethernet controller viene integrato mediante un DMA (Direct Memory Access), in questo modo per l'invio dei dati tra il master e la rete non vengono utilizzate le risorse della CPU. Gli slave scrivono i dati prodotti e leggono quelli a loro indirizzati mentre il telegramma li attraversa, facendo così al master arrivare l'immagine già ordinata correttamente, la CPU quindi non è più responsabile dell'ordinamento.

I dispositivi *slave* invece utilizzano ESC (*Ethercat slave controller*), solitamente di costo contenuto oppure integrato in un microcontrollore standard, esistono slave semplici che non richiedono nemmeno la presenza di un microcontrollore per il fatto che gli ingressi e le uscite digitali possono essere direttamente collegati all'ESC, mentre per quelli più complessi viene usato un controllore a 8-bit.

5.3 CME2

CME2 è un software prodotto da Copley control, e serve per la configurazione degli azionamenti. Le funzioni principali riguardano *auto-phasing* e *auto-tuning*, che van-

no a semplificare la realizzazione del sistema. Oltre a queste due funzioni principali abbiamo anche le tabelle Cam che forniscono un buon approccio per la produzione di movimenti sincronizzati e ripetitivi ad un dispositivo esterno, e la possibilità di definire sequenze fino a 32 indici o sequenze indicizzate. Sono anche predisposte funzioni per l'analisi degli strumenti, la configurazione dei motori dei filtri e dei guadagni, è possibile anche andare a cambiare la modalità operativa, scegliendo quindi se lavorare con un anello in posizione, corrente o coppia. Il collegamento con gli azionamenti è avvenuto tramite cavo ethernet, come si può notare dalle foto gli

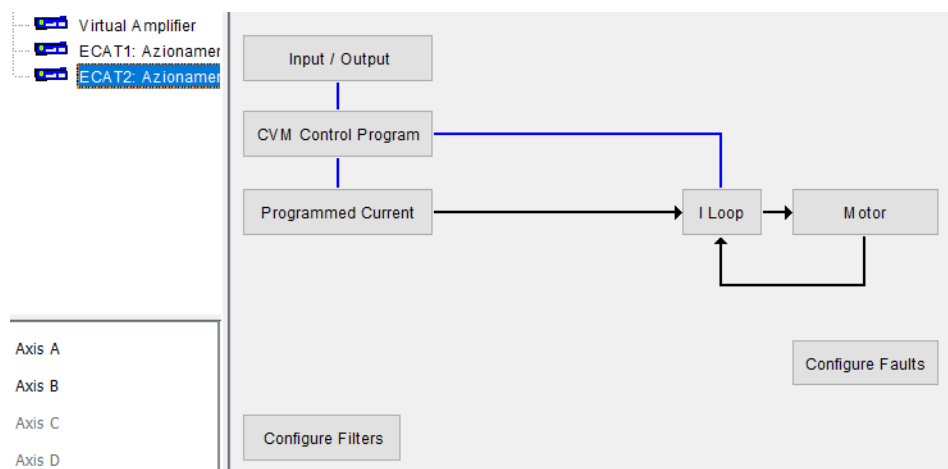


Figura 24: Schema azionamento

azionamenti sono controllati in corrente. Un'operazione fondamentale è stata l'analisi dei registri nella quale si è visto a cosa corrispondeva ogni registro. In particolare i registri osservati sono stati quelli dei finecorsa. Per le braccia i registri da guardare sono stati l'ottavo e il quindicesimo, in particolare per vincoli di progetto i finecorsa osservati sono stati quelli al lato destro, per la vite invece il registro osservato è stato il quindicesimo, ed era riferito alla movimentazione superiore della vite.

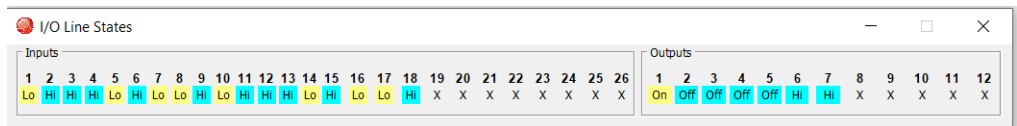


Figura 25: Registri azionamento

5.4 EC-Engineer

EC-Engineer è uno strumento software utilizzato per la configurazione, diagnostica e monitoraggio delle reti EtherCAT, è nato con lo scopo di aver tutto il necessario su un solo ambiente. Usando questo programma è possibile generare due tipologie di configurazioni EtherCAT, online oppure offline. Il primo tipo di configurazione viene fatto quando siamo direttamente connessi sulla macchina alla rete EtherCAT quindi è un'operazione fatta in real-time, la seconda tipologia invece può essere fatta in laboratorio o ufficio e non richiede la connessione alla rete in quel preciso momento. Per andare a fare una connessione online non è necessario che gli slave siano direttamente connessi al PC locale, questo grazie alla funzione *bus scan* che permette di andare a determinare la topologia della rete facilmente. Nel nostro caso abbiamo scelto una configurazione *offline*, come metodo di comunicazione si è scelto di usare le PDO⁵. Per concludere la configurazione viene esportato un file ENI, ovvero un file XML che descrive la topologia della rete, il comando di inizializzazione per ogni dispositivo ed i comandi che devono essere inviati ciclicamente, il file ENI viene fornito al master che invia i comandi in base a questo file. Una volta creato è stato inserito su simulink nell'*EtherCAT init*.

⁵*Process Data Object*, ovvero dati trasmetti dal/al *MotionController*, in tempo reale ad ogni nodo ad ogni tempo di campionamento.

6 Sistema reale

Avendo introdotto anche le tecnologie implementate, andiamo ora a discutere del sistema reale, in particolare concentrandoci sulla sua struttura, compresa la configurazione, le operazioni che hanno consentito la movimentazione e le tecniche di controllo

6.1 Struttura del robot

Il manipolatore PKM è un manipolatore a cinematica parallela, composto da due braccia ed un end-effector. Alle braccia sono collegati due motori, uno per il link motorizzato sinistro e l'altro per il link motorizzato destro, i link distali si muovono in conseguenza al movimento di quelli motorizzati. Anche l'end-effector è composto da due motori, il primo motore permette di far salire/scendere la vite, il secondo invece genera un moto elicoidale che permette la rotazione della vite con conseguente salita/discesa. Per quanto riguarda la parte elettronica abbiamo la presenza di due azionamenti che sono collegati uno ai motori delle braccia e l'altro ai motori della vite ed un modulo beckhoff che si occupa della gestione degli input digitali. Per

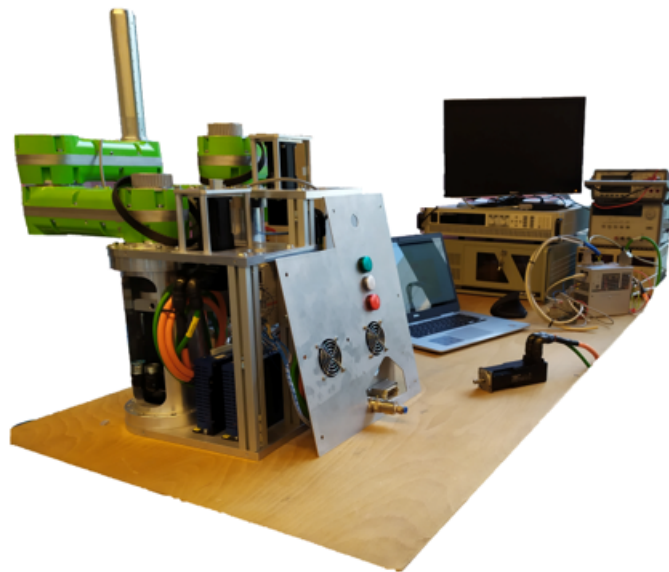


Figura 26: Banco di test

funzionare il sistema ha bisogno di due alimentatori, uno che serve ad alimentare la logica, alimentato a 24 Volt e un altro che serve ad alimentare i quattro motori, 80 Volt.

6.1.1 Azionamenti

Gli azionamenti utilizzati sono gli accelnet plus a 2 assi BE2, sono progettati appositamente per EtherCAT, operano con tensioni da 14 a 90 volt, riescono a fornire in uscita fino a 30A.

Sono predisposti per controllo in posizione, velocità e coppia di motori brushless, per la configurazione utilizzano il software CME 2 e la comunicazione avviene mediante l'interfaccia seriale RS-232. Il BE2 opera come ethercat slave, utilizzando il layer applicativo CAN su ethercat CoE. Inoltre, viene fornito un input AuxHV che permette in casi critici di tener vivo l'azionamento anche quando non c'è alimentazione senza perdere le informazioni sulla posizione o le comunicazioni con il sistema di controllo. Per la comunicazione con ethercat invece sono predisposti due cavi RJ-45, la porta d'ingresso IN permette la connessione ad un master o alla porta d'uscita OUT di un dispositivo che nella gerarchia è interposto tra il master e l'azionamento. Inoltre, se l'accelnet è l'ultimo nodo della rete non vi è bisogno di un terminatore sulla porta d'uscita.

6.1.2 Beckhoff EK1814

Il beckhoff EK1814 è un accoppiatore EtherCAT che fa da *link* tra il protocollo EtherCAT a livello di bus di campo e il terminali EtherCAT. Inoltre, su questo modello sono anche integrati quattro input digitali e quattro output digitali. La sua struttura lo rende ideale per applicazioni con pochi input/output. L'accoppiatore converte i telegrammi che passano da Ethernet *100BASE-TX* a rappresentazioni di segnali *E-bus*. Una stazione EtherCAT è formata da un accoppiatore e da un numero N di terminali che vengono identificati automaticamente.

Inoltre, l'EK1814 ha due connessioni RJ45, l'interfaccia Ethernet superiore è utilizzata per collegare l'accoppiatore alla rete, mentre quella posteriore serve per il collegamento di altri dispositivi EtherCAT nello stesso commento. Nel nostro pro-

getto è stato usato come master, a questo sono stati connessi gli slave (ovvero gli azionamenti), inoltre gli input e output digitali sono stati usati per controllare la pressione del fungo di emergenza e le luci di segnalazione delle fasi del manipolatore.

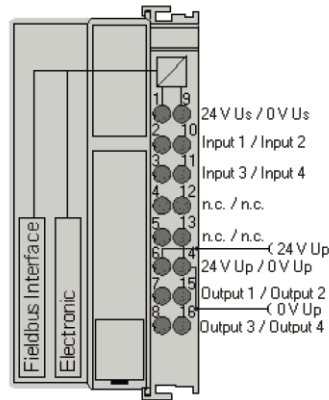


Figura 27: Schema modulo bechhoff

6.1.3 Configurazione della rete

La configurazione della rete prevede alla base il PC Target, in questo vi è una chiavetta USB che fa *runnare* sul pc un sistema operativo simulink real time. Il target è il master della rete, ha due uscite ethernet, la prima è collegata direttamente al modulo bechhoff, il quale prende l'identità di primo slave, e come abbiamo visto precedentemente, al bechhoff sono attaccati e i due azionamenti che si comportano come slave aggiuntivi slave. Invece, alla seconda porta ethernet, vi è collegato il

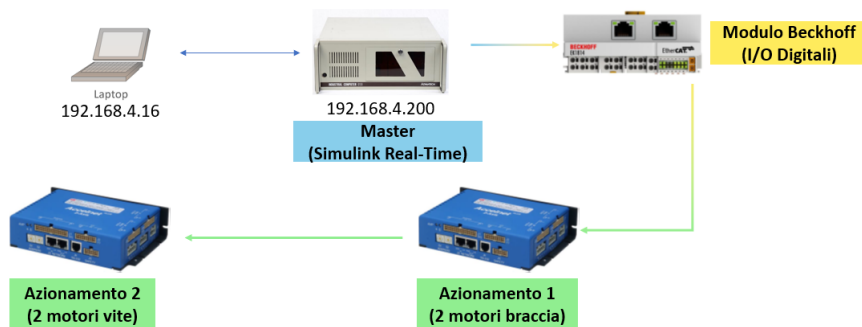


Figura 28: Topologia della rete

PC dell'utente, il quale provvede a generare, compilare, e caricare ed i programmi sul PC target. Da User-PC è anche possibile vedere i grafici e fare delle analisi sui movimenti e le traiettorie eseguite dal manipolatore. La connessione avviene tramite una rete ethernet, l'indirizzo del target è 192.168.4.200, invece per User-PC: Va precisato che il pc dell'utente non fa parte della rete ethercat, ma la rete inizia

```
Ethernet adapter Ethernet 3:
Connection-specific DNS Suffix . : 
Link-local IPv6 Address . . . . . : fe80::61ef:4e4:4849:4a71%10
IPv4 Address. . . . . : 192.168.4.16
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :
```

Figura 29: Configurazione rete ethernet user PC

soltanto dal pc target in poi, infatti, ad esclusione delle operazioni viste prima il manipolatore non ha bisogno del pc utente per funzionare. Una volta configurata la

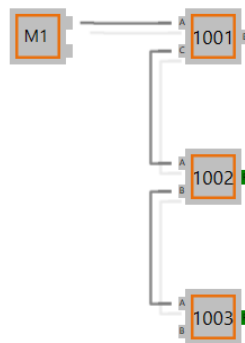


Figura 30: Topologia rete mediante Ec-engineer

rete, il passo successivo è stato quello della configurazione dei messaggi, come è stato anticipato nel capitolo precedente il metodo di comunicazione sono le PDO. Le PDO possono essere in input o in output, la differenza sta nel fatto che le primo sono PDO che gli azionamenti trasmettono al master, di conseguenza il master le riceve, quelle di output invece sono PDO che il master trasmette e che gli azionamenti ricevono. Nelle immagini seguenti sono elencate le PDO di input, in particolare:

- PDO1 e PDO2 contengono i parametri di posizione, velocità coppia effettiva e modalità operativa che vengono trasmesse dagli azionamenti

- PDO3, contiene input generici che sono indipendenti dalla modalità operativa del motore, in particolare è presente il *general purpose inputs* che è il registro che permette la visione dei finecorsa
- PDO4 contiene *status word* e *control word*, sono registri importanti che servono per verificare la modalità operativa e lo stato dell'azionamento, quindi sono utili per capire se l'azionamento è in fase pre-operativa, operativa o in errore

<input checked="" type="checkbox"/> Transmit PDO 1		
Name	Index	Bit Length
Axis A Modes of operation	0x6060:00	8
Axis A Actual motor position	0x6064:00	32
Axis A Actual motor velocity	0x606C:00	32
Axis A Torque actual value	0x6077:00	16

(a) PDO Input 1

<input checked="" type="checkbox"/> Transmit PDO 2		
Name	Index	Bit Length
Axis B Modes of operation	0x6860:00	8
Axis B Actual motor position	0x2A40:00	32
Axis B Actual motor velocity	0x6869:00	32
Axis B Torque actual value	0x6877:00	16

(b) PDO Input 2

<input checked="" type="checkbox"/> Transmit PDO 3		
Name	Index	Bit Length
General purpose inputs	0x2190:00	16
Axis A Peak current limit	0x2110:00	16
Axis B Peak current limit	0x2910:00	16

(c) PDO Input 3

<input checked="" type="checkbox"/> Transmit PDO 4		
Name	Index	Bit Length
Axis A Status word	0x6041:00	16
Axis B Status word	0x6841:00	16
Axis A Control word	0x6040:00	16
Axis B Control word	0x6840:00	16

(d) PDO Input 4

Figura 31: PDO in input

Per quanto riguarda le PDO che riceve l'azionamento sono solo due, ed i parametri ricevuti sono:

- *Modes of operation*, è un registro che specifica la modalità con la verrà controllato l'azionamento, per esempio coppia, posizione, velocità o ciclica
- *Target torque*, specifica il valore di coppia che l'azionamento dovrà fornire al motore

<input checked="" type="checkbox"/> Receive PDO 1			0x1600
Name	Index	Bit Length	
Axis A Modes of operation	0x6060:00	8	
Axis A Target Torque	0x6071:00	16	
<input checked="" type="checkbox"/> Receive PDO 2			0x1601
Name	Index	Bit Length	
Axis B Modes of operation	0x6860:00	8	
Axis B Target Torque	0x6871:00	16	

Figura 32: PDO Output 1 e 2

6.2 Implementazione nel sistema reale

Una volta ottenuto il file ENI contenente la topologia della rete è stato utilizzato simulink real-time per implementare la logica di controllo del manipolatore Il

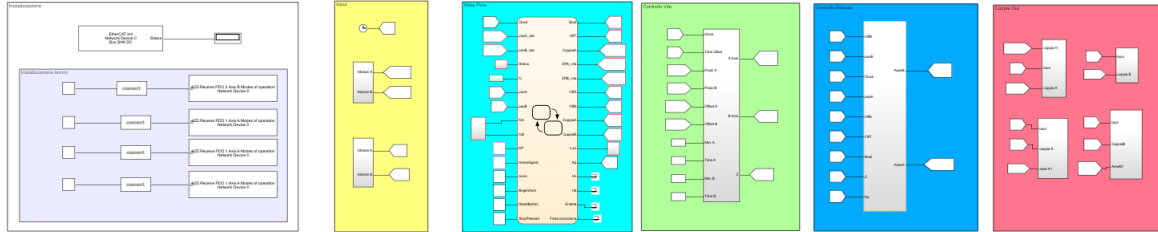


Figura 33: Schema generale simulink

programma è stato diviso in sei stati diversi, andiamo ad analizzare ora i vari stati.

Inizializzazione

La prima fase è quella di inizializzazione, in questa fase viene inserito il file ENI e viene specificata la modalità operativa degli azionamenti. Oltre al file ENI viene

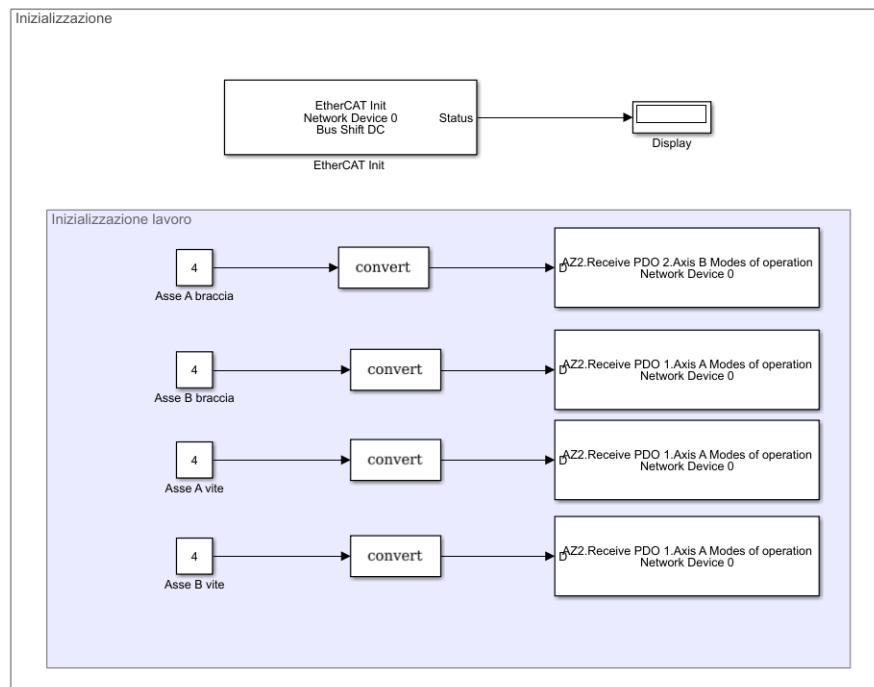


Figura 34: Fase 1: Inizializzazione

anche specificata la porta di comunicazione ed il bus che verranno utilizzati per

lo scambio di dati. Ogni azionamento ha poi una determinata modalità operativa, come abbiamo visto nelle sezioni precedenti il controllo è effettuato in coppia.

Modalità	Descrizione
1	modalità profilo in posizione
3	modalità profilo in velocità
4	modalità profilo in coppia
6	modalità homing
7	modalità posizione interpolata

Tabella 4: Tipologie controllo azionamenti

Input

La seconda fase è quella di input, in questa fase andiamo a prendere tutti i valori di posizione dei motori sia della vite che delle braccia I valori vengono presi mediante

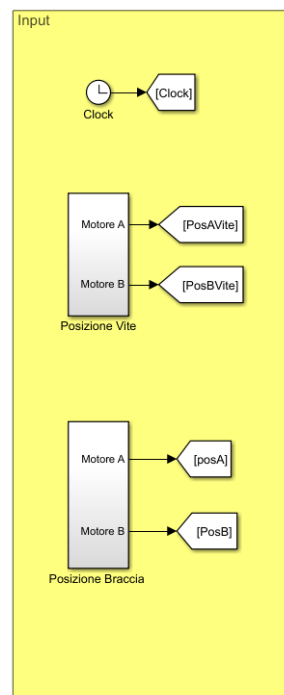


Figura 35: Fase 2: Input

i messaggi dalle PDO ed hanno bisogno di essere convertiti, proprio per questo la struttura di ricezione di un messaggio è la seguente:

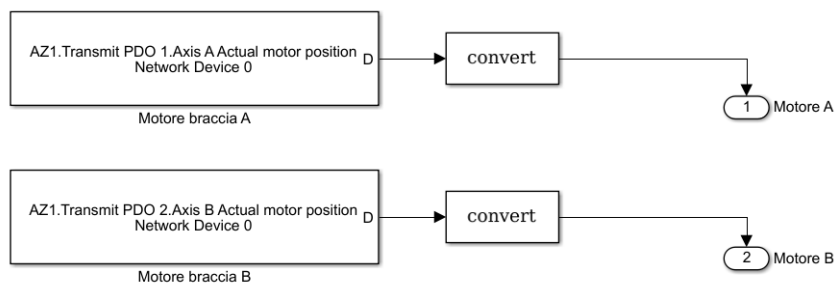


Figura 36: Conversione e lettura motori

Stateflow

Lo stateflow è un blocco dove viene posta la logica fondamentale dell'applicazione, in particolare in input avremo tutti i dati come ad esempio il clock, lo stato dei finecorsa, le posizioni dei motori e degli input che serviranno per l'interfaccia grafica. In output invece abbiamo gli offset, che ci serviranno per capire di quanto ci siamo

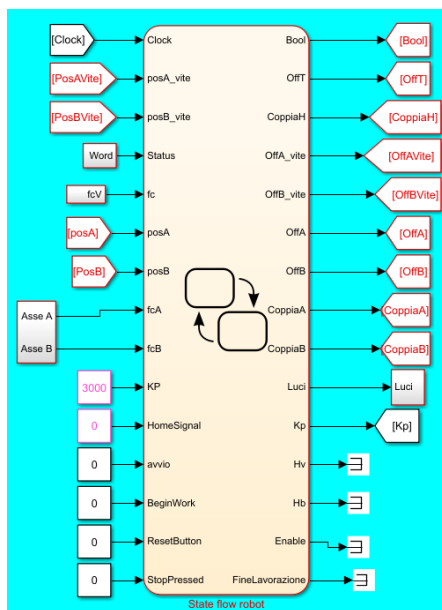


Figura 37: Fase 3: Stateflow

mossi nelle varie fasi, e quindi avere un riferimento sia di posizione che temporale, le coppie di homing dei motori e della vite, e un blocco relativo alla gestione delle luci⁶.

⁶abbiamo la presenza di tre luci: rosso, bianco e verde, nei paragrafi successivi verrà introdotto il loro comportamento

Controllo vite

Il blocco Controllo Vite, contiene lo schema del controllore implementato per controllare la vite, in ingresso abbiamo il clock, le posizioni dei motori con relativi offset e la movimentazione da far eseguire ad entrambi i motori della vite con il tempo di esecuzione. Grazie a tutti questi parametri potremo definire le leggi di moto che

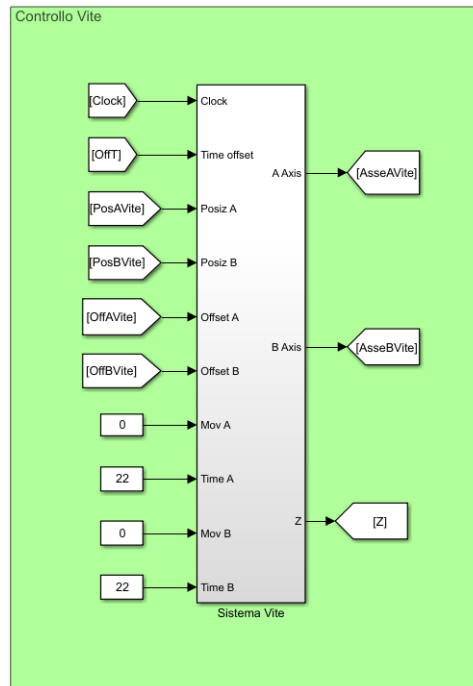


Figura 38: Fase 4: Schema di controllo della vite

ci permetteranno il movimento nell'asse Z. Importante è sapere che il movimento eseguito dalla vite a ricircolo di sfere è dipendente anche dal movimento eseguito dalla guida lineare, infatti la rotazione provoca anche un abbassamento della vite, per risolvere questo problema il motore della guida dovrà essere sempre pronto a rispondere e correggere questa situazione; in caso che i due motori vadano alla stessa velocità si avrà una rotazione senza traslazione.

Controllo braccia

Il blocco Controllo Braccia è quello responsabile della movimentazione dei link motorizzati, come per il blocco della vite prende in ingresso le posizioni, il *clock* con

i relativi offset e dentro vengono svolte le operazioni di generazione della legge di moto e di controllo. In uscita avremo le coppie che saranno assegnate ai due assi.

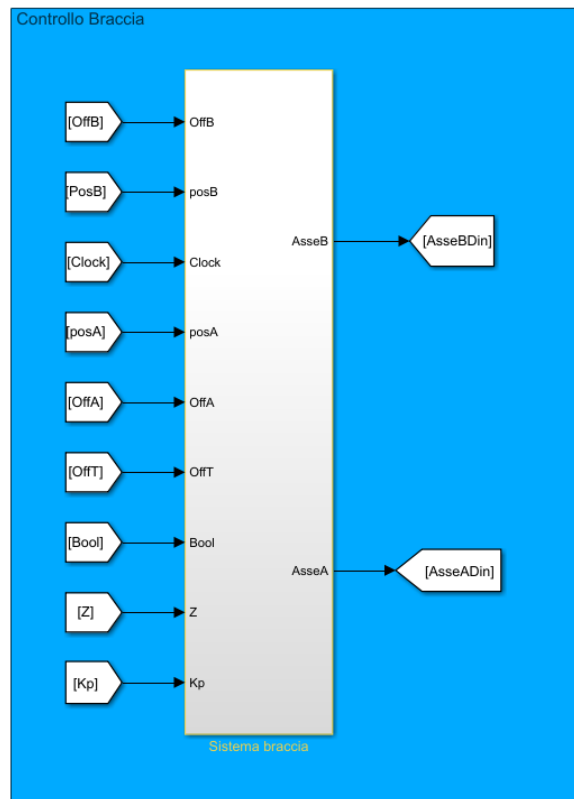


Figura 39: Fase 5: Schema di controllo delle braccia

Coppie uscita

Dopo aver ottenuto le coppie di homing dallo stateflow e le coppie dei motori dagli schemi di controllo è venuto il momento di inviare le coppie agli azionamenti e di conseguenza ai motori; per far questo usiamo un blocco per ogni motore, oltre alle coppie in entrata avremo anche una variabile di controllo. Nella figura precedente

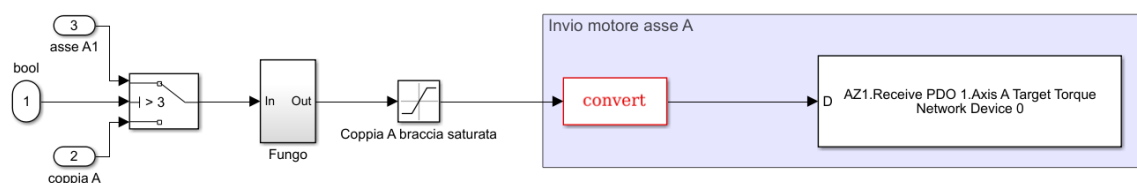


Figura 40: Schema espanso invio coppia

è possibile vedere lo schema espanso come spiegato, lo switch permette la scelta in base alla variabile bool, fintanto che è minore o uguale a 3 verrà erogata solo la coppia di Homing, quando arriva a 4 invece vuol dire che siamo nella fase di controllo, di conseguenza verrà erogata la coppia di controllo. Successivo allo switch c'è un blocco che serve per la gestione delle emergenze, infatti, una volta premuto il fungo verrà assegnata una coppia costante uguale a 0 che fermerà la lavorazione ed anche dopo che verrà sbloccato il fungo la coppia per sicurezza rimarrà a zero, l'unico modo per resettare questa condizione è il riavvio del programma. Successivo al blocco fungo abbiamo un saturatore, questo serve per evitare di danneggiare il manipolatore in caso che le coppie computate siano molto alte, è stato trovato sperimentalmente un limite che coincide con la coppia nominale che non può essere superato, per concludere l'ultima parte è quella che si occupa di inviare mediante PDO il valore di coppia convertito all'azionamento.

Il ragionamento appena discusso si va ad applicare ai restanti tre casi, possiamo quindi definire lo schema totale come segue: Andiamo ora a concentrarci sulle parti

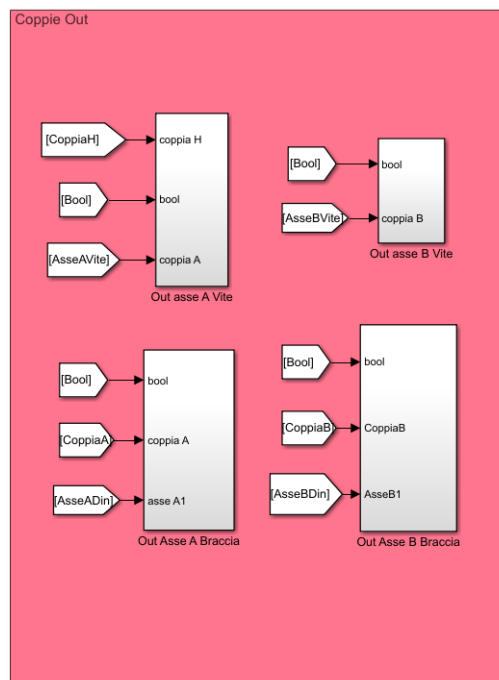


Figura 41: Fase 6: Copie in uscita

principali di questo programma, in particolare andremo a trattare lo stateflow nello

specifico, vedremo poi l'interfaccia grafica e le tipologie di controllore implementate per la vite e per le braccia.

6.3 Stateflow

Stateflow si occupa di fornire diagrammi di transizione, stato e di flusso utilizzando un linguaggio grafico. Nel caso del manipolatore è stato utilizzato per la progettazione di diagrammi di transizione in base agli stati del robot. In questa sezione andremo a vedere le fasi gli stadi di evoluzione che sono stati costruiti.

6.3.1 Fase di Zero

Appena il robot viene acceso non possiamo sapere dove si trova, di conseguenza abbiamo bisogno di uno stadio che ci vada a trovare una posizione di riferimento nella quale sappiamo dove è collocato effettivamente. Il primo stadio è quindi quello di *zero*, consiste nel portare i motori a toccare i fincorsa indicandogli che quello è il loro punto di partenza. I motori utilizzati per questa fase sono stati quelli delle braccia e quello di traslazione della vite.

L'approccio iniziale è stato quello di fornire una coppia costante che in automatico si occupava di andare a toccare i fincorsa, e dopo che li toccava si passava nello stato successivo. Però, per motivi di sicurezza e considerando che lasciando fermo il manipolatore per diverso tempo la stessa coppia magari potrebbe non farlo muovere si è deciso di chiudere l'anello in posizione, in particolare per eseguire la fase di zero è stata data una rampa con pendenza negativa in quanto lo zero viene fatto verso destra. Per controllare questa rampa è stato fatto un controllo proporzionale e

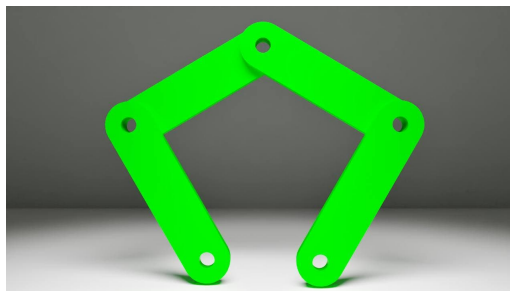


Figura 42: DA CAMBIARE CON CONF HOME

integrale sull'errore tra la posizione attuale ed il riferimento, la legge implementata è quindi del tipo:

$$PI = \frac{K_p s + K_i}{s}$$

Nella figura successiva andiamo a vedere lo schema effettivamente implementato per questa prima fase:

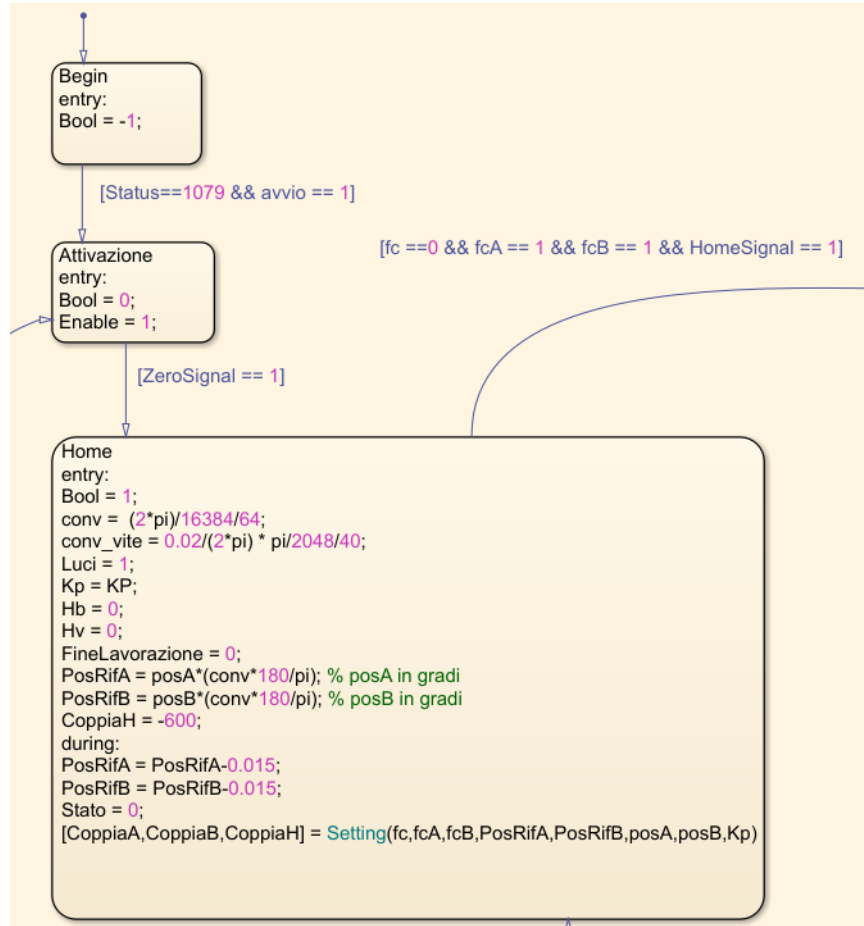


Figura 43: Fase di zero

Il primo stato è quello di begin, appena il programma si avvia ci entriamo in automatico, per passare allo stato successivo, ovvero attivazione abbiamo bisogno di due condizioni, la prima si verifica quando la *Status Word* è uguale a 1079, ovvero quando gli azionamenti sono usciti dalla fase *pre-operational* e sono quindi pronti all'uso, la seconda condizione invece è quando il segnale avvio è vero, nello stato attivazione i motori sono ancora fermi però viene abilitato il loro utilizzo, passare quindi in questo stato è obbligatorio. Per passare allo stato successivo il segnale

ZeroSignal deve essere vero, questo è un segnale gestito mediante un bottone da interfaccia grafica, appena viene premuto entriamo nello stato **Home**. In questo stato vengono definite variabili per la conversione dei valori⁷, viene poi salvata la posizione di riferimento dell'asse A e B dei motori delle braccia. La fase successiva è quella del *during*, rimanendo in quello stato quella fase viene eseguita ad ogni ciclo (1ms), in questa abbiamo la rampa che decresce di 0.015 gradi al millisecondo (15 gradi al secondo) e abbiamo una funzione simulink che è quella che si occupa del controllo PI. Una volta raggiunta la posizione del finecorsa le coppie vengono settate a 0, impedendo quindi un'ulteriore movimentazione. Per passare alla fase successiva abbiamo bisogno che tutti gli elementi siano arrivati a finecorsa.

6.3.2 Fase di Homing

La fase successiva è quella di homing, definita anche fase di posizionamento, per non lasciare il robot nella configurazione di zero si è scelto di spostarlo in una configurazione standard lontana dai punti di singolarità e che sarà comoda per le movimentazioni e traiettorie successive. La configurazione scelta prevede che i giunti siano messi a 100° e 80° , anche qua come prima il primo approccio è stato quello di utilizzare una coppia costante per il movimento; la fase di zero lasciava i link a 60° e -30° , vi era quindi la necessità di fare $+40^\circ$ per il motore a sinistra e $+110^\circ$ per il motore a destra, la coppia costante veniva erogata finché non si arrivava a quelle condizioni, dopodiché eravamo sicuri che il posizionamento era stato fatto in maniera corretta. La vite invece ha due modalità di configurazione, è possibile farla abbassare o lasciarla alta in finecorsa, si è scelto di proseguire in questo modo in quanto alla vite come abbiamo visto precedentemente è possibile collegare degli utensili e quindi l'abbassamento la predisponeva al disegno per traiettorie bidimensionali. L'approccio di controllo finale però non è stato quello della coppia costante, anche in questo caso per motivi di sicurezza ma, sapendo le posizioni finali che si volevano raggiungere, si è optato per definire una legge di moto che si occupava di portarci in quella condizione. Inserisci descrizione schema

⁷I valori presi dai motori sono espressi tutti in conti, per quello è stata necessaria una fase di analisi dei motori per capire come convertirli, successivamente c'è stato un passaggio da counts a radianti e da radianti a gradi.

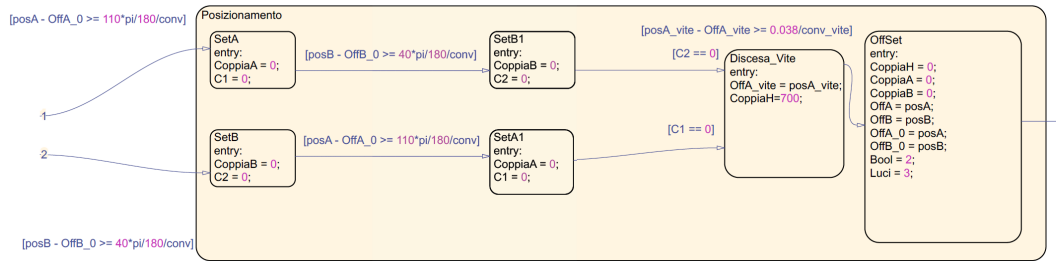


Figura 44: DA CAMBIARE

6.3.3 Fase di traiettoria

L'ultima fase è quella della traiettoria, in questa fase ri-settiamo gli offset della vite in quanto il posizionamento è stato appena concluso ed andiamo a settare l'offset del tempo, questo ci permette di far partire virtualmente il tempo da zero dopo che è stata raggiunta la configurazione finale di *homing*. Durante tutte le fasi

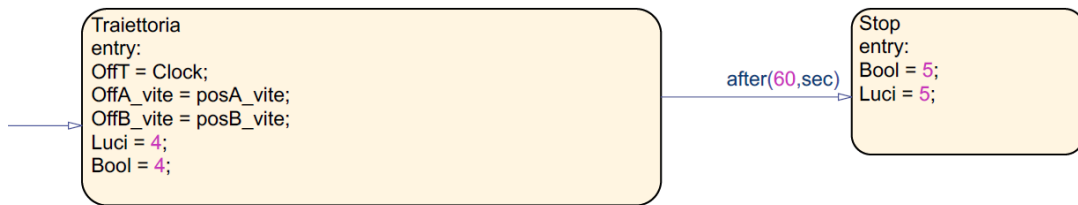


Figura 45: State flow: traiettoria

anche due variabili si evolvono costantemente, queste sono bool e luci, la prima ci va ad indicare lo stato preciso in cui ci troviamo in base alla seguente tabella: la seconda invece serve a pilotare le luci in modo che si ha un feedback visuale

Valore	Stato
0	pre-operazionale
1	Homing
2	Manipolatore posizionato
3	Traiettoria
4	Manipolatore fermo

Tabella 5: Valori di *bool*

della fase in cui è il manipolatore. Dopo che il manipolatore entra nello stato della

Valore	Colore	Bool
1	Bianco	1
2	Bianco Rosso	1
3	Rosso	2
4	Bianco Verde	3
5	Verde	4

Tabella 6: Valori delle luci

traiettoria quest'ultima viene eseguita in base ad una selezione, per passare allo stato successivo invece dovranno passare 60 secondi, l'ultimo stato è uno stato di fermo, il manipolatore ha finito la traiettoria e di conseguenza va in "riposo" mettendo a zero tutte le coppie.

6.4 Struttura del manipolatore

6.4.1 Sistema vite

6.4.2 Sistema braccia

6.4.3 Gestione della traiettoria

6.5 Controllo

In questa sezione andremo a vedere le tipologie di controllo che sono state utilizzate, verranno mostrati anche dei grafici e delle analisi relative a questi metodi.

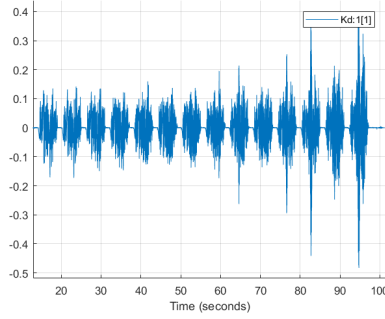
6.5.1 Controllo Vite

6.5.2 Controllo PD

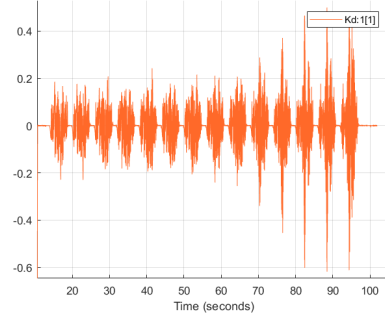
6.5.3 Controllo in dinamica inversa

Il primo controllo per il robot è stato fatto mediante l'utilizzo di un proporzionale che comprendeva la somma di posizione e velocità, si è riscontrato che a causa del mancato contributo derivativo ed integrativo la traiettoria non veniva seguita in maniera del tutto corretta, ed alzano il valore di K_p troppo le vibrazioni aumentavano, si è quindi cercato inizialmente di trovare un punto nel quale il manipolatore

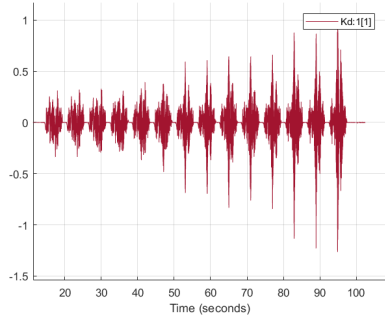
non vibrava e non oscillava. Successivamente si è andato ad applicare un controllo in dinamica inversa, la prima tipologia di controllo scelta è stata quella in anello aperto [inserisci schema], dopo i test effettuati con questa tipologia di controllo si è riscontrato che le coppie risultanti erano basse, all'incirca $0.5C_m$, con C_m la coppia per far muovere il manipolatore. L'approccio successivo è stato quello di continuare con la dinamica inversa ma chiudendo l'anello, [inserisci schema] Un test che è stato fatto per andare a ricercare i parametri è stato quello di far variare K_p , in un range di valori compreso tra 350 e 950 ad intervalli di 50, in base ad un valore fisso di K_d , andiamo ora a vedere un risultato di questo metodo. Si può chiaramente vedere



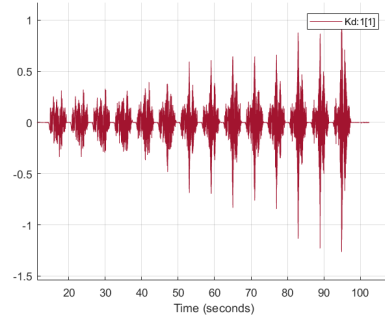
(a) Test con $K_d = 1.5$



(b) Test con $K_d=2$



(c) Test con $K_d=3$



(d) Test con $K_d=3$

Figura 46: Test su K_d

come all'aumentare di K_d i valori di K_p iniziano ad vibrare sempre prima. Come valori finali per questo controllore sono stati scelti $K_d = \frac{3}{2}$ e $K_p = 720$

6.6 Traiettorie eseguite

In questa sezione andremo a vedere ed analizzare tutte le traiettorie che sono state create, le traiettorie variano da bidimensionali a tridimensionali, andando anche ad

includere due pattern.

7 Conclusioni

7.1 Confronto modellazione teorica-pratica

7.2 Sviluppi futuri

Elenco delle figure

1	Robot PKM	5
2	Rappresentazione fisica del robot	8
3	Equazioni alle circonferenze	9
4	Confronto dinamica leggi di moto su accelerazioni	21
5	Confronto dinamica leggi di moto su velocità	21
6	Confronto dinamica leggi di moto su posizioni	21
7	Modello simulink manipolatore	22
8	Modello Adams manipolatore 5R	22
9	Coppie in uscita Simulink	23
10	Coppie in uscita Adams	23
11	Errore Simulink-Adams	24
12	Caso 1 singolarità	25
13	Caso 2 singolarità	25
14	Caso 3 singolarità	26
15	Caso 4 e 5 singolarità	27
16	Punti di singolarità	28
17	Numero di condizionamento	29
18	Angoli di movimentazione dei giunti motorizzati	29
19	<i>Workspace</i> 5R	30
20	Numero di condizionamento workspace	30
21	Struttura vite	32
22	Utensile da disegno	32
23	Coppie e posizione end-effector	34
24	Schema azionamento	38
25	Registri azionamento	38
26	Banco di test	40
27	Schema modulo bechoff	42
28	Topologia della rete	42
29	Configurazione rete ethernet user PC	43

30	Topologia rete mediante Ec-engineer	43
31	PDO in input	44
32	PDO Output 1 e 2	44
33	Schema generale simulink	45
34	Fase 1: Inizializzazione	45
35	Fase 2: Input	46
36	Conversione e lettura motori	47
37	Fase 3: Stateflow	47
38	Fase 4: Schema di controllo della vite	48
39	Fase 5: Schema di controllo delle braccia	49
40	Schema espanso invio coppia	49
41	Fase 6: Copie in uscita	50
42	DA CAMBIARE CON CONF HOME	51
43	Fase di zero	52
44	DA CAMBIARE	54
45	State flow: traiettoria	54
46	Test su K_d	56