

UNIVERSITÀ DEGLI STUDI DI BERGAMO

Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Corso di Laurea Magistrale in Ingegneria Informatica

Classe n. LM-32- Classe delle lauree magistrali in ingegneria informatica

**INDAGINE TEORICA E Sperimentale sul
Controllo centralizzato per un
Manipolatore a cinematica parallela**

Relatore:

Chiar.mo Prof. Paolo Righettini

Correlatore:

Chiar.mo Prof. Roberto Strada

Tesi di Laurea Magistrale

Daniele RAVASIO

Matricola n. 1045934

Simone CORTINOVIS

Matricola n. 1047201

ANNO ACCADEMICO 2020 / 2021

Mistakes are not shackles that halt one from stepping forward.

Rather, they are that which sustain and grow one's heart.

– Mavis Vermillion

Ringraziamenti

Prima di procedere con la trattazione, vorrei dedicare qualche riga a tutti coloro che mi sono stati vicini in questo percorso di crescita personale.

Desidero ringraziare il Prof. Paolo Righettini ed il Prof. Roberto Strada per avermi aperto gli occhi al mondo della meccatronica.

Ringrazio il team del laboratorio di meccatronica dell'università degli studi di Bergamo, in particolare Jasmine Santinelli e Filippo Cortinovis per avermi seguito durante tutto questo percorso di tesi.

Ringrazio il mio collega Simone con il quale ho condiviso questi nove mesi di tesi; Andrea e Raul per aver affrontato questi due anni di magistrale insieme.

Non posso non ringraziare i miei genitori **Claudio** e Roberta, i miei nonni **Antonio**, Marisa, Giovanni e Lisetta e mia sorella Monica per avermi sempre sostenuto nella realizzazione dei miei progetti.

Infine, ringrazio Alessia, la persona che più di tutte nei momenti difficili ha creduto in me grazie al suo ottimismo. Grazie ad Alessia ho avuto il coraggio di sperimentare nuove idee, mettermi in gioco e capire che, gli ostacoli esistono per essere superati.

Indice

1	Introduzione	5
1.1	Obiettivo	5
1.2	Stato dell'arte	5
1.3	Ambiti applicativi	6
1.4	Software modellazione teorica	7
1.4.1	Matlab	7
1.4.2	Adams	7
2	Cinematica Manipolatore	9
2.1	Cinematica Diretta	10
2.1.1	Posizione	10
2.1.2	Velocità	13
2.1.3	Accelerazione	15
2.2	Cinematica inversa	16
2.2.1	Posizione	16
2.2.2	Velocità	17
2.2.3	Accelerazione	18
3	Dinamica Manipolatore	19
3.1	Prerequisiti per il calcolo della dinamica	19
3.1.1	Jacobiana J_{34} e calcolo $\dot{\theta}_3, \dot{\theta}_4$	19
3.1.2	Jacobiana J_{34} e calcolo di $\ddot{\theta}_3, \ddot{\theta}_4$	20
3.1.3	Matrici di inerzia	21
3.2	Principio dei lavori virtuali	22
3.3	Dinamica inversa	23
3.4	Dinamica diretta	24
	Modellazione su Adams	27
	Validazione e confronto	28
4	Analisi cinetostatica	31
4.1	Punti di singolarità	31

Primo e secondo caso	32
Terzo caso	33
Terzo e quarto caso	35
4.2 Manipolabilità	36
4.3 Workspace	37
5 Modellazione end-effector	41
5.1 Cinematica end-effector	42
5.2 Dinamica della vite	43
6 Tecnologie implementate	45
6.1 Simulink Real time	45
6.2 EtherCAT	45
6.2.1 Proprietà	46
6.2.2 Gestione della rete	46
6.2.3 Implementazione interfacce	47
6.3 CME2	47
6.4 EC-Engineer	49
7 Sistema reale	51
7.1 Struttura del robot	51
7.1.1 Azionamenti	52
7.1.2 Beckhoff EK1814	52
7.1.3 Configurazione della rete	53
7.2 Implementazione nel sistema reale	56
Inizializzazione	56
Input	57
Stateflow	58
Controllo vite	59
Controllo braccia	60
Coppie uscita	61
7.3 Stateflow	63

7.3.1	Fase di Homing	63
7.3.2	Fase di posizionamento	65
7.3.3	Fase di controllo	67
	Gestione variabile di stato e luci	69
7.3.4	Interfaccia grafica	70
7.4	Controllo vite	71
7.4.1	Controllo proporzionale	72
7.4.2	Controllo proporzionale derivativo	73
7.4.3	Confronto controllori vite	75
7.5	Controllo braccia	76
7.5.1	Controllo proporzionale derivativo	76
7.5.2	Controllo feed-forward con coppia pre-computata	79
7.5.3	Controllo in dinamica inversa	82
	Test Kp e Kd	84
7.5.4	Controllo robusto	86
	Implementazione controllore robusto	92
7.6	Confronto approcci controllori	93
7.7	Problema della velocità	96
	Filtro primo ordine	96
	Filtro secondo ordine	97
	Filtro di Kalman	98
8	Conclusioni	103
	Sviluppi futuri	103
Appendice A	Formule estese	105
A.1	Formule cinematica diretta accelerazione	105
A.2	Prerequisiti dinamica	107
Appendice B	Cenni sistemi dinamici	109
B.1	Stabilità asintotica	110
B.2	Metodo delle differenze finite	111

INDICE

Appendice C Quadro elettrico	113
Bibliografia	127
Elenco delle figure	129
Elenco delle tabelle	131

1 Introduzione

1.1 Obiettivo

L’obiettivo di questa tesi riguarda lo studio teorico e l’implementazione pratica di approcci di controllo centralizzato ad un manipolatore a cinematica parallela, avente 4 gradi di libertà, con l’obiettivo di andare a trovare la tecnica migliore. In primis verrà eseguita una modellazione meccanica teorica, successivamente mediante un’attività sperimentale che comprende l’implementazione degli schemi di controllo analizzati e la generazione di traiettorie bidimensionali e tridimensionali, si andranno ad analizzare i risultati ottenuti.

1.2 Stato dell’arte

Il lavoro di tesi è stato svolto in collaborazione con il laboratorio di meccatronica dell’università degli studi di Bergamo. Alla base di questa tesi vi è il manipolatore PKM (*parallel kinematic manipulator*) costruito nel 2013. 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 cinque giunti rotoidali, quattro link, di cui due motorizzati e all’estermità è connessa una vite che consente traslazione e rotazione nell’asse Z, in questa configurazione il robot arriva ad avere 4 gradi di libertà. Escludendo l’introduzione, il lavoro di tesi sarà articolato in sette capitoli: nel secondo capitolo verrà presentata l’analisi cinematica del manipolatore, in particolare verranno analizzate sia cinematica diretta che inversa, nel terzo capitolo si parlerà di dinamica, nel quarto tratteremo i punti di singolarità e manipolabilità del robot; il quinto capitolo mostrerà la modellazione dell’*end-effector*, nel nostro caso la vite; il sesto capitolo andrà a presentare tutte le tecnologie implementate a livello pratico, nel settimo capitolo verrà presentato

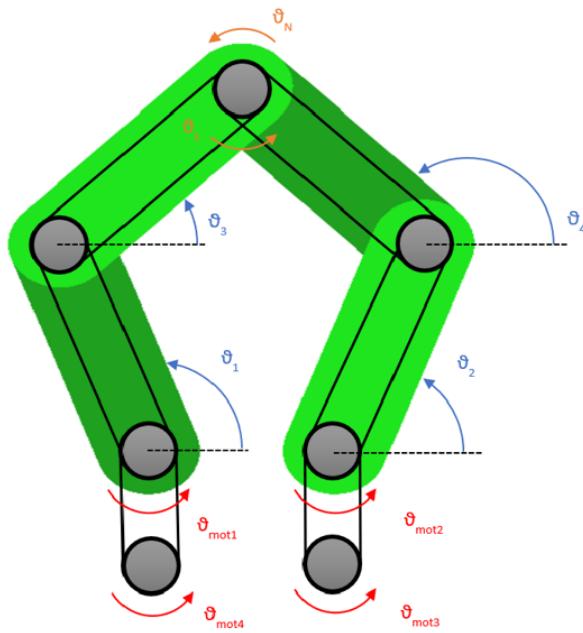


Figura 1.1: Robot PKM

il sistema reale, includendo la struttura, le modalità di comunicazione, il software implementato, il sistema di controllo, l’interfaccia grafica ed i problemi riscontrati con le relative soluzioni. Infine, nell’ottavo capitolo, verranno esposte le conclusioni, e gli sviluppi futuri.

1.3 Ambiti applicativi

Grazie alle loro caratteristiche, i manipolatori a cinematica parallela vengono utilizzati in diversi ambiti, di seguito è riportato solo qualche esempio:

- Simulatori di volo
- Simulatori di guida
- Allineamento e posizionamento della fibra ottica
- Ambito medicale
- Assemblamento *PCB*
- Operazioni di *pick & place*

1.4 Software modellazione teorica

Per la modellazione teorica del manipolatore, abbiamo bisogno di utilizzare strumenti software specifici; in particolare verranno utilizzati 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.

1.4.1 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'ambiente, 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.

1.4.2 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. È 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:

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

Tutti e quattro i link hanno lunghezza, massa, posizione del centro di massa e momento di inerzia uguali, per questo si è deciso di rappresentare i dati una sola volta.

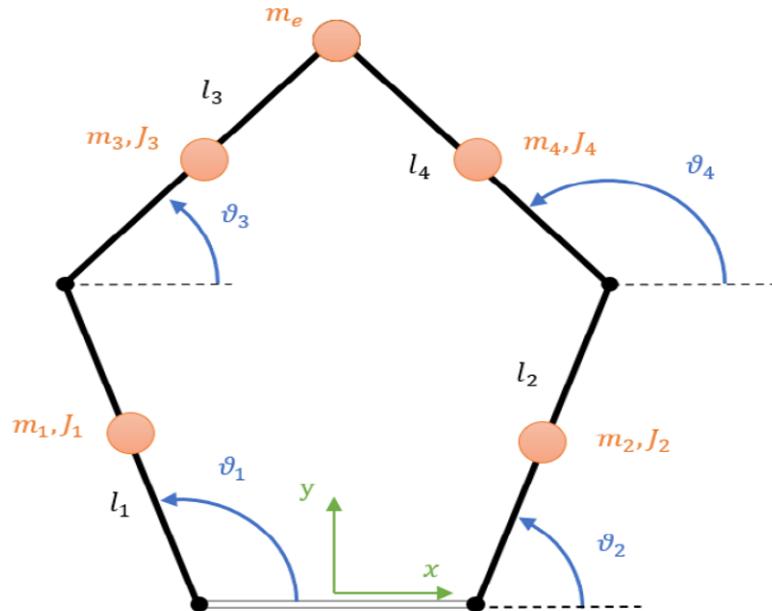


Figura 2.1: Rappresentazione fisica del robot

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

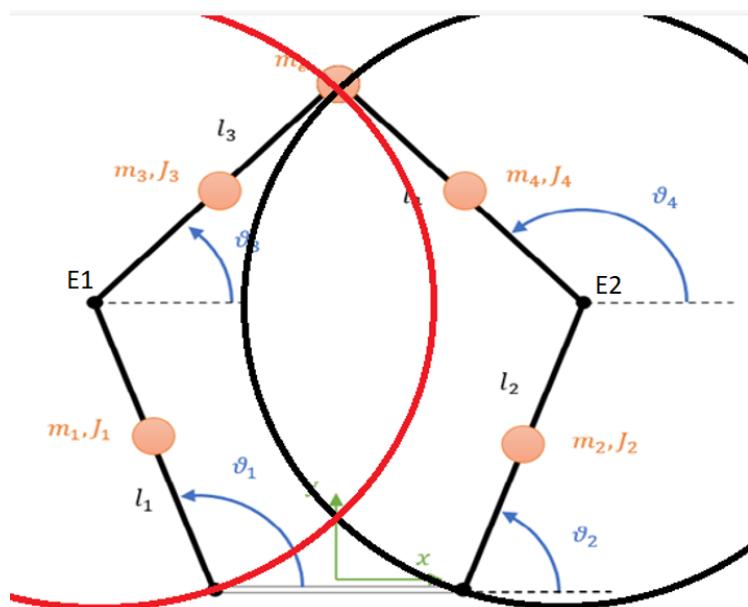


Figura 2.2: Equazioni alle circonferenze

Dalla combinazione di queste due equazioni otteniamo il seguente sistema:

$$\begin{cases} (x - \frac{l}{2} - l \cos \theta_1)^2 + (y - l \sin \theta_1)^2 = l^2 \\ (x + \frac{l}{2} - l \cos \theta_2)^2 + (y - l \sin \theta_2)^2 = l^2 \end{cases} \quad (2.1)$$

Da queste, andando a sviluppare i calcoli otteniamo:

$$\begin{cases} x^2 + x(-2d - 2l \cos \theta_2) + y^2 + y(-2d \sin \theta_2) + d(d + 2l \cos \theta_2) = 0 \\ x^2 + x(2d - 2l \cos \theta_1) + y^2 + y(-2d \sin \theta_1) + d(d - 2l \cos \theta_1) = 0 \end{cases} \quad (2.2)$$

Andando a sottrarre le equazioni:

$$x(-4d - 2l(\cos \theta_2 - \cos \theta_1)) + y(-2l(\sin \theta_2 - \sin \theta_1)) + 2ld(\cos \theta_2 + \cos \theta_1) = 0$$

Una volta risolto è possibile ricavare x in funzione di y come:

$$x = \frac{y \cdot l(\sin \theta_2 - \sin \theta_1) - l \cdot d(\cos \theta_2 + \cos \theta_1)}{-2d - l(\cos \theta_2 - \cos \theta_1)} \quad (2.3)$$

Sostituendo la x nell'equazione 2.1 possiamo ricavare y come:

$$y_{1,2}^1 = \frac{-B \pm \Delta}{2A}$$

Dove i parametri A,B e C sono:

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

$$\begin{aligned} B = & -2l^2d(\sin \theta_2 - \sin \theta_1)(\cos(\theta_2 + \theta_1) + l(\sin \theta_2 - \sin \theta_1)(-2d - l(\cos \theta_2 - \cos \theta_1))) \\ & (-2d - 2l \cos \theta_2) - 2l \sin \theta_2(-2d - l(\cos \theta_2 - \cos \theta_1))^2 \end{aligned}$$

$$\begin{aligned} C = & l^2d^2(\cos \theta_2 + \cos \theta_1)^2 - ld(\cos \theta_2 + \cos \theta_1)(-2d - l(\cos \theta_2 - \cos \theta_1)) \\ & (-2d - 2l \cos \theta_2) + (d^2 + 2dl \cos \theta_2)(-2dl(\cos \theta_2 - \cos \theta_1))^2 \end{aligned}$$

¹ $\Delta = \sqrt{B^2 - 4AC}$

Il passo successivo è quello di ricavare la posizione $P = [x, y]$ dell'end-effector. La formula della y fornisce due risultati diversi, in uno caso quando si utilizza $+\Delta$ e l'altro quando si usa $-\Delta$. Per come è modellato il manipolatore, l'unica soluzione fisicamente raggiungibile è quella con delta positivo. Possiamo poi andare a trovare le posizioni dei giunti non motorizzati mediante relazioni geometriche nel seguente modo:

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

e

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

Possiamo andare a trovare poi gli angoli θ_3, θ_4 in funzione di θ_1 e θ_2 partendo dall'equazioni dei *loop vettoriali*:

$$\begin{cases} l \cos \theta_1 + l \cos \theta_3 - l \cos \theta_2 - l \cos \theta_4 - 2d = 0 \\ l \sin \theta_1 + l \sin \theta_3 - l \sin \theta_2 - l \sin \theta_4 = 0 \end{cases} \quad (2.4)$$

Elevando al quadrato:

$$\begin{cases} \cos^2 \theta_4 = (\cos \theta_1 - \cos \theta_2 - 2\frac{d}{l})^2 + 2 \cos \theta_3 (\cos \theta_1 - \cos \theta_2 - 2\frac{d}{l}) + \cos^2 \theta_3 \\ \sin^2 \theta_4 = (\sin \theta_1 - \sin \theta_2)^2 + 2 \sin \theta_3 (\sin \theta_1 - \sin \theta_2) + \sin^2 \theta_3 \end{cases}$$

Andando poi a sommare le equazioni troviamo:

$$(\cos \theta_1 - \cos \theta_2 - 2\frac{d}{l})^2 + (\sin \theta_1 - \sin \theta_2)^2 + 2 \cos \theta_3 (\cos \theta_1 - \cos \theta_2 - 2\frac{d}{l}) + 2 \sin \theta_3 (\sin \theta_1 - \sin \theta_2) = 0$$

Sono state introdotte equazioni parametriche in sostituzione di seno e coseno:

$$\cos \theta_3 = \frac{1 - t^2}{1 + t^2}, \quad \sin \theta_3 = \frac{2t}{1 + t^2}, \quad t = \tan \frac{\theta_3}{2}$$

Da questi possiamo ricavare un'equazione di secondo grado con incognita t dalla quale risolvendo con il delta positivo possiamo ricavare θ_3 in funzione di θ_1, θ_2

$$\theta_3 = -2 \cdot \operatorname{tg}^{-1}$$

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

Il ragionamento usato per ricavare θ_3 si può applicare in maniera analoga per ottenere θ_4 in funzione di θ_1, θ_2 come:

$$\theta_4 = -2 \cdot \operatorname{tg}^{-1}$$

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

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 al calcolo delle velocità, mediante la cinematica diretta possiamo ricavare le velocità sulle coordinate x e y dell'*end-effector*. A partire dall'equazione 2.1 ed facendo la derivata rispetto al tempo troviamo:

$$\begin{cases} \dot{x}(2x - 2d - 2l \cos \theta_2) + \dot{y}(2y - 2l \sin \theta_2) + \dot{\theta}_2(2lx \sin \theta_2 - 2ly \cos \theta_2 - 2dl \sin \theta_2) = 0 \\ \dot{x}(2x + 2d - 2l \cos \theta_1) + \dot{y}(2y - 2l \sin \theta_1) + \dot{\theta}_2(2lx \sin \theta_1 - 2ly \cos \theta_1 + 2dl \sin \theta_1) = 0 \end{cases}$$

Possiamo infine definire una jacobiana che ci permette di trovare il rapporto appena espresso. Dalla prima equazione ricaviamo \dot{x} come:

$$\dot{x} = \frac{-\dot{y}(2y - 2l \sin \theta_2) - \dot{\theta}_2(2lx \sin \theta_2 - 2ly \cos \theta_2 - 2dl \sin \theta_2)}{2x - 2d - 2l \cos \theta_2}$$

Sostituendo \dot{x} nella seconda equazione possiamo ricavare \dot{y} .

$$\dot{y} = \frac{N_{21}}{D_2} \dot{\theta}_1 + \frac{N_{22}}{D_2} \dot{\theta}_2$$

Dove:

$$\begin{aligned} N_{21} &= -l \sin \theta_1 (x + d - l \cdot \cos \theta_1 + l \cdot \cos \theta_1 (y - l \sin \theta_1)) \\ N_{22} &= -l \cos \theta_2 \cdot \frac{y - l \sin \theta_2 (x + d - l \cos \theta_1)}{x - d - l \cos \theta_2} + l \sin \theta_2 (x + d - l \cos \theta_1) \\ D_2 &= \frac{y - l \sin \theta_2 (x + d - l \cos \theta_1)}{x - d - l \cos \theta_2} \end{aligned}$$

Sostituendo \dot{y} nella prima equazione otteniamo

$$\dot{x} = \frac{N_{11}}{D_1} \dot{\theta}_1 + \frac{N_{12}}{D_1} \dot{\theta}_2$$

Dove:

$$\begin{aligned} N_{11} &= y - l \sin \theta_2 \cdot N_{21} \\ N_{12} &= -y + l \sin \theta_2 \cdot N_{22} - l \sin \theta_2 (x - d - l \cos \theta_2) + l \cos \theta_2 (y - l \sin \theta_2) \\ D_1 &= (x - d - l \cos \theta_2) D_2 \end{aligned}$$

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

$$J_{11} = \frac{N_{11}}{D_1}$$

$$J_{12} = \frac{N_{12}}{D_1}$$

$$J_{21} = \frac{N_{21}}{D_2}$$

$$J_{22} = \frac{N_{22}}{D_2}$$

Posizionando i termini della matrice possiamo quindi definire J come:

$$J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \quad (2.5)$$

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

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = J \cdot \dot{\Theta} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} J_{11}\dot{\theta}_1 + J_{12}\dot{\theta}_2 \\ J_{21}\dot{\theta}_1 + J_{22}\dot{\theta}_2 \end{bmatrix} \quad (2.6)$$

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:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = J\dot{\Theta} + J\ddot{\Theta} = J \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + J \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} \quad (2.7)$$

In questa equazione l'unico elemento non noto è J che è la derivata rispetto al tempo dello jacobiano J.

$$J = \begin{bmatrix} j_{11} & j_{12} \\ j_{21} & j_{22} \end{bmatrix}$$

Analiticamente essendo le derivate troppo lunghe ed onerose sono state trovate mediante il calcolo simbolico fornito da matlab, successivamente state confrontate con le derivate numeriche per verificarne la correttezza.

Presentiamo ora i risultati semplificati relativi alle equazioni delle accelerazioni, introducendo i parametri:

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

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

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

$$\ddot{x} = -\frac{\ddot{y}(y - l \sin \theta_1)}{x + d - l \cos \theta_1} \quad (2.8)$$

$$\ddot{y} = \frac{\frac{B_{acc} \cdot (x - d - l \cos \theta_2)}{x + d l \cos \theta_1 - A_{acc}}}{y - l \sin \theta_2 - \frac{x - d - l \cos \theta_2}{(x + d - l \cos \theta_1) \cdot (y - l \sin \theta_1)}} \quad (2.9)$$

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

A partire dall'equazione 2.2, raccogliendo i termini rispetto a θ_1 e θ_2 troviamo:

$$\begin{bmatrix} p \cos \theta_1 + e \sin \theta_1 = f \\ a \cos \theta_2 + b \sin \theta_2 = c \end{bmatrix}$$

Dove:

$$p = 2dl + 2xl$$

$$e = 2yl$$

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

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 .

Introducendo le equazioni parametriche per θ_1 :

$$\cos \theta_1 = \frac{1 - t^2}{1 + t^2}, \quad \sin \theta_1 = \frac{2t}{1 + t^2}, \quad t = \tan \frac{\theta_1}{2}$$

ricaviamo la seguente equazione di secondo grado:

$$t^2(p + f) - 2et + f - p = 0 \rightarrow t_{1,2} = \frac{e \pm \sqrt{p^2 + e^2 - f^2}}{p + f}$$

ritornando alla variabile originale $t \rightarrow \theta_1$ troviamo:

$$\theta_1 = 2 \arctan \frac{e + \sqrt{p^2 + e^2 - f^2}}{p + f} \quad (2.10)$$

in modo analogo si può applicare lo stesso ragionamento per θ_2

$$\theta_2 = 2 \arctan \frac{b - \sqrt{a^2 + b^2 - c^2}}{a + c} \quad (2.11)$$

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 posizionamento non raggiungibile fisicamente.

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

2.5 andiamo poi ad invertirla e ricaviamo le velocità $\dot{\theta}_1$ e $\dot{\theta}_2$ nel seguente modo:

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = J^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \quad (2.12)$$

Con

$$J^{-1} = \frac{1}{J_{11}J_{22} - J_{12}J_{21}} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} \quad (2.13)$$

In particolare nel caso in cui il determinante si annulla lo jacobiano non è invertibile, di conseguenza siamo in presenza di un caso singolare². 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 invertire l'equazione:

$$\ddot{P} = J\ddot{\Theta} + J\dot{\Theta}$$

ottenendo quindi:

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = J^{-1} \left(\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} - J \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \right) \quad (2.14)$$

²I punti di singolarità sono punti nei quali il manipolatore non si comporta in modo standard, potrebbero originarsi rotture, verranno descritti in modo approfondito nel capitolo riguardante l'analisi cinetostatica

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 . Dall'equazione dei loop vettoriali 2.4 derivandola otteniamo:

$$\begin{cases} -\sin \theta_1 \dot{\theta}_1 - \sin \theta_3 \dot{\theta}_3 + \sin \theta_2 \dot{\theta}_2 + \sin \theta_4 \dot{\theta}_4 = 0 \\ \cos \theta_1 \dot{\theta}_1 + \cos \theta_3 \dot{\theta}_3 - \cos \theta_2 \dot{\theta}_2 - \cos \theta_4 \dot{\theta}_4 = 0 \end{cases} \quad (3.1)$$

Considerando anche le velocità $\dot{\theta}_1$ e $\dot{\theta}_2$ possiamo dobbiamo ricavare $\dot{\theta}_3, \dot{\theta}_4$, per far questo dalla seconda equazione ricaviamo $\dot{\theta}_4$ in funzione di $\dot{\theta}_3$:

$$\dot{\theta}_4 = \frac{\cos \theta_1 \dot{\theta}_1 + \cos \theta_3 \dot{\theta}_3 - \cos \theta_2 \dot{\theta}_2}{\cos \theta_4}$$

Sostituendo $\dot{\theta}_4$ nella prima equazione otteniamo:

$$\cos \theta_4 (-\sin \theta_1 \dot{\theta}_1 - \sin \theta_3 \dot{\theta}_3 + \sin \theta_2 \dot{\theta}_2) + \tan \theta_4 (\cos \theta_1 \dot{\theta}_1 + \cos \theta_3 \dot{\theta}_3 - \cos \theta_2 \dot{\theta}_2) = 0$$

Da questa possiamo ricavare $\dot{\theta}_3$ come:

$$\dot{\theta}_3 = \frac{N_{13}}{D_{13}}\dot{\theta}_1 + \frac{N_{23}}{D_{13}}\dot{\theta}_2 \quad (3.2)$$

dove:

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

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

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

Avendo ottenuto questo valore possiamo anche esprimere $\dot{\theta}_4$ come:

$$\dot{\theta}_4 = \frac{N_{14}}{D_{14}}\dot{\theta}_1 + \frac{N_{24}}{D_{14}}\dot{\theta}_2 \quad (3.3)$$

dove:

$$N_{14} = \frac{\sin \theta_1 \cos \theta_3}{\sin \theta_3} - \frac{\cos \theta_4}{\sin \theta_4} + \frac{\cos \theta_4}{\sin \theta_4} \sin \theta_1 - \cos \theta_1$$

$$N_{24} = -\frac{\sin \theta_2 \cos \theta_3}{\sin \theta_3} - \frac{\cos \theta_4}{\sin \theta_4} - \frac{\cos \theta_4}{\sin \theta_4} \sin \theta_2 + \cos \theta_2$$

$$D_{14} = \frac{\sin \theta_4 \cos \theta_3}{\sin \theta_3} - \frac{\cos \theta_4}{\sin \theta_4}$$

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:

$$J_{34} = \begin{bmatrix} \frac{N_{13}}{D_{13}} & \frac{N_{23}}{D_{13}} \\ \frac{N_{14}}{D_{14}} & \frac{N_{24}}{D_{14}} \end{bmatrix} \quad (3.4)$$

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 giunti non motorizzati occorre derivare tutti gli elementi visti in precedenza, in particolare:

$$\dot{N}_{13} = \frac{-1}{\sin^2 \theta_4 \cdot \dot{\theta}_4 \sin \theta_1} + \frac{\cos \theta_4}{\sin \theta_4} \cos \theta_1 \dot{\theta}_1 + \sin \theta_1 \dot{\theta}_1$$

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

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

$$\dot{J}_{34} = \begin{bmatrix} \frac{\dot{N}_{13}D_{13} - N_{13}\dot{D}_{13}}{D_{13}^2} & \frac{\dot{N}_{23}D_{13} - N_{23}\dot{D}_{13}}{D_{13}^2} \\ \frac{\dot{N}_{14}D_{14} - N_{14}\dot{D}_{14}}{D_{14}^2} & \frac{\dot{N}_{24}D_{14} - N_{24}\dot{D}_{14}}{D_{14}^2} \end{bmatrix} \quad (3.5)$$

Per concludere andiamo a trovare:

$$\begin{bmatrix} \ddot{\theta}_3 \\ \ddot{\theta}_4 \end{bmatrix} = \dot{J}_{34} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + J_{34} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} \quad (3.6)$$

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:

$$\begin{aligned}
 J_1 &= \begin{bmatrix} -0.5l \sin \theta_1 & 0 \\ 0.5l \cos \theta_1 & 0 \end{bmatrix} \Rightarrow \dot{J}_1 = \begin{bmatrix} -0.5l \cos \theta_1 \dot{\theta}_1 & 0 \\ -0.5l \sin \theta_1 \dot{\theta}_1 & 0 \end{bmatrix} \\
 J_2 &= \begin{bmatrix} 0 & -0.5l \sin \theta_2 \\ 0 & 0.5l \cos \theta_2 \end{bmatrix} \Rightarrow \dot{J}_2 = \begin{bmatrix} 0 & -0.5l \cos \theta_2 \dot{\theta}_2 \\ 0 & -0.5l \sin \theta_2 \dot{\theta}_2 \end{bmatrix} \\
 J_3 &= \begin{bmatrix} -l \sin \theta_1 + 0.5 \sin \theta_3 \frac{N_{13}}{D_{13}} & -0.5l \sin \theta_3 \frac{N_{23}}{D_{13}} \\ l \cos \theta_1 + 0.5 \cos \theta_3 \frac{N_{13}}{D_{13}} & 0.5l \cos \theta_3 \frac{N_{13}}{D_{23}} \end{bmatrix}
 \end{aligned}$$

$$J_4 = \begin{bmatrix} -0.5l \sin \theta_4 \frac{N_{14}}{D_{14}} & -l \sin \theta_2 + 0.5 \sin \theta_4 \frac{N_{24}}{D_{14}} \\ 0.5l \cos \theta_4 \frac{N_{14}}{D_{14}} & l \cos \theta_2 + 0.5 \cos \theta_4 \frac{N_{24}}{D_{14}} \end{bmatrix}$$

In particolare, le matrici assumo il significato di jacobiano che consente il passaggio dalle coordinate dei giunti alle coordinate baricentriche dei quattro link.

$$\begin{bmatrix} \dot{x}_{g1} \\ \dot{y}_{g1} \end{bmatrix} = J_1 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}, \quad \begin{bmatrix} \ddot{x}_{g1} \\ \ddot{y}_{g1} \end{bmatrix} = J_1 \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \dot{J}_1 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x}_{g2} \\ \dot{y}_{g2} \end{bmatrix} = J_2 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}, \quad \begin{bmatrix} \ddot{x}_{g2} \\ \ddot{y}_{g2} \end{bmatrix} = J_2 \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \dot{J}_2 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x}_{g3} \\ \dot{y}_{g3} \end{bmatrix} = J_3 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}, \quad \begin{bmatrix} \ddot{x}_{g3} \\ \ddot{y}_{g3} \end{bmatrix} = J_3 \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \dot{J}_3 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x}_{g4} \\ \dot{y}_{g4} \end{bmatrix} = J_4 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}, \quad \begin{bmatrix} \ddot{x}_{g4} \\ \ddot{y}_{g4} \end{bmatrix} = J_4 \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \dot{J}_4 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

Importanti nel calcolo della dinamica sono anche le derivate delle matrici che abbiamo appena visto, ovvero \dot{J}_3, \dot{J}_4 , riportate in appendice A.2

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 corpi 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.

$$\sum_{i=1}^m F_j \delta q_j \quad (3.7)$$

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:

$$\begin{aligned} \begin{bmatrix} \delta\theta_1 & \delta\theta_2 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = & \begin{bmatrix} \delta\theta_1 & \delta\theta_2 \end{bmatrix} J_r \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} \delta x_{g1} & \delta y_{g1} \end{bmatrix} m \begin{bmatrix} \ddot{x}_{g1} \\ \ddot{y}_{g1} \end{bmatrix} + \\ \begin{bmatrix} \delta x_{g2} & \delta y_{g2} \end{bmatrix} m \begin{bmatrix} \ddot{x}_{g2} \\ \ddot{y}_{g2} \end{bmatrix} + & \begin{bmatrix} \delta\theta_3 & \delta\theta_4 \end{bmatrix} I_m \begin{bmatrix} \ddot{\theta}_3 \\ \ddot{\theta}_4 \end{bmatrix} + \begin{bmatrix} \delta x_{g2} & \delta y_{g2} \end{bmatrix} m \begin{bmatrix} \ddot{x}_{g3} \\ \ddot{y}_{g3} \end{bmatrix} + \quad (3.8) \\ & + \begin{bmatrix} \delta x_{g4} & \delta y_{g4} \end{bmatrix} m \begin{bmatrix} \ddot{x}_{g4} \\ \ddot{y}_{g4} \end{bmatrix} + \begin{bmatrix} \delta x & \delta y \end{bmatrix} m_e \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} \end{aligned}$$

Andando ora ad introdurre le matrici $J_1, J_2, J_3, J_4, J_{34}$ e J possiamo esprimere i termini al secondo membro come funzioni di $\delta\theta_1$ e $\delta\theta_2$:

$$\begin{aligned} \begin{bmatrix} \delta\theta_1 & \delta\theta_2 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = & \begin{bmatrix} \delta\theta_1 & \delta\theta_2 \end{bmatrix} \left[J_r \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + J_1^T \left(mJ_1 \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + mJ_1 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \right) + \right. \\ & J_2^T \left(mJ_2 \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + mJ_2 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \right) + J_{34}^T \left(I_m J_{34} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + I_m J_{34} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \right) \\ & J_3^T \left(mJ_3 \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + mJ_3 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \right) J_4^T \left(mJ_4 \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + mJ_4 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \right) \\ & \left. J^T \left(mJ \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + m_e J \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \right) \right] \end{aligned}$$

Semplificando il termine dello spostamento virtuale $\begin{bmatrix} \delta\theta_1 & \delta\theta_2 \end{bmatrix}$ e raccogliendo possiamo esprimere l'equazione come:

$$\tau = M\ddot{\theta} + C\dot{\theta} \quad (3.9)$$

Che possiamo vedere come:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = M \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + C \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \quad (3.10)$$

Dove:

$$M = J_r I_2 + m(J_1^T J_1 + J_2^T J_2 + J_3^T J_3 + J_4^T J_4) + J_r J_{34}^T J_{34} + m_v J^T J \quad (3.11)$$

$$C = m(J_1^T \dot{J}_1 + J_2^T \dot{J}_2 + J_3^T \dot{J}_3 + J_4^T \dot{J}_4) + J_r J_{34}^T \dot{J}_{34} + m_v J^T \dot{J} \quad (3.12)$$

Le equazioni viste fino ad ora vanno a considerare la dinamica ai giunti. Note le coppie ai giunti, l'inerzia del motore e il rapporto di trasmissione è possibile calcolare le coppie motrici come:

$$\begin{bmatrix} \tau_{m1} \\ \tau_{m2} \end{bmatrix} = \left(\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} + J_m i^2 \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} \right) \frac{1}{i} \quad (3.13)$$

Dove:

Nome	Descrizione	Valore
i	Rapporto di riduzione	64
J_m	Inerzia del motore	$1.7 \cdot 10^{-5}$

Tabella 2: Parametri dei motori

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 motorizzati.

Andiamo ora a definire l'equazione della dinamica diretta andando ad invertire l'equazione 3.9 in questo modo:

$$\ddot{\theta} = M^{-1}(-C\dot{\theta} + \tau) \quad (3.14)$$

La dinamica del sistema non è lineare, per risolvere l'equazione differenziale del secondo ordine è necessario procedere mediante integrazione numerica. Note le condizioni iniziali identificate dal vettore Θ_0 e definite come segue:

$$\Theta_0 = \begin{bmatrix} \theta_1(t_0) \\ \theta_2(t_0) \\ \dot{\theta}_1(t_0) \\ \dot{\theta}_2(t_0) \end{bmatrix}$$

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 ricalcolare le equazioni 3.11 e 3.12 riferite ad M e K nell'istante di tempo iniziale. Viene poi implementato un algoritmo che note posizioni e velocità in un istante calcola quelle dell'istante successivo.

$$\begin{bmatrix} \ddot{\theta}_1(n) \\ \ddot{\theta}_2(n) \end{bmatrix} = M^{-1}(n) \left(-C(n) \begin{bmatrix} \dot{\theta}_1(n) \\ \dot{\theta}_2(n) \end{bmatrix} + \tau(n) \right)$$

ricaviamo la velocità all'istante successivo come:

$$\begin{bmatrix} \dot{\theta}_1(n+1) \\ \dot{\theta}_2(n+1) \end{bmatrix} = \begin{bmatrix} \dot{\theta}_1(n) \\ \dot{\theta}_2(n) \end{bmatrix} + \begin{bmatrix} \ddot{\theta}_1(n) \\ \ddot{\theta}_2(n) \end{bmatrix} dt$$

e la posizione come:

$$\begin{bmatrix} \theta_1(n+1) \\ \theta_2(n+1) \end{bmatrix} = \begin{bmatrix} \theta_1(n) \\ \theta_2(n) \end{bmatrix} + \begin{bmatrix} \dot{\theta}_1(n) \\ \dot{\theta}_2(n) \end{bmatrix} dt$$

Facciamo ora il confronto fra posizione, velocità ed accelerazione dei *setpoint* rispetto

3.4 Dinamica diretta

a quelle ottenute tramite integrazione numerica:

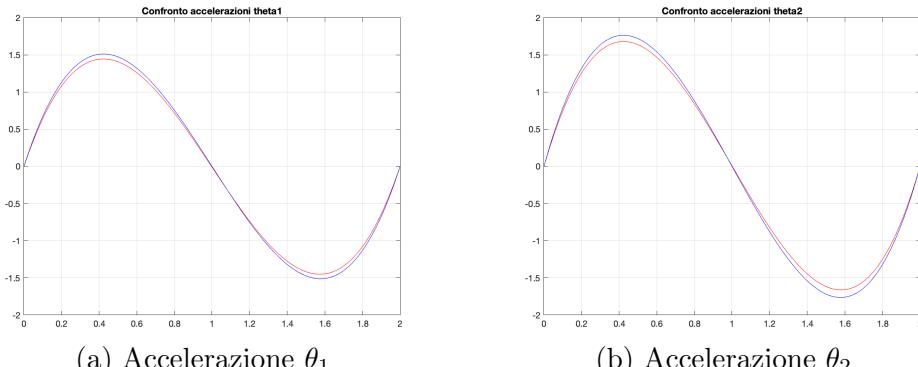


Figura 3.1: Confronto dinamica leggi di moto su accelerazioni

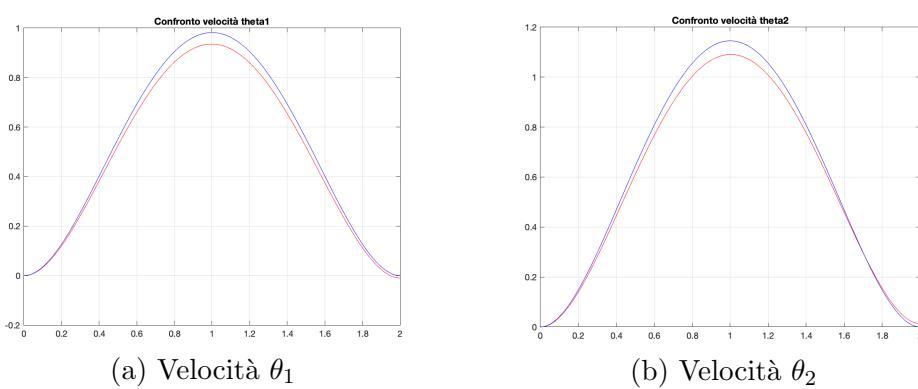


Figura 3.2: Confronto dinamica leggi di moto su velocità

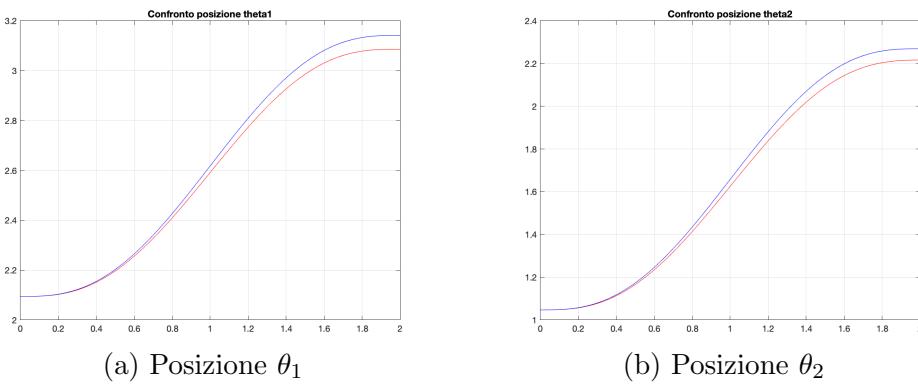


Figura 3.3: Confronto dinamica leggi di moto su posizioni

Si può notare un errore tra le due curve dovuto al metodo di integrazione utilizzato. Una volta calcolati tutti i parametri di nostro interesse andiamo a costruire il modello simulink che sarà utile per una visione d'insieme e per effettuare la validazione: Il modello implementa l'equazione della dinamica diretta 3.14, infatti riusciamo ad

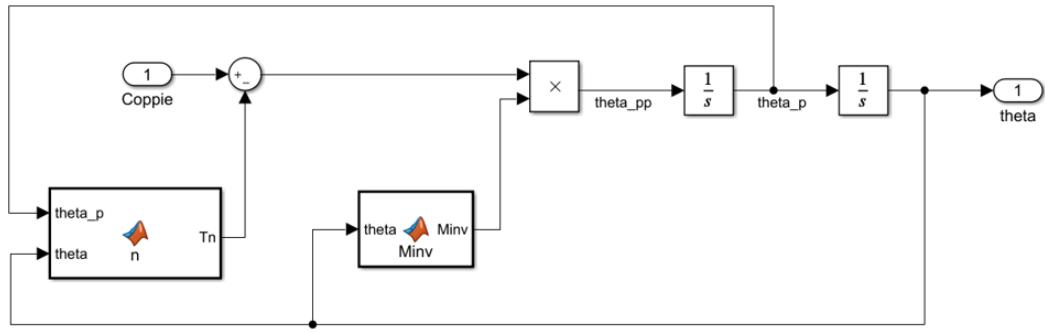


Figura 3.4: Modello simulink manipolatore

ottenere l'accelerazione a partire dalla moltiplicazione fra la matrice d'inerzia e la somma delle coppie con la matrice dei termini di Coriolis. Andando poi ad integrare otteniamo velocità e posizione che ci serviranno per trovare le accelerazioni negli step successivi.

Modellazione su Adams

Per quanto riguarda la modellazione su Adams, è stato realizzato un prototipo dei link del manipolatore composto da aste rigide, i due link motorizzati sono fissati a terra mediante delle cerniere. Una volta definiti i vincoli e le modalità di movimento

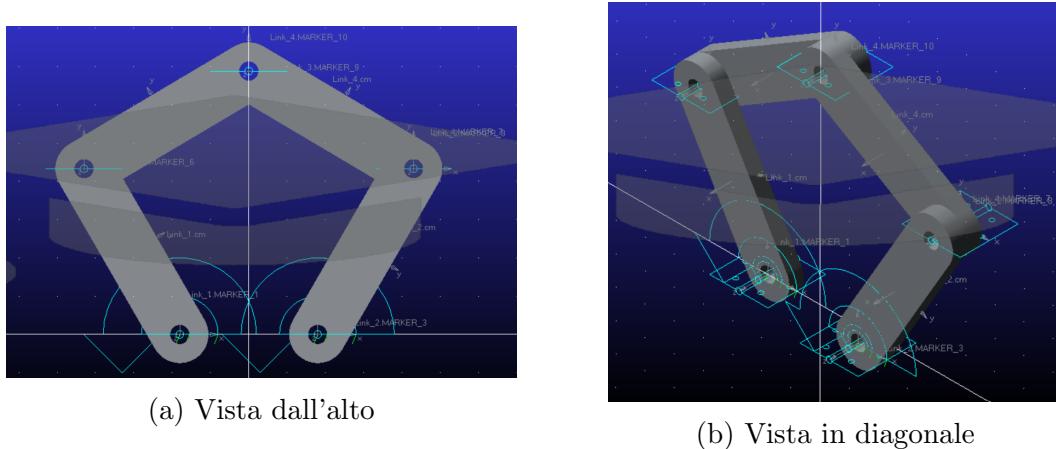


Figura 3.5: Modello Adams manipolatore 5R

del manipolatore si è passati alla fase successiva ovvero quella dell'analisi del modello

e della simulazione, andando ad assegnare le leggi di moto al modello Adams e verificando il suo comportamento rispetto al modello matlab.

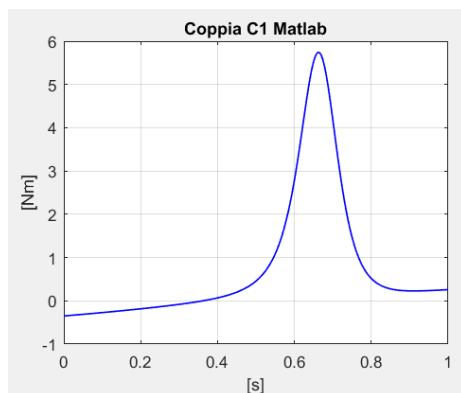
Validazione e confronto

Le leggi di moto utilizzate per la validazione sono:

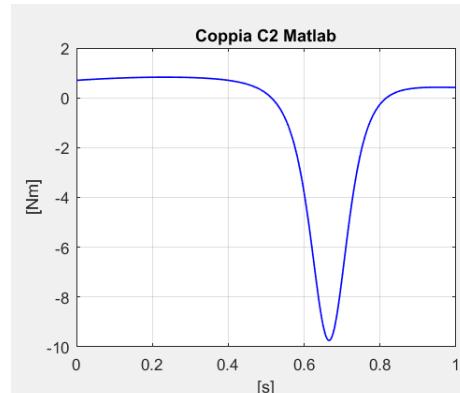
Motore	Legge di moto
Motore 1 (θ_1)	Sinusoidale \sin
Motore 2 (θ_2)	Sinusoidale $2 \sin$

Tabella 3: Leggi di moto validazione

Una volta assegnata la legge ed eseguita la simulazione si è fatto un confronto:

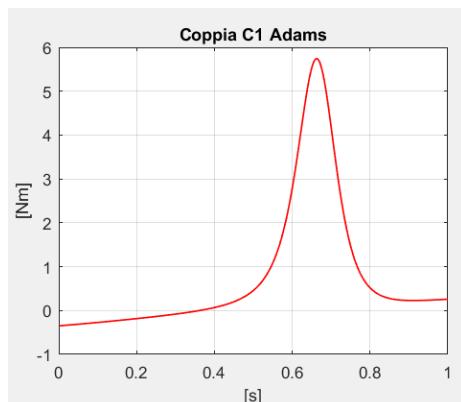


(a) Coppia C1 Matlab

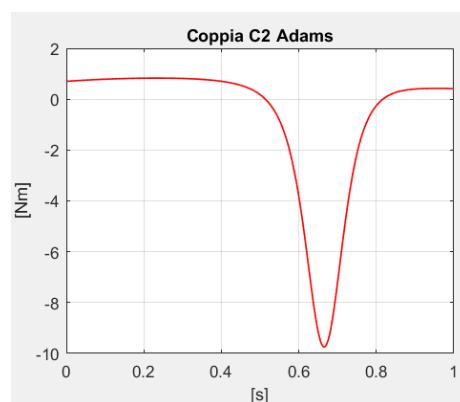


(b) Coppia C2 Matlab

Figura 3.6: Copie in uscita Matlab



(a) Coppia C1 Adams



(b) Coppia C2 Adams

Figura 3.7: Copie in uscita Adams

Andando poi a fare una differenza tra questi due grafici riusciamo a trovare l'andamento dell'errore sulle coppie:

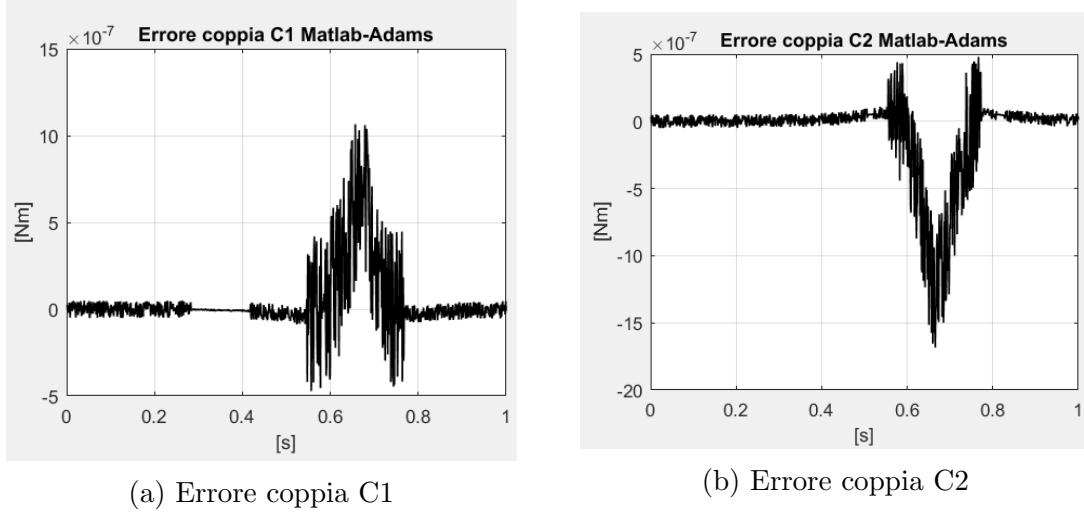


Figura 3.8: Errore Simulink-Adams

Analizzando il grafico vediamo che l'errore è nell'ordine di 10^{-7} di conseguenza è possibile vedere che la validazione ha portato un risultato positivo in quanto le coppie ottenute da matlab e quelle da Adams sono molto simili a parte un fattore d'errore dato dalle diverse modalità di calcolo due due software.

4 Analisi cinetostatica

Una volta effettuata la cinematica e dinamica del sistema in questo capitolo presentiamo l'analisi cinetostatica. In particolare andremo a concentrarci sui punti di singolarità, sul numero di condizionamento e per concludere analizzeremo lo spazio di lavoro del manipolatore.

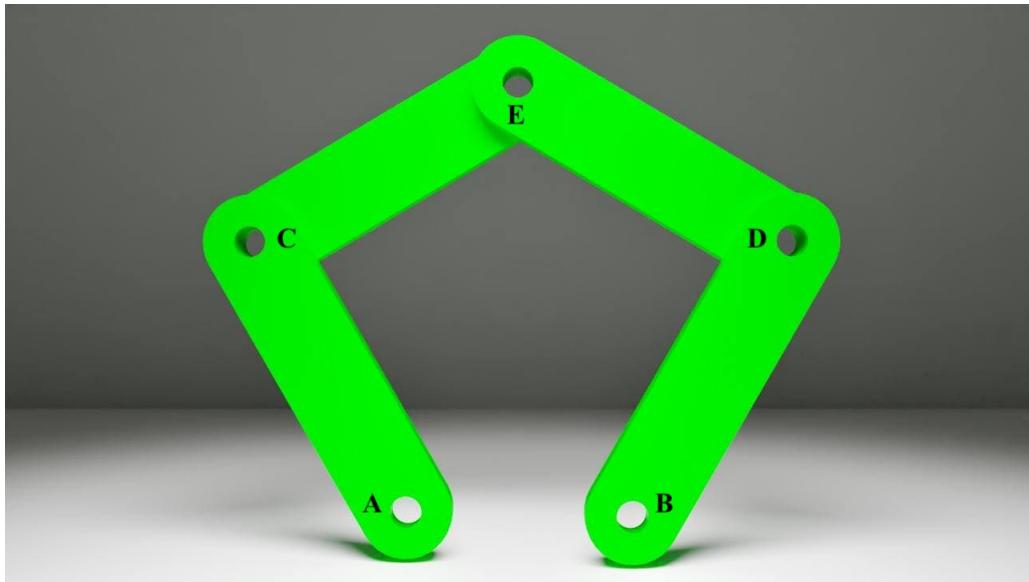


Figura 4.1: Posizione standard manipolatore

4.1 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 sono punti che vanno a delimitare lo spazio di lavoro del robot. Definiamo *workspace* del manipolatore una parte del piano $[x,y]$ nel quale il robot ha un funzionamento normale e non presenta problematiche³. Considerando l'immagine 1.1 si possono identificare cinque casi di singolarità. Di conseguenza il robot avrà come spazio di lavoro, tutto lo spazio che è interno (delimitato) da queste configurazioni.

³Nel caso in cui il manipolatore dovesse passare per un punto di singolarità potrebbero verificarsi problematiche sia nel seguire la traiettoria che a livello meccanico, arrivando nel peggior dei casi alla rottura.

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

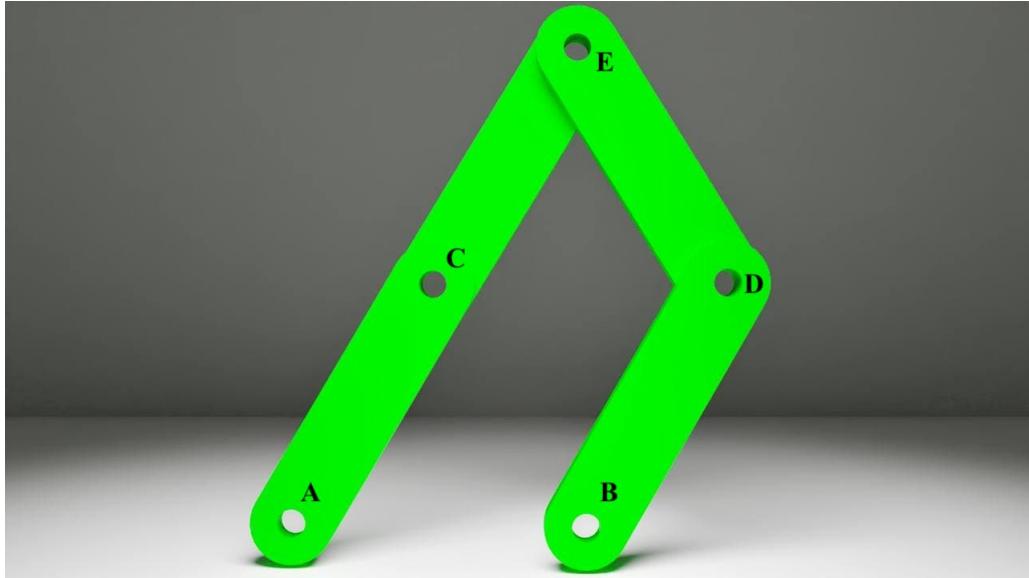


Figura 4.2: Caso 1 singolarità

Analiticamente: $||\overrightarrow{AC}||^2 = (l + l)^2$ quindi:

$$\left| \begin{bmatrix} -d \\ 0 \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix} \right|^2 = 4l^2$$

$$\left| \begin{bmatrix} -d - x \\ -y \end{bmatrix} \right|^2 = 4l^2$$

Svolgendo il modulo troviamo

$$d^2 + 2dx + x^2 + y^2 = 4l^2$$

Facendo variare x ricaviamo la y come:

$$y_1 = \sqrt{4l^2 - (x - d)^2} \quad (4.1)$$

Per quanto riguarda il secondo caso è simmetrico al primo, abbiamo la catena \overrightarrow{AB} parallela a \overrightarrow{BC} :

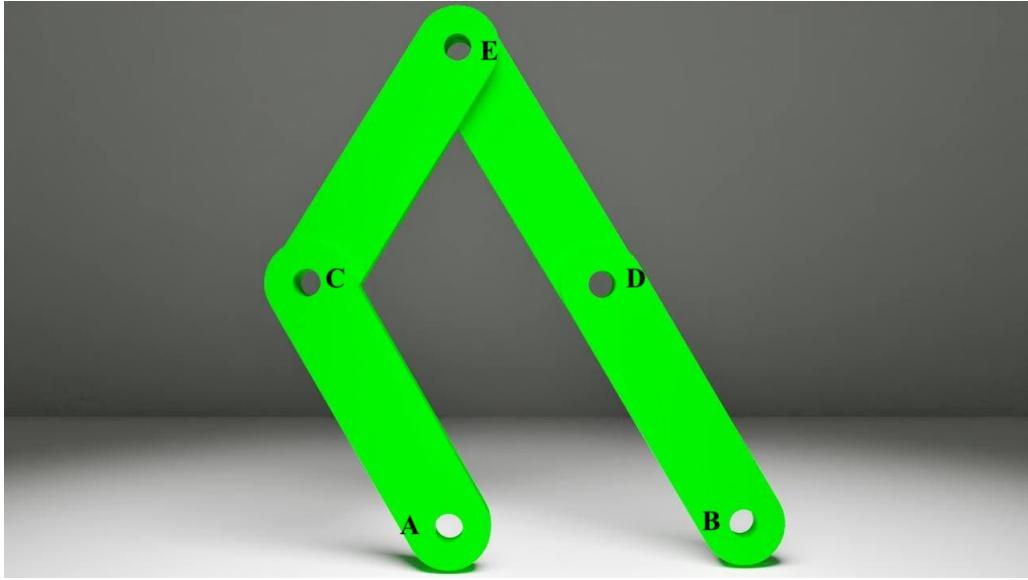


Figura 4.3: Caso 2 singolarità

Il procedimento è simile a prima, lasciando sempre la x libera possiamo trovare la y come:

$$y_2 = \sqrt{4l^2 - (x + d)^2} \quad (4.2)$$

Entrambi i casi producono come risultato una circonferenza.

Terzo caso

Il terzo caso di singolarità avviene quando i due link non motorizzati sono paralleli, abbiamo tre giunti allineati tra di loro rendendo la configurazione labile. Per poter uscire da questa singolarità abbiamo la necessità di una maggior coppia.

Per quanto riguarda la soluzione, andiamo inizialmente a trovare la posizione del punto D $[x_D, y_D]$: facciamo l'intersezione tra due circonferenze la prima ha centro in E con raggio l, la seconda ha centro in B con raggio 2l; possiamo esprimere le equazioni come:

$$\begin{cases} (x_D - d)^2 + y_D^2 = l \\ (x_D - l \cos \theta_1 + d)^2 + (y_D - l \sin \theta_1)^2 = 4l^2 \end{cases} \quad (4.3)$$



Figura 4.4: Caso 3 singolarità

Svolgendo i quadrati:

$$\begin{cases} x_D^2 - 2dx_D + d^2 + y_D^2 = l^2 \\ x_D^2 + l^2 + d^2 - 2ld \cos \theta_1 - 2x_Dl \cos \theta_1 + 2x_Dd + y_D^2 - 2y_Dl \sin \theta_1 = 4l^2 - l^2 \end{cases}$$

Sottraendo la prima equazione nella seconda otteniamo:

$$x_D(-l \cos \theta_1 + 2d) - y_Dl \sin \theta_1 - ld \cos \theta_1 = l^2$$

Otteniamo x in funzione di y come:

$$x_D = \frac{l^2 + y_Dl \sin \theta_1 + ld \cos \theta_1}{2d - l \cos \theta_1}$$

Sostituendo la x nella prima equazione possiamo ottenere y in funzione di θ_1 :

$$y_D = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (4.4)$$

dove:

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

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

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

Facendo variare θ_1 troviamo x_D e y_D . Per poter trovare la posizione dell'end-effector facciamo la media tra la posizione del punto D e del punto B.

$$x_3 = \frac{x_D - d + l \cos \theta_1}{2}, y_3 = \frac{y_B + l \sin \theta_1}{2} \quad (4.5)$$

Quarto e quinto caso

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

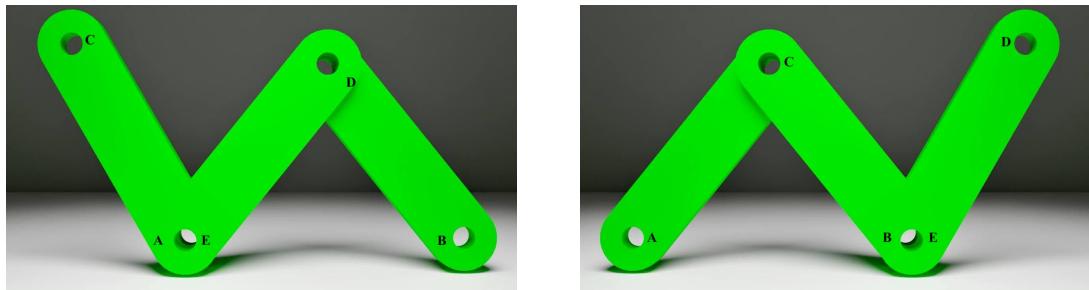


Figura 4.5: Caso 4 e 5 singolarità

Possiamo trovare le soluzione del quarto caso imponendo

$$|\overrightarrow{EC}| = 0 \rightarrow x = d, y = 0 \quad (4.6)$$

e per il quinto caso

$$|\overrightarrow{AC}| = 0 \rightarrow x = -d, y = 0 \quad (4.7)$$

Per delimitare i confini di operabilità del manipolatore sono stati uniti tutti e cinque i casi di singolarità. Nella figura seguente è fornita una rappresentazione sul piano cartesiano dell'unione di questi luoghi di singolarità, facciamo notare che non tutti

i punti sono fisicamente raggiungibili dal robot per come è stato progettato; in particolare la posizione y dell'*end-effector* non potrà mai essere negativa.

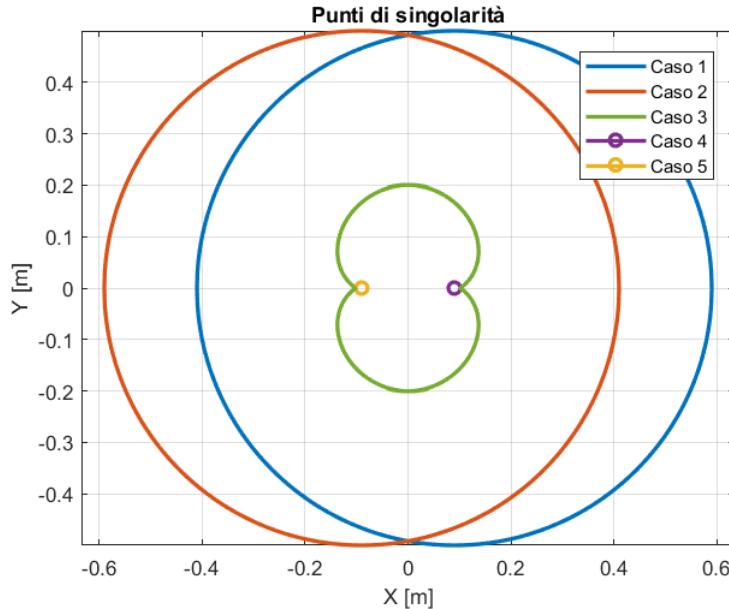


Figura 4.6: Punti di singolarità

4.2 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 2.5, vista nella sezione 2.1.2.

Andiamo a definire la matrice

$$J_{man} = J J^T \quad (4.8)$$

Da questa possiamo ricavare gli autovalori λ_1, λ_2 . Definiamo l'indice di manipolabilità r come:

$$r = \frac{\max(\lambda_1, \lambda_2)}{\min(\lambda_1, \lambda_2)} \quad (4.9)$$

Questo numero può variare tra 1 e $+\infty$, più è piccolo e meno si rischia di andare in singolarità. Lo spazio analizzato ha come estremi:

$$-0.3 \leq x \leq 0.3$$

$$0.1 \leq y \leq 0.5$$

Rappresentiamo il numero di condizionamento tramite i seguenti grafici:

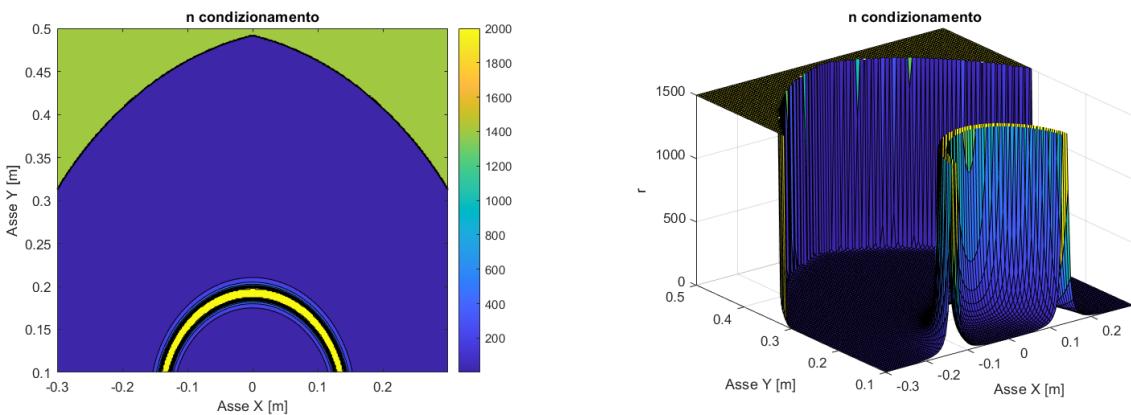


Figura 4.7: Numero di condizionamento

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 possiamo vedere 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.

4.3 Workspace

Dalle analisi appena effettuate sui punti di singolarità e sul numero di condizionamento è stato possibile descrivere i limiti di lavoro del manipolatore. In prima battuta ci si è concentrato sull'analizzare gli angoli di movimentazione dei giunti e i loro vincoli, in particolare il link sinistro con angolo θ_1 può effettuare un movimento da 60° fino a 210° , mentre il link destro da -30° e 150° . Questa limitata mobilità è data dai vincoli che ci sono tra le due braccia, infatti, se a livello teorico si possono

4.3 Workspace

ottenere configurazioni particolari come quella vista in figura 4.5 nella pratica non è possibile muovere né manualmente né automaticamente i giunti per ottenere quei casi, l'unico modo possibile è quello di smontare e rimontare il manipolatore al contrario.

Possiamo andare a rappresentare gli angoli di movimentazione mediante la seguente immagine:

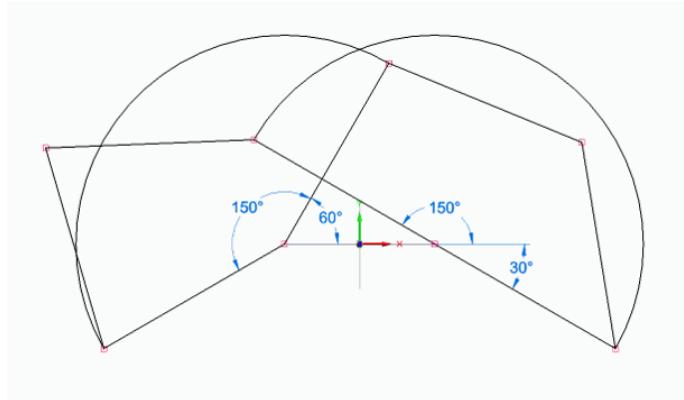


Figura 4.8: Angoli di movimentazione dei giunti motorizzati

A partire dai limiti di lavoro è stato identificato uno spazio buono dove il manipolatore può operare non raggiungendo i punti singolari. Questo luogo è stato descritto mediante un rettangolo:

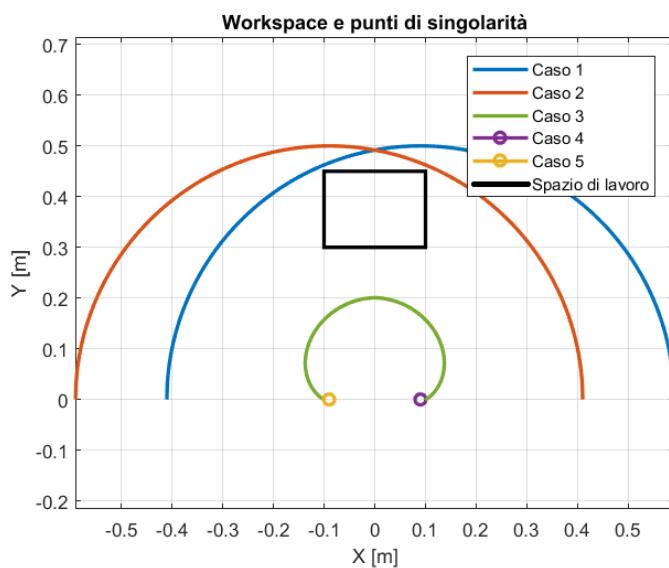


Figura 4.9: *Workspace 5R*

Con estremi:

$$-0.1 \leq x \leq 0.1$$

$$0.3 \leq y \leq 0.45$$

Successivamente, per verificare che lo spazio di lavoro fosse corretto e rispettasse tutte le condizioni è stata fatta un'analisi sul numero di condizionamento del *workspace*:

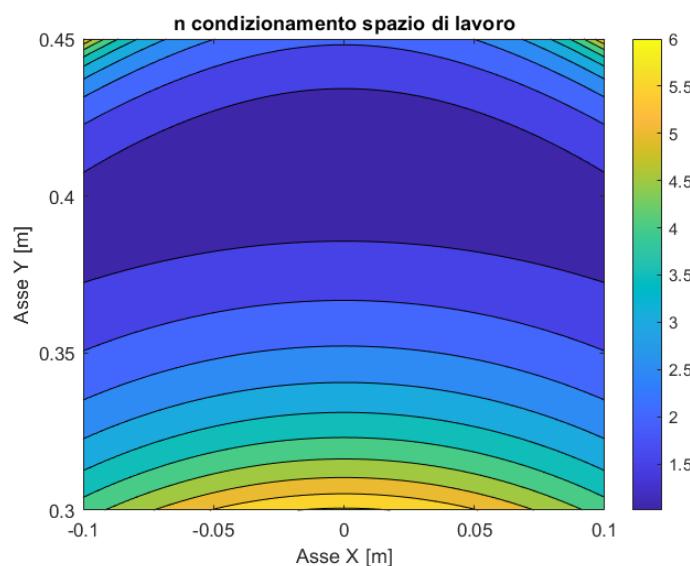


Figura 4.10: Numero di condizionamento workspace

Su tutto il piano x,y si nota come il numero di condizionamento assume valori 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 punti di singolarità.

5 Modellazione end-effector

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

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

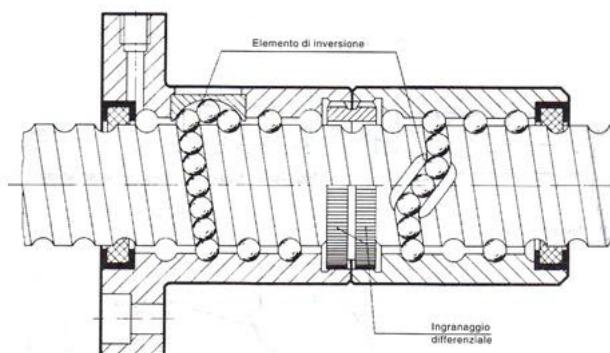


Figura 5.1: Struttura vite

La prima è una vite a ricircolo di sfere che permette un movimento rotatorio, 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 un foglio.



Figura 5.2: 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 4: Parametri end-effector

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

5.1 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, con velocità ed accelerazione che 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 cinematicaabbiamo bisogno della legge di moto e della jacobiana della vite, che possiamo definire come:

$$J_e = \begin{bmatrix} \frac{p_v}{2\pi} & \frac{p_v}{2\pi} \\ 0 & 1 \end{bmatrix} \quad (5.1)$$

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

$$V = J_e \begin{bmatrix} z_{ee} \\ \varphi_v \end{bmatrix}, \dot{V} = J_e \begin{bmatrix} \dot{z}_{ee} \\ \dot{\varphi}_v \end{bmatrix}, \ddot{V} = J_e \begin{bmatrix} \ddot{z}_{ee} \\ \ddot{\varphi}_v \end{bmatrix} \quad (5.2)$$

5.2 Dinamica della vite

Per la definizione della dinamica della vite è stato usato, come per la dinamica dei link il principio dei lavori virtuali. Per questa soluzione abbiamo bisogno delle accelerazioni dei due elementi quindi \ddot{Z}_{ee} e $\ddot{\varphi}_v$. Introduciamo la matrice di massa della vite, composta dalla massa e dall'inerzia, definita come:

$$M_v = \begin{bmatrix} m_e & 0 \\ 0 & I_v \end{bmatrix} \quad (5.3)$$

La soluzione della dinamica la troviamo quindi come:

$$C_{ee} = J_e^T M_v J_e \begin{bmatrix} \ddot{z}_{ee} \\ \ddot{\varphi}_v \end{bmatrix} \quad (5.4)$$

6 Tecnologie implementate

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

6.1 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 e l'esecuzione del modello in tempo reale ovvero il dispositivo target. Grazie a questo il software permette raggiungere tempi di campionamento molto più veloci per uno stesso modello (vi è la possibilità di raggiungere frequenze fino a 20 kHz) rispetto ad *kernel real-time* che condivide le risorse hardware con il computer. Un'altra caratteristica fondamentale è il fatto che su SLRT è possibile far funzionare il dispositivo target in modalità *stand-alone*, si può riuscire a far avviare il dispositivo senza il bisogno di una vera e propria procedura d'installazione.

6.2 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 ed introdotta nell'aprile 2003 da Beckhoff automation. Il protocollo è stato pubblicato nello standard IEC61158, e regolato dalla società ETG (*EtherCAT Technology Group*), soddisfa i requisiti *hard* e *soft* real time specifici per l'ambito dell'automazione. Una caratteristica molto nota riguarda la velocità dei tempi di ciclo, la durata media di un periodo di ciclo è inferiore a $100\mu s$.

6.2.1 Proprietà

Il principio base di funzionamento di EtherCAT si basa sul concetto di *Master/Slave*. Il master 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; inoltre, utilizza un Media Access Controller (MAC) standard, senza alcun processore dedicato alla comunicazione. Questo consente di implementare un dispositivo master su una qualunque piattaforma hardware dotata di una porta di rete, indipendentemente dal sistema operativo o dal software applicativo utilizzato. I dispositivi EtherCAT slave integrano un 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 predibili e indipendenti dalla particolare implementazione dei dispositivi slave.

6.2.2 Gestione della rete

Caratteristica fondamentale della rete è la sicurezza, 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 sezione considerata non rilevante ai fini della sicurezza; viene utilizzato un solo canale per trasferire sia i dati standard che quelli di sicurezza, in particolare i frame di sicurezza vengono identificati come *safety container*, e contengono i dati critici del processo e l'informazione necessaria per garantirne l'integrità. Il *safety container* viene mappato dentro i dati di processo che sono ciclici durante la comunicazione; possono viaggiare tramite cavi di rame, fibre ottiche e connessioni wireless. La variabilità del mezzo trasmissivo introduce flessibilità, e rende più semplice e sicura la connessione tra 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 mediante tecnologie diverse.

6.2.3 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 schede 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 arriva l’immagine già ordinata correttamente, la CPU quindi non è più responsabile dell’ordinamento. I dispositivi *slave* invece utilizzano l’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.

6.3 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 vanno 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 oppure 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 avviene tramite cavo ethernet

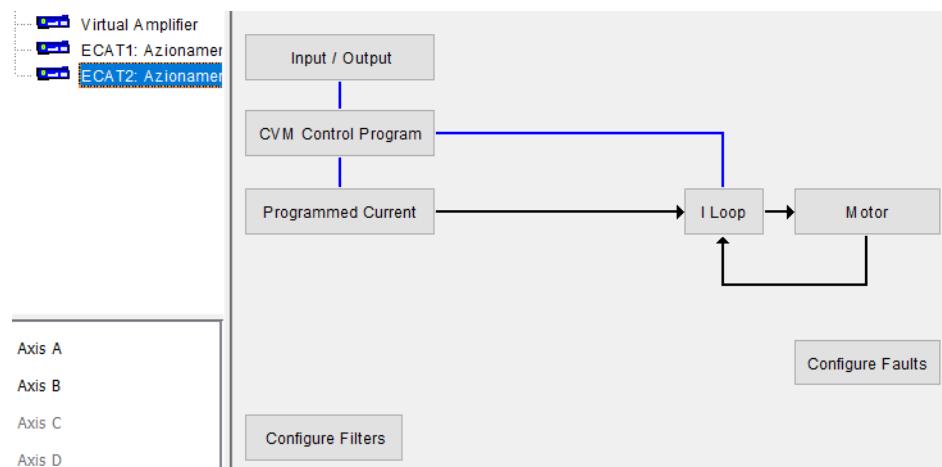


Figura 6.1: Schema azionamento

come si può notare dalle foto gli azionamenti sono controllati in coppia, in particolare viene chiuso l'anello in corrente grazie ai parametri del motore. Un'operazione fondamentale è stata l'analisi dei registri alla ricerca di quelli dei finecorsa⁴ Per le braccia, i registri da guardare sono l'ottavo e il quindicesimo, in particolare per vincoli di progetto i finecorsa osservati sono stati quelli al lato destro, quindi quando le braccia sono entrambe a destra; per la vite invece il registro osservato è stato il quindicesimo, riferito a quando la vite arriva a toccare il lato superiore.

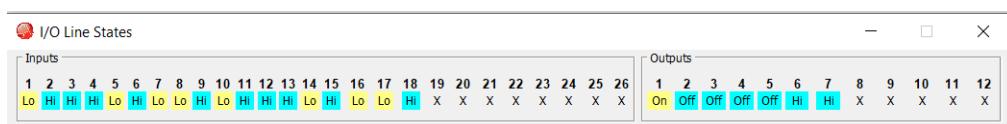


Figura 6.2: Registri azionamento

⁴Il registro del finecorsa passa da *LOW* a *HIGH* dopo che il manipolatore lo raggiunge.

6.4 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 la configurazione è stata fatta *offline*, e come metodo di comunicazione si è scelto di usare le PDO⁵. L'ultimo passo della configurazione è l'esportazione di un file ENI, un file XML che descrive la topologia della rete specificando il comando di inizializzazione per ogni dispositivo ed i comandi che devono essere inviati ciclicamente.

⁵Process Data Object, ovvero dati trasmessi dal/al MotionController, in tempo reale a tutti i nodi ogni ciclo del tempo di campionamento.

7 Sistema reale

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

7.1 Struttura del robot

Il manipolatore PKM è un manipolatore a cinematica parallela, composto da due braccia e un end-effector. Alle braccia sono collegati due motori, uno per il link motorizzato sinistro e l'altro per quello 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 a quelli della vite e un modulo beckhoff che si occupa della gestione degli input/output digitali.

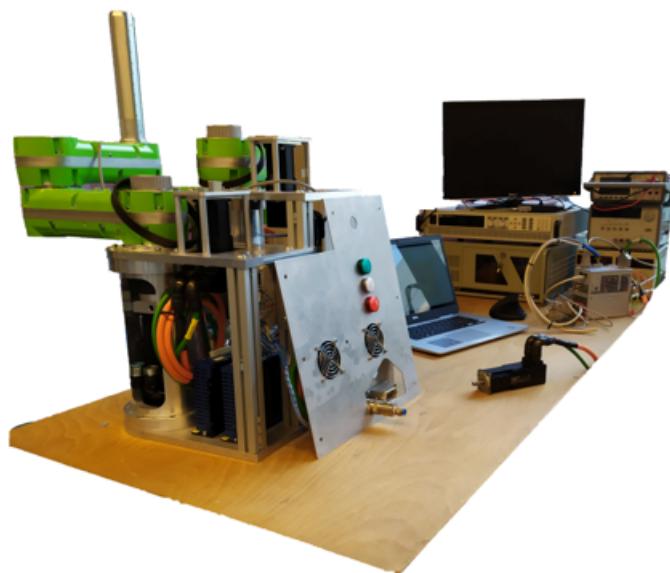


Figura 7.1: Banco di test

Per poter utilizzare il sistema sono necessarie due fonti di alimentazione, una che serve ad alimentare la logica con tensione pari a 24 Volt, e l'altra che alimenta i motori con tensione pari a 80 Volt.

7.1.1 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 CME2 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.

7.1.2 Beckhoff EK1814

Il beckhoff EK1814 è un accoppiatore EtherCAT che fa da *link* tra il protocollo EtherCAT a livello di bus di campo e i terminali EtherCAT. Inoltre, su questo modello sono 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 slave, a questo sono stati connessi gli azionamenti (come slave in cascata), e il PC Target (come master). Gli input e output digitali sono stati usati per controllare la pressione del fungo di emergenza e le luci di segnalazione per le fasi del manipolatore.

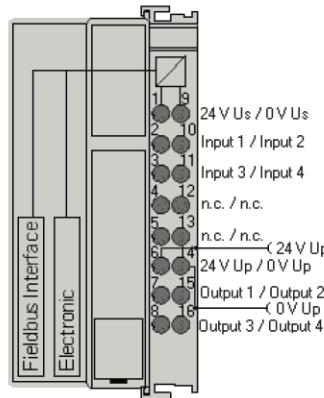


Figura 7.2: Schema modulo bechoff

7.1.3 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.

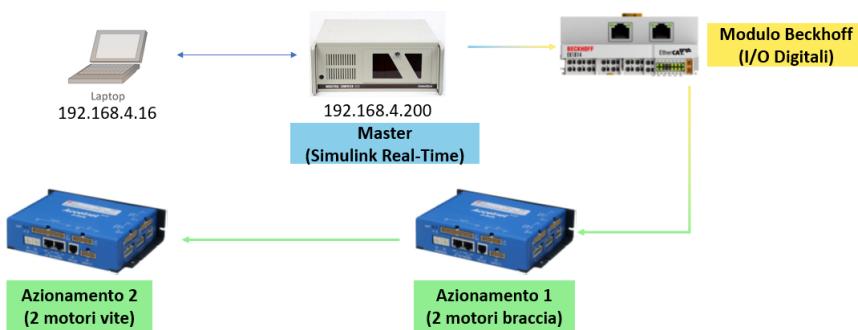


Figura 7.3: Topologia della rete

7.1 Struttura del robot

Alla seconda porta ethernet, vi è collegato il PC dell'utente, il quale provvede a generare, compilare, e caricare i programmi sul PC target. Da User-PC è anche possibile vedere i grafici e fare delle analisi sulle movimentazioni 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:

```
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 7.4: Configurazione rete ethernet user PC

Va precisato che il pc dell'utente non fa parte della rete ethercat, ma la rete inizia soltanto dal pc target in poi, infatti, ad esclusione delle operazioni viste precedentemente il manipolatore non ha bisogno del pc utente per funzionare.

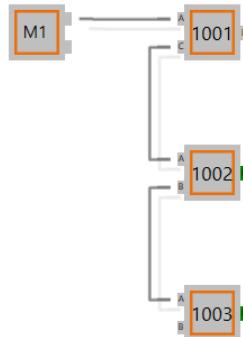


Figura 7.5: Topologia rete mediante Ec-engineer

Una volta configurata la 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 prime sono trasmesse al master dagli azionamenti, di conseguenza il master le riceve, quelle in 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

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

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

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

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 7.6: PDO in input

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

Receive PDO 1		
Name	Index	Bit Length
Axis A Modes of operation	0x6060:00	8
Axis A Target Torque	0x6071:00	16

Receive PDO 2		
Name	Index	Bit Length
Axis B Modes of operation	0x6860:00	8
Axis B Target Torque	0x6871:00	16

Figura 7.7: PDO Output 1 e 2

- *Modes of operation*, è un registro che specifica la modalità con la quale verrà controllato l'azionamento, per esempio coppia, posizione, velocità o ciclica

7.2 Implementazione nel sistema reale

- *Target torque*, specifica il valore di coppia che l'azionamento dovrà fornire al motore

7.2 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

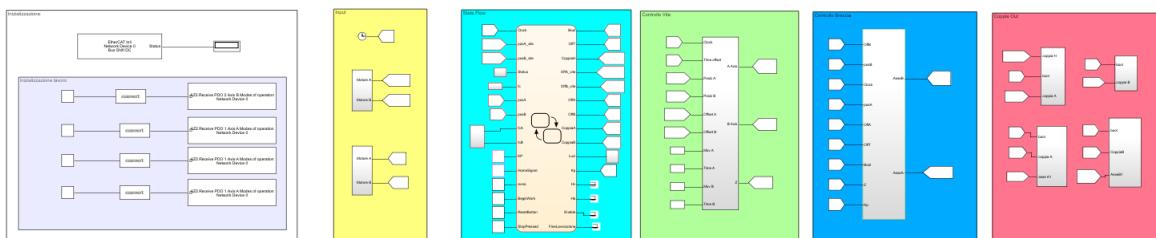


Figura 7.8: Schema generale simulink

Il 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 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 (mod. 4).

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 5: Tipologie controllo azionamenti

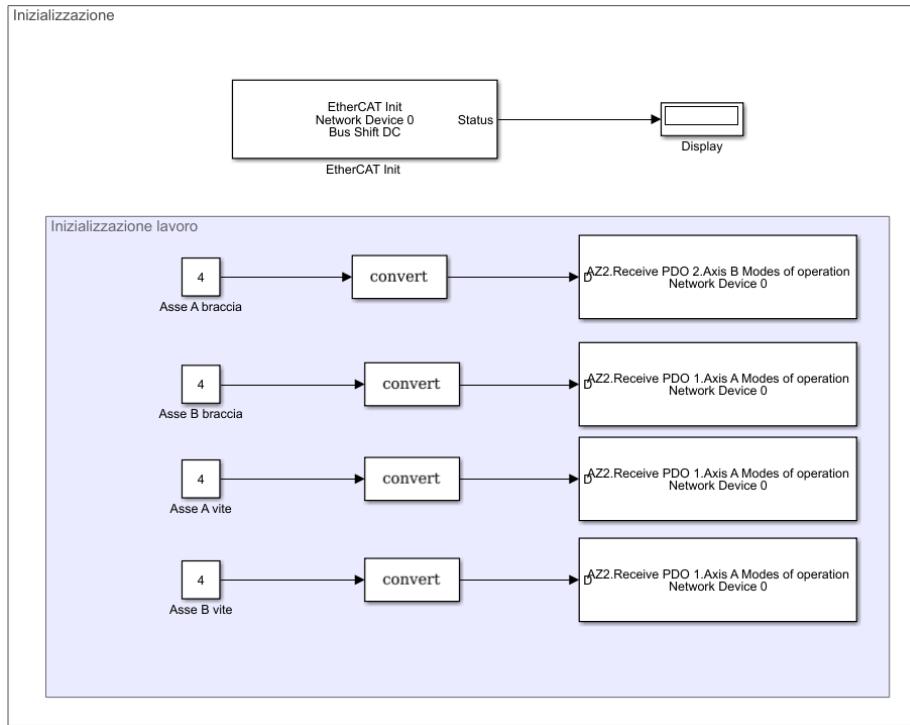


Figura 7.9: Fase 1: Inizializzazione

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 lo scambio di PDO ed hanno bisogno di essere convertiti, proprio per questo la struttura di ricezione di un messaggio è la seguente:

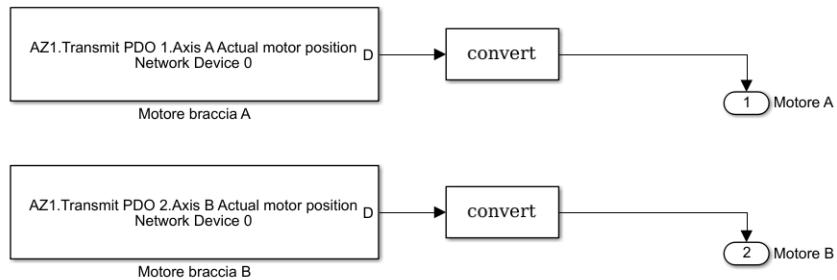


Figura 7.10: Conversione e lettura motori

Vediamo invece nella figura successiva la struttura del blocco di ricezione degli input,

tutti i valori dopo una conversione entrano in blocchi *goto* e verranno ripescati negli stati successivi da blocchi *from*.

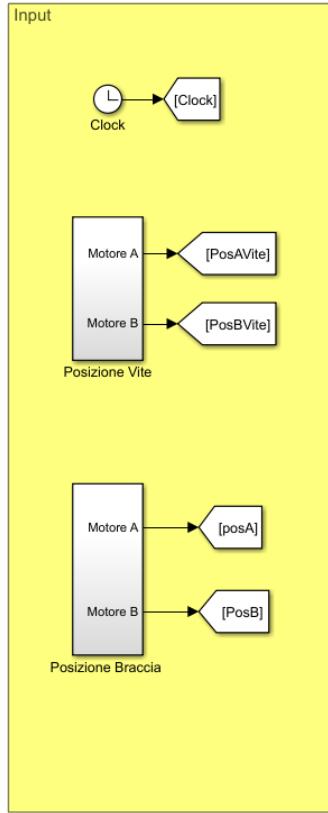


Figura 7.11: Fase 2: Input

Stateflow

Il blocco stateflow è un blocco posto la logica fondamentale dell'applicazione, in ingresso abbiamo dati che servono per il controllo delle traiettorie e degli stati del manipolatore, in particolare il clock, lo stato dei finecorsa, le posizioni dei motori e delle variabili di input che serviranno per gestire interamente l'interfaccia grafica. In uscita invece abbiamo gli offset, utili per avere un riferimento spaziale di posizione e temporale, in quanto ci aiutano a capire di quanto ci siamo mossi e in quanto tempo, le coppie di *homing* dei motori, un blocco per la gestione delle luci e variabili di output che serviranno per l'accensione e lo spegnimento dei led nell'interfaccia grafica.

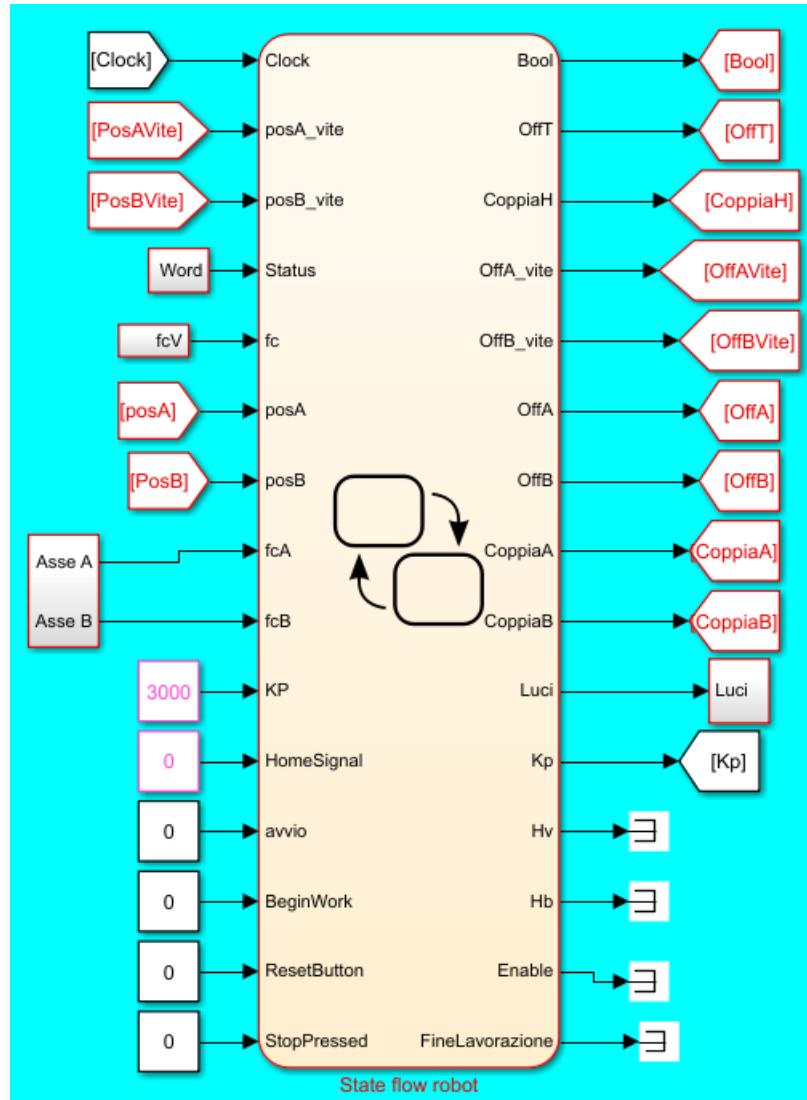


Figura 7.12: Fase 3: Stateflow

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 con il tempo di esecuzione. Grazie a tutti questi parametri possiamo definire le leggi di moto che ci permettono di eseguire il movimento nell'asse Z. Importante è sapere che il movimento eseguito dalla vite a ricircolo di sfere è dipendente anche da quello della 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

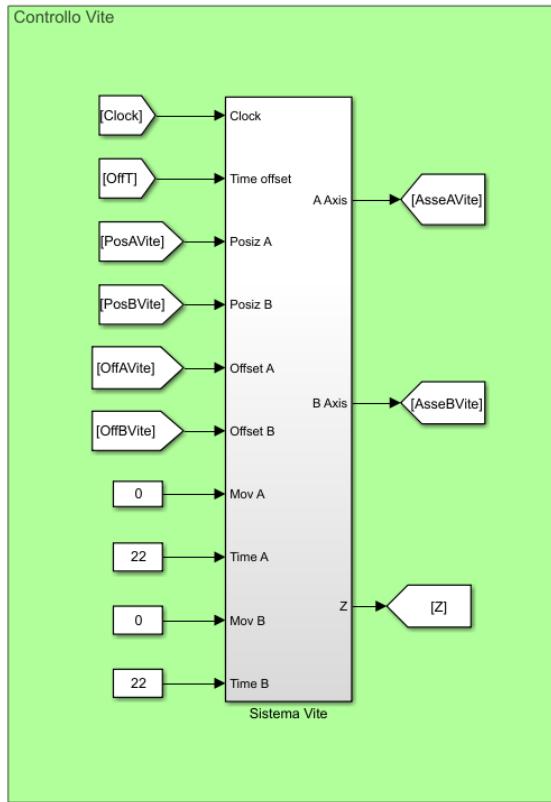


Figura 7.13: Fase 4: Schema di controllo della vite

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 ed internamente vengono svolte le operazioni di selezione della traiettoria, generazione della legge di moto con conseguente creazione degli angoli di riferimento (grazie alle relazioni di cinematica e dinamica) ed implementazione dello schema di controllo. In uscita abbiamo le coppie da assegnare ai due motori delle braccia. La scelta di dividere il controllo delle braccia e quello della vite in due blocchi diversi è stata fatta in quanto i due sono indipendenti l'uno dall'altro. Inizialmente ci si è concentrati sul controllo della vite, per poi passare a quello delle braccia ed infine dopo vari test i due controlli sono stati implementati contemporaneamente.

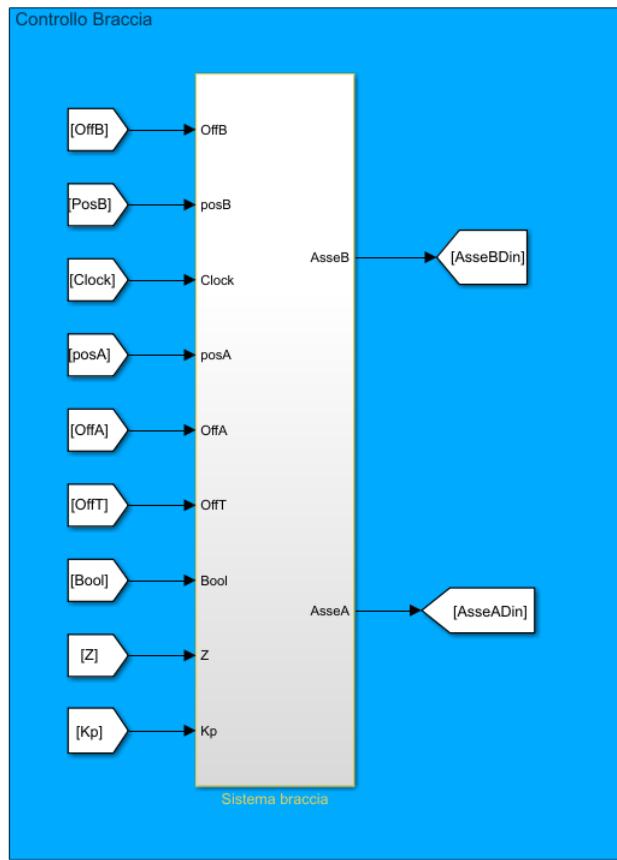


Figura 7.14: Fase 5: Schema di controllo delle braccia

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

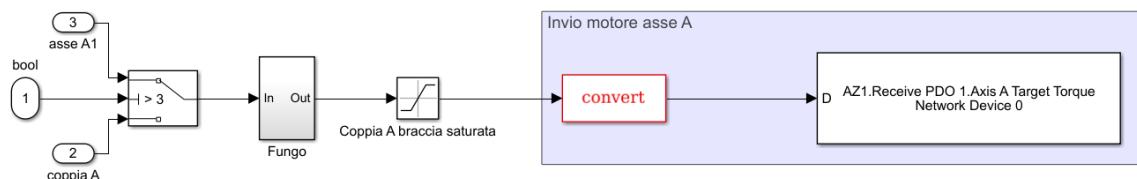


Figura 7.15: Schema espanso invio coppia

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

7.2 Implementazione nel sistema reale

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.

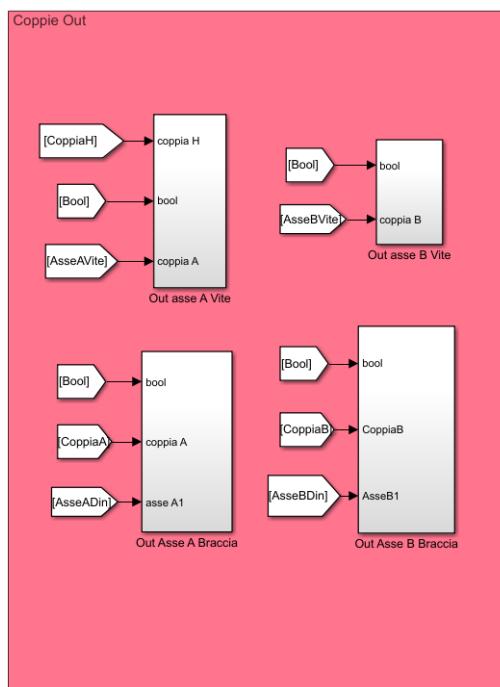


Figura 7.16: Fase 6: Copie in uscita

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.

Andiamo ora a concentrarci sulle parti principali di questo programma, in particolare andremo a trattare lo stateflow il metodo di funzionamento ed i vari stati, passeremo poi all'interfaccia grafica che permette di comandare il manipolatore, e per concludere andremo a vedere i controllori implementati per la vite e per le braccia, andando a vedere la struttura, lo schema e i risultati ottenuti per ogni approccio.

7.3 Stateflow

Stateflow si occupa di fornire diagrammi di transizione, di stato e di flusso utilizzando un linguaggio grafico. Nel caso del manipolatore è stato utilizzato per la progettazione dei 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, per comodità facciamo riferimento allo schema seguente:

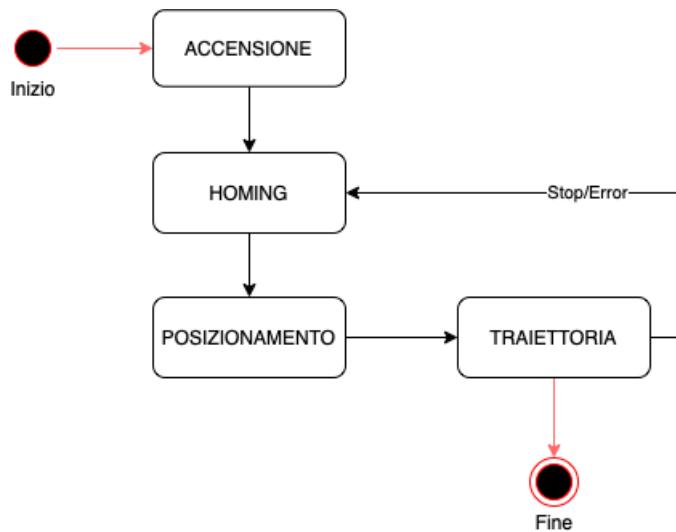


Figura 7.17: Macchina a stati stateflow

7.3.1 Fase di Homing

Appena il robot viene acceso non vi è alcuna conoscenza relativa alla sua posizione, di conseguenza abbiamo bisogno di uno stadio che ci porti in una posizione di riferimento nella quale sappiamo dov'è collocato effettivamente. Il primo stadio è quindi quello di *homing*, consiste nel portare i motori a toccare i finecorsa indicando quello come 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 finercorsa, dopo che erano stati toccati si passava nello stato successivo. Però, per motivi di sicurezza e, considerando che lasciando fermo il manipolatore per diverso tempo la stessa coppia costante potrebbe non essere in grado di farlo muovere si è deciso di chiudere l'anello in posizione,

in particolare per eseguire la fase di *homing* è stata data una rampa con pendenza negativa⁶. Per controllare la rampa è stato fatto un controllo proporzionale e integrale sull'errore tra la posizione attuale ed il riferimento, la legge di controllo implementata è del tipo:

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

Nella figura 7.18 possiamo vedere lo schema implementato per questa prima fase:

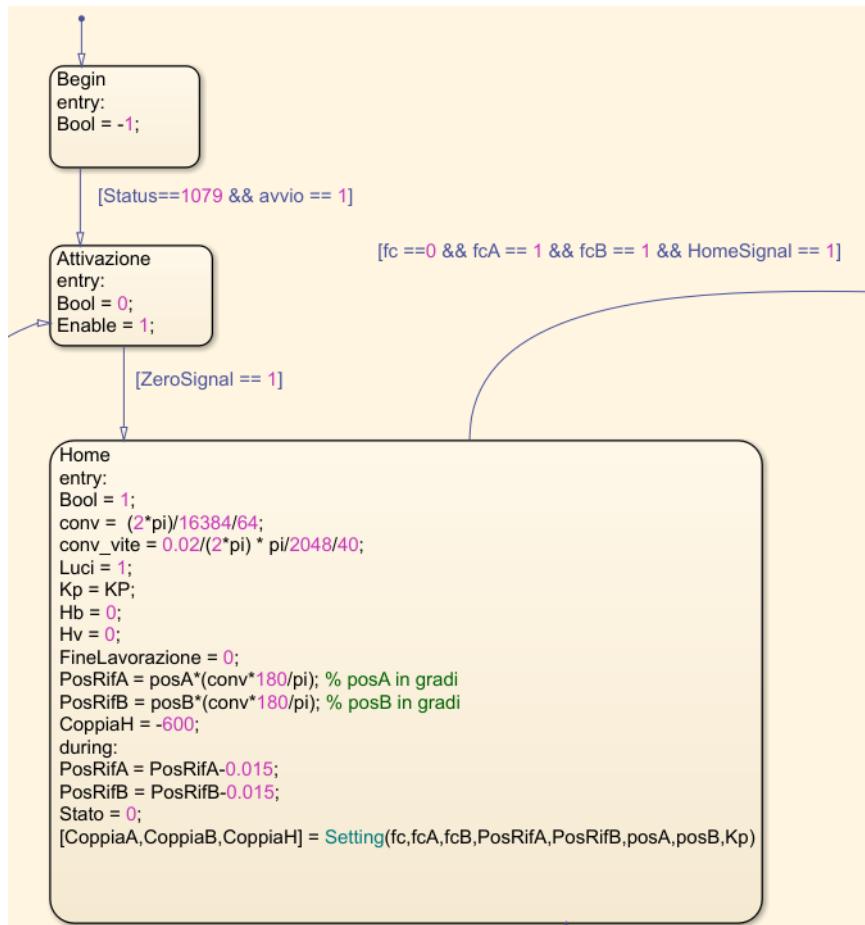


Figura 7.18: Fase di Homing

Il primo stato è quello di **Begin**, appena il programma si avvia ci si entra 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

⁶i finecorsa utilizzati per questa fase vengono rilevati quando i link sono totalmente a destra, quindi la direzione negativa per i motori delle braccia.

all'uso, la seconda invece 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. Proseguiamo nello stato successivo quando il segnale *ZeroSignal* è vero, questo viene gestito mediante un bottone da interfaccia grafica, appena premuto entriamo automaticamente nello stato **Home**.

In questo stato abbiamo la definizione di variabili per la conversione dei valori presi dai motori⁷, viene poi salvata la posizione di riferimento degli assi A e B dei motori delle braccia. La fase successiva è quella del *during*⁸, in questa abbiamo la rampa che decresce di 0.015 gradi al millisecondo, quindi 15 gradi al secondo e abbiamo una funzione simulink che si occupa del controllo PI. Una volta raggiunta la posizione del finecorsa le coppie vengono settate a 0, impedendo quindi un'ulteriore movimentazione, oltre al movimento delle braccia c'è anche quello della vite, che raggiunge la posizione di Homing salendo con una coppia negativa. Per passare alla fase successiva abbiamo bisogno che tutti gli elementi siano arrivati a finecorsa.

7.3.2 Fase di posizionamento

La fase successiva è quella di posizionamento, per non lasciare il robot nella configurazione di homing si è scelto di spostarlo in una configurazione standard lontana dai punti di singolarità che sarà comoda per le movimentazioni successive. La configurazione scelta prevede che i giunti siano messi a 100° e 80°, anche in questo caso, come prima il primo approccio è stato quello di utilizzare una coppia costante per il movimento; la fase di homing lasciava i link a 60° e -30°, vi era quindi la necessità di fare +40° per il braccio sinistro e +110° per quello destro, la coppia costante veniva erogata finché la condizione non era vera, dopodiché eravamo sicuri che il posizionamento fosse stato effettuato in maniera corretta.

L'approccio di posizionamento finale però non è stato quello della coppia costante, anche in questo caso per motivi di sicurezza ma, sapendo le posizioni finali che si vogliono raggiungere, si è optato per definire una legge di moto che si occupa di portarci nella condizione desiderata.

⁷I valori sono espressi tutti in *counts*, è stata necessaria una fase di analisi dei motori per capire come convertirli, effettuando quindi un passaggio da *counts* a radianti e da radianti a gradi.

⁸Finché rimaniamo in quello stato, le operazioni vengono eseguite ad ogni ciclo (1ms).

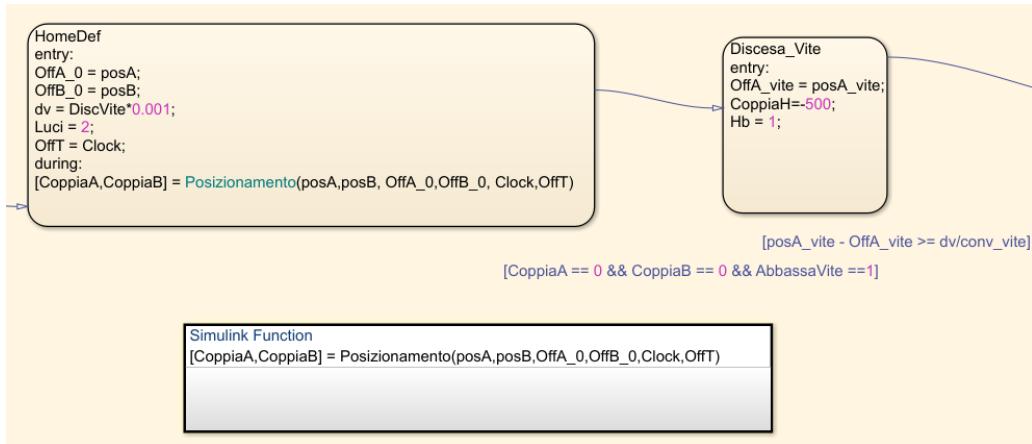


Figura 7.19: Fase di posizionamento

A partire dallo schema precedente, per passare allo stato **HomeDef** bisogna aver raggiunto tutti i finecorsa; non appena entriamo nello stato vengono salvati gli offset della posizione delle braccia, del tempo e la posizione desiderata di discesa della vite. Ad ogni ciclo, quindi ogni millisecondo viene eseguita la funzione **Posizionamento(…)** la quale, mediante una legge di moto, ci consente di eseguire in maniera corretta il posizionamento. Analizzando nello specifico il blocco simulink del posizionamento, vengono costruite due leggi polinomiali, una per ogni motore che producono un *setpoint* in posizione in gradi, che verrà confrontato con la posizione attuale. Per ottenere la coppia, e garantire errore a transitorio esaurito viene introdotto un sistema di controllo proporzionale integrativo, con legge di controllo:

$$\tau = K\tilde{\theta} = K_p\tilde{\theta} + K_i\tilde{\theta}$$

Le coppie di controllo uscenti andranno al manipolatore e si occuperanno della movimentazione. Possiamo vedere lo schema della legge di moto implementato in figura 7.20. Per quanto riguarda la vite sono state implementate due possibili scelte: la prima è l'opzione di farla abbassare la seconda invece è quella di lasciarla in posizione di finecorsa, si è scelto di proseguire in questo modo in quanto alla vite come abbiamo visto precedentemente nel capitolo 5, è possibile collegare utensili e quindi l'abbassamento di una determinata quantità la predispone al disegno per traiettorie bidimensionali. Una volta eseguito il posizionamento, lo stato successivo riguarda

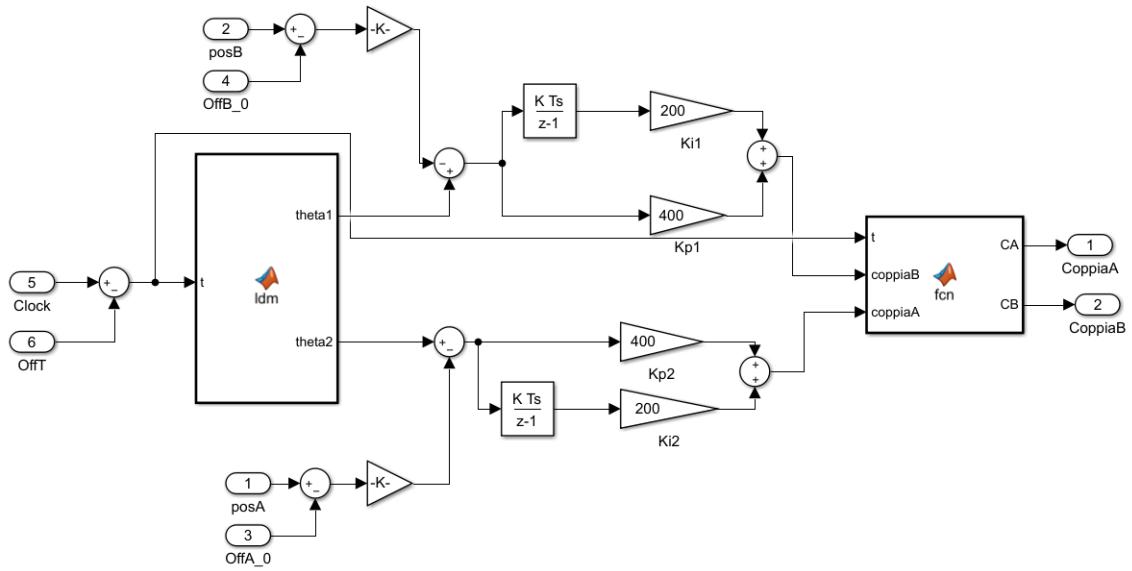


Figura 7.20: Implementazione funzione di posizionamento

la discesa della vite, per entrarci le coppie delle braccia dovranno essere pari a zero e sull’interfaccia grafica dovrà essere premuto il bottone relativo all’abbassamento della vite⁹. Per particolare per la discesa verrà utilizzata la variabile **dv** salvata precedentemente che indica di quanti centimetri la vite deve scendere; per farla scendere viene fornita una coppia al motore di traslazione finché non arriva alla posizione desiderata.

7.3.3 Fase di controllo

L’ultima fase è quella di controllo ed esecuzione della traiettoria, in questa fase andiamo a settare l’offset dei motori della vite, del tempo e dei motori delle braccia; resettare l’offset della vite è necessario in quanto il posizionamento (nel caso di discesa) è appena terminato; l’offset del tempo ci permette di far partire virtualmente il tempo da zero negli schemi di controllo dopo essere arrivati nella configurazione di *Posizionamento*. I figura 7.21 lo schema implementato di quest’ultima fase.

Come abbiamo anticipato precedentemente nello stato **OffSet** avviene il settaggio degli *offset* e tutte le coppie vengono messe a zero per evitare eventuali

⁹Sia che si voglia abbassare di una quantità che si voglia lasciare nella posizione alta il bottone va premuto.

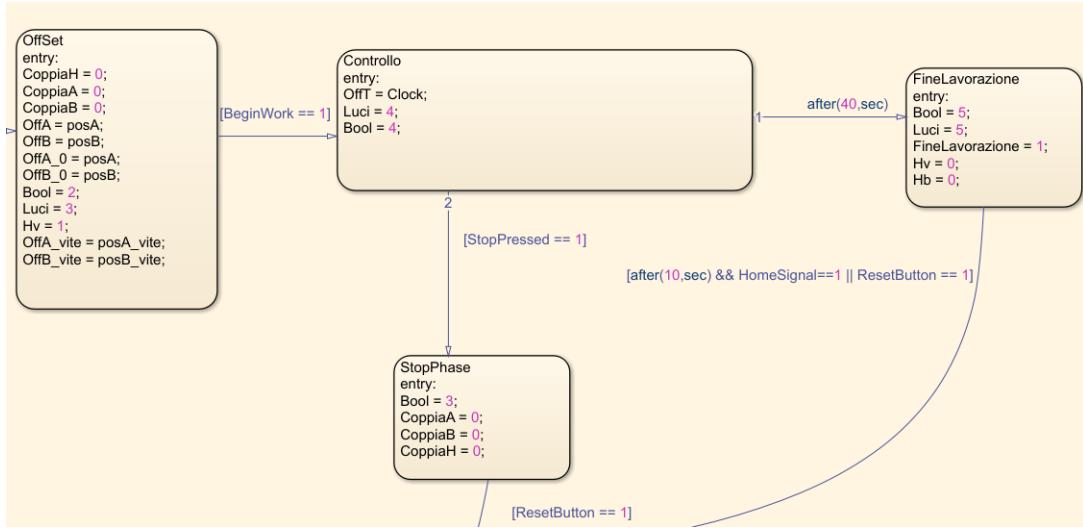


Figura 7.21: Fase di controllo

movimentazioni indesiderate. Per passare alla fase successiva abbiamo bisogno che venga premuto il bottone **Work** sull’interfaccia grafica, questo garantisce il passaggio allo stato *Controllo*, può darsi che il bottone non venga premuto subito, di conseguenza il clock potrebbe aumentare, per questo l’offset viene settato nella fase di **Controllo**. Per quanto riguarda la scelta della traiettoria tramite interfaccia grafica vi è la possibilità di scegliere quella desiderata. A livello di implementazione le traiettorie non sono altro che leggi di moto costruite in due o tre dimensioni. Dopo che il manipolatore inizia ad eseguire la traiettoria possono esserci due evoluzioni:

- la traiettoria viene eseguita correttamente
- la traiettoria da problemi

Nel primo caso abbiamo 40 secondi per eseguire la traiettoria (il tempo può essere personalizzato in base alla tipologia di traiettorie), alla fine di questo tempo passiamo allo stato **FineLavorazione** dove il manipolatore è fermo ed ha concluso la sua traiettoria, da questo, grazie al bottone reset è possibile ritornare alla fase di **homing**, oppure in automatico dopo un determinato periodo di tempo se il tasto di *posizionamento* è stato lasciato attivo il manipolatore torna nella configurazione predefinita. Nel secondo caso ci si accorge che la traiettoria sta dando problemi, ad esempio vibrazioni o si nota che il manipolatore rischia di entrare in singolarità, per

risolvere questi problemi vi è un bottone denominato **STOP** che permette l'arresto immediato del manipolatore, azzerando tutte le coppie. A differenza della pressione del fungo, che dopo lo sbloccaggio richiede il riavvio del dispositivo, nel caso in cui si entri nella fase di stop mediante il bottone di reset è possibile far tornare il manipolatore nella fase di attivazione, da questa poi sarà possibile far partire di nuovo la fase di homing e successivamente quella di posizionamento.

Gestione variabile di stato e luci

Durante tutte le fasi è possibile vedere nello stateflow che due variabili si evolvono costantemente: *bool* e *luci*.

La prima serve per indicare lo stato di lavoro nel quale si trova il manipolatore, in particolare l'evoluzione segue la tabella degli stati 6.

Valore	Stato
-1	Pre-operativo
0	Attivo
1	Homing
2	Manipolatore posizionato
3	Traiettoria
4	Fine lavorazione

Tabella 6: Valori variabile *bool*

Il valore di *luci*, serve esattamente a pilotare le luci presenti nel manipolatore¹⁰ secondo la tabella 7, in modo tale da avere un feedback visuale che indica fase in cui è il manipolatore.

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

Tabella 7: Valori luci

¹⁰Inizialmente le luci erano solo fisiche, montate nella parte posteriore del manipolatore, per comodità visiva sono state anche implementate nell'interfaccia grafica.

7.3.4 Interfaccia grafica

Per gestire al meglio le varie impostazioni, e per essere sicuri del passaggio tra i vari stati è stata creata un'interfaccia grafica *user-friendly* mediante *instrument panel* fornito da *Simulink real-time explorer*, possiamo vederla in figura 7.22.

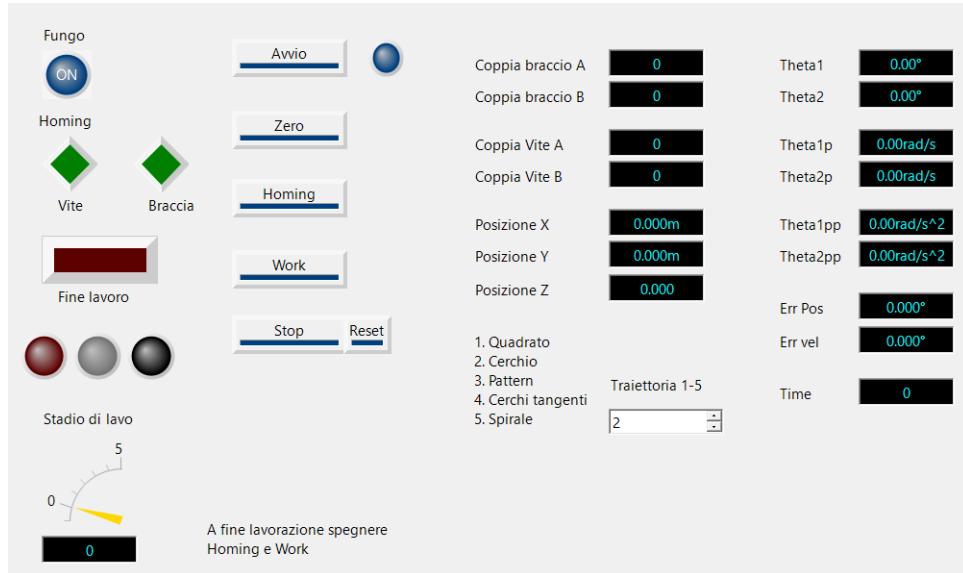


Figura 7.22: Interfaccia grafica

A sinistra, dall'alto in basso possiamo vedere dei LED che hanno la funzionalità di:

- Fungo: rimane acceso fintantoché il fungo non è premuto
- Homing: si accendono quando i motori hanno raggiunto il finecorsa
- Fine lavoro: appena la traiettoria assegnata finisce questo si accende
- Luci: sono i led visti nella tabella 7, che vanno ad indicare in che fase di lavoro è il manipolatore, per comodità visiva il led bianco nell'interfaccia è stato sostituito da uno ocra

Sotto questi led abbiamo un indicatore che va a specificare le fasi di lavoro come viste nella tabella 6. Al centro abbiamo i bottoni che permettono di passare tra le varie fasi dello stateflow e di fare tutte le operazioni quindi avvio, homing, posizionamento, lavoro stop e reset. In particolare i bottoni funzionano tramite il collegamento a determinate variabili; quelle di riferimento sono le ultime costanti collegate in input

allo stateflow visto in figura 7.12. A destra abbiamo invece una parte di visualizzazione dove vediamo tutti i parametri di interesse del manipolatore, in particolare le coppie fornite e le posizioni sia nel piano $[x, y, z]$ che quelle ai link motorizzati quindi θ_1, θ_2 . Infine, abbiamo il selettore di traiettoria, il quale permette la scelta fra le sei opzioni possibili (di default viene eseguito il cerchio).

1. Quadrato
2. Cerchio
3. Pattern
4. Cerchi tangenti
5. Spirale
6. Solo asse Z

Adesso che abbiamo introdotto la logica di funzionamento a stati, proseguiamo la nostra trattazione andando a vedere le tipologie di controllo che sono state implementate, in particolare inizieremo col guardare il controllo applicato alla vite e successivamente quello per le braccia.

7.4 Controllo vite

Lo scopo di questa sottosezione è quello di introdurre le due tipologie di controllo per la vite, da un punto di vista teorico e andare ad analizzare il comportamento pratico. Il sistema, essendo formato da due parti, e potendo fare solo due movimenti è ad un solo grado di libertà non risulta essere quindi molto complesso. Gli approcci di controllo utilizzati sono di tipo centralizzato, l'analisi di questi controllori verrà fatta mediante funzioni di trasferimento, proseguire con andiamo quindi ad introdurre la legge di controllo di un controllore PID generico:

$$\tau_v(t) = K_p \left(q_v(t) + \frac{1}{T_I} \int_0^t q_v(\tau) d\tau + T_D \frac{dq_v(t)}{dt} \right) \quad (7.1)$$

7.4.1 Controllo proporzionale

Il primo controllo implementato è quello proporzionale, questa tipologia di controllo si basa sull'idea che ingresso ed uscita siano legati in modo algebrico da un coefficiente K_p chiamato anche guadagno proporzionale. La legge di controllo è quindi definita come

$$\tau_v = K_p q_v \quad (7.2)$$

L'azione proporzionale è utile in quanto più grande è l'errore all'ingresso del control-

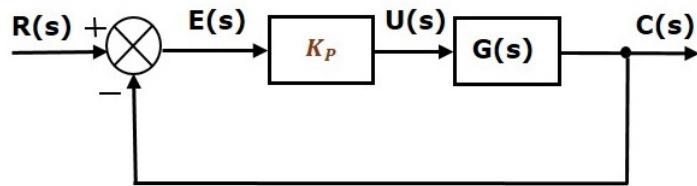


Figura 7.23: Schema teorico controllore proporzionale

lore e maggiore sarà l'azione svolta da esso. Il regolatore, riprendendo l'equazione 7.1 si può notare che solo con il contributo proporzionale la velocità di risposta del sistema aumenta, però con guadagno elevato diminuisce la stabilità e quindi aumentano le oscillazioni¹¹. Andiamo ora a vedere l'implementazione del controllore:

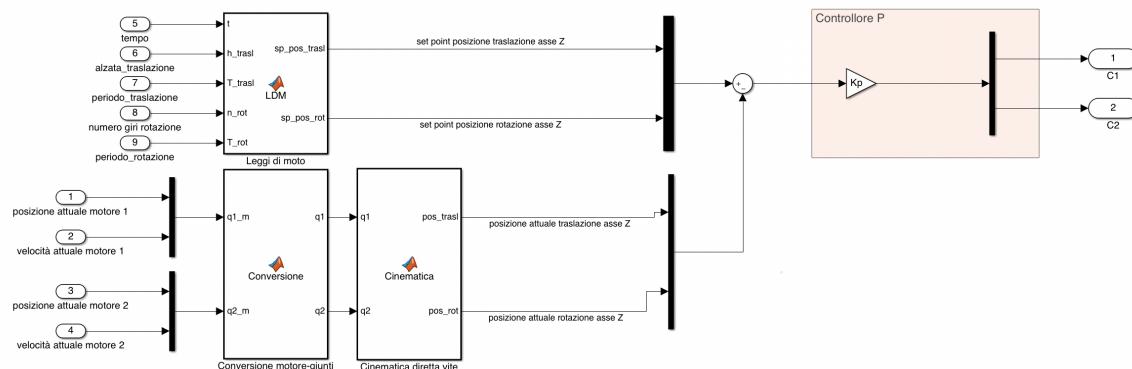


Figura 7.24: Controllore proporzionale vite

¹¹Il parametro proporzionale viene definito ed analizzato molto in letteratura, nella pratica però è solo un parametro teorico, infatti solitamente ci si riferisce alla banda proporzionale, definita come la variazione minima dell'ingresso che porta l'uscita al valore minimo percentuale.

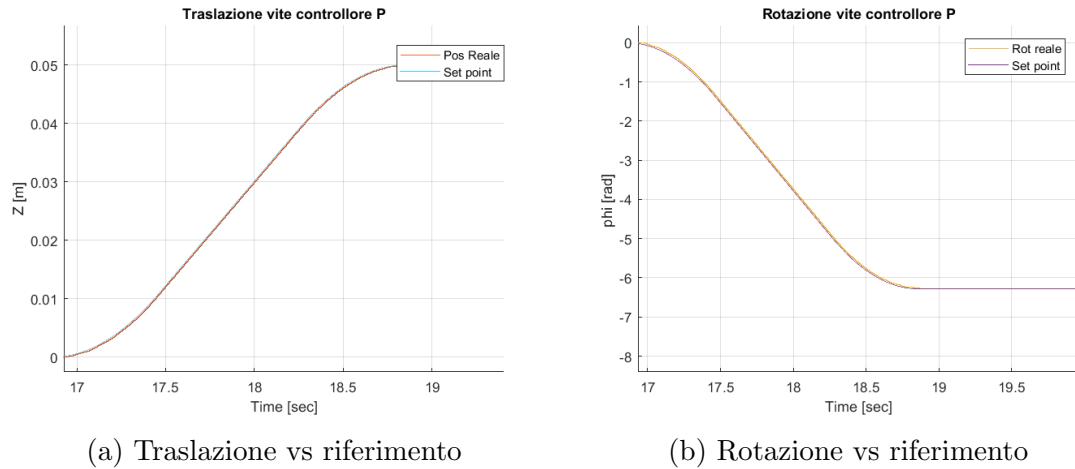


Figura 7.25: Andamento vite rispetto a riferimento

Il movimento effettuato è composto da una discesa di 5 centimetri in contemporanea ad una rotazione di 360° in due secondi.

7.4.2 Controllo proporzionale derivativo

A partire dal controllore precedente andiamo ad aggiungere una parte: l'azione derivativa. In uscita fornisce la derivata rispetto al tempo dell'errore, se dovessimo analizzarla singolarmente potremmo definire una legge del tipo:

$$\tau_v = K_D \dot{q}_v \quad (7.3)$$

Abbiamo la presenza della velocità, infatti il controllore derivativo viene anche chiamato controllore di velocità, il suo comportamento è sostanzialmente diverso da quello proporzionale in quanto l'uscita dipende dalla velocità con la quale varia l'errore e riesce a fornire un anticipo di fase. Riprendendo l'equazione 7.1 il parametro che controlla questo anticipo è T_D infatti nel caso in cui $K_P = K_I = 0$ andando ad analizzare solo il suo comportamento si nota che la stabilità del sistema peggiora sia aumentando che diminuendo il valore. Andando ora ad unire i due contributi otteniamo un controllore PD. Possiamo scrivere l'azione combinata dei due controllori considerando il rapporto ingresso uscita e ottenendo:

$$\tau_v = K_P(1 + sT_D) \quad (7.4)$$

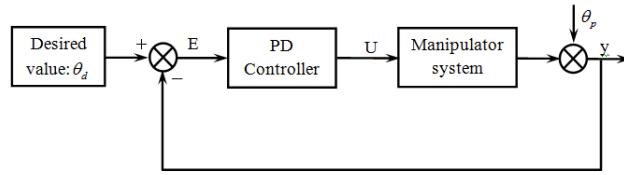


Figura 7.26: Schema teorico controllore PD

con $T_D = \frac{K_D}{K_P}$. In caso di ingresso definito come uno scalino, la presenza del termine derivativo va ad introdurre uno zero e ad aumentare il coefficiente di s, andando a ridurre le oscillazioni e di conseguenza stabilizzando il sistema.

Andiamo ora a vedere l'implementazione sul manipolatore del controllore PD:

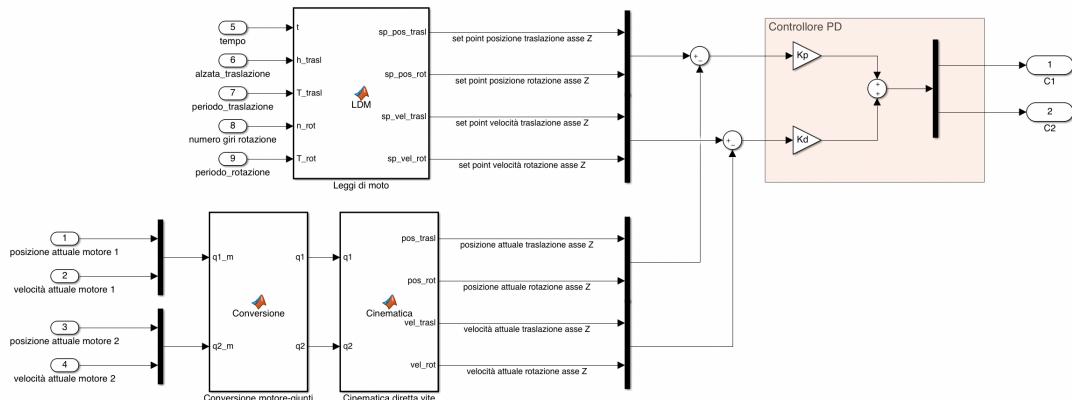


Figura 7.27: Controllore PD Vite

A partire da una legge di moto per entrambi i motori che fornisce quattro set point, abbiamo: una prima fase di raccolta dei dati nei quali si prendono le posizioni e velocità dei motori della vite, essendo i motori connessi a riduttori, ed operando sui giunti abbiamo un blocco che si occupa di fare la conversione; vi è poi la necessità di effettuare la cinematica diretta per poter passare dagli angoli alle coordinate dell'asse z. In seguito verrà fatta una differenza tra le posizioni e velocità per poter trovare gli errori. Questi, verranno poi moltiplicati per K_p (posizione) e K_d (velocità) per ottenere i valori della legge di controllo da assegnare ai motori. Presentiamo ora il risultato di una simulazione di una traiettoria che comprendeva la discesa della vite in due secondi di 5 centimetri ed in contemporanea una rotazione di 360° .

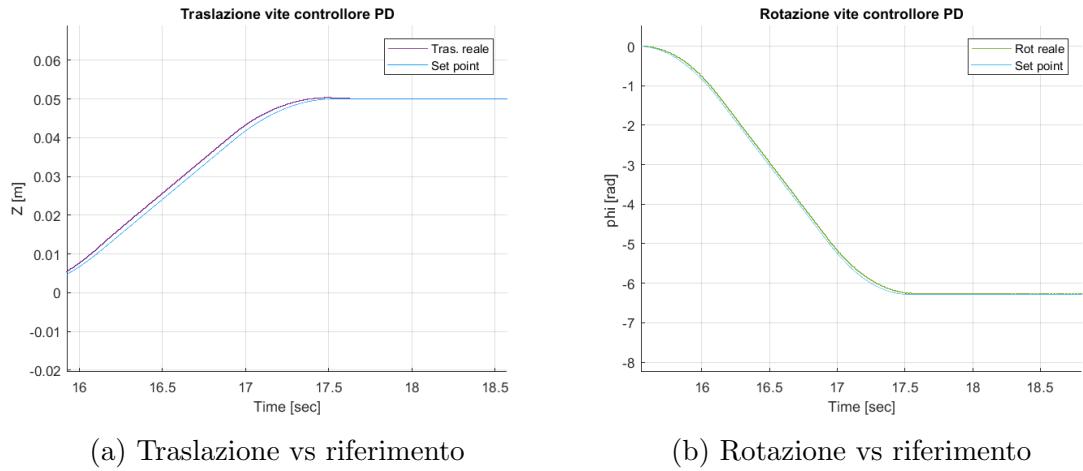


Figura 7.28: Andamento vite rispetto a riferimento

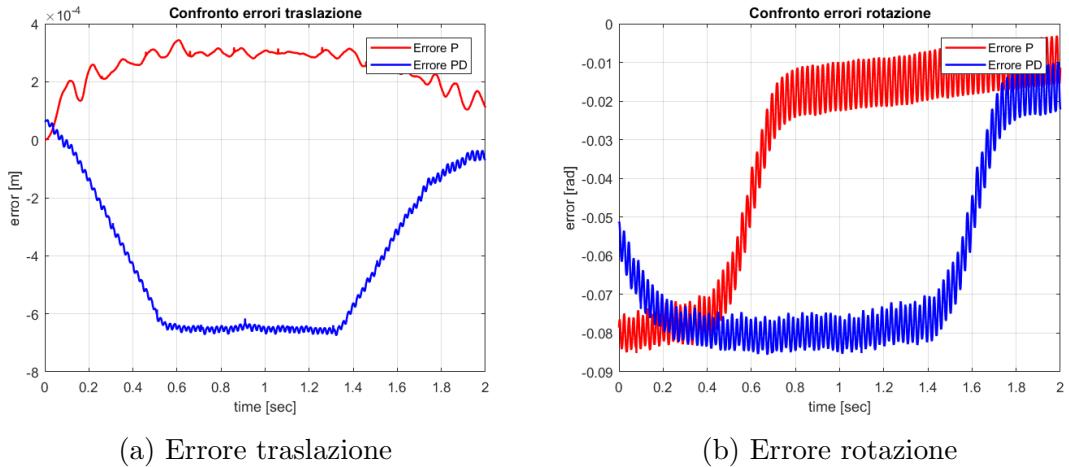


Figura 7.29: Errori controllori vite

7.4.3 Confronto controllori vite

Possiamo vedere confronto fra gli errori di traslazione e rotazione tra i due approcci di controllo implementati per la vite in figura . A differenza del controllore P quello PD integra la velocità, seguendo questo approccio le traiettorie hanno un ritardo minore; il problema principale è che la velocità ricevuta in output dagli azionamenti presenta vibrazioni e di conseguenza il controllo risulta imperfetto. Per la scelta del controllore bisogna fare un *trade-off* tra ritardo e vibrazioni.

7.5 Controllo braccia

Questa sottosezione si pone come obiettivo l'introduzione di approcci di controllo noti in letteratura e applicarli ai link motorizzati del manipolatore; in particolare andando ad analizzare il loro funzionamento a livello teorico, la loro implementazione pratica ed i risultati ottenuti, cercando quindi il controllore migliore. Tutte le tipologie di controllo introdotte saranno di tipo centralizzato, gli algoritmi di questa tipologia sfruttano una conoscenza più approfondita del modello dinamica del manipolatore in modo tale da compensare i termini di accoppiamento non lineari, verranno quindi introdotte coppie di compensazione per i termini NL, inoltre, qualsiasi incertezza della struttura e qualunque imprecisione nella misura della posizione daranno origine ad una perdita di accuratezza e quindi a problemi di controllo.

7.5.1 Controllo proporzionale derivativo

La prima tipologia di controllo applicata è stata quella proporzionale derivativa. L'obiettivo di questa tecnica è quello di risolvere il problema della regolazione, ovvero assegnare la posizione corretta all'end-effector rispetto ad un riferimento di equilibrio costante. Non viene risolto il problema della dinamica con la quale si raggiunge quella configurazione, però come obiettivo ci si pone di trovare la struttura del controllore che ci assicuri una stabilità asintotica in quella specifica posa desiderata. Definiamo lo stato come:

$$\tilde{q} = q_m^0 - q_m \quad (7.5)$$

Con q_m^0 che rappresenta il *set-point* (posizione desiderata) e q_m la posizione attuale, per poter andare a risolvere il problema abbiamo bisogno di introdurre il metodo diretto di Lyapunov, il quale dice che:

L'analisi della stabilità di un punto di equilibrio viene fatta utilizzando, oltre alle equazioni di stato del sistema, opportune funzioni scalari, dette funzioni di Lyapunov, definite sullo spazio degli stati.

Definiamo quindi la funzione come:

$$V(q : m, \tilde{q}) = \frac{1}{2} \dot{q}_m M(q_m) \dot{q}_m + \frac{1}{2} \tilde{q} K_p \tilde{q} > 0 \quad (7.6)$$

Il termine K_p rappresenta la rigidezza del sistema ed è una matrice ($n \times n$) simmetrica e definita positiva. Considerando che il *set-point* è un termine costante possiamo andare a derivare la funzione V rispetto al tempo ottenendo:

$$\dot{V} = \ddot{q}_m M(q_m) \dot{q}_m + \frac{1}{2} \dot{q}_m M(q_m) \dot{q}_m - \dot{q}_m K_p \tilde{q}$$

e sapendo che $\tau_m = M(q_m) \ddot{q}_m + C(q_m, \dot{q}_m) \dot{q}_m$ ¹² possiamo riscrivere la funzione come

$$\dot{V} = \dot{q}_m [\tau_m - K_p \tilde{q}] \quad (7.7)$$

Se adesso consideriamo la legge di controllo PD definita come:

$$\tau_m = K_p \tilde{q} - K_d \dot{q}_m \quad (7.8)$$

e andiamo a sostituirla nell'equazione 7.7 otteniamo

$$\dot{V} = -\dot{q}_m K_d \dot{q}_m \leq 0$$

Se K_d è una matrice definita positiva allora \dot{V} è semidefinita negativa, la dinamica del sistema è quindi:

$$M(q_m) \ddot{q}_m + C(q_m, \dot{q}_m) \dot{q}_m = K_p \tilde{q} - K_d \dot{q}_m \quad (7.9)$$

In particolare abbiamo che velocità ed accelerazioni sono nulla in corrispondenza di $\dot{V} = 0$. Lo schema teorico di controllo è uguale a quello visto in figura 7.26 nella sezione della vite.

Possiamo vedere lo schema di controllo implementato sul manipolatore: Avendo la possibilità di scegliere tra più traiettorie, a partire dal tempo attuale e dalla traiettoria desiderata andiamo a definire una legge di moto che fornirà i *set-point* in posizione, velocità all'end-effector. Per andare a convertire questi in coordinate ai giunti vi è un blocco che si occupa di fare la cinematica inversa. Vengono poi

¹²Nella trattazione di Lyapunov appare anche il termine di compensazione gravitazionale però nel manipolatore analizzato questo termine è costante e pari a zero.

7.5 Controllo braccia

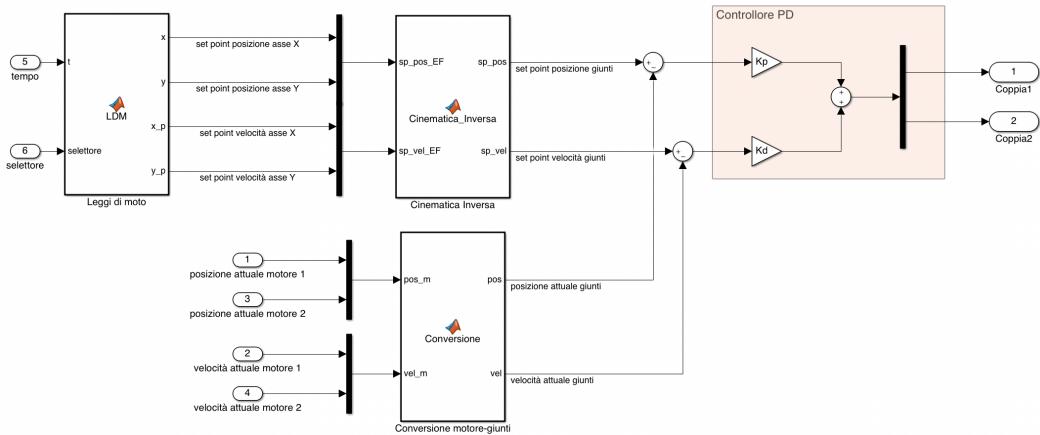


Figura 7.30: Controllore PD braccia

lette le posizioni e velocità attuali dei motori e verranno successivamente convertite ai giunti; adesso che abbiamo tutto a livello di giunti possiamo fare la differenza tra riferimento e reale ottenendo un errore in posizione e uno in velocità¹³. Gli errori verranno moltiplicati per K_p e K_d per ottenere la legge di controllo che verrà assegnata ai motori.

Per poter andare a testare il controllore è stata assegnata una legge di moto che disegna un cerchio di raggio 5cm in 5 secondi. Andiamo ora a vedere le coppie assegnate ai giunti θ_1 e θ_2 :

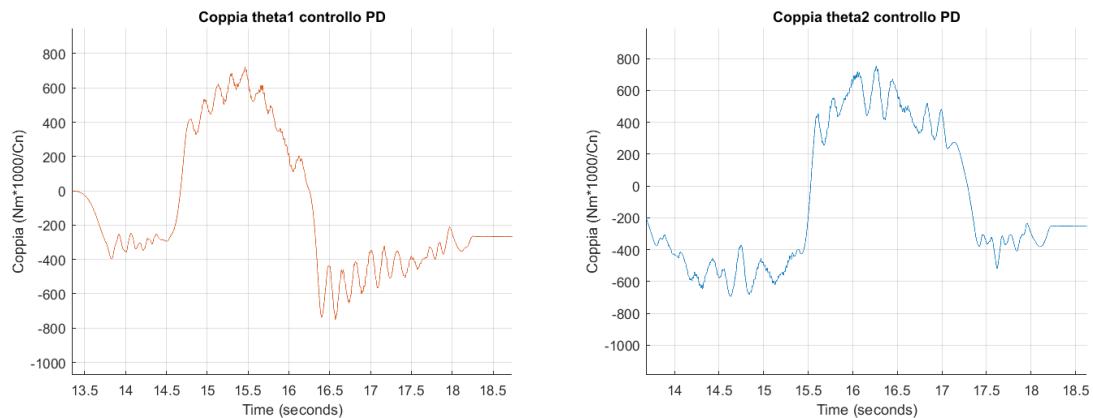


Figura 7.31: Coppie controllore PD braccia

Possiamo poi andare ad analizzare la posizione ottenuta con quelle coppie rispetto

¹³gli errori sono dei vettori [2x1] contenenti le due componenti relative ai due angoli θ_1, θ_2 .

al riferimento, e di conseguenza trovare anche l'errore:

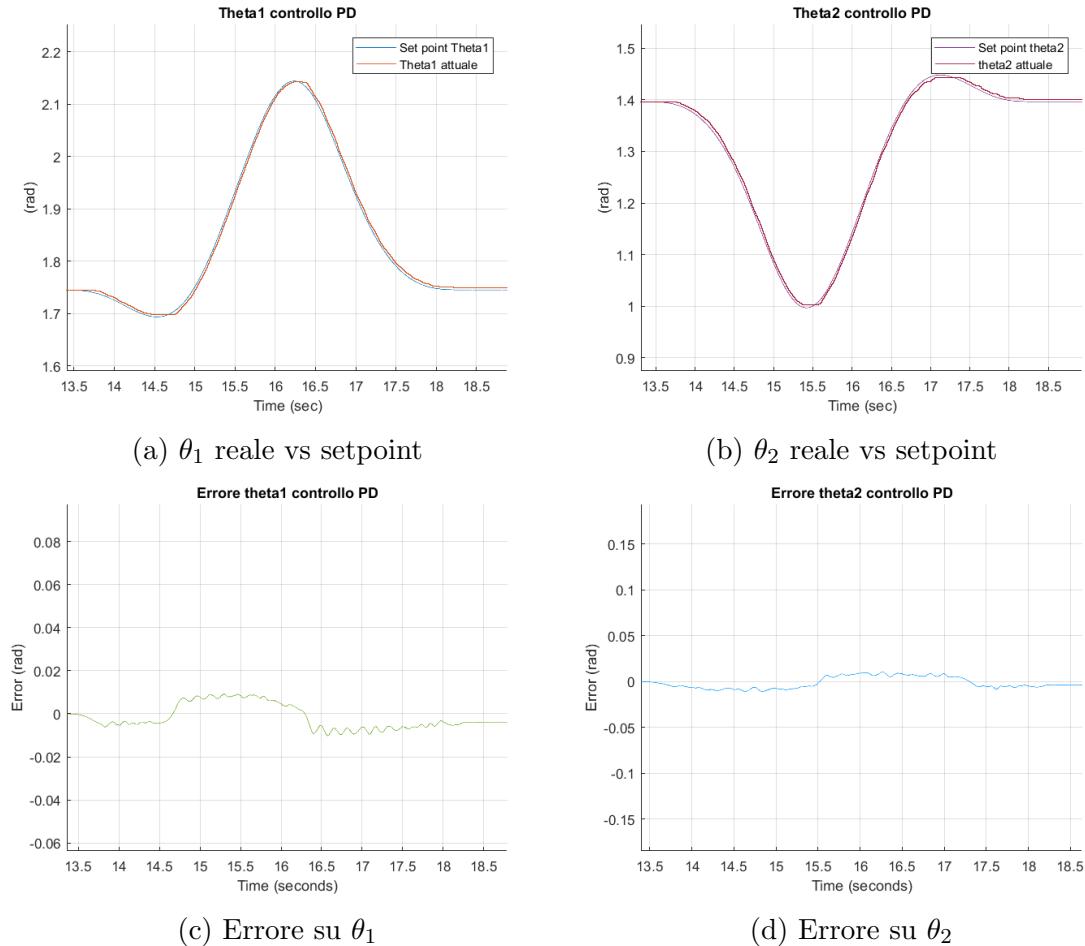


Figura 7.32: Andamenti posizione ed errori controllore PD

Andiamo poi a rappresentare la traiettoria reale e il riferimento anche sull'asse [X,Y] in figura 7.33

7.5.2 Controllo feed-forward con coppia pre-computata

Il passo successivo è stato quello di studiare ed implementare un controllore che opera in anello aperto; in particolare questo controllore calcola le coppie di disturbo basate sul modello matematico del sistema con parametri d'ingresso il *set-point* in posizione e velocità e accelerazione. L'introduzione di questi termini riesce a risolvere in maniera corretta il problema del tracciamento della traiettoria desiderata, di conseguenza gli elementi introdotti riescono a compensare gli effetti di accoppiamento

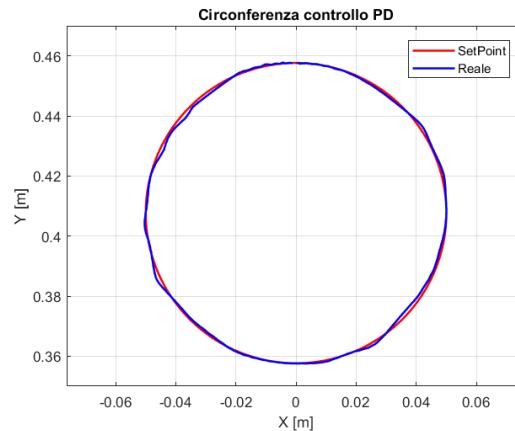


Figura 7.33: Cerchio assi [x,y] controllore PD

presenti nel modello dinamico del sistema.

$$g_d = \Delta M(q_m^0) \ddot{q}_m^0 + C(q_m^0, \dot{q}_m^0) \dot{q}_m^0 \quad (7.10)$$

L'idea è che i termini vengono calcolati considerando i valori di posizione, velocità ed accelerazione del riferimento, poiché grazie ad una legge di moto sono sempre ben noti. L'elemento g_d compensa i termini di accoppiamento non lineari dovuti a forza d'inerzia, di Coriolis e centrifughe che dipendono dalla struttura e di conseguenza variano durante il movimento del manipolatore; in generale però calcolare questo termine è molto impegnativo, di conseguenza l'utilizzo di questo approccio su un sistema online può richiedere molto tempo, per questo solitamente vengono compensati solo i termini più importanti come quelli inerziali.

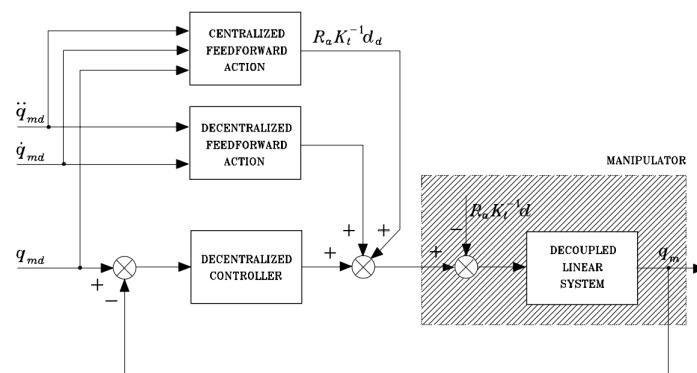


Figura 7.34: Schema teorico controllo feedforward

A livello implementativo possiamo vederlo come:

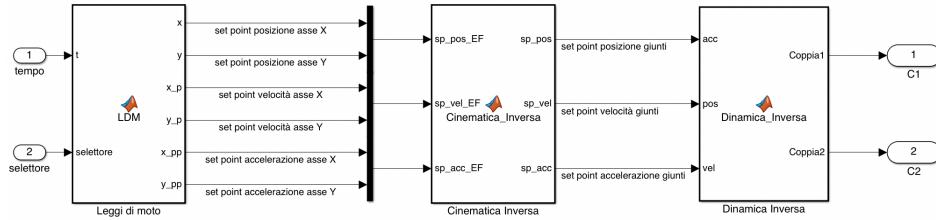


Figura 7.35: Controllore feedforward

A partire dalla legge di moto selezionata si ottengono i *set-point* all'end-effector, che grazie alla cinematica inversa vengono convertiti in riferimento ai giunti. Una volta ottenuti i riferimenti di posizione, velocità ed accelerazione si va ad applicare la funzione di dinamica inversa per calcolare le coppie di cui abbiamo bisogno. Le coppie sono quindi calcolate esclusivamente mediante dai riferimenti, è possibile vedere questo controllore come una versione di quello a dinamica inversa in anello aperto. Eseguendo traiettorie con una dinamica maggiore, con una conseguente diminuzione

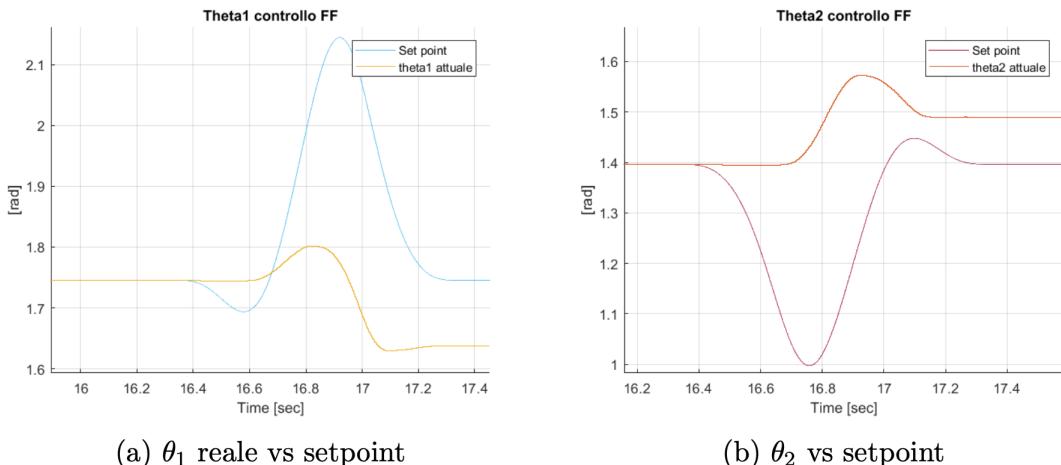


Figura 7.36: Andamenti controllo feed-forward

del tempo di esecuzione il manipolatore riesce ad eseguire un movimento, però non segue la traiettoria desiderata; questo è dovuto alla mancata stima degli attriti nel modello considerato.

7.5.3 Controllo in dinamica inversa

Riprendendo le tecniche di controllo centralizzato caratteristiche della letteratura l'approccio successivo è stato quello del controllo in dinamica inversa. Il primo step è stato quello di prendere l'equazione della dinamica e ridefinirla come:

$$M(q_m)\ddot{q}_m + n(q_m, \dot{q}_m) = \tau_m \quad (7.11)$$

Dove $n(q_m, \dot{q}_m)$ raccoglie i termini centrifughi e di Coriolis (in letteratura raccoglie anche i termini gravitazionali, ma come anticipato precedentemente nel nostro caso equivalgono a zero). L'idea del controllo a dinamica inversa si basa sul trovare il vettore di coppie τ_m come funzione dello stato del sistema, cercando di creare una relazione ingresso-uscita di tipo lineare. Considerando che l'equazione della dinamica è lineare nel controllo e che la matrice d'inerzia è invertibile in ogni configurazione del manipolatore abbiamo la garanzia di trovare un controllore linearizzato di questo tipo. Possiamo andare a riscrivere il controllo come:

$$\tau_m = M(q_m)y + n(q_m, \dot{q}_m) \quad (7.12)$$

Dove $y = \ddot{q}_m$ rappresenta un vettore d'ingresso con espressione ancora da determinare. La legge di controllo è basata sul calcolo della dinamica inversa del manipolatore, il sistema è lineare e disaccoppiato rispetto al nuovo ingresso, questo implica che le componenti y_k con $k = -\infty \dots m-1$ influenzano solo la variabile q_m indipendentemente dal movimento degli altri giunti. In questo modo il problema del controllo è quello di trovare una legge \mathbf{y} stabilizzante. Viene quindi scelta:

$$y = K_p \tilde{q} + K_d \tilde{\dot{q}} + \dot{q}_m^0 \quad (7.13)$$

Dove con i termini \tilde{q} e $\tilde{\dot{q}}$ si indica la differenza tra il *setpoint* e la misurazione attuale di posizione e velocità.

$$\tilde{q} = q_m^0 - q_m \quad \tilde{\dot{q}} = \dot{q}_m^0 - \dot{q}_m \quad (7.14)$$

Possiamo andare ora a sostituire le definizioni di 7.14 in 7.13 e troviamo l'espressione:

$$\ddot{\tilde{q}} + K_d + \tilde{q} = 0 \quad (7.15)$$

Chiaramente l'errore si verifica quando uno o entrambi i termini sono diversi da zero. Però, volendo assegnare la dinamica a ciascun giunto ci basta selezionare i guadagni delle matrici K_p e K_d , inoltre, se le due matrici sono definite positive è possibile ottenere un sistema asintoticamente stabile. Possiamo scegliere le matrici come:

$$K_p = \text{diag}(\omega_{01}^2, \dots, \omega_{0n}^2) \bar{M} \quad K_d = \text{diag}(2\xi_1\omega_{01}, \dots, 2\xi_n\omega_{0n}) \bar{M}$$

Grazie al controllo a dinamica inversa i termini di compensazioni vengono calcolati ad ogni iterazione, quindi a piccoli intervalli temporali, il loop interno serve per ottenere una relazione ingresso/uscita lineare e disaccoppiata mentre quello esterno grazie alla dinamica desiderata serve a stabilizzare il sistema. L'utilizzo di que-

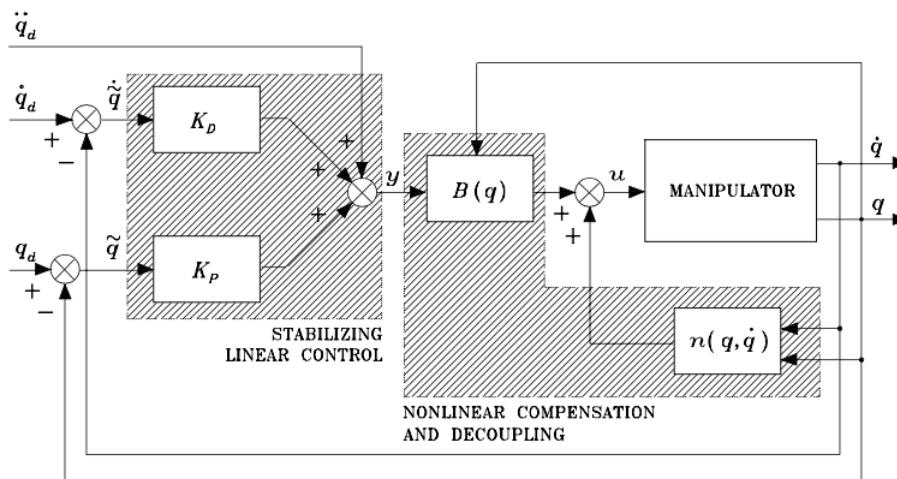


Figura 7.37: Schema controllore dinamica inversa

sto approccio è basato sull'ipotesi della cancellazione perfetta dei termini dinamici, quindi, i parametri dinamici del sistema devono essere accuratamente conosciuti e l'equazione del moto deve essere calcolata in real-time. Possiamo andare a vedere lo schema implementato sul controllore come: Come per tutti i controllori visti fino ad ora a partire da selettore e dal tempo ed applicando la cinematica inversa

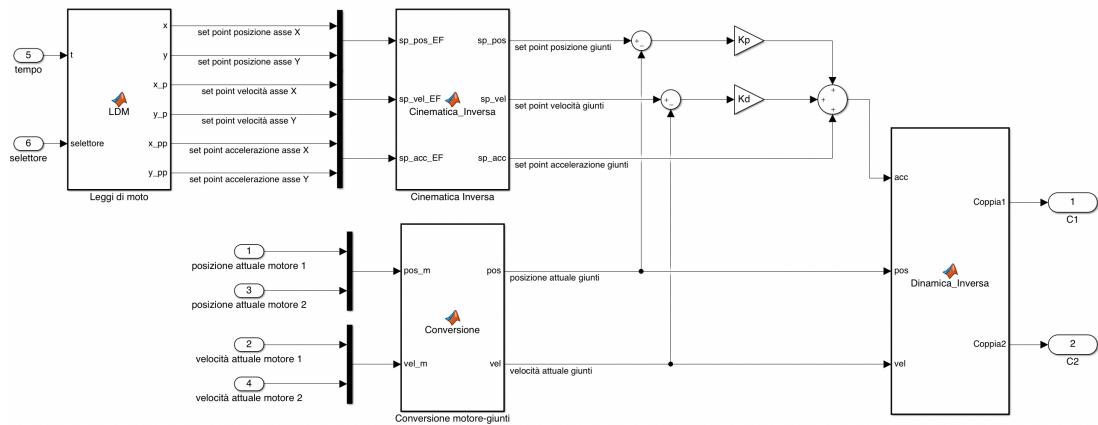


Figura 7.38: Controllore dinamica inversa

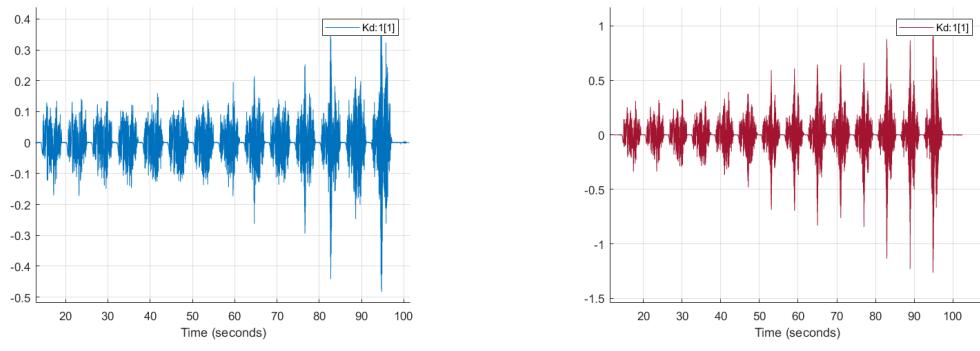
andiamo ad ottenere i riferimenti di posizione, velocità ed accelerazione; andiamo poi a prendere le posizioni e le velocità attuali dei motori e le convertiamo da motori ai giunti. Successivamente andiamo per posizione e velocità a trovare gli errori che verranno moltiplicati rispettivamente per K_p e K_d . Per ottenere la y dobbiamo sommare questi due componenti insieme all’accelerazione del riferimento. Una volta ottenuto questo andiamo ad applicare la funzione di dinamica inversa con ingressi pari a $ID(y, q, \dot{q})$, da questa otteniamo la legge di controllo e le relative coppie che andranno assegnate ai link del manipolatore.

Test K_p e K_d

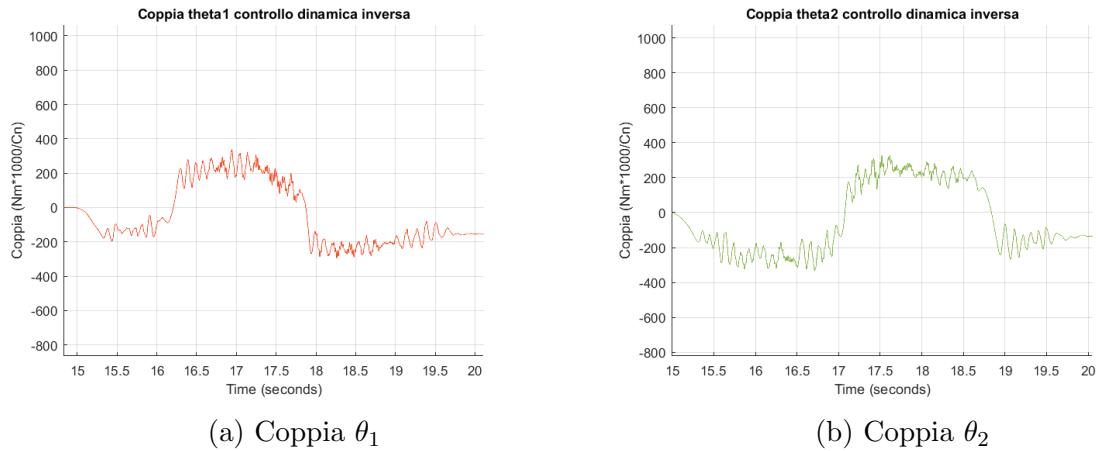
Per andare a scegliere K_p e K_d in modo preciso è stato adottato un approccio sperimentale, ovvero far variare K_p , in un range di valori compreso ad incrementi di 50, in base ad un valore fisso di K_d . La legge di moto selezionata è stata quella del cerchio in cinque secondi, una legge molto lenta andando ad analizzare quando il controllo iniziava a dar origine a vibrazioni. Vediamo in figura 7.39 un risultato di questo metodo, dai test condotti si vede che all’aumentare di entrambi i parametri le vibrazioni iniziano a crescere, e sono visibili prima

Analisi

Anche in questo caso, come nei precedenti la traiettoria principale testata è stata quella del cerchio raggio 5cm in 5 secondi, possiamo andare a vedere le coppie


 (a) Test con $K_d = 1.5$

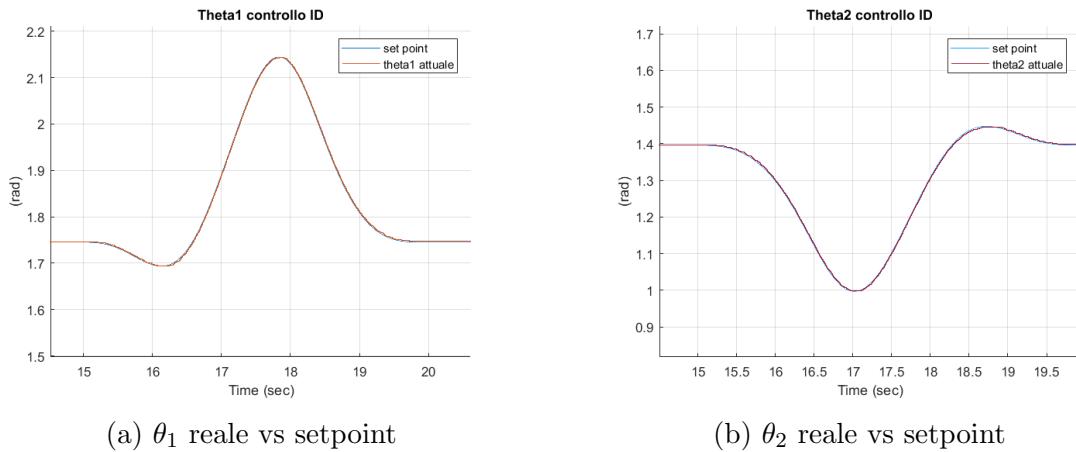
 (b) Test con $K_d=3$

 Figura 7.39: Ricerca di K_d al variare di K_p

 (a) Coppia θ_1

 (b) Coppia θ_2

Figura 7.40: Coppie giunti controllo dinamica inversa

fornite ai giunti, come per il controllore PD possiamo andare a vedere e comparare la posizione attuale rispetto al riferimento e trovare l'errore:


 (a) θ_1 reale vs setpoint

 (b) θ_2 reale vs setpoint

Figura 7.41: Posizione vs riferimento controllore ID

7.5 Controllo braccia

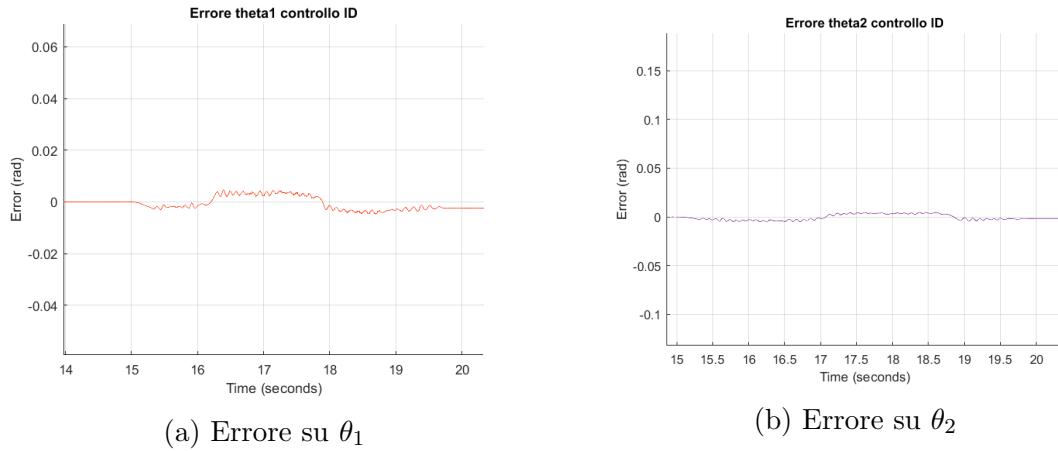


Figura 7.42: Errori controllore ID

Come per il controllore PD anche qua possiamo andare a rappresentare la traiettoria nel piano cartesiano, ottenendo:

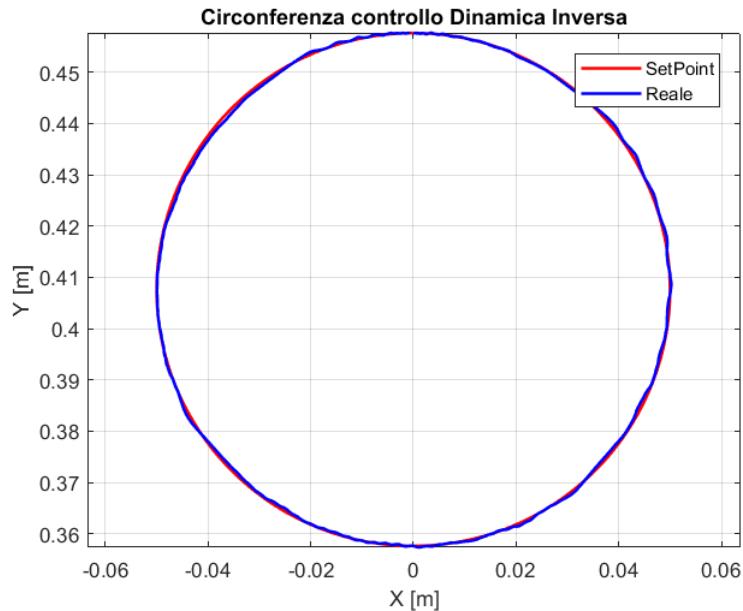


Figura 7.43: Cerchio [X,Y] controllo ID

7.5.4 Controllo robusto

L'effetto di incertezze sul modello induce in errore il sistema di controllo, soprattutto nel caso reale bisogna supporre che la compensazione del modello dinamico risulti imperfetta magari a causa di approssimazioni oppure per semplificazioni. Possiamo

andare quindi a riscrivere l'equazione 7.11 come:

$$\tau_m = \hat{M}(q_m)y + \hat{n}(q_m, \dot{q}_m) \quad (7.16)$$

Dove \hat{M} e \hat{n} rappresentano i parametri stimati del modello dinamico, è possibile rappresentare l'incertezza come:

$$\overline{M} = \hat{M} - M \quad \overline{n} = \hat{n} - n$$

Riprendendo la legge di controllo vista prima possiamo riscriverla come:

$$M(q_m)\ddot{q}_m + n(q_m, \dot{q}_m) = \hat{M}(q_m)y + \hat{n}(q_m, \dot{q}_m) \quad (7.17)$$

Essendo la matrice M invertibile in ogni configurazione possiamo ricavare \ddot{q}_m :

$$\ddot{q}_m = y + (M^{-1}\hat{M} - I)y + M^{-1}\tilde{n} = y - \eta$$

Dove η è una funzione non lineare definita come:

$$\eta = (I - M^{-1}\hat{M})y - M^{-1}\tilde{n} \quad (7.18)$$

Adottando la legge 7.14 vista nel caso del controllore a dinamica inversa possiamo ottenere che l'errore dinamico è gestito dall'equazione:

$$\ddot{\tilde{q}} + K_d\dot{\tilde{q}} + K_p\tilde{q} = \eta \quad (7.19)$$

Otteniamo quindi un sistema non lineare e accoppiato, di conseguenza implementare un semplice controllore PD non basta; per risolvere questo problema occorre inserire un termine non lineare, che sia funzione dell'errore e creato appositamente per fornire robustezza al controllo. Come per il controllore PD si ricerca una funzione grazie al

metodo diretto di Lyapunov. Andiamo a definire lo stato del sistema come:

$$\xi = \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix}$$

sostituendo poi lo stato nell'equazione $\ddot{q}_m = y - \eta$ si ottiene un'equazione differenziale del primo ordine

$$\dot{\xi} = H\xi + D(\ddot{q}_m^0 - y + \eta) \quad (7.20)$$

Dove H e D sono definite come: $H = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{(2n \times 2n)}$, $D = \begin{bmatrix} 0 \\ I \end{bmatrix} \in \mathbb{R}^{(2n \times n)}$

Si può vedere il problema di inseguimento della traiettoria come la soluzione che va a stabilizzare il sistema non lineare di qui sopra. Nella letteratura del controllore robusto, anche se l'incertezza η non è nota è comunque disponibile un suo intervallo di variazione. La legge \mathbf{y} dovrebbe garantire stabilità di $\dot{\xi}$ per ogni η nell'intervallo.

Di conseguenza vengono formulate tre assunzioni:

1. $\sup_{t \geq 0} \|\ddot{q}_m^0\| < Q_M < \infty \forall \ddot{q}_m^0$
2. $\|I - M^{-1}\hat{M}(q_m)\| \leq \alpha \leq 1 \forall q_m$
3. $\|\tilde{n}\| \leq \Phi(\|\xi\|) < \infty \forall q_m, \dot{q}_m$

Andando ad analizzare le assunzioni possiamo dire che:

- La prima assunzione è soddisfatta sempre, in quanto per ogni traiettoria definita le accelerazioni non possono essere infinite
- La seconda assunzione conferma che M (e di conseguenza M^{-1}) sia superiormente e inferiormente limitata, infatti

$$0 < M_m < \|M^{-1}(q_m)\| \leq M_M < \infty \forall q_m \quad (7.21)$$

Quindi, esiste sempre una scelta di \hat{M} che soddisfa la condizione, selezionando per esempio

$$\hat{M} = \frac{2}{M_M + M_m} I$$

otteniamo

$$||M^{-1}(q_m)\hat{M}(q_m) - I| \leq \alpha = \frac{M_M - M_m}{M_M + M_m} < 1 \quad (7.22)$$

Il limite inferiore si ha quando $\hat{M} = M$ in quanto $\alpha = 0$. Concentrandoci sull'assunzione 3, si può osservare che \tilde{n} è funzione di q_m, \dot{q}_m , nel primo caso, in base alla tipologia di giunto (rotoidale o prismatico) abbiamo intervalli che sono limitati e quindi il contributo è limitato. Anche la velocità è limitata grazie all'effetto della saturazione (esistente sulle velocità massime dei motori). Di conseguenza, prendiamo per Φ una funzione calcolata nella norma dello stato come:

$$\Phi(||\xi||) = \alpha_0 + \alpha_1||\xi|| + \alpha_2||\xi||^2 \quad (7.23)$$

Andando a riprendere la legge di controllo 7.13 ed ampliandola possiamo definirla come:

$$y = \ddot{q}_m^0 + K_p \tilde{q} + K_d \dot{\tilde{q}} + \omega \quad (7.24)$$

Il termine PD ci assicura la stabilizzazione della dinamica della matrice dell'errore, \ddot{q}_m^0 fornisce un termine di previsione e ω è progettato in modo da combattere l'incertezza fornendo robustezza al sistema.

$$\dot{\xi} = \tilde{H}\xi + D(\eta - \omega) \quad (7.25)$$

Dove:

$$\tilde{H} = H - DK = \begin{bmatrix} 0 & I \\ -K_p & -K_d \end{bmatrix} \in \mathbb{R}^{(2n \times 2n)}$$

$$K_p = diag(\omega_{01}^2, \dots, \omega_{0n}^2) \quad K_d = diag(2\xi_1\omega_{01}^2, \dots, 2\xi_n\omega_{0n}^2)$$

Per andare a definire *omega* si procede col metodo diretto di Lyapunov; consideriamo come funzione:

$$V(\xi) = \xi^T Q \xi > 0, \forall \xi \neq 0 \quad (7.26)$$

con Q matrice simmetrica definita positiva. Facendo la derivata lungo la traiettoria si trova che

$$\dot{V} = \xi^T P \xi + 2z^T (\eta - \omega) \quad (7.27)$$

Considerando che \tilde{H} ha tutti gli autovalori con parte reale negativa, possiamo scegliere una qualunque P definita positiva che soddisfi:

$$\tilde{H}^T Q + Q \tilde{H} = -P$$

Consideriamo ora l'equazione 7.27 , $z = D^T Q \xi$, il primo termine sulla parte di destra è definito negativo, di conseguenza la soluzione converge solo se $\xi \in \mathbb{N}(D^T Q)$, se invece non appartiene, il controllo ω andrà scelto per rendere il secondo termine minore o uguale a zero utilizzando la legge di controllo:

$$\omega = \rho(|\xi|) \frac{z}{\|z\|}, \rho > 0 \quad (7.28)$$

con ρ funzione positiva da determinare. Scegliendo ω in questo modo otteniamo:

$$z^T (\eta - \omega) = z^T \eta - \rho(|\xi|) \frac{z z^T}{\|z\|} \leq \|z\| [|\eta| \cdot \rho(|\xi|)]$$

se poi garantiamo che: $|\xi| > |\eta|$ per ogni valore di posizione, velocità e velocità di riferimento, otteniamo che questo termine e \dot{V} sono negativi lungo tutte le traiettorie dell'errore; per poter soddisfare la diseguaglianza, dalla definizione di η troviamo:

$$\eta = (I - M^{-1} \hat{M})y - M^{-1} \tilde{n} \text{ e } y = \ddot{q}_m^0 + K\xi + \omega \quad (7.29)$$

sapendo poi che $|\omega| = \rho$ troviamo che:

$$|\eta| \leq \|I - M^{-1} \hat{M}\| (\|\ddot{q}_m^0\| + \|K\| |\xi| + |\omega|) + \|M^{-1}\| \|\tilde{n}\|$$

Andando a maggiorare questa quantità troviamo che:

$$|\eta| \leq \alpha Q_M + \alpha \|K\| |\xi| + \alpha \rho(|\xi|) + M_M \Phi(|\xi|) < \rho(|\xi|) \quad (7.30)$$

è quindi possibile andare a selezionare un *rho* in modo tale che:

$$\rho(|\xi|) \geq \frac{1}{1-\alpha} [\alpha Q_M + \alpha \|K\| |\xi| + M_M \Phi(|\xi|)] \quad (7.31)$$

Si può osservare che $\Phi(||\xi||) = \alpha_0 + \alpha_1||\xi|| + \alpha_2||\xi||^2$ per soddisfare 7.31 basta che si selezionino:

$$\rho(||\xi||) = \beta_0 + \beta_1||\xi|| + \beta_2||\xi||^2$$

Con i valori:

$$\begin{cases} \beta_0 \geq \frac{\alpha Q_M + \alpha_0 M_M}{1-\alpha} \\ \beta_1 \geq \frac{\alpha K + \alpha_1 M_M}{1-\alpha} \\ \beta_2 \geq \frac{\alpha_2 M_M}{1-\alpha} \end{cases}$$

Andando poi a sostituire tutto nell'equazione 7.27 troviamo:

$$\dot{V} = -\xi^T P \xi + 2z^T (\eta - \rho(||\xi||) \frac{z}{||z||}) < 0 \quad \forall \xi \neq 0 \quad (7.32)$$

Di conseguenza $\xi = 0$ è un equilibrio di stato asintoticamente e globalmente stabile. La legge di controllo è quindi formata da tre termini, un primo termine che compensa i termini non lineari e gli accoppiamenti tra i link, un secondo termine che stabilizza il sistema dinamico grazie ad una retroazione con previsione per l'errore e un terzo termine che fornisce robustezza contrastando le incertezze e calcolando i termini non lineari che dipendono dallo stato del manipolatore.

Ad alte frequenze però potrebbe esserci la commutazione della variabile di controllo, che potrebbe portare ad un'elevata presenza di "sali/scendi" a causa del terzo termine, per questo possiamo approssimarla come:

$$\omega = \begin{cases} \rho(||\xi||) \frac{z}{||z||} & \text{for } ||z|| \geq \varepsilon \\ \rho(||\xi||) \frac{z}{\varepsilon} & \text{for } ||z|| < \varepsilon \end{cases} \quad (7.33)$$

Possiamo andare quindi a rappresentare lo schema del controllo robusto teorico nel seguente modo: Data la difficoltà implementativa del controllore robusto sul modello reale è stata condotta un'analisi e simulazione del controllore sul modello simulink del manipolatore.

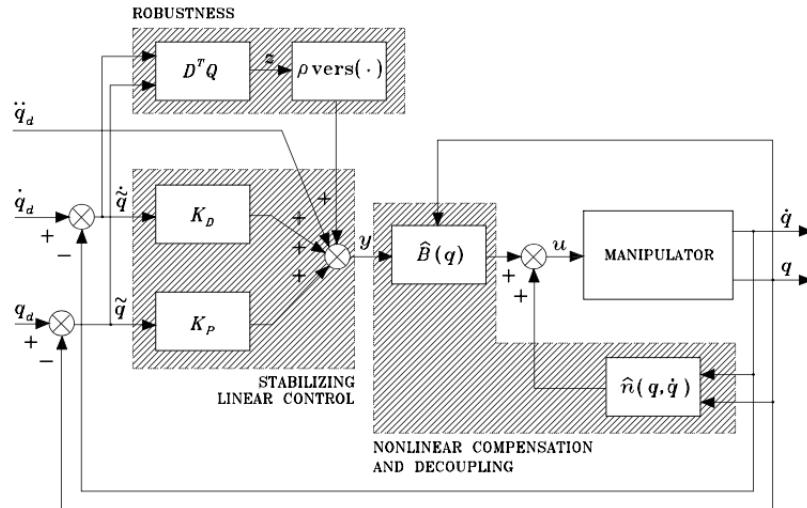


Figura 7.44: Schema controllore robusto

Implementazione controllore robusto

L'implementazione su simulink del controllore robusto consiste nel seguente schema:
Anche per questo approccio è stata condotta un'analisi su una traiettoria circolare

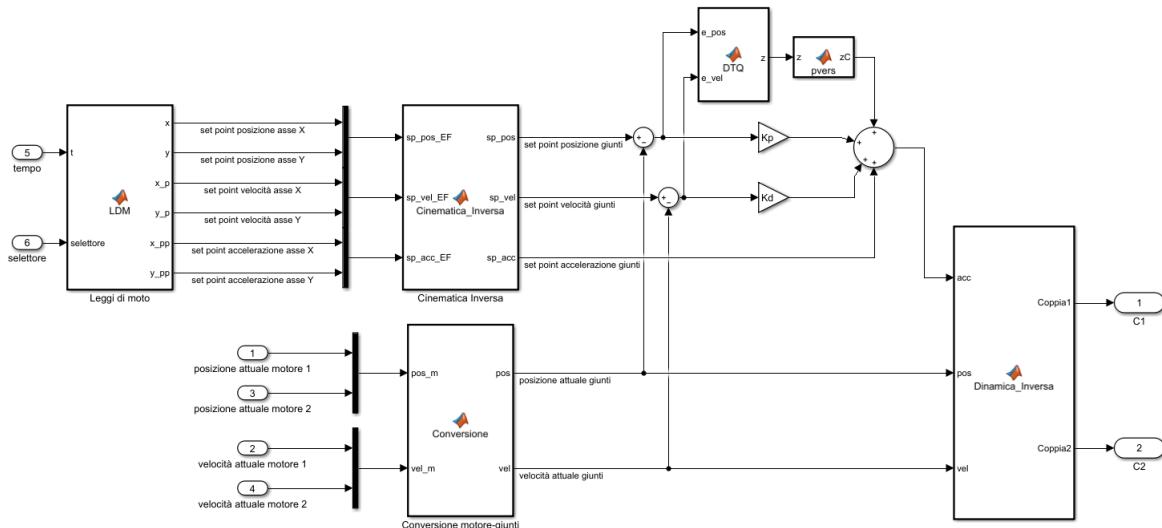


Figura 7.45: Controllore robusto

di raggio 5 cm da eseguire in 3 secondi. Di rilevanza in tutti questi test è stato il parametro ε , visto nell'equazione 7.33 presentiamo ora il risultato all'uscita del controllo robusto, quindi zV e le coppie di controllo ottenute al variare del parametro. Si può vedere che per valori di ε relativamente piccoli il valore di zV presenta alti

fenomeni vibratori causati dalla commutazione ad alta frequenza della variabile di controllo.

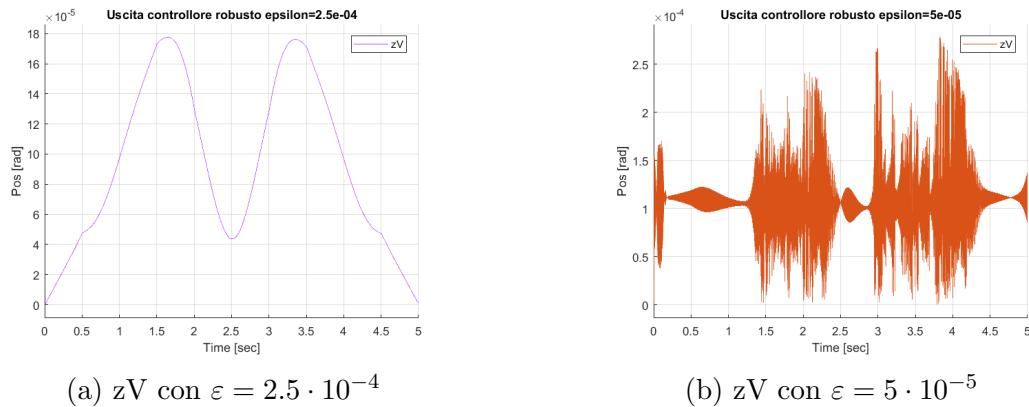


Figura 7.46: Uscite parte robusta del controllo

Per quanto riguarda le coppie di controllo, si può facilmente notare che nel primo caso non vibrano in quanto la variabile d'uscita dal controllo robusto non presenta vibrazioni, nel secondo caso invece le coppie in uscita sono vibrate e questo non le rende ideali per il controllo.

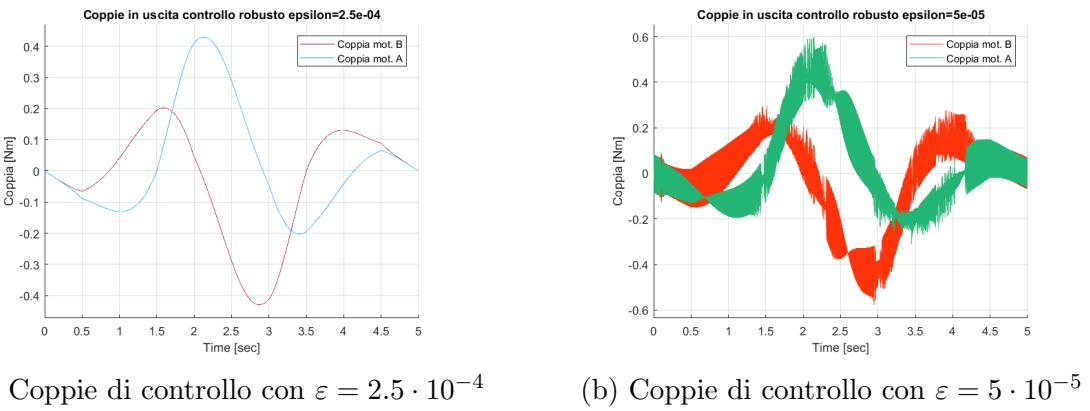


Figura 7.47: Coppie in uscita al variare di ϵ

A differenza dei controlli precedenti, l'unità di misura della coppia è il [Nm] in quanto questi test sono stati condotti esclusivamente su simulink.

7.6 Confronto approcci controllori

Avendo introdotto e verificato le principali tipologie di controllo centralizzato facciamo ora un confronto andando a vedere quale tra gli approcci analizzati è il migliore.

7.6 Confronto approcci controllori

In particolare, sulla traiettoria effettuata andiamo a confrontare l'andamento degli errori tra il controllore proporzionale derivativo e quello a dinamica inversa:

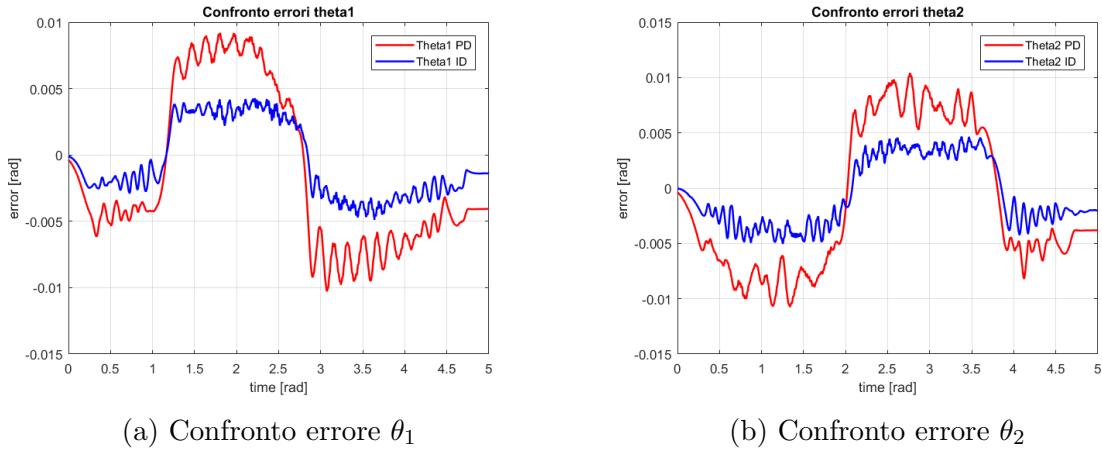


Figura 7.48: Confronto controllore PD e ID

Gli errori sono espressi in radianti, il picco di errore è circa due decimi di grado. Da entrambe le immagini è chiaro come il controllore a dinamica inversa risulti migliore. Di conseguenza si è scelto questo come controllore definitivo e sono stati analizzati altri andamenti con leggi di moto diverse al fine di valutarne il comportamento.

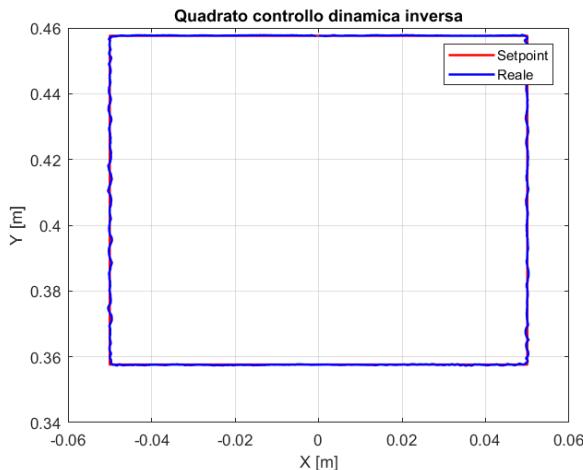


Figura 7.49: Quadrato controllo dinamica inversa

La prima traiettoria eseguita è stata quella del quadrato, nei lati orizzontali *setpoint* e curva reale sono pressoché uguali, in quelli verticali invece il manipolatore segue ma non alla perfezione, questo è dovuto alla contrapposizione dei motori in fase di lavoro.

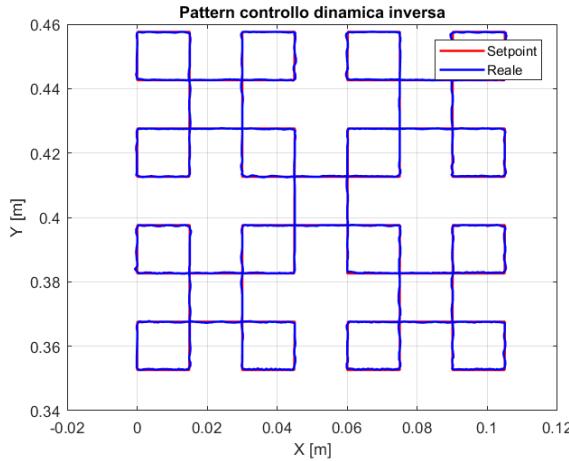


Figura 7.50: Pattern controllo dinamica inversa

La seconda figura è ancora una traiettoria in 2D ed è un pattern di quadrati nei quali il manipolatore non passa mai più di una volta per lo stesso punto, anche qui si può notare, come abbiamo visto in precedenza per il quadrato che i lati orizzontali sono eseguiti correttamente mentre in quelli verticali possono originarsi vibrazioni. Nella parte inferiore possiamo vedere più vibrazioni in quanto come visto [metti ref. a workspace] il numero di condizionamento è più alto e quindi ci si sta avvicinando ad una configurazione singolare.

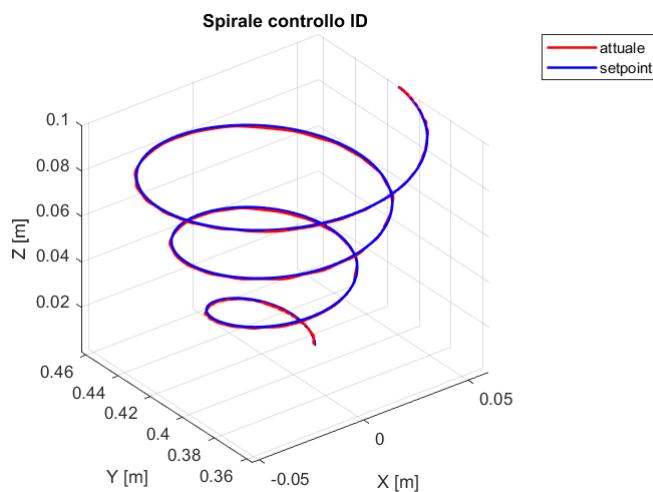


Figura 7.51: Spirale controllo dinamica inversa

Come ultima traiettoria si è scelto di eseguirne una tridimensionale: la spirale, in

questa è compresa la movimentazione sia delle braccia che della vite, è possibile selezionare quanti cerchi fare e di quanto far scendere la vite; si può vedere che il *setpoint* e la traiettoria reale sono molto simili.

7.7 Problema della velocità

Implementando e studiando le tecniche di controllo centralizzato, si può vedere chiaramente come in tutte queste abbiamo bisogno della velocità \dot{q} . Ci si è quindi concentrati su di essa, notando che la velocità ricavata dagli encoder soffriva di rumore, questa situazione poteva essere critica in quanto un parametro "rumoroso" da origine ad una legge di moto vibrante e di conseguenza non è possibile alzare il contributo K_d in maniera adeguata in quanto aumenterebbero le vibrazioni. Proprio per questo si è scelto di ricercare un metodo che potesse filtrare la velocità o predirla in modo da avere un risultato migliore da poter dare in ingresso al manipolatore.

Filtro primo ordine

L'idea iniziale è stata quella di implementare un filtro del primo ordine passa basso, con funzione di trasferimento:

$$T(s) = \frac{a_0}{s + \frac{b_0}{b_1}} = G_0 \frac{P_0}{s + P_0} \quad (7.34)$$

In questa tipologia di filtri c'è un polo che deve essere reale e negativo per garantire la stabilità (possiamo vedere questa condizione nell'appendice B.1). Andando ad analizzare in frequenza l'andamento del filtro $s = j\omega$ si trova che a basse frequenze l'input viene fatto passare completamente, mentre ad alte frequenze viene tagliato. Possiamo vedere in figura la velocità reale e due filtri. Il filtro con coefficiente 10^{-2} riesce ad approssimare meglio la curva ma è chiaramente in ritardo, invece, il filtro con coefficiente 10^{-3} è molto simile alla curva e non riesce a rimuovere le vibrazioni che essa porta.

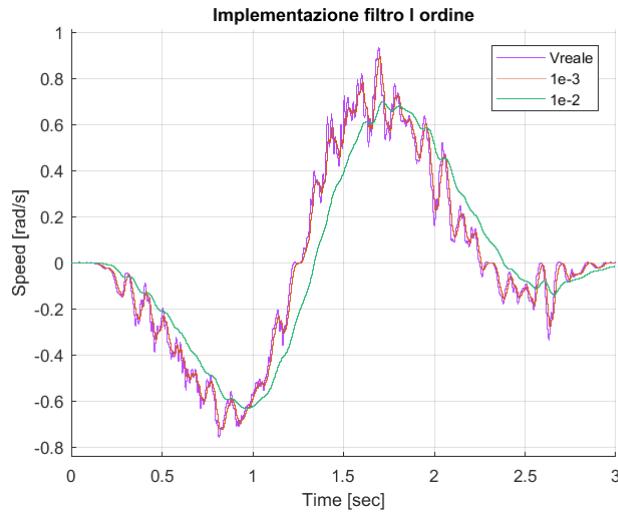


Figura 7.52: Filtro primo ordine

Filtro secondo ordine

Un filtro del secondo ordine si differenzia dal primo aggiungendo un polo, permette quindi di arrivare ad avere una pendenza del diagramma di Bode del modulo fino a $40 \frac{dB}{dec}$ nella banda attenuata, mentre in banda passante a seconda dell'approssimazione applicata possono verificarsi vari casi. Dal punto di vista pratico si realizzano mettendo più filtri del primo ordine in serie tra di loro.

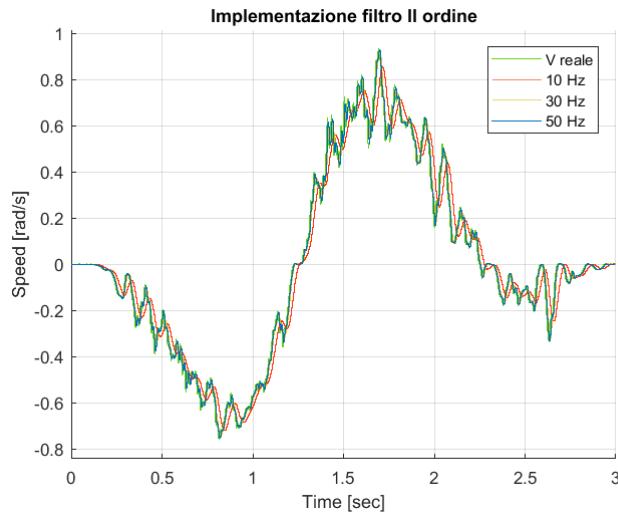


Figura 7.53: Filtro secondo ordine

In figura possiamo vedere la velocità reale e tre filtri, anche in questo caso nessuno

dei filtri riesce a fornire un output della velocità adeguato.

Filtro di Kalman

Il filtro di Kalman è un efficiente filtro ricorsivo che valuta lo stato di un sistema dinamico a partire da una serie di misure soggette a rumore, gli approcci implementativi del filtro sono stati due, il primo metodo consisteva nell'approccio al problema mediante tecniche di *sensor fusion*, andando a ricercare i valori delle matrici di covarianza in modo sperimentale e valutando il comportamento del filtro rispetto alla velocità originale; il secondo approccio è stato l'utilizzo del filtro di kalman esteso implementandolo in maniera più teorica.

Approccio sensor fusion

Per l'approccio *sensor fusion* è stato implementato un codice che permette di andare a trovare i valori migliori delle matrici di covarianza in base. Andiamo prima a definire le matrici che ci serviranno per la trattazione e poi riportare ora i due algoritmi fondamentali:

$$A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ T \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \end{bmatrix}, P_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (7.35)$$

con $T = 0.001\text{sec}$. Le variabili incognite che andremo a ricavare sono:

$$R, Q = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} \quad (7.36)$$

Algorithm 1 Creazione del filtro

```

for dof  $\leftarrow 1 : 1000$                                  $\triangleright$  Costruzione filtro con 1000 punti
     $S \leftarrow C \cdot P_m \cdot C^T + R$ 
     $K \leftarrow P_m \cdot C' \cdot S^{-1}$ 
     $P_p \leftarrow P_m - K \cdot C \cdot P_m$ 
     $P_m = A \cdot P_p \cdot A^T + Q$                            $\triangleright$  Passo di predizione
end for

```

Una volta definito il filtro il passo successivo è quello di avviarlo, per far questo

andiamo a far ciclare i valori di Q_1, Q_2 ed R in modo da andare a trovare i migliori:
Alla fine del ciclo otteniamo i valori di [i,j,k] che ci permettono di trovare le matrici

Algorithm 2 Avvio filtro

```

Require:  $Q_1, Q_2, R, Y, \dot{Y}_{rif}$ 
for  $i \leftarrow 1 : N$  do
    for  $j \leftarrow 1 : N$  do
         $Q \leftarrow [Q_1(i), 0, 0, Q_2(j)]$ 
        for  $k \leftarrow 1 : N$  do
             $r \leftarrow R(k)$ 
             $P_m = P_0$ 
             $em \leftarrow [0, 0]^T$ 
            for  $dog \leftarrow 1 : length(Y)$ 
                 $ep \leftarrow em + K \cdot (Y(k) - C * em)$ 
                 $[pos, vel] \leftarrow [sep(1), sep(2)]$ 
                 $em \leftarrow A \cdot ep + B \cdot Y(k)$ 
                 $err(k) \leftarrow \dot{Y}_{rif} - vel$ 
            end for
            for  $dok \leftarrow 1 : length(err)$ 
                if  $\text{thenmean}(abs(err(k))) < my$ 
                     $my = \text{mean}(abs(err(k)))$ 
                     $\text{Save}[i, j, k]$ 
                end if
            end for
        end for
    end for
end for

```

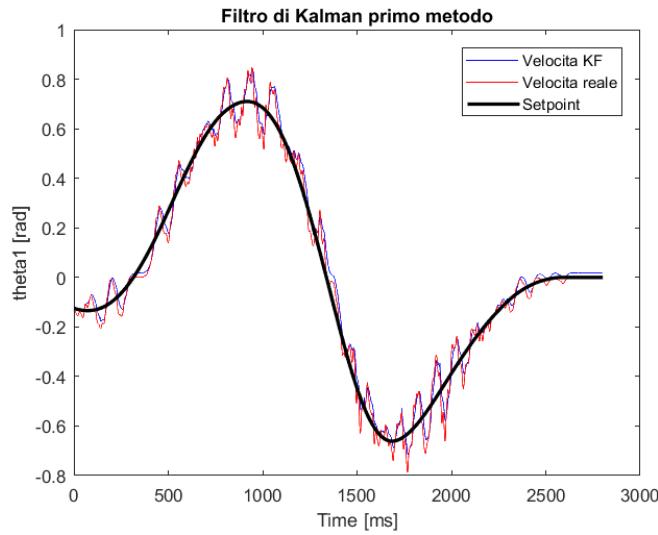
Q ed R a minimo errore tra riferimento e stima. Grazie alla formula:

$$\frac{|\dot{y}_{rif} - \dot{y}|}{N} \quad (7.37)$$

Per completezza sono stati condotti anche dei test scegliendo come risultato migliore la varianza minore:

$$\sum_{i=0}^N \frac{[(\dot{y}_{rif} - \dot{y}) - \mu_y]^2}{N} \quad (7.38)$$

Si è visto però che il risultato finale è identico.


 Figura 7.54: Kalman *sensor fusion*

Filtro di Kalman esteso

Il filtro di Kalman esteso Extended Kalman Filter (EKF) è una versione non-lineare del filtro di Kalman usata quando l'evoluzione o l'osservazione dello stato del sistema sono non-lineari, un sistema a tempo discreto, non lineare può essere descritto come:

$$\begin{cases} \xi_{k+1} = f(\xi_k, u_k) \\ y_k = h(\xi_k) \end{cases} \quad (7.39)$$

Per definire il filtro di Kalman partiamo dalla definizione dello stato come

$$\xi = \begin{bmatrix} \dot{\theta} \\ \theta \end{bmatrix} \quad (7.40)$$

Definiamo ora la derivata dello stato come:

$$\dot{\xi} = \begin{bmatrix} \ddot{\theta} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & -M^{-1}C \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix} \tau_m \quad (7.41)$$

Possiamo esprimere questa equazione come:

$$\dot{\xi} = \varphi(\xi, u)$$

Con u ingresso corrispondente alle coppie. Il filtro, a partire dal valore attuale ci consente di predire il valore successivo come:

$$\xi_{k+1} - \xi_k = \varphi(\xi_k, u_k) \Delta t \rightarrow \xi_{k+1} = \xi_k + \varphi(\xi_k, u_k) \Delta t = f(\xi_k, u_k) \quad (7.42)$$

L'uscita y_k sarà data da:

$$y_k = \mathbb{F}\xi_k \quad (7.43)$$

Dove:

$$\mathbb{F} = \frac{\partial f}{\partial \xi} \Bigg|_{\hat{\xi}_k, u_k} = \begin{bmatrix} \frac{\partial f}{\partial \xi_1} & \frac{\partial f}{\partial \xi_2} & \frac{\partial f}{\partial \xi_3} & \frac{\partial f}{\partial \xi_4} \end{bmatrix}$$

Sfruttando il metodo delle differenze finite(B.2) possiamo trovare:

$$\frac{\partial}{\partial \xi_d} f(\xi, u) \approx \frac{f(\xi + \Delta \xi \delta_d) - f(\xi - \Delta \xi \delta_d)}{2 \Delta \xi} \quad (7.44)$$

Con δ_d definito come la colonna della matrice identità calcolata col metodo di Sarrus.

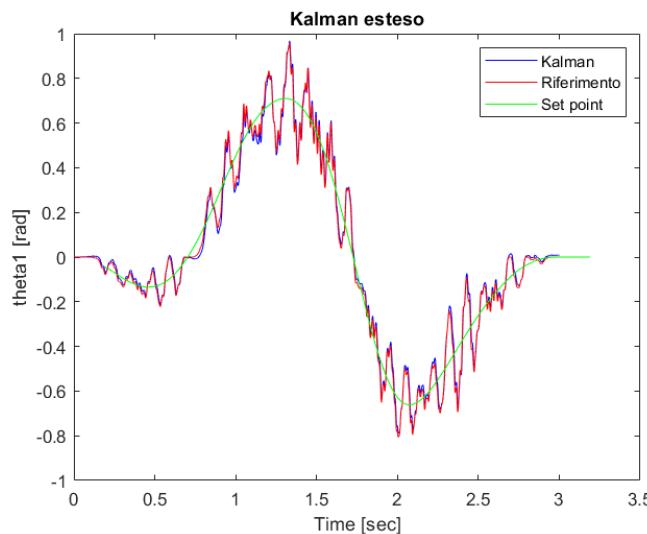


Figura 7.55: Filtro di Kalman esteso

8 Conclusioni

L’obiettivo principale della tesi è stato lo studio e l’analisi di approcci di controllo centralizzato al fine di trovare quello migliore da implementare nel manipolatore. Dopo la trattazione sulla cinematica, dinamica e cinetostatica del sistema è stato possibile definire le relazioni matematiche che caratterizzano le due parti del manipolatore, braccia e vite, il workspace ed i possibili punti raggiungibili implementando quanto visto su un modello simulato. Il modello una volta validato grazie a processi di cosimulazione (Matlab e Adams) ha fornito una base per l’implementazione degli schemi di controllo.

L’implementazione di differenti tecniche ha permesso di ottenere una maggior conoscenza nell’ambito della manipolazione del 5R e la fase di simulazione ha permesso di definire quale tra le strategie analizzate fosse la più consona, andando a scartare l’impiego di altri criteri; uno in particolare è stato il controllo robusto, influenzato da fenomeni vibratori generati dalla commutazione ad alta frequenza della variabile di controllo.

Per concludere, la validazione sperimentale ha permesso di evidenziare che il miglior criterio di controllo è quello a dinamica inversa, in quanto si riesce ad avere un *trade-off* tra prestazioni e robustezza. L’ultima fase è stata quella di test, implementando traiettorie bidimensionali e tridimensionali ed analizzando il comportamento del controllo.

Sviluppi futuri

Gli obiettivi futuri legati al progetto di tesi riguardano nuovi criteri di controllo, andando ad analizzare tipologie di controllo diverse che potrebbero essere implementate ed dare risultati migliori di quello a dinamica inversa. Un’altra tematica riguarda il telecontrollo del manipolatore mediante interfaccia aptica (che assumerà il ruolo di master).

A Formule estese

In riferimento al capitolo della cinematica e della dinamica andiamo ora a vedere le formule per esteso:

A.1 Formule cinematica diretta accelerazione

Andiamo a concentrarci sulle formule viste nel capitolo della cinematica diretta di accelerazione. [metti riferimento]

$$\dot{J}_{11} = A_{11} - B_{11} - C_{11} - D_{11} \quad (\text{A.1})$$

Andiamo ora a definire i parametri:

$$\begin{aligned}
A_{11} &= \frac{l(y - l \sin \theta_2)(d \sin \theta_1 - y \cos \theta_1 + x \sin \theta_1) \cdot (\dot{y} - l \dot{\theta}_1 \cos \theta_1 + A_{1.1.2})}{(d - x + l \cos \theta_2)(y - l \sin \theta_1 + \frac{(y - l \sin \theta_2)(d + x - l \cos \theta_1)}{d - x + l \cos \theta_2})^2} \\
A_{1.1.2} &= \frac{(y - l \sin \theta_2)(\dot{x} + l \dot{\theta}_1 \sin \theta_1)}{d - x + l \cos \theta_2} + \frac{\dot{y} - l \dot{\theta}_2 \cos \theta_2(d + x - l \cos \theta_1)}{d - x + l \cos \theta_2} + \\
&\quad + \frac{(y - l \sin \theta_2)(\dot{x} + l \dot{\theta}_2 \sin \theta_2)(d + x - l \cos \theta_1)}{(d - x + l \cos \theta_2)^2} \\
B_{11} &= \frac{l(y - l \sin \theta_2)(\dot{x} \sin \theta_1 - \dot{y} \cos \theta_1 + d \dot{\theta}_1 \cos \theta_1 + \dot{\theta}_1 x \cos \theta_1 + \dot{\theta}_1 y \sin \theta_1)}{(d - x + l \cos \theta_2) \cdot (y - l \sin \theta_1 + \frac{(y - l \sin \theta_2)(d + x - l \cos \theta_1)}{d - x + l \cos \theta_2})} \\
C_{11} &= \frac{l(\dot{y} - l \dot{\theta}_2 \cos \theta_2)(d \sin \theta_1 - y \cos \theta_1 + x \sin \theta_1)}{(d - x + l \cos \theta_2)(y - l \sin \theta_1 + \frac{(y - l \sin \theta_2)(d + x - l \cos \theta_1)}{d - x + l \cos \theta_2})} \\
D_{11} &= \frac{l(y - l \sin \theta_2)(\dot{x} + l \dot{\theta}_2 \sin \theta_2)(d \sin \theta_1 - y \cos \theta_1 + x \sin \theta_1)}{(d - x + l \cos \theta_2)^2(y - l \sin \theta_1 + \frac{(y - l \sin \theta_2)(d + x - l \cos \theta_1)}{d - x + l \cos \theta_2})} \\
\dot{J}_{12} &= \frac{A_{1.2}}{AD_{1.2}} + B_{1.2} + C_{1.2} + D_{1.2} + E_{1.2} + F_{1.2} \quad (\text{A.2}) \\
A_{1.2} &= (y - l \sin \theta_2)(l \sin \theta_2(\dot{x} + l \dot{\theta}_1 \sin \theta_1) + l \dot{\theta}_2 \cos \theta_2(d + x - l \cos \theta_1) + A_{1.2.1}) \\
A_{1.2.1} &= \frac{l \cos \theta_2(\dot{y} - l \dot{\theta}_2 \cos \theta_2)(d + x - l \cos \theta_1)}{d - x + l \cos \theta_2} + \frac{l \cos \theta_2(y - l \sin \theta_2)(\dot{x} + l \dot{\theta}_1 \sin \theta_1)}{d - x + l \cos \theta_2} - \\
&\quad - \frac{l \dot{\theta}_2 \sin \theta_2(y - l \sin \theta_2)(d + x - l \cos \theta_1)}{d - x + l \cos \theta_2} + A_{1.2.2}
\end{aligned}$$

$$\begin{aligned}
A_{1.2.2} &= \frac{l \cos \theta_2 (y - l \sin \theta_2) (\dot{x} + l \dot{\theta}_2 \sin \theta_2) (d + x - l \cos \theta_1)}{(d - x + l \cos \theta_2)^2} \\
AD_{1.2} &= (d - x + l \cos \theta_2) (y - l \sin \theta_1 + \frac{(y - l \sin \theta_2) (d + x - l \cos \theta_1)}{d - x + l \cos \theta_2}) \\
B_{1.2} &= -\frac{l \cos \theta_2 (\dot{y} - l \dot{\theta}_2 \cos \theta_2)}{d - x + l \cos \theta_2} - l \dot{\theta}_2 \cos \theta_2 + \frac{l \dot{\theta}_2 \sin \theta_2 (y - l \sin \theta_2)}{d - x + l \cos \theta_2} \\
C_{1.2} &= -\frac{l \cos \theta_2 (y - l \sin \theta_2) (\dot{x} + l \dot{\theta}_2 \sin \theta_2)}{(d - x + l \cos \theta_2)^2} \\
D_{1.2} &= \frac{l (\dot{y} - l \dot{\theta}_2 \cos \theta_2) (d \sin \theta_2 + y \cos \theta_2 - x \sin \theta_2) (d + x - l \cos \theta_1)}{(d - x + l \cos \theta_2)^2 (y - l \sin \theta_1 + \frac{(y - l \sin \theta_2) (d + x - l \cos \theta_1)}{d - x + l \cos \theta_2})} \\
E_{1.2} &= -\frac{l (y - l \sin \theta_2) (d \sin \theta_2 + y \cos \theta_2 - x \sin \theta_2) (d + x - l \cos \theta_1) (E_{1.2.1} + E_{1.2.2})^2}{ED_{1.2}} \\
E_{1.2.1} &= \dot{y} - l \dot{\theta}_1 \cos \theta_1 + \frac{(y - l \sin \theta_2) (\dot{x} + l \dot{\theta}_1 \sin \theta_1)}{d - x + l \cos \theta_2} \\
E_{1.2.2} &= \frac{(\dot{y} - l \dot{\theta}_2 \cos \theta_2) (d + x - l \cos \theta_1)}{d - x + l \cos \theta_2} + \frac{(y - l \sin \theta_2) (\dot{x} + l \dot{\theta}_2 \sin \theta_2) (d + x - l \cos \theta_1)}{(d - x + l \cos \theta_2)} \\
ED_{1.2} &= (d - x + l \cos \theta_2)^2 (y - l \sin \theta_1 + \frac{(y - l \sin \theta_2) (d + x - l \cos \theta_1)}{d - x + l \cos \theta_2})^2 \\
F_{1.2} &= \frac{l (y - l \sin \theta_2) (\dot{x} + l \dot{\theta}_2 \sin \theta_2) (d \sin \theta_2 + y \cos \theta_2 - x \sin \theta_2) (d + x - l \cos \theta_1)}{(d - x + l \cos \theta_2)^3 (y - l \sin \theta_1 + \frac{(y - l \sin \theta_2) (d + x - l \cos \theta_1)}{d - x + l \cos \theta_2})} \\
J_{21} &= \frac{l (d \sin \theta_1 - y \cos \theta_1 + x \sin \theta_1) (\dot{y} - l \dot{\theta}_1 \cos \theta_1 + A_{1.1.2})}{(y - l \sin \theta_1 + \frac{(y - l \sin \theta_2) (d + x - l \cos \theta_1)}{d - x + l \cos \theta_2})^2} + B_{2.1.1} \quad (A.3) \\
B_{2.1.1} &= -\frac{l (\dot{x} \sin \theta_1 - \dot{y} \cos \theta_1 + d \dot{\theta}_1 \cos \theta_1 + \dot{\theta}_1 x \cos \theta_1 + \dot{\theta}_1 y \sin \theta_1)}{y - l \sin \theta_1 + \frac{(y - l \sin \theta_2) (d + x - l \cos \theta_1)}{d - x + l \cos \theta_2}} \\
J_{22} &= \frac{l \sin \theta_2 (\dot{x} + l \dot{\theta}_1 \sin \theta_1) + l \dot{\theta}_2 \cos \theta_2 (d + x - l \cos \theta_1) + A_{2.2.1} + A_{2.2.2} + A_{2.2.3}}{y - l \sin \theta_1 + \frac{(y - l \sin \theta_2) (d + x - l \cos \theta_1)}{d - x + l \cos \theta_2}} \quad (A.4) \\
A_{2.2.1} &= \frac{l \cos \theta_2 (y - l \sin \theta_2) (\dot{x} + l \dot{\theta}_1 \sin \theta_1)}{d - x + l \cos \theta_2} - \frac{l \dot{\theta}_2 \sin \theta_2 (y - l \sin \theta_2) (d + x - l \cos \theta_1)}{d - x + l \cos \theta_2} \\
A_{2.2.2} &= \frac{l \cos \theta_2 (\dot{y} - l \dot{\theta}_2 \cos \theta_2) (d + x - l \cos \theta_1)}{d - x + l \cos \theta_2} \\
A_{2.2.3} &= \frac{l \cos \theta_2 (y - l \sin \theta_2) (\dot{x} + l \dot{\theta}_2 \sin \theta_2) (d + x - l \cos \theta_1)}{(d - x + l \cos \theta_2)^2}
\end{aligned}$$

$$B_{2.2.1} = -\frac{l(d \sin \theta_2 + y \cos \theta_2 - x \sin \theta_2)(d + x - l \cos \theta_1) + B_{2.2.2}}{(d - x + l \cos \theta_2)(y - l \sin \theta_1 + \frac{(y - l \sin \theta_2)(d + x - l \cos \theta_1)}{d - x + l \cos \theta_2})^2}$$

$$B_{2.2.2} = \left(\dot{y} - l \dot{\theta}_1 \cos \theta_1 + \frac{(y - l \sin \theta_2)(\dot{x} + l \dot{\theta}_1 \sin \theta_1)}{d - x + l \cos \theta_2} + \frac{(\dot{y} - l \dot{\theta}_2 \cos \theta_2)(d + x - l \cos \theta_1)}{d - x + l \cos \theta_2} + \right.$$

$$\left. \frac{(y - l \sin \theta_2)(\dot{x} + l \dot{\theta}_2 \sin \theta_2)(d + x - l \cos \theta_1)}{(d - x + l \cos \theta_2)^2} \right)$$

A.2 Prerequisiti dinamica

Le matrici importati sono \dot{J}_3 e \dot{J}_4 che andiamo a riprendere ora come:

$$\dot{J}_3 = \begin{bmatrix} \dot{J}_{31} & \dot{J}_{32} \\ \dot{J}_{33} & \dot{J}_{34} \end{bmatrix} \quad (\text{A.5})$$

Con i relativi parametri pari a:

$$\begin{aligned} \dot{J}_{31} &= -l \cos \theta_1 \dot{\theta}_1 + \frac{1}{2} \cos \theta_3 \dot{\theta}_3 J_{34} + \frac{1}{2} \sin \theta_3 \dot{J}_{34} \\ \dot{J}_{32} &= -l \frac{1}{2} \cos \theta_3 \dot{\theta}_3 J_{34} + \frac{1}{2} \sin \theta_3 \dot{J}_{34} \\ \dot{J}_{33} &= -l \sin \theta_1 \dot{\theta}_1 - \frac{1}{2} \sin \theta_3 \dot{\theta}_3 J_{34} + \frac{1}{2} \cos \theta_3 \dot{J}_{34} \\ \dot{J}_{34} &= -\frac{1}{2} l \sin \theta_3 \dot{\theta}_3 J_{34} + \frac{1}{2} \cos \theta_3 \dot{J}_{34} \end{aligned}$$

Invece, definiamo \dot{J}_4 in modo analogo come:

$$\begin{bmatrix} \dot{J}_{41} & \dot{J}_{42} \\ \dot{J}_{43} & \dot{J}_{44} \end{bmatrix} \quad (\text{A.6})$$

Con parametri:

$$\begin{aligned} \dot{J}_{41} &= -l \frac{1}{2} \cos \theta_4 \dot{\theta}_4 J_{34} + \frac{1}{2} \sin \theta_4 \dot{J}_{34} \\ \dot{J}_{42} &= -l \cos \theta_2 \dot{\theta}_2 + \frac{1}{2} \cos \theta_4 \dot{\theta}_4 J_{34} + \frac{1}{2} \sin \theta_4 \dot{J}_{34} \\ \dot{J}_{43} &= -l \frac{1}{2} \sin \theta_4 \dot{\theta}_4 J_{34} + \frac{1}{2} \cos \theta_4 \dot{J}_{34} \\ \dot{J}_{44} &= -l \sin \theta_2 \dot{\theta}_2 - \frac{1}{2} \sin \theta_4 \dot{\theta}_4 J_{34} + \frac{1}{2} \cos \theta_4 \dot{J}_{34} \end{aligned}$$

B Cenni sistemi dinamici

In questa sezione dell'appendice andremo a considerare un sistema non lineare come quello del manipolatore e analizzeremo la stabilità dell'equilibrio a tempo discreto. Dato un sistema dinamico, a dimensione finita, stazionario, di tipologia MIMO (*multiple input multiple output*), non lineare e a tempo discreto possiamo descriverlo mediante l'equazione:

$$\xi(k+1) = f(\xi(k), u(k)) \quad (\text{B.1})$$

Il sistema può andare ad evolversi in due stati:

- Movimento d'equilibrio, chiamato anche movimento nominale: $\tilde{\xi}(k) = \bar{\xi}(k)$, si ottiene applicando l'ingresso nominale di equilibrio $\tilde{u}(k) = \bar{u}(k)$ al sistema posto nello stato iniziale $\tilde{\xi}(k_0 = 0) = \bar{\xi}$ in questo modo il movimento soddisfa

$$\tilde{\xi}(k+1) = \bar{\xi} = f(\bar{\xi}, \bar{u}) \quad (\text{B.2})$$

- Ingresso perturbato $\xi(k)$ ottenuto applicando la stessa tipologia di ingresso nominale però in uno stato differente $\xi_0 \neq \bar{\xi}$, allora il movimento soddisfa il seguente sistema di equazioni:

$$\begin{cases} \xi(k+1) = f(\xi(k), u(k)) \\ \xi(k_0 = 0) = \xi_0 \end{cases} \quad (\text{B.3})$$

Possiamo andare a definire la perturbazione sullo stato del sistema come la differenza fra i due movimenti in particolare possiamo esprimere la come:

$$\delta\xi(k) = \xi(k) - \bar{\xi} \in \mathbb{R}^n \quad (\text{B.4})$$

procedendo in maniera incrementale troviamo:

$$\delta\xi(k+1) = \xi(k+1) - \bar{\xi} = f(\xi(k), \bar{u}) - \bar{\xi} = f(\bar{\xi} + \delta\xi(k), \bar{u}) - \bar{\xi} \quad (\text{B.5})$$

risulta essere non lineare, e possiamo trovare la condizione iniziale come:

$$\delta\xi(k_0 = 0) = \xi_0 - \bar{\xi} \rightarrow \delta\xi \neq 0$$

La soluzione all'equazione non è facile da trovare in quanto dipende da un punto preciso chiamato punto di equilibrio, identificato da $(\bar{\xi}, \bar{u})$; in caso di sistemi non lineari stazionari, la proprietà di stabilità riguarda esclusivamente un intorno dello stato di equilibrio considerato e non l'intero sistema.

B.1 Stabilità asintotica

A partire dallo stato di equilibrio ξ del sistema dinamico a tempo discreto visto precedentemente, la condizione sufficiente perché risulti asintoticamente stabile è che

$$\forall i : |\lambda_i(A)| < 1 \quad (\text{B.6})$$

$$\text{Con } A = \left. \frac{\partial f(\xi, \bar{u})}{\partial \xi} \right|_{\xi=\bar{\xi}}$$

In questo caso esiste un intorno dell'equilibrio $(\Psi_{\bar{\xi}})$ tale per cui, data qualunque perturbazione iniziale $\delta\xi(k) \in \Psi_{\bar{\xi}}$ la perturbazione sullo stato $\delta\xi(k)$ rimane limitata nel tempo ed asintoticamente tende a zero.

$$\delta\xi(k+1) = f(\bar{\xi} + \delta\xi(k), \bar{u}) = A\delta\xi(k) + h(\delta\xi(k)) \quad (\text{B.7})$$

Da tutto questo possiamo derivare anche la condizione di instabilità, esprimibile come:

$$\exists i : |\lambda_i(A)| > 1 \quad (\text{B.8})$$

con A definita come prima. In questo caso, non esiste alcun intorno dell'equilibrio che fa rimanere la perturbazione dello stato limitata a partire da una perturbazione iniziale $\delta\xi_0 \in \Psi_{\bar{\xi}}$.

B.2 Metodo delle differenze finite

In caso di equazioni differenziali ordinarie l'interesse è quello di risolvere in un preciso dominio, in particolare, per poterle risolvere numericamente dobbiamo imporre la condizione che il dominio sia limitato. Le differenze finite vengono utilizzate per risolvere equazioni differenziali che descrivono fenomeni stazionari, ovvero quando le quantità di riferimento variano nello spazio (ipotizzando che non varino nel tempo). La trattazione seguente utilizzerà come esempio l'equazione di Poisson

$$-\frac{d^2u(\gamma)}{d\gamma^2} = f(\gamma) \quad (\text{B.9})$$

con $\gamma \in [0, R]$ e $f : \mathbb{R} \rightarrow \mathbb{R}$ una funzione con condizioni $u(0) = a, u(R) = b$; per poterla risolvere dobbiamo garantire un minimo di regolarità per la funzione f . La prima idea che applichiamo è la discretizzazione del dominio, ovvero dividiamo l'intervallo $[0, R]$ in intervalli sufficientemente piccoli la cui unione restituisce il dominio iniziale. Definiamo quindi la discretizzazione come:

$$\Gamma = \{\gamma_1 = 0 < \gamma_2 < \dots < \gamma_{N-1} < \gamma_N = R\} \quad (\text{B.10})$$

tale che $\cup_{i=1}^{N-1} [\gamma_i, \gamma_{i+1}] = [0, R]$. Essendo la discretizzazione costruita con distanze uguali chiamiamo $\Delta\gamma_k = \gamma_{k+1} - \gamma_k$. Ricordandoci il concetto di rapporto incrementale per la definizione della derivata prima, sapendo che la derivata di una funzione è definita come il limite del rapporto incrementale in caso che sia finito, ne consegue che possiamo definire la derivata seconda come:

$$u''(\gamma) = \lim_{\Delta\gamma \rightarrow 0} \frac{u'(\gamma + \Delta\gamma) - u'(\gamma)}{\Delta\gamma}$$

Introduciamo quindi $u(\gamma_k) \approx u_k$, cioè, quello che noi andiamo a calcolare è l'approssimazione della soluzione calcolata in γ_k . Per fornire un'approssimazione alle differenze finite di queste derivate utilizziamo l'idea di trovare un intervallo di discretizzazione piccolo tale per cui prendendo due elementi γ_k e γ_{k+1} distano sufficientemente

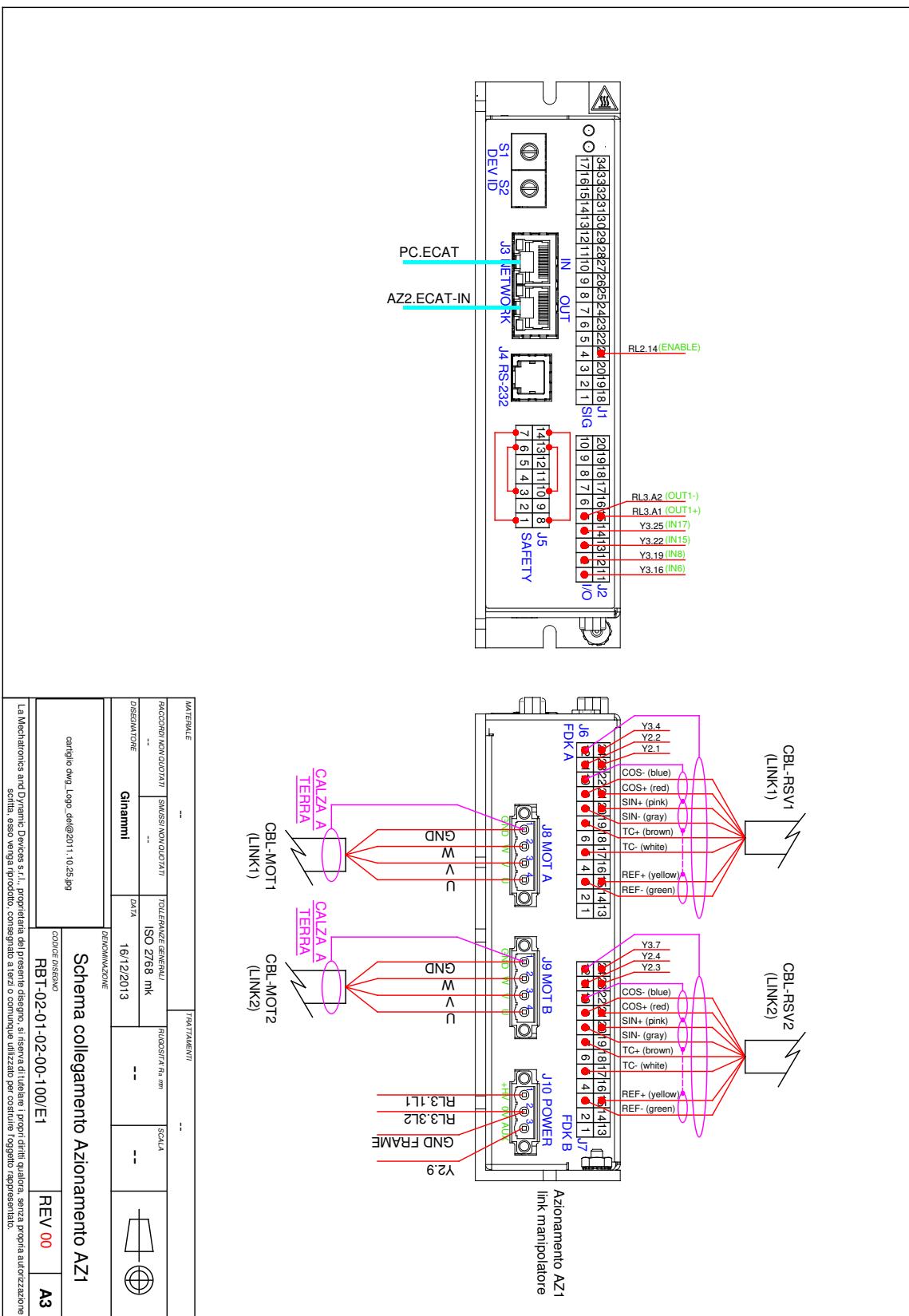
poco. Possiamo quindi approssimare la derivata come:

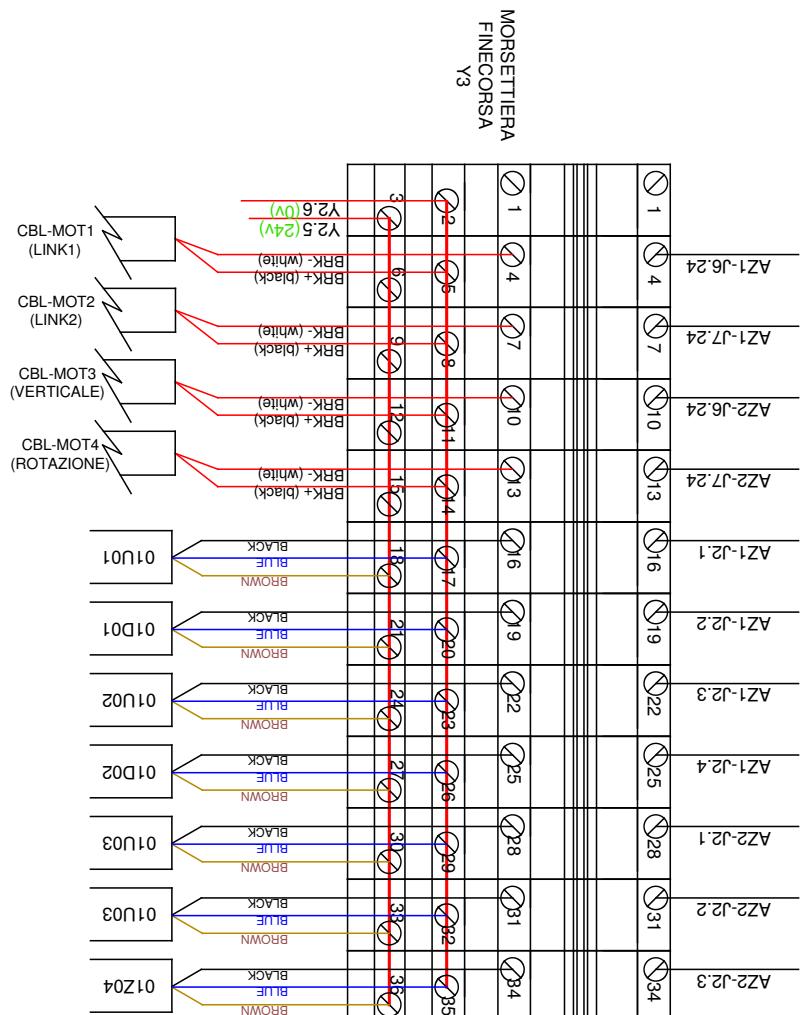
$$u'(\gamma_k) \approx \frac{u_{k+1} + u_k}{\Delta\gamma} \quad (\text{B.11})$$

L'approssimazione ottenuta però non è molto raffinata, si può ottenere un'espressione migliore, del secondo ordine effettuando una differenza finita centrata ottenendo quindi:

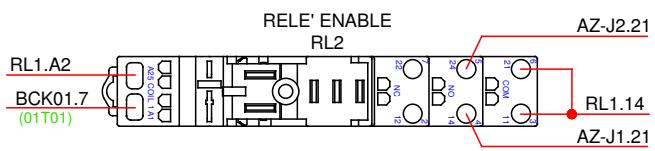
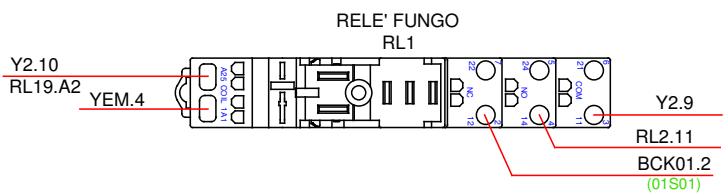
$$u'(\gamma_k) \approx \frac{u_{k+1} - u_{k-1}}{2\Delta\gamma} \quad (\text{B.12})$$

C Quadro elettrico

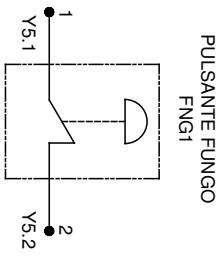
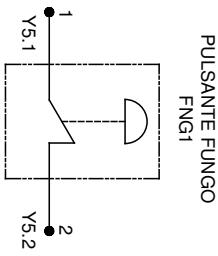
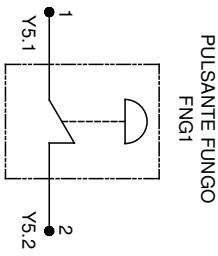
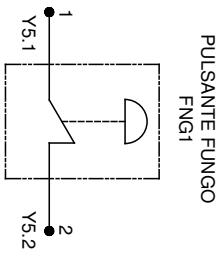
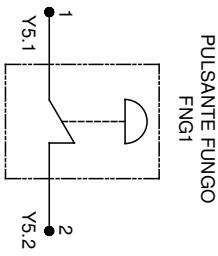




MATERIALE		TRATTAMENTI		---	
PAZZOCCHI NON QUADRATI	SAUSSI NON QUADRATI	TOLLENZEE GENERALI	RUGOSITÀ (Ra mm)	SCALA	---
--	--	ISO 2768 mR	--	--	
DISEGNATORE	Gianmatti	DATA			
		16/12/2013			
DENOMINAZIONE		Schema collegamento MORSETTERA Y3			
carlo.dmg_Logo_delt@2011.1.025.jpg					
COPPIE INSEGNAZO					
RBT-02-01-02-00-100/E10					
		REV 00		A3	
La Mechatronics and Dynamic Devices s.r.l., proprietario del presente disegno, si riserva di tutelare i propri diritti quadra, senza propria autorizzazione scritta, esso venga riprodotto, consegnato a terzi o comunicato all'utilizzo per corso di studio regolare rappresentato.					



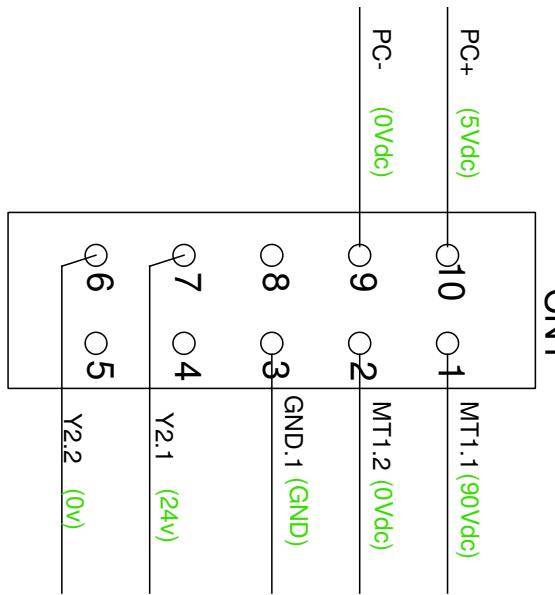
MATERIALE		TRATTAMENTI	
RACCORDI NON QUOTATI	SMASSI NON QUOTATI	TOLLEFANZE GENERALI	RUGOSITÀ Ra mm
--	--	ISO 2768 mk	--
DISEGNATORE		DATA	SCALA
Gianmimi			16/12/2013
DENOMINAZIONE			
carriaggio dwg_Logo_delt@2011.10.25.jpg			
Schema collegamento RELe		REV 00	A3
RBT-02-01-02-00-100/E11		CODICE DISEGNO	
La Mechanic's and Dynamic Devices s.r.l. propria di questo disegno, si riserva i propri diritti qualora, senza propria autorizzazione scritta, esso venga riprodotto, conegnato a terzi o comunque utilizzato per costituire oggetto rappresentato.			



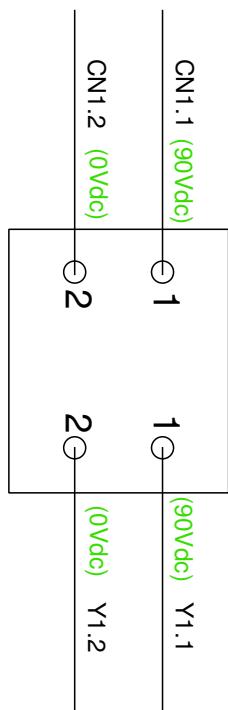
MATERIALE		TRATTAMENTI	
--	--	--	--
RACCORDI NON QUOTATI	SMASSI NON QUOTATI	TOLLEFANZE GENERALI	
--	--	ISO 2768 mk	
DISEGNATORE	Gianmimi	DATA	16/12/2013
DENOMINAZIONE			
carrello dwg_Logo_dal@2011.10.25.jpg			
Schema collegamento ACCESSORI		REV 00	A3
RBT-02-01-02-00-100/E12		CODICE DISEGNO	

La Mechanic's and Dynamic Devices s.r.l., proprietaria del presente disegno, si riserva di tutelare i propri diritti qualora, senza propria autorizzazione scritta, esso venga riprodotto, conegnato a terzi o comunque utilizzato per costituire oggetto rappresentato.

**CONNETTORE
HARTING 10POLI
CN1**



**MAGNETOTERMICO
MT1**



MATERIALE

--

RACCORDI NON QUOTATI

--

SMASSI NON QUOTATI

--

TOLLEFANZE GENERALI

--

RUGOSITÀ Ra mm

--

SCALA

--

DISEGNATORE

--

DATA

--

ISO 2768 mk

--

GTIN

--

Schema collegamento CONNETTORE

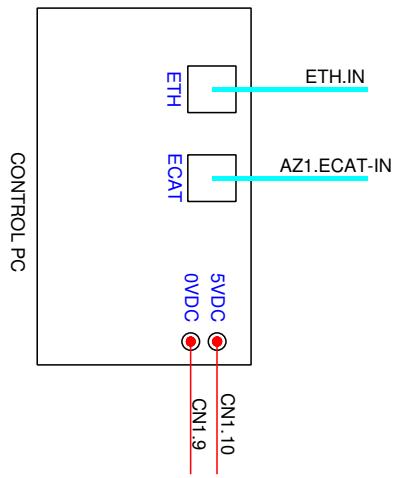
codice disegno

RBT-02-01-02-00-100/E13

REV 00

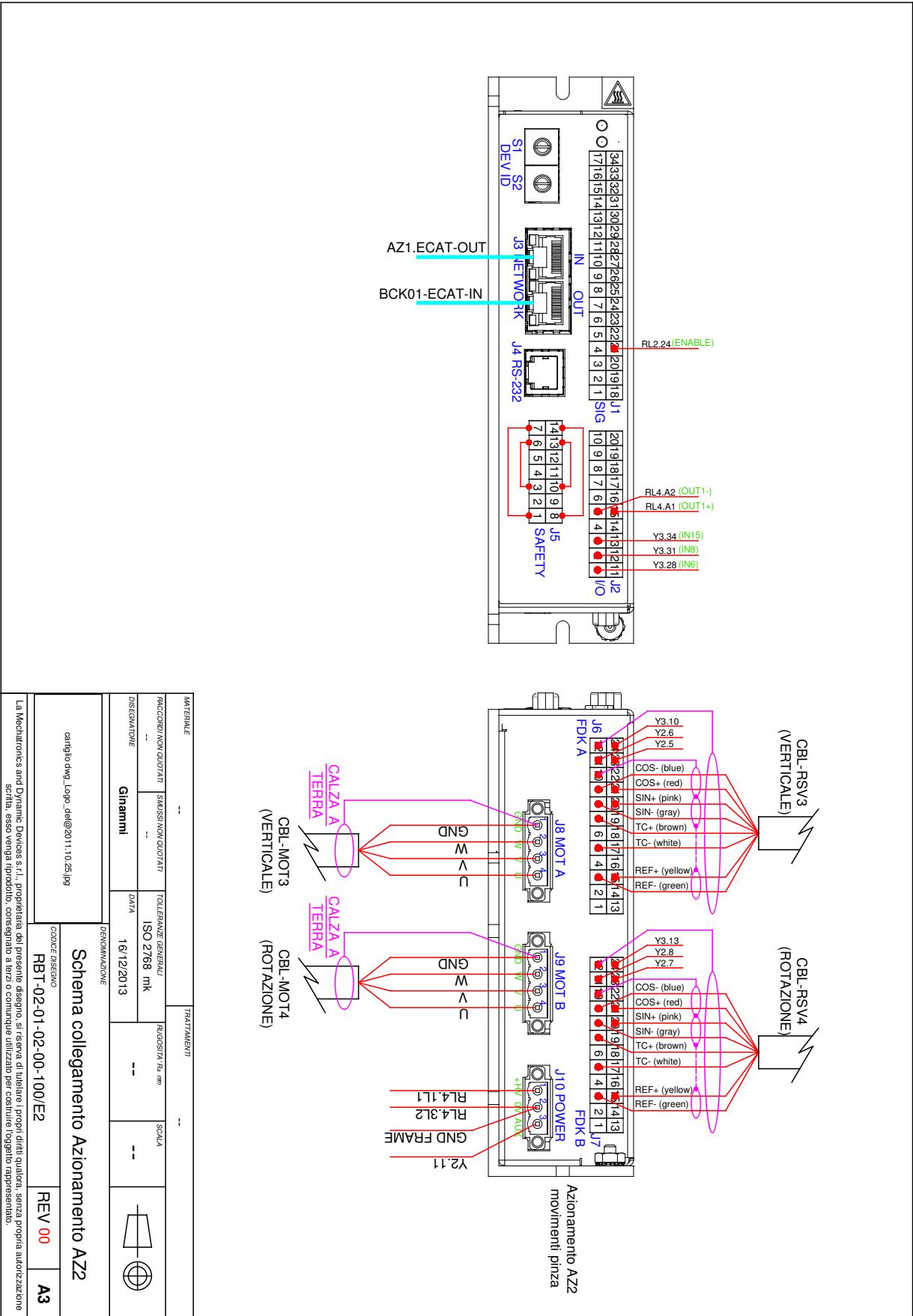
A3

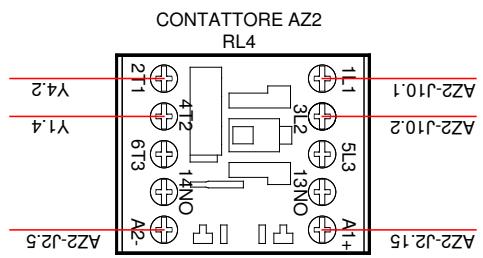
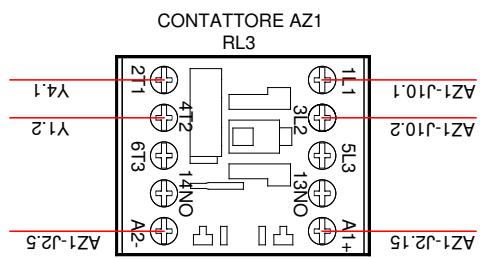
carreglio dwg_logo_dal@2011.10.25.jpg
La Mechanic's and Dynamic Devices s.r.l., proprietaria del presente disegno, si riserva i propri diritti qualunque sia il soggetto, senza propria autorizzazione scritta, esso venga riprodotto, conegnato a terzi o comunque utilizzato per costituire oggetto rappresentato.



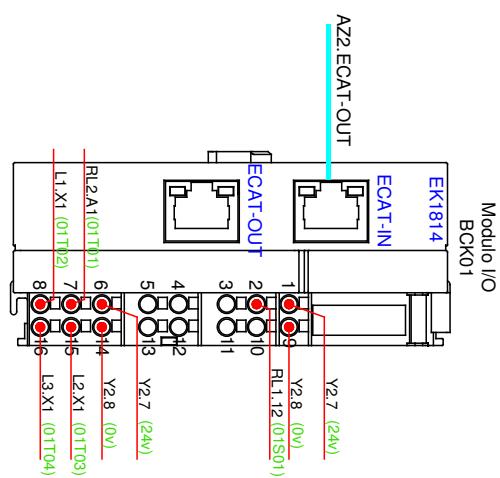
CONTROL PC

MATERIALI		TRATTAMENTI	
PACCHETTI NON QUOTATI	SIQUISI NON QUOTATI	TOLLENANZE GENERALI	RIGUARDI A R ₀ mm
DISGOMBI	DATA	ISO 2678 mk	SCALA
Ginammì	16/12/2013		--
DENOMINAZIONE			
cariglio dwg_Logo_deli@2011.10.25.jpg	CONCEP. DISEGNO	RBT-02-01-02-00-100/E14	REV 00 A3
La Mechantronics è Dynamic Devices s.r.l. proprietaria dei presenti disegni, si riserva di tutelare i propri diritti di autorità, senza propria autorizzazione scritta, esso venga riprodotto, consegnato a terzi o comunque utilizzato per così dire tolgendo apposito riferimento.			





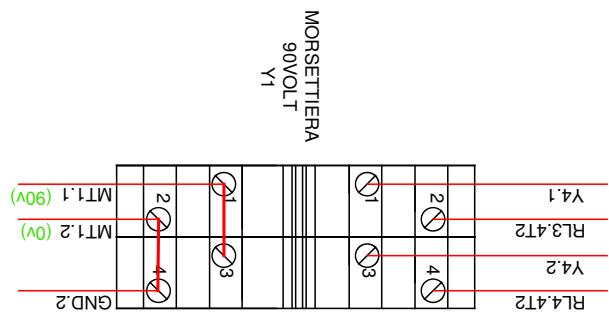
MATERIALE		TRATTAMENTI		
--	--	--	--	
RACCORDI NON QUOTATI	SMASSI NON QUOTATI	TOLLEFANZE GENERALI	RUGOSITÀ Ra mm	
--	--	ISO 2768 mk	SCALA	
DISEGNATORE	DATA	-- --		
Gianmimi	16/12/2013			
DENOMINAZIONE				
carriagio dwg_Logo_dal@2011.10.25.jpg		Schema collegamento CONTATTORI		
CODICE DISEGNO		REV 00	A3	
RBT-02-01-02-00-100/E3				
La Mecantronics and Dynamic Devices srl., proprietaria del presente disegno, si riserva di tutelare i propri diritti qualora, senza propria autorizzazione scritta, esso venga riprodotto, conegnato a terzi o comunque utilizzato per costituire oggetto rappresentato.				



Modulo I/O
BCK01

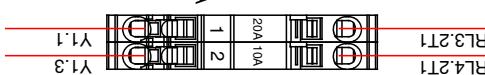
MATERIALE		TRATTAMENTI	
--	--	--	--
RACCORDI NON QUOTATI	SMASSI NON QUOTATI	TOLLEFANZE GENERALI	RUGOSITÀ Ra mm
--	--	ISO 2768 mk	--
DISEGNATORE			SCALA
Gianmimi			--
DATA			--
16/12/2013			
DENOMINAZIONE			
Schema collegamento BECKHOFF			
carreggi dwg_Logo_delt@2011.10.25.jpg		CODICE DISEGNO	REV 00
RBT-02-01-02-00-100/E4		A3	

La Mechanic's and Dynamic Devices s.r.l., proprietaria del presente disegno, si riserva i propri diritti qualunque, senza propria autorizzazione scritta, esso venga riprodotto, conegnato a terzi o comunque utilizzato per costituire oggetto rappresentato.



MATERIALE		TRATTAMENTO	
RACCORDI NON QUOTATI	SIMMETRIE NON QUOTATI	TOLERANZE GENERALI	RUGOSITÀ Ra mm
--	--	ISO 2768 mK	
DISGNAIORE	Ginammi	DATA	
		16/12/2013	
		DEMONSTRAZIONE	
cariglio dwg_Logo_deli@2011.10.25.jpg	COCCHE DISGNAIO	RBT-02-01-02-00-100/E5	REV 00 A3
La Mechatronics and Dynamic Devices s.r.l. proprietaria del presente disegno, si riserva di tutelare i propri diritti qualora, senza propria autorizzazione scritta, esso venga in prodotto, consegnato a terzi o comunque utilizzato per costituire l'oggetto di esproprio.			

MORSETTIERA
FUSIBILI
Y4



20A 10A	1	2
---------	---	---

MATERIALE		TRATTAMENTI	
RACCORDI NON QUOTATI	---	--	--
---	SMASSI NON QUOTATI	TOLLEFANZE GENERALI	RUGOSITÀ Ra mm
---	---	ISO 2768 mk	SCALA

DISEGNATORE	DATA	REV	SCALA
Gianmimi	16/12/2013	--	--

DENOMINAZIONE

Schema collegamento MORSETTIERA Y4

cattiglio dwg_Logo_dal@2011.10.25.jpg

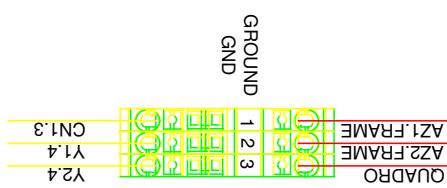
CODICE DISEGNO

RBT-02-01-02-00-100/E6

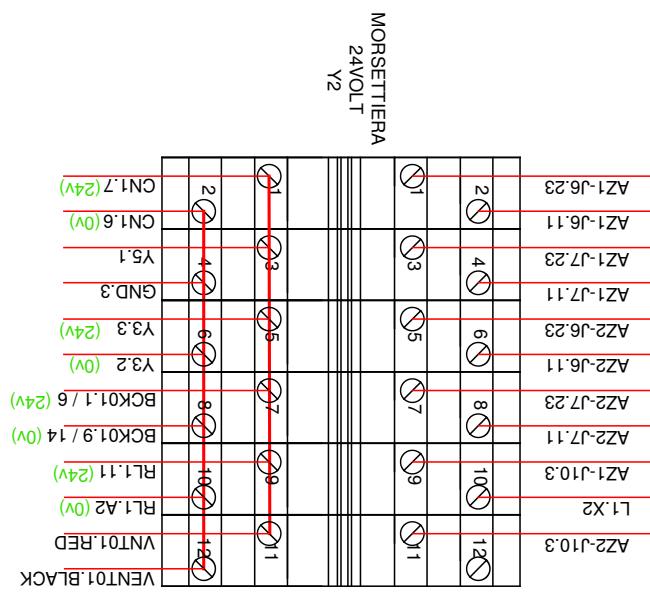
REV 00

A3

La Mechanic's and Dynamic Devices s.r.l., proprietaria del presente disegno, si riserva i propri diritti qualunque, senza propria autorizzazione scritta, esso venga riprodotto, concesso a terzi o comunque utilizzato per costituire oggetto rappresentato.

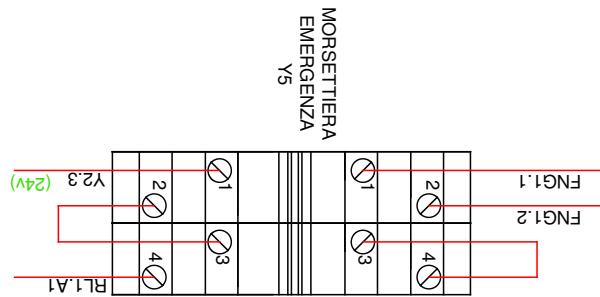


MATERIALE		TRATTAMENTI	
RACCORDI NON QUOTATI	SMASSI NON QUOTATI	TOLLEFANZE GENERALI	RUGOSITÀ Ra mm
--	--	ISO 2768 mk	--
DISEGNATORE		DATA	SCALA
Gianmimi			--
16/12/2013			--
DENOMINAZIONE			
carriagio dwg_Logo_delt@2011.10.25.jpg			
Schema collegamento MORSETTIERA GND		REV 00	A3
RBT-02-01-02-00-100/E7		CODICE DISEGNO	
La Mechanic's and Dynamic Devices s.r.l. proprietaria del presente disegno, si riserva i propri diritti qualunque, senza propria autorizzazione scritta, esso venga riprodotto, conegnato a terzi o comunque utilizzato per costituire oggetto rappresentato.			



MATERIALE		TRATTAMENTI	
RACCORDI NON QUOTATI	SMASSI NON QUOTATI	TOLLEFANZE GENERALI	RUGOSITÀ Ra mm
--	--	ISO 2768 mk	--
DISEGNATORE		DATA	SCALA
Gianmimi		16/12/2013	--
DINAMICHEZIONE			
carriagio dwg_Logo_delt@2011.10.25.jpg			
Schema collegamento MORSETTIERA Y2		REV 00	A3
RBT-02-01-02-00-100/E8		CODICE DISEGNO	

La Mechantronics and Dynamic Devices s.r.l., proprietaria del presente disegno, si riserva di tutelare i propri diritti qualora, senza propria autorizzazione scritta, esso venga riprodotto, conegnato a terzi o comunque utilizzato per costituire oggetto rappresentato.



MORSETTIERA
EMERGENZA
Y5

MATERIALE		TRATTAMENTI	
RACCORDI NON QUOTATI	SMASSI NON QUOTATI	TOLLEFANZE GENERALI	RUGOSITÀ Ra mm
--	--	ISO 2768 mk	--
DISEGNATORE	DATA	--	--
Gianmimi		16/12/2013	
DENOMINAZIONE			
carriagio dwg_Logo_delt@2011.10.25.jpg			
Schema collegamento MORSETTIERA Y5			
CODICE DISEGNO		REV 00	A3
RBT-02-01-02-00-100/E9			
La Mechanic's and Dynamic Devices s.r.l. proprietaria del presente disegno, si riserva i propri diritti qualunque, senza propria autorizzazione scritta, esso venga riprodotto, conegnato a terzi o comunque utilizzato per costituire oggetto rappresentato.			

Riferimenti bibliografici

- [1] IEC-International Electrotechnical Commission et al. “IEC 61158-1:2019”. In: (2019).
- [2] IEC-International Electrotechnical Commission et al. “IEC 61784-3-3”. In: (2016).
- [3] Dirk Jansen e Holger Buttner. “Real-time Ethernet: the EtherCAT solution”. In: *Computing and Control Engineering* 15.1 (2004), pp. 16–21.
- [4] Ortiz Jose. “Introduction to Adams/Solver C++”. In: (2011).
- [5] Ingo Lütkebohle. *BWorld Robot Control Software*. <http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software/>. [Online; accessed 19-July-2008]. 2008.
- [6] Suresh Mikkili, Anup Kumar Panda e Jayanthi Prattipati. “Review of real-time simulator and the steps involved for implementation of a model from MATLAB/SIMULINK to real-time”. In: *Journal of The Institution of Engineers (India): Series B* 96.2 (2015), pp. 179–196.
- [7] CC Ng, SK Ong e Andrew YC Nee. “Design and development of 3-DOF modular micro parallel kinematic manipulator”. In: *The International Journal of Advanced Manufacturing Technology* 31.1-2 (2006), pp. 188–200.
- [8] Bruno Siciliano et al. “Modelling, planning and control”. In: *Advanced Textbooks in Control and Signal Processing*. Springer, (2009).
- [9] Shishan Yang e Marcus Baum. “Extended Kalman filter for extended object tracking”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017, pp. 4386–4390. DOI: 10.1109/ICASSP.2017.7952985.

Elenco delle figure

1.1	Robot PKM	6
2.1	Rappresentazione fisica del robot	9
2.2	Equazioni alle circonferenze	10
3.1	Confronto dinamica leggi di moto su accelerazioni	26
3.2	Confronto dinamica leggi di moto su velocità	26
3.3	Confronto dinamica leggi di moto su posizioni	26
3.4	Modello simulink manipolatore	27
3.5	Modello Adams manipolatore 5R	27
3.6	Coppie in uscita Matlab	28
3.7	Coppie in uscita Adams	28
3.8	Errore Simulink-Adams	29
4.1	Posizione standard manipolatore	31
4.2	Caso 1 singolarità	32
4.3	Caso 2 singolarità	33
4.4	Caso 3 singolarità	34
4.5	Caso 4 e 5 singolarità	35
4.6	Punti di singolarità	36
4.7	Numero di condizionamento	37
4.8	Angoli di movimentazione dei giunti motorizzati	38
4.9	<i>Workspace</i> 5R	38
4.10	Numero di condizionamento workspace	39
5.1	Struttura vite	41
5.2	Utensile da disegno	41
6.1	Schema azionamento	48
6.2	Registri azionamento	48
7.1	Banco di test	51
7.2	Schema modulo bechoff	53

ELENCO DELLE FIGURE

7.3	Topologia della rete	53
7.4	Configurazione rete ethernet user PC	54
7.5	Topologia rete mediante Ec-engineer	54
7.6	PDO in input	55
7.7	PDO Output 1 e 2	55
7.8	Schema generale simulink	56
7.9	Fase 1: Inizializzazione	57
7.10	Conversione e lettura motori	57
7.11	Fase 2: Input	58
7.12	Fase 3: Stateflow	59
7.13	Fase 4: Schema di controllo della vite	60
7.14	Fase 5: Schema di controllo delle braccia	61
7.15	Schema espanso invio coppia	61
7.16	Fase 6: Copie in uscita	62
7.17	Macchina a stati stateflow	63
7.18	Fase di Homing	64
7.19	Fase di posizionamento	66
7.20	Implementazione funzione di posizionamento	67
7.21	Fase di controllo	68
7.22	Interfaccia grafica	70
7.23	Schema teorico controllore proporzionale	72
7.24	Controllore proporzionale vite	72
7.25	Andamento vite rispetto a riferimento	73
7.26	Schema teorico controllore PD	74
7.27	Controllore PD Vite	74
7.28	Andamento vite rispetto a riferimento	75
7.29	Errori controllori vite	75
7.30	Controllore PD braccia	78
7.31	Coppie controllore PD braccia	78
7.32	Andamenti posizione ed errori controllore PD	79
7.33	Cerchio assi [x,y] controllore PD	80

7.34	Schema teorico controllo feedforward	80
7.35	Controllore feedforward	81
7.36	Andamenti controllo feed-forward	81
7.37	Schema controllore dinamica inversa	83
7.38	Controllore dinamica inversa	84
7.39	Ricerca di K_d al variare di K_p	85
7.40	Coppie giunti controllo dinamica inversa	85
7.41	Posizione vs riferimento controllore ID	85
7.42	Errori controllore ID	86
7.43	Cerchio [X,Y] controllo ID	86
7.44	Schema controllore robusto	92
7.45	Controllore robusto	92
7.46	Uscite parte robusta del controllo	93
7.47	Coppie in uscita al variare di ε	93
7.48	Confronto controllore PD e ID	94
7.49	Quadrato controllo dinamica inversa	94
7.50	Pattern controllo dinamica inversa	95
7.51	Spirale controllo dinamica inversa	95
7.52	Filtro primo ordine	97
7.53	Filtro secondo ordine	97
7.54	Kalman <i>sensor fusion</i>	100
7.55	Filtro di Kalman esteso	101

Elenco delle tabelle

1	Parametri manipolatore	9
2	Parametri dei motori	24
3	Leggi di moto validazione	28
4	Parametri end-effector	42
5	Tipologie controllo azionamenti	56
6	Valori variabile <i>bool</i>	69

ELENCO DELLE TABELLE

7	Valori luci	69
---	-----------------------	----