



UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA MECCANICA

**Algoritmi di identificazione dei
parametri dinamici applicati ad un
robot collaborativo**

**Identification algorithms of dynamic
parameters applied to a collaborative robot**

Relatore:
Prof. Matteo Claudio Palpacelli

Candidato:
Sergio Pacioni

Anno Accademico 2022-2023



UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA MECCANICA

Algoritmi di identificazione dei parametri dinamici applicati ad un robot collaborativo

Identification algorithms of dynamic
parameters applied to a collaborative robot

Relatore:
Prof. Matteo Claudio Palpacelli

Candidato:
Sergio Pacioni

Anno Accademico 2022-2023

UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA MECCANICA
Via Brezze Bianche – 60131 Ancona (AN), Italy

Sommario

L'importanza di un controllo preciso e facilmente attuabile su un robot ha portato allo sviluppo di diverse tecniche di modellazione ed identificazione dinamica, molte consistono in algoritmi basati su relazioni in forma chiusa. Il seguente lavoro è basato sullo studio e sull'utilizzo di specifici algoritmi applicandoli ad un robot collaborativo FANUC a 6 gradi di libertà appartenente alla serie CRX. Verrà prima effettuata e validata la modellazione, sia cinematica sia dinamica, usando i software *Matlab* e *Adams* per poi passare all'introduzione degli algoritmi presi in esame, spiegandone i passaggi e gli eventuali pregi per poter essere integrati in *Matlab*.

Le routine considerate riguardano la formulazione della matrice di regressione, utile ad eseguire l'identificazione dei parametri del robot e l'individuazione dei parametri base. Infine, una volta applicati, saranno confrontati i risultati ottenuti dai modelli con quelli misurati in ambiente virtuale *Adams*.

Indice

1	Introduzione	1
1.1	Concetti base	1
1.2	Identificazione parametrica	3
1.3	Obiettivo della tesi	4
2	Modellazione	7
2.1	Convenzione di Denavit-Hartenberg	7
2.2	Modellazione della cinematica	10
2.3	Modellazione della dinamica	13
3	Verifica dei modelli	19
3.1	Modello ADAMS	19
3.2	Verifica modello cinematico	22
3.3	Verifica modello dinamico	27
4	Identificazione	33
4.1	Formulazione del regressore	33
4.2	Identificazione dei parametri base	39
4.3	Verifica dei risultati	42
5	Conclusioni	47

Elenco delle figure

1.1	Schemi delle strategie di controllo possibili.	1
1.2	Robot collaborativo FANUC CRX-10iA/L.	4
1.3	Dati principali del FANUC CRX-10iA/L [1].	5
1.4	Geometria FANUC CRX-10iA/L [1].	5
2.1	Parametri di Denavit-Hartenberg [2].	8
2.2	Terne di giunto su FANUC CRX-10iA/L [3].	9
2.3	Configurazione FANUC CRX-10iA/L.	11
2.4	Variazione coordinate terminale nel tempo.	11
2.5	Variazione spostamenti sui giunti nel tempo secondo la legge <i>STEP</i>	15
2.6	Variazione velocità sui giunti nel tempo secondo la legge <i>STEP</i>	15
2.7	Variazione accelerazione sui giunti nel tempo secondo la legge <i>STEP</i>	16
2.8	Andamento delle coppie sui giunti 1 e 2	16
2.9	Andamento delle coppie sui giunti 3 e 4	17
2.10	Andamento delle coppie sui giunti 5 e 6	17
3.1	Rappresentazione dei giunti definiti sul robot in <i>Adams</i>	20
3.2	Rappresentazione degli attuatori definiti sul robot in <i>Adams</i>	22
3.3	Andamento posizione terminale risultante dal modello <i>Matlab</i>	23
3.4	Andamento posizione terminale risultante dal modello <i>Adams</i>	23
3.5	Andamento velocità terminale risultante dal modello <i>Matlab</i>	24
3.6	Andamento velocità terminale risultante dal modello <i>Adams</i>	24
3.7	Configurazione finale ottenuta in <i>Matlab</i>	25
3.8	Configurazione finale ottenuta in <i>Adams</i>	25
3.9	Configurazione finale ottenuta in <i>Adams</i> per verifica dinamica.	27
3.10	Coppia sul giunto 1 calcolata dal modello <i>Matlab</i>	28
3.11	Coppia sul giunto 1 ottenuta dal modello <i>Adams</i>	28
3.12	Coppia sul giunto 2 calcolata dal modello <i>Matlab</i>	28
3.13	Coppia sul giunto 2 ottenuta dal modello <i>Adams</i>	29
3.14	Coppia sul giunto 3 calcolata dal modello <i>Matlab</i>	29
3.15	Coppia sul giunto 3 ottenuta dal modello <i>Adams</i>	29
3.16	Coppia sul giunto 4 calcolata dal modello <i>Matlab</i>	30
3.17	Coppia sul giunto 4 ottenuta dal modello <i>Adams</i>	30
3.18	Coppia sul giunto 5 calcolata dal modello <i>Matlab</i>	30
3.19	Coppia sul giunto 5 ottenuta dal modello <i>Adams</i>	31
3.20	Coppia sul giunto 6 calcolata dal modello <i>Matlab</i>	31

Elenco delle figure

3.21	Coppia sul giunto 6 ottenuta dal modello <i>Adams</i>	31
4.1	Confronto valori grandezze dinamiche identificate	42
4.2	Confronto coppia al giunto 1.	43
4.3	Confronto coppia al giunto 2.	43
4.4	Confronto coppia al giunto 3.	44
4.5	Confronto coppia al giunto 4.	44
4.6	Confronto coppia al giunto 5.	45
4.7	Confronto coppia al giunto 6.	45

Elenco delle tabelle

2.1	Parametri DH per del FANUC CRX-10iA/L.	8
2.2	Valori variabili di giunto simulazione.	12
2.3	Valori variabili di giunto e tempi simulazione.	18
3.1	Valori variabili di giunto simulazione per la validazione.	26
3.2	Valori variabili di giunto e tempi simulazione per la validazione.	32

Capitolo 1

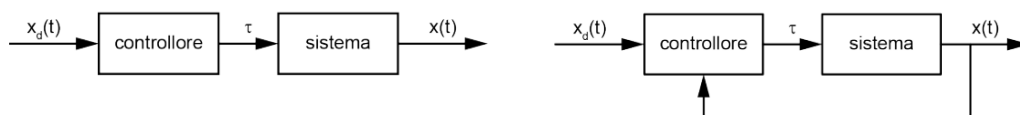
Introduzione

1.1 Concetti base

Tra la fine degli anni 70 e gli inizi degli anni 80, l'interesse nei confronti di possibili applicazioni di robot manipolatori con lo scopo di automatizzare la produzione aumentò vertiginosamente. A causa di complessità tecniche ed economiche l'idea di utilizzare robot manipolatori per lo svolgimento di alcuni compiti in una linea di produzione si rivelò inizialmente fallimentare.

Ancora oggi il controllo di robot rappresenta un problema rilevante nell'utilizzo di manipolatori per finalità industriali; infatti, per il corretto controllo dei movimenti di un robot è richiesta un'analisi accurata delle caratteristiche meccaniche di tutte le componenti della macchina. Il controllo di un sistema meccanico avviene tramite un controllore che va a regolare le azioni da intraprendere sul sistema; nel caso di un robot vengono regolate le coppie sugli attuatori della macchina facendo sì che il terminale si muova rispettando le leggi desiderate. Si possono attuare due tipologie differenti di controllo: controllo in anello aperto e controllo in anello chiuso.

- Controllo in anello aperto: una volta definito, e quindi noto il comportamento desiderato, il controllore agisce sugli attuatori imponendo i valori di coppia opportuni per ottenere il comportamento desiderato, senza guardare al comportamento attuale del sistema;
- Controllo in anello chiuso: note le traiettorie desiderate il controllore definisce e regola le coppie ai motori basandosi unicamente sulle informazioni relative allo stato attuale della macchina;



(a) Controllo in anello aperto.

(b) Controllo in anello chiuso.

Figura 1.1: Schemi delle strategie di controllo possibili.

Come intuibile, sia nel caso di controllo in catena aperta, sia nel caso di controllo in catena chiusa, è indispensabile essere a conoscenza del modello matematico del sistema

che si vuole controllare, e tanto più questo sarà accurato tanto più si otterrà precisione nell'azione di controllo. Dunque, il problema di poter controllare una macchina con elevata precisione passa per l'esigenza di trovare gli andamenti nel tempo delle forze e delle coppie agenti sui giunti che garantiscano l'esecuzione delle traiettorie programmate. Viene naturale dedurre che la modellazione del comportamento dinamico di una macchina gioca un ruolo importante nella progettazione di un algoritmo di controllo. Inoltre, una volta in possesso di un modello dinamico affidabile, sarà possibile effettuare simulazioni tramite le quali mettere a punto tecniche di controllo, o studiare e progettare nuovi giunti, attuatori o trasmissioni.

Lo studio del comportamento dinamico permette di ottenere le equazioni del moto del sistema, le quali consentono di modellare il comportamento dinamico dello stesso quando sollecitato da azioni esterne e azioni sui giunti. Le equazioni possono essere ricavate con approcci differenti in base alle caratteristiche del sistema o in base alle finalità dello studio, le due formulazioni più utilizzate sono quella di Newton-Eulero e quella di Lagrange, che a differenzia dalla prima segue un approccio energetico.

Generalmente la dinamica di una macchina può essere sempre ricondotta alla seguente equazione vettoriale

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (1.1)$$

dove q, \dot{q}, \ddot{q} rappresentano i vettori contenenti le posizioni, le velocità e le accelerazioni sui giunti e τ è il vettore delle coppie agli attuatori; mentre la matrice B e C sono la matrice di inerzia e la matrice contenente i termini di Coriolis. Infine G rappresenta il contributo della forza gravitazionale. A prescindere dalla formulazione utilizzata, una volta ricavate, le equazioni del moto possono essere utilizzate per risolvere due problemi differenti:

- Analisi dinamica diretta: determinare il moto del sistema quando è sottoposto a forze e coppie assegnate;
- Analisi dinamica inversa: determinare le forze e coppie sui giunti per la realizzazione di una traiettoria assegnata;

Però, come facilmente intuibile da (1.1), l'utilizzo delle equazioni del moto richiede la conoscenza dei valori dei parametri dinamici del manipolatore, che rispetto alle caratteristiche geometriche risultano più difficili da valutare. Se alcuni parametri dinamici non sono noti si potrebbero ricavare replicando i componenti di interesse tramite software di modellazione 3D oppure disassemblare il manipolatore ed eseguire una serie di misurazione sui componenti per dedurre le caratteristiche inerziali di ognuno. La prima strada porterebbe a risultati poco accurati a causa delle semplificazioni che tipicamente si introducono nella modellazione 3D; mentre la seconda tecnica è ovviamente molto complessa per essere attuata e potrebbe risultare molto difficoltoso realizzare le misure corrette di tutti i parametri necessari.

Una tecnica che permette facilmente di ricavare il corretto valore dei parametri dinamici di un manipolatore, è quella dell'identificazione dinamica, o calibrazione. Questa tecnica permette di ricavare i valori di ogni singolo parametro inerziale del sistema tramite l'elaborazione dell'equazioni del moto e permette di calcolare facilmente tutte le caratteristiche dinamiche del sistema, basandosi sulla possibilità di linearizzare la relazione fra i parametri dinamici e le coppie ai giunti.

1.2 Identificazione parametrica

Considerando un generico manipolatore formato da n elementi, e considerando che per ogni elemento sono presenti 10 parametri dinamici sarà necessario calcolare un totale di $r = n \times 10$ parametri dinamici per poter descrivere in maniera completa la dinamica del robot; infatti per ogni link i -esimo è possibile definire le seguenti caratteristiche inerziali:

$$m_i, m_i p_{iG_i x}, m_i p_{iG_i y}, m_i p_{iG_i z}, I_{ixx}, I_{ixy}, I_{ixz}, I_{iyy}, I_{iyz}, I_{izz},$$

Tramite l'identificazione parametrica questi parametri di tutti i corpi verranno raccolti in un vettore $\pi \in \mathbb{R}^r$, che può essere calcolato valutando tutte le coppie ai giunti τ e la matrice $Y(q, \dot{q}, \ddot{q}) \in \mathbb{R}^{n \times r}$, chiamata matrice di regressione, o regressore. Questo è possibile poiché, se opportunamente definito, il vettore π , risulta essere lineare il legame con le generiche equazioni del moto (1.1) per mezzo della matrice di regressione. Dunque sarebbe possibile calcolare il valore del vettore π riscrivendo le equazioni del moto nella forma

$$Y(q, \dot{q}, \ddot{q})\pi = \tau \quad (1.2)$$

Ipotizzando di poter misurare o calcolare con accuratezza i valori delle coppie ai giunti e, considerando che la matrice di regressione è solamente funzione di grandezze cinematiche facilmente calcolabili, l'unica grandezza incognita nella (4.33) risulta essere il vettore contenente i valori delle grandezze dinamiche. Per poter ricavare correttamente π , è però necessario valutare la matrice di regressione e le coppie ai giunti in più istanti di tempo, impilando i risultati computati in modo tale da ottenere

$$\bar{Y}(q, \dot{q}, \ddot{q})\pi = \bar{\tau} \quad (1.3)$$

tale che $\bar{Y}(q, \dot{q}, \ddot{q})$ sia una matrice $Mn \times r$ e $\bar{\tau}$ un vettore $Mn \times 1$ con $M \gg n$. Così facendo il regressore sarà caratterizzato da molte più righe che colonne e verranno evitati mal condizionamenti della matrice. Sfruttando la tecnica dei minimi quadrati è possibile calcolare il vettore π :

$$\pi = (\bar{Y}^T \bar{Y})^{-1} \bar{Y}^T \bar{\tau} \quad (1.4)$$

Non tutti i parametri racchiusi in π contribuiscono alla dinamica del sistema, ma solo alcuni ne influenzano il comportamento dinamico. Oltre a ciò, come verrà analizzato successivamente, è possibile individuare un vettore, detto vettore dei parametri base π_B , che contiene al suo interno b combinazioni lineari degli r parametri inerziali, meno quelli influenti, le quali risultano essere sufficienti per descrivere il comportamento del sistema. È quindi possibile raggruppare i parametri e, modificando opportunamente la matrice di regressione, si ottiene un modello matematico più snello e meno pesante dal punto di vista computazionale, ma comunque in grado di descrivere in maniera completa la dinamica della macchina costituito dalla relazione:

$$Y_B \pi_B = \tau \quad (1.5)$$

dove Y_B è una matrice $Mn \times b$ e π_B un vettore $b \times 1$, con τ che rimane invariato.

1.3 Obiettivo della tesi

L'obiettivo di questo lavoro è quello di effettuare un'analisi approfondita su un robot collaborativo industriale, studiandone sia la cinematica sia la dinamica, per poi definire il vettore dei parametri base e verificare la possibilità di descrivere la dinamica del sistema ricorrendo alla formula (1.5). Il robot preso in esame è un FANUC CRX-10iA/L, (Figura 1.2), un robot che, oltre ad un'eccellente collaboratività, fornisce tutti i requisiti di affidabilità industriale.



Figura 1.2: Robot collaborativo FANUC CRX-10iA/L.

Il robot FANUC CRX-10iA/L è dotato di sei giunti rotoidali ed è caratterizzato da sei gradi di libertà nello spazio; si tratta di una catena cinematica aperta in cui i giunti fungono da collegamento tra un link e l'altro. Una delle sue particolarità sta nel polso roll-pitch-roll, che ha configurazione non sferica, dato che gli ultimi

tre giunti hanno assi che sono due per due perpendicolari e incidenti. Appartenente alla CRX series della casa produttrice, gamma caratterizzata da cinque modelli differenti per capacità di carico (payload) e sbraccio (reach), il CRX-10iA/L presente le seguenti caratteristiche tecniche:

Robot			Max. load capacity at wrist (kg)	***Reach (mm)	Controlled axes	**Repeatability (mm)	Mechanical weight (kg)	Motion range (°)						*Maximum speed (°/s)	
Series	Version	Type						J1	J2	J3	J4	J5	J6	Collaborative Mode (mm/sec)	High Speed Mode (mm/sec)
CRX	iA/L	10	10	1418	6	±0.04	40	360	360	540	380	360	380	1000	2000

Figura 1.3: Dati principali del FANUC CRX-10iA/L [1].

Mentre le principali informazioni riguardanti le geometrie dei componenti della macchina, che saranno utili per effettuare l'analisi cinematica, sono riportati nella figura seguente.

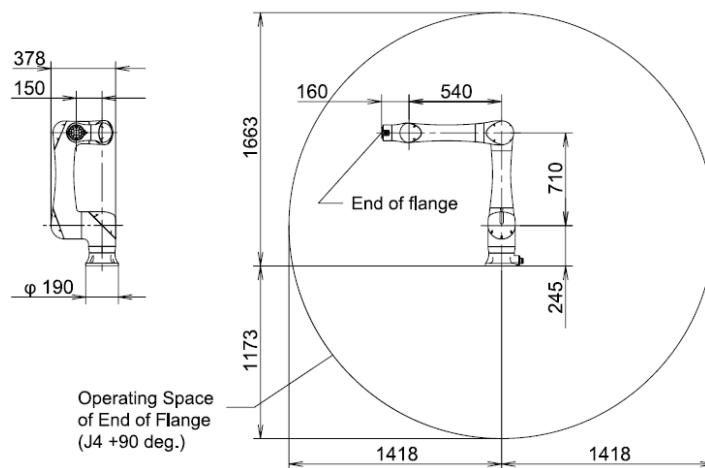


Figura 1.4: Geometria FANUC CRX-10iA/L [1].

Il primo step è stato quello di studiare la cinematica del sistema, andando a comporre un modello matematico caratterizzato dalle equazioni della cinematica dettate dalla geometria e dalla struttura del robot. Questo è stato fatto tramite l'utilizzo del software *Matlab*, creando uno script utile a risolvere le equazioni di cinematica diretta; per poter verificare la correttezza di tali equazioni invece, è stato utilizzato il software di modellazione multibody *MSC Adams*. Stessa cosa è stata fatta per le equazioni della dinamica, inizialmente definite e risolte in *Matlab* e successivamente verificate con il modello multibody del sistema. Una volta verificate le equazioni costituenti i modelli matematici è stato possibile iniziare a lavorare per realizzare l'identificazione parametrica.

Per ottenere la matrice di regressione sono stati valutati due algoritmi spiegati in [4] e [5] per poi utilizzare quello descritto in [5], che permette di ricavare la struttura

simbolica della matrice per un sistema meccanico partendo dalle energie associate ai corpi di esso. La procedura di entrambi gli algoritmi considerati sono spiegate nella sezione 4.1. Per ricavare in seguito il vettore dei parametri base sono stati presi in considerazione altri due differenti algoritmi presentati in [6] e [7], i quali presentano due differenti approcci per individuare il set minimo di parametri utili a descrivere la dinamica di un robot. Anche in questo caso le procedure saranno spiegate più nel dettaglio nella sezione 4.1. Ricavata la matrice di regressione e applicati gli algoritmi, si è finalmente giunti alla forma espressa dalla (1.5), che è stata verificata confrontato le coppie risultanti con quelle simulate e calcolate dal software *Adams*.

Capitolo 2

Modellazione

2.1 Convenzione di Denavit-Hartenberg

La modellazione di un meccanismo seriale parte dallo studio dei suoi giunti, con lo scopo di individuarne la tipologia ed il posizionamento, per poi passare alla definizione delle coordinate di giunto. Uno dei metodi più utilizzati per lo studio della cinematica di sistemi di corpi rigidi è quello proposto da Denavit e Hartenberg; questa metodologia, che prende il nome dai due studiosi, permette di ottenere le equazioni cinematiche del sistema attraverso la scrittura di apposite matrici di trasformazione omogenea 4×4 . Il metodo risulta essere sistematico, applicabile a qualsiasi tipo di sistema, e facendo uso di coordinate relative, si dimostra particolarmente utile per lo studio di robot seriali.

Come prima cosa i membri della macchina devono essere numerati da 1 a n partendo dal telaio, che convenzionalmente viene indicato come membro 0, analogamente sono indicate le coppie cinematiche che uniscono i vari membri, in seguito si dota ogni membro della catena cinematica di un sistema di riferimento a esso solidale. La relazione cinematica tra una coppia di membri connessi in maniera seriale verrà espressa attraverso i 4 parametri caratteristici del giunto $a_i, d_i, \alpha_i, \theta_i$; quindi per un manipolatore con n giunti si avranno un totale di $4 \times n$ parametri, che possono essere raccolti in una tabella, chiamata tabella di Denavit-Hartenberg. Prima di poter definire i parametri per ogni giunto è necessario individuare gli assi di giunto e definire le terne solidali ai membri secondo la seguente procedura:

- si sceglie l'asse z_i giacente lungo l'asse del giunto $i+1$;
- si individua l'origine O_i della terna i nell'intersezione dell'asse del giunto $i+1$ con la normale comune tra l'asse stesso e l'asse del giunto precedente (se i due assi sono paralleli l'origine può essere scelta in un qualsiasi punto della retta che funge da distanza);
- si sceglie l'asse x_i diretto lungo la normale comune del passo precedente con verso positivo dal giunto i al giunto $i+1$;
- si sceglie l'asse y_i in modo da completare la terna ortonormale destrorsa;

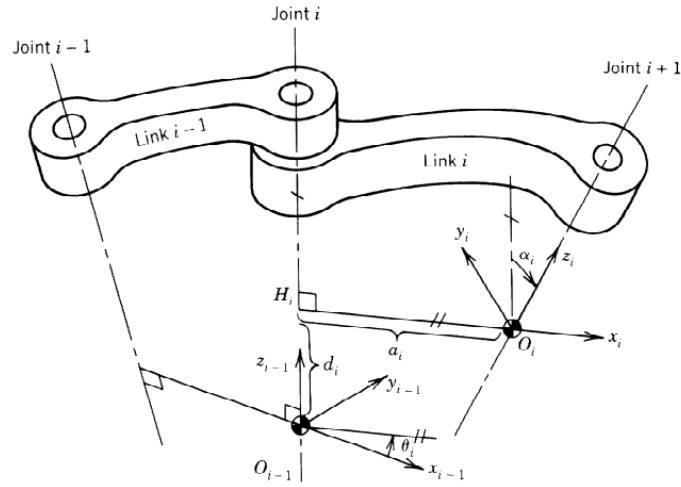


Figura 2.1: Parametri di Denavit-Hartenberg [2].

A questo punto sarà possibile determinare il valore dei parametri di giunto e comporre racchiuderli nella tabella di Denavit-Hartenberg, rispettando le loro definizioni:

- a_i , distanza tra l'asse z_i e l'asse z_{i-1} , misurata lungo la direzione positiva di x_i ;
- d_i , distanza tra l'asse x_i e l'asse X_{i-1} , misurata lungo la direzione positiva di z_i ;
- α_i , angolo intorno a x_i tra l'asse z_{i-1} e l'asse z_i , positivo in senso antiorario;
- θ_i , angolo intorno a z_{i-1} tra l'asse x_{i-1} e l'asse x_i , positivo in senso antiorario;

Nel caso del robot oggetto di questo lavoro i parametri θ_i corrisponderanno alle variabili di giunto q , mentre tutti gli altri parametri sono stati calcolati e riportati nella tabella 2.1, mentre in figura 2.2 è rappresentato il posizionamento delle terne di giunto sul robot.

Tabella 2.1: Parametri DH per del FANUC CRX-10iA/L.

i	$\alpha_i [^\circ]$	$a_i [m]$	$d_i [m]$	$\theta_i [^\circ]$
1	90	0	0.2503	q_1
2	-180	0.71	0.2604	q_2
3	-90	0	0.2604	q_3
4	-90	0	0.540	q_4
5	90	0	0.150	q_5
6	0	0	0.160	q_6

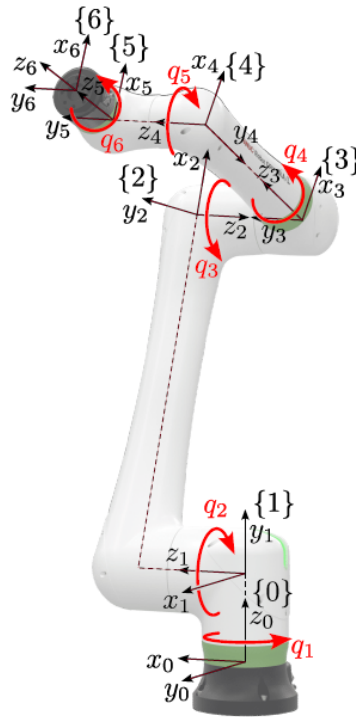


Figura 2.2: Terne di giunto su FANUC CRX-10iA/L [3].

La trasformazioni omogenea, fra un sistema di riferimento al successivo, nasce dalla composizione di due rototraslazioni: una attorno l'asse z e l'altra attorno all'asse x, rappresentate dalle due matrici:

$${}_{int}^{i-1}T = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}_{int}^{i-1}T = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

Dove *int* rappresenta una terna intermedia centrata nel punto H_i di figura 2.1. Dalla composizione delle due trasformazioni si ha quella complessiva tra il sistema di riferimento *i*-esimo e quello (*i*-1)-esimo.

$${}_{i-1}^i T = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

2.2 Modellazione della cinematica

Combinando tutte le terne di trasformazione di un sistema meccanico, partendo dal telaio e risalendo fino all'ultimo link, è possibile ottenere la matrice di trasformazione ${}^n_0T(q_1, \dots, q_n)$ che permette di esprimere la posizione del terminale rispetto alla terna assoluta a telaio, risolvendo così il problema di cinematica diretta. Dunque sarà necessario calcolare tutte le matrici di trasformazione omogenea del robot per ottenere l'espressione della matrice ${}^n_0T(q)$.

Come anticipato nella sezione 1.3, lo studio cinematico del regressore è stato realizzato con *Matlab* e si basa principalmente su una funzione definita per poter calcolare la matrice di trasformazione ${}^{i-1}_i T$ una volta indicati i valori dei parametri a_i , d_i , α_i , θ_i , ovvero q_i . Creata correttamente la funzione, è stato realizzato lo script nel quale viene effettivamente risolta la cinematica diretta del manipolatore, calcolando le coordinate del terminale in funzione delle 6 variabili di giunto $q_1, q_2, q_3, q_4, q_5, q_6$.

Il modello esegue un'analisi cinematica del meccanismo nel tempo basandosi su un'interpolazione di giunto; ovvero definito il tempo della simulazione t si fissano le coordinate di giunto iniziali q_0 e finali q_f per poi ricavare le velocità angolari su ogni giunto secondo la formula $\frac{q_f - q_0}{t}$, che quindi risultano essere costanti nel tempo. Per effettuare un'analisi cinematica nel tempo è stato necessario definire un numero di step di calcolo, di conseguenza è possibile calcolare la variazione delle coordinate di giunto per ogni step come $\Delta q = \frac{q_f - q_0}{m}$; poiché i giunti sono caratterizzati da velocità angolari costanti. A questo punto il valore delle coordinate q è calcolabile per ogni step di calcolo con la seguente formula, dove m rappresenta lo step di calcolo in cui si desidera calcolare q .

$$q = q_0 + m\Delta q \quad (2.3)$$

Ora, grazie ad un ciclo *for*, si può risolvere la cinematica per tutti gli m step in cui è stato diviso il tempo t della simulazione, ed ottenere l'andamento della cinematica del sistema. Come spiegato ad inizio sezione per ottenere la posa del manipolatore occorre calcolare le matrici di trasformazione, così per ogni step di calcolo sono state calcolate le matrici ${}^0_1T, {}^1_2T, {}^2_3T, {}^3_4T, {}^4_5T, {}^5_6T$, grazie alla funzione appositamente definita, usando come dati di input i parametri della tabella 2.1 e i valori di q calcolati tramite la 2.3. A questo punto è possibile calcolare le coordinate ed il loro andamento nel tempo dei punti che si desiderano: ad esempio per studiare il terminale basta ricavare la matrice 0_6T , combinando tutte le matrici di trasformazione, e moltiplicarla per il vettore $[0;0;0;1]$.

$$P_6 = {}^0_6 T \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = {}^0_1 T {}^1_2 T {}^2_3 T {}^3_4 T {}^4_5 T {}^5_6 T \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.4)$$

Conclusa la parte computazionale è possibile ottenere il plot raffigurante la variazione delle coordinate del terminale nel tempo (Fig.2.4) e anche una visualizzazione grafica della configurazione finale che il robot assume (Fig.2.3), con evidenziata in rosso la traiettoria compita nello spazio dal terminale.

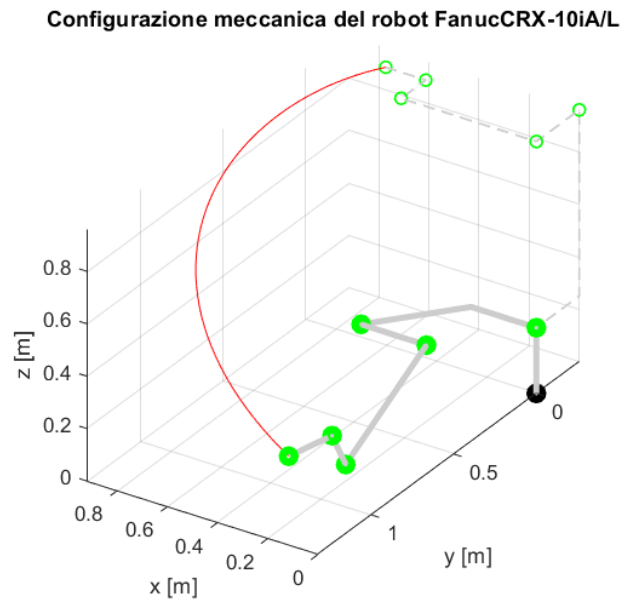


Figura 2.3: Configurazione FANUC CRX-10iA/L.

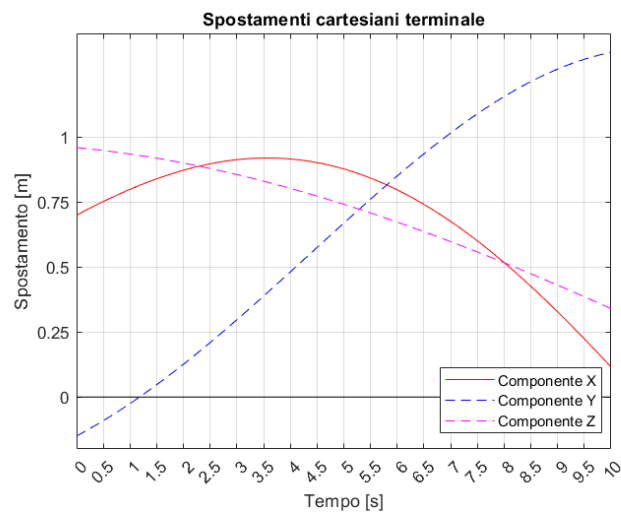


Figura 2.4: Variazione coordinate terminale nel tempo.

I risultati mostrati sono ottenuti impostando il tempo di simulazione pari a 10 secondi, con 50 step di calcolo, e i seguenti valori delle variabili di giunto q_0 e q_f , con conseguente derivazione delle velocità \dot{q} .

Tabella 2.2: Valori variabili di giunto simulazione.

$q_0[^\circ]$	$q_f[^\circ]$	$\dot{q}[\text{rad/s}]$
0	90	0.1571
90	-70	-0.1222
0	-45	-0.0785
0	-85	-0.1484
0	-40	-0.0698
0	15	0.0262

In conclusione il modello cinematico del robot FANUC CRX-10iA/L è composto da tre file *Matlab*:

- *Geometria_CRX10.mlx*, al suo interno sono definiti tutti i parametri DH caratteristici del manipolatore e viene richiamato all'interno del file *Cinematica_CRX10.mlx* per effettuare le operazioni numeriche;
- *MaTrasf.mlx*, è la funzione che permette di ricavare la matrice di trasformazione ${}^i T$ inserendo come input $a_i, d_i, \alpha_i, \theta_i, q_i$;
- *Cinematica_CRX10.mlx*, questo è il file al cui interno viene effettivamente risolto il problema di cinematica diretta e si ottengono i plot dei risultati mostrati in figura 2.3 e 2.4;

2.3 Modellazione della dinamica

Nella sezione 1.1 è stata introdotta la forma generale in grado di descrivere la dinamica di un sistema meccanico (1.1), come già spiegato è possibile ricavare tali equazioni utilizzando diversi metodi a seconda delle finalità. Come suggerito in [4], col fine di ricavare la matrice di regressione, è conveniente scrivere la dinamica del manipolatore utilizzando la formulazione di Lagrange, seguendo dunque un approccio che prevede il calcolo delle energie in gioco nel sistema.

Definendo come $T^{(i)}$ e $U^{(i)}$ l'energia potenziale e cinetica del membro i -esimo, per un manipolatore seriale a n gradi di libertà, la funzione lagrangiana può essere espressa come:

$$L(q, \dot{q}) = T(q, \dot{q}) - U(q) = \sum_{i=1}^n (T^{(i)}(q, \dot{q}) - U^{(i)}(q)) = \sum_{i=1}^n L^{(i)}(q, \dot{q}) \quad (2.5)$$

dove q e \dot{q} rappresentano i vettori $n \times 1$ contenenti gli spostamenti e le velocità sui giunti. Adottando la formulazione di Lagrange si può scrivere la dinamica della catena seriale nella forma seguente.

$$\sum_{i=1}^n \left[\frac{d}{dt} \frac{\partial L^{(i)}}{\partial \dot{q}} - \frac{\partial L^{(i)}}{\partial q} \right]^T = \tau \quad (2.6)$$

Per definizione della funzione lagrangiana (2.5) occorre calcolare le espressioni dell'energia cinetica e potenziale di ogni link per poi comporre le equazioni del moto secondo la formulazione di Lagrange (2.6).

Considerando il generico link i -esimo l'energia cinetica totale del corpo sarà ottenuta sommando l'energia cinetica di traslazione a quella rotazionale.

$$T^{(i)} = \frac{1}{2} m_i v_{c_i}^T v_{c_i} + \frac{1}{2} \omega_i^T I_i \omega_i \quad (2.7)$$

dove m_i rappresenta la massa del link, v_{c_i} la velocità del baricentro, ω_i la velocità angolare del corpo e I_i la matrice di inerzia dello stesso. Tutte le quantità sono espresse rispetto alla terna assoluta 0. L'energia potenziale di ogni corpo può essere invece espressa come:

$$U^{(i)} = m_i g^0 r_{c_i} \quad (2.8)$$

dove g è il vettore dell'accelerazione gravitazionale e ${}^0 r_{c_i}$ è il vettore posizione del baricentro del membro i -esimo espresso rispetto alla terna di base 0.

Nel caso specifico di questa analisi, le equazioni della dinamica, verranno utilizzate per risolvere il problema di dinamica inversa, cioè occorrerà ricavare l'espressione del vettore τ in funzione delle variabili di giunto q, \dot{q}, \ddot{q} . Anche in questo caso è stato utilizzato il software *Matlab*, sfruttando però il *Symbolic Math Toolbox*, che permette di risolvere, rappresentare e manipolare equazioni di matematica simbolica.

Prima di applicare la (2.7) e la (2.8) a tutti e 6 i membri, e poter ottenere la funzione lagrangiana del manipolatore è necessario calcolare le espressioni delle velocità ${}^0v_{c_i}$, ${}^0\omega_i$, delle matrici 0I_i e del vettore ${}^0r_{c_i}$:

$${}^0v_{c_i} = {}^0v_i + {}^0\omega_i \times_i {}^0R^i r_{c_i}, \quad {}^0v_i = {}^0v_{i-1} + {}^0\omega_{i-1} \times {}^0r_{i-1,i} \quad (2.9)$$

$${}^0\omega_i = {}^0\omega_{i-1} + {}^0\dot{q}_i R \begin{bmatrix} 0 \\ 0 \\ \dot{q}_i \end{bmatrix} \quad (2.10)$$

$${}^0I_i = {}^0C_i R {}^0C_i I_i {}^0T_{C_i} R \quad (2.11)$$

$${}^0r_{c_i} = {}^0T^i r_{c_i} \quad (2.12)$$

Le grandezze appena definite sono strettamente legate a grandezze cinematiche, come la matrice di rotazione in (2.9),(2.10)e(2.11) o matrice di trasformazione stessa per (2.12), quindi occorre aver prima validato le equazioni cinematiche per poter scrivere una corretta dinamica del sistema. Tutti gli aspetti legati alla validazione e al controllo della cinematica saranno trattati in maniera più approfondita nella sezione 3.2. A questo punto, dopo aver ricavato l'espressione della funzione lagrangiana relativa al robot, è finalmente possibile ricavare in forma simbolica le coppie applicate ai giunti secondo la (2.6).

Grazie all'utilizzo della funzione *matlabFunction* l'espressione simbolica delle coppie sui giunti è stata trasformata in una funzione di *Matlab* da poter utilizzare per computare numericamente le coppie e quindi per poter risolvere la dinamica inversa. Le grandezze richieste come input dalla funzione sono ovviamente tutti i parametri inerziali del manipolatore, i parametri di Denavit-Hartenberg ed i valori dei tre vettori q , \dot{q} , \ddot{q} . Per questo modello, a differenza di quello relativo alla cinematica, sono state definite sui giunti delle leggi di movimento differenti seguendo la definizione data in [8] della funzione *STEP* presente in *Adams*, che permette di approssimare attraverso polinomi del terzo ordine la funzione a gradino di Heaviside; ha derivata prima continua, mentre la derivata seconda è discontinua in due punti (x_0, x_1) . La funzione è applicata sugli spostamenti q e richiede quattro grandezze in input, che nella guida *Adams* sono nominate x, x_0, x_1, h_0, h_1 :

- x , la variabile indipendente;
- x_0 , una variabile reale che specifica il valore x in corrispondenza del quale inizia la funzione STEP;
- x_1 , una variabile reale che specifica il valore x in corrispondenza del quale termina la funzione STEP;
- h_0 , il valore iniziale dello step;
- h_1 , il valore finale dello step;

In questo caso si considererà x come il tempo t e h come spostamento sul giunto q . La funzione calcolerà l'andamento di q nel tempo secondo la seguente formula.

$$a = h_1 - h_0, \quad \Delta = \frac{x-x_0}{x_1-x_0}$$

$$STEP = \begin{cases} h_0 & : x \leq x_0 \\ h_0 + a\Delta^2(3 - 2\Delta) & : x_0 < x < x_1 \\ h_1 & : x \geq x_1 \end{cases} \quad (2.13)$$

Da cui si possono ricavare anche gli andamenti di \dot{q} e \ddot{q} derivando la (2.13) rispetto al tempo. Graficamente si ottengono i seguenti andamenti degli spostamenti (Fig.2.5) velocità (Fig.2.6) e (Fig.2.7) accelerazioni sui giunti:

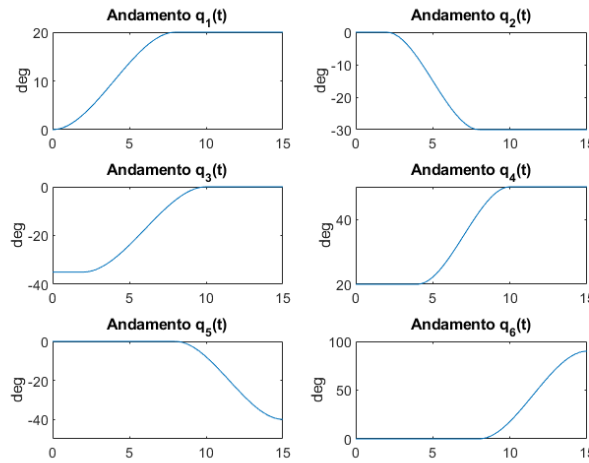


Figura 2.5: Variazione spostamenti sui giunti nel tempo secondo la legge *STEP*.

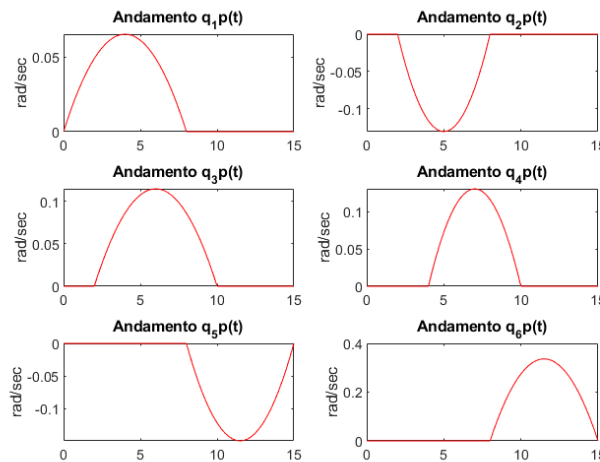


Figura 2.6: Variazione velocità sui giunti nel tempo secondo la legge *STEP*.

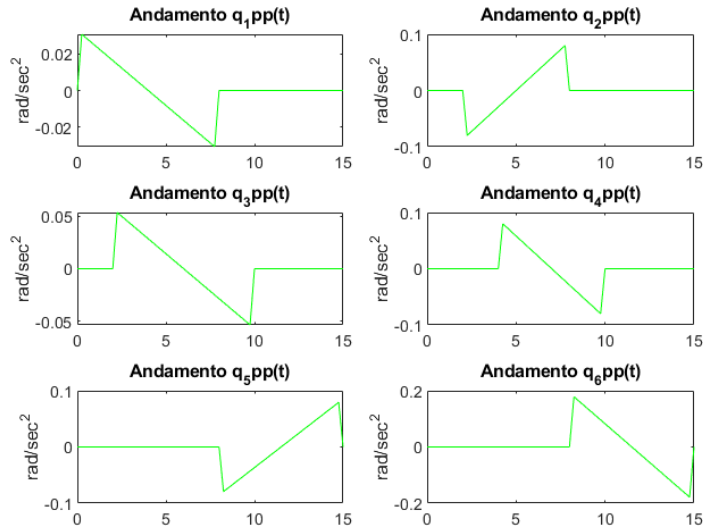


Figura 2.7: Variazione accelerazione sui giunti nel tempo secondo la legge *STEP*.

Ovviamente per ottenere tali andamenti la funzione *STEP* è stata definita anche in *Matlab* secondo la (2.13) in un apposito file. Noti i valori delle variabili sui giunti è possibile utilizzare la funzione ricavata dallo script simbolico per il calcolo di τ e, come per la soluzione della cinematica, anche per la dinamica inversa si risolveranno le equazioni in un intervallo di tempo t per un fissato numero di step, ottenendo i seguenti andamenti delle coppie nel tempo sui sei giunti del manipolatore.

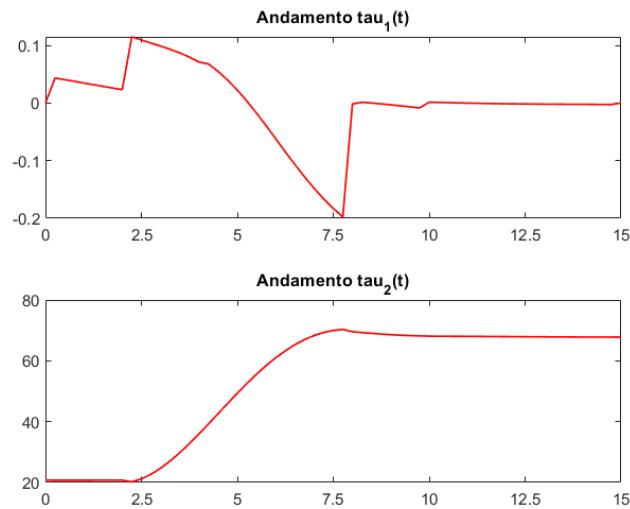


Figura 2.8: Andamento delle coppie sui giunti 1 e 2

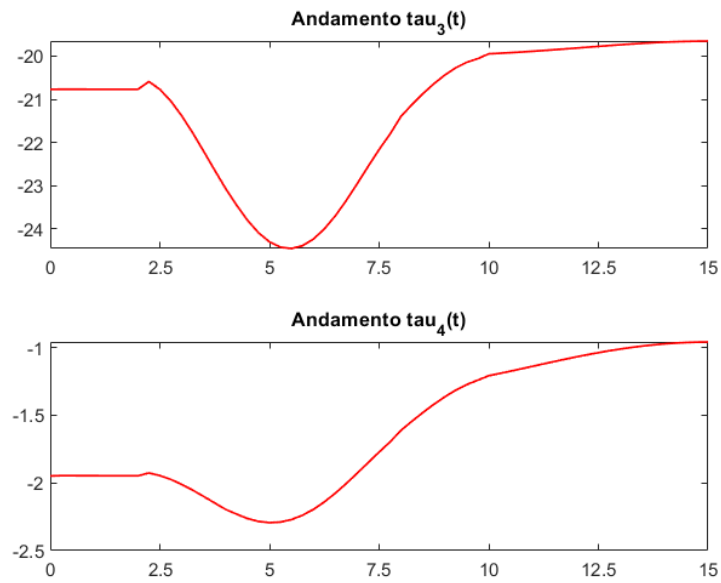


Figura 2.9: Andamento delle coppie sui giunti 3 e 4

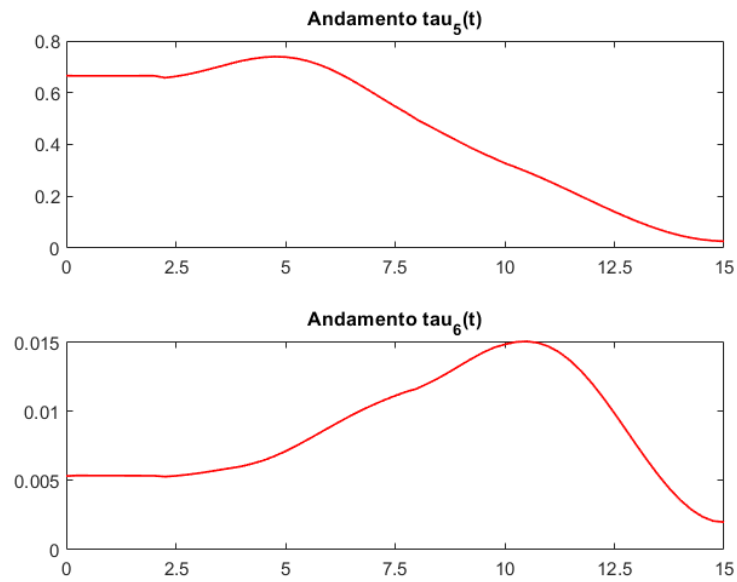


Figura 2.10: Andamento delle coppie sui giunti 5 e 6

I risultati mostrati sono ottenuti impostando il tempo di simulazione pari a 15 secondi, con 60 step di calcolo, e i seguenti valori delle variabili di giunto q_0 e q_f e tempi t_0 , t_f per la funzione *STEP*, con conseguente derivazione delle velocità \dot{q} e accelerazioni \ddot{q} .

Tabella 2.3: Valori variabili di giunto e tempi simulazione.

q_0 [°]	q_f [°]	t_0 [s]	t_f [s]
0	20	0	8
0	-30	2	8
-35	0	2	10
20	50	4	10
0	-40	8	15
0	90	8	15

In conclusione il modello dinamico del robot FANUC CRX-10iA/L è composto da cinque file *Matlab*:

- *Formulazione_Symb_Lagrange.mlx*, il file permette di ricavare l'espressione simbolica delle τ calcolando l'energia cinetica e potenziale del manipolatore e ricavandone la funzione lagrangiana;
- *Masse_e_Inerzie_CRX10.mlx*, al suo interno sono definite tutte le grandezze inerziali del sistema e viene richiamato all'interno del file *Dinamica_CRX.mlx* per effettuare le operazioni numeriche;
- *R_eulero.mlx*, è la funzione che permette di calcolare la matrice di rotazione utile ad esprimere le matrici di inerzia dei corpi rispetto alla terna assoluta $\{0\}$. Necessaria in quanto le matrici di inerzia sono note solamente rispetto alle terne baricentriche;
- *Legge_qt.mlx*, è la funzione che ricrea in ambiente *Matlab* la legge *STEP* definita in *Adams*, permettendo di ottenere gli andamenti delle variabili di giunto mostrati nelle figure (Fig.2.5),(Fig.2.7) e (Fig.2.7)
- *Dinamica_CRX.mlx*, questo è il file al cui interno viene effettivamente risolto il problema di dinamica inversa e si ottengono i plot dei risultati mostrati nelle figure (2.8), (2.9) e (2.10).

Il file *tau_function.mlx* è generato dal comando *matlabFunction* nello script simbolico e rappresenta la funzione per il calcolo numerico delle coppie in *Dinamica_CRX.mlx*.

Capitolo 3

Verifica dei modelli

3.1 Modello ADAMS

Il programma *MSC Adams* è proposto dall'azienda *MSC Software* come strumento per la simulazione della dinamica multicorpo e risulta essere uno fra gli strumenti più utilizzati nell'industria per la sua polivalenza e per l'ampio range di utilizzo che lo caratterizza. *Adams* migliora l'efficienza della progettazione e riduce i costi di sviluppo del prodotto, consentendo una prima convalida della progettazione a livello di sistema; oltre alle ampie capacità di analisi, il software è ottimizzato per affrontare problemi su larga scala, sfruttando gli ambienti di calcolo ad alte prestazioni. Per ogni specifica applicazione sono previsti dei relativi pacchetti *plug-in* che ottimizzano l'utilizzo del simulatore Multibody Dynamics (MDB): come ad esempio *Adams Car* per lo studio approfondito della dinamica del veicolo o *Adams Mechatronics* che può essere utilizzato per integrare facilmente i sistemi di controllo nei modelli meccanici.

Il software permette la modellazione o l'importazione di sistemi meccanici di qualsiasi complessità e da la possibilità di definirne le caratteristiche cinematiche e dinamiche in ambiente 3D, con lo scopo di effettuare simulazioni nell'ambiente virtuale e condurre le analisi desiderate sul meccanismo modellato. Il vantaggio di un modello ben definito nell'ambiente *Adams* sta nella possibilità di utilizzare tale modello per risparmiare sui costi di sviluppo di un prodotto o per validare dei modelli matematici realizzati con altri strumenti. Proprio per quest'ultima ragione il programma risulta di grande interesse per lo svolgimento di questo progetto: ottenere un modello affidabile e ben costituito del FANUC CRX-10iA/L consentirà di deliberare le equazioni della cinematica e della dinamica scritte su *Matlab*.

Per validare, appunto, i modelli appena descritti sarà necessario ricreare nell'ambiente 3D la geometria del manipolatore, assegnando ai corpi i dovuti valori di massa, inerzie e posizione del baricentro, per poi passare alla definizione dei 6 giunti rotoidali e relativi attuatori presenti. Una volta terminata la modellazione sarà possibile confrontare i risultati forniti dalle simulazioni *Adams* con quelli ottenuti dai modelli matematici e verificare la bontà di questi ultimi.

La realizzazione del modello è partita importando all'interno dell'ambiente *Adams* la geometria effettiva del robot FANUC CRX-10iA/L, evitando di modellare la geometrie dei singoli link e permettendo una visualizzazione più concreta e vicina alla realtà della macchina. Ottenuta la geometria si può passare alla definizione dei *joint*, che in questo caso sono 6 e tutti di tipo rotazionale, per ognuno dei quali il software chiede di selezionare i due corpi che verranno connessi ed il punto in cui effettivamente creare il giunto. Per ogni *joint* vengono aggiunte al modello due terne, cioè le due terne di giunto, una sul corpo i -esimo e l'altra appartenente al corpo $i-1$ -esimo; queste due terne verranno rinominate rispetto la seguente codifica:

- "DHi-1": terna del giunto i -esimo definita sul link $i - 1$;
- "MARKER_DHi_i": terna del giunto i -esimo definita sul link i ;

per mantenere la coerenza con le terne definite sul robot prima di scrivere le equazioni di dinamica e cinematica. Si arriverà ad avere sul modello la stessa disposizione delle terne mostrata nella figura 2.2.

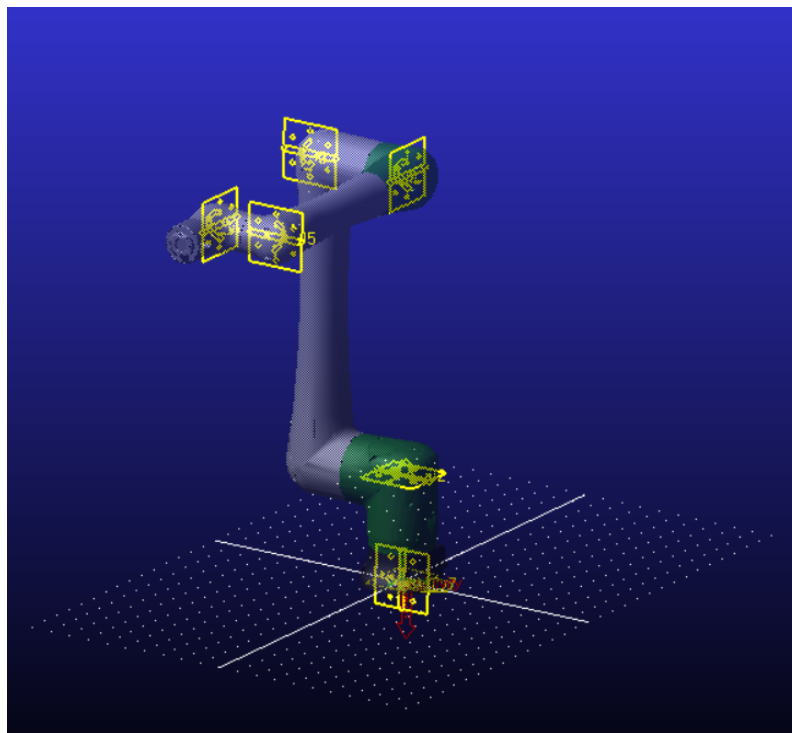


Figura 3.1: Rappresentazione dei giunti definiti sul robot in *Adams*

Ogni giunto permetterà il moto relativo fra le terne DH_i e $MARKER_DH_{i+1}$, e quindi fra i corpi. In particolare si avrà:

- J1: definito fra base e corpo 1, (DH_0 e $MARKER_DH_{0_1}$);
- J2: definito fra corpo 1 e corpo 2, (DH_1 e $MARKER_DH_{1_2}$);
- J3: definito fra corpo 2 e corpo 3, (DH_2 e $MARKER_DH_{2_3}$);
- J4: definito fra corpo 3 e corpo 4, (DH_3 e $MARKER_DH_{3_4}$);
- J5: definito fra corpo 4 e corpo 5, (DH_4 e $MARKER_DH_{4_5}$);
- J6: definito fra corpo 5 e corpo 6, (DH_5 e $MARKER_DH_{5_6}$);

La terna in corrispondenza del terminale viene rinominata DH_6 .

Visivamente la presenza del giunto nel modello è rappresentata dal simbolo di una cerniera posto nel punto in cui è stato definito, una volta definiti tutti e 6 i giunti il modello apparirà come mostrato in figura 3.1.

Dato che nel CAD rappresentante la geometria è presente anche la base fissa tramite la quale il robot può essere meccanicamente fissato ad un piano sarà necessario applicare un vincolo di incastro fra la base e la terna assoluta del sistema, che è definita da *Adams* stesso, per definirla come parte integrante di quello che è stato considerato telaio ed evitare moti relativi fra base e terna assoluta.

Impostata la struttura cinematica del sistema è ora necessario definire gli attuatori, chiamati *motion*, che permettono al robot di muoversi. Basterà semplicemente definire un *rotational joint motion* per ogni giunto presente e verrà associato al *joint* selezionato il corrispettivo *motion* (Figura 3.2). Per ogni *motion* definito sarà possibile indicare una funzione rappresentante la legge di moto da applicare, indicando se riferita a spostamento, velocità o accelerazione.

Definiti tutti e 6 gli attuatori, con le relative leggi di moto, la modellazione è conclusa ed è possibile effettuare una simulazione tramite il pannello *Simulation*. Entrando nella sezione *SimulationControll* si hanno a disposizione diversi parametri ed informazioni riguardanti la simulazione che si sta per effettuare, come la durata o il numero di step di calcolo da effettuare che ovviamente possono essere definiti dall'utente. Questi parametri, assieme alle leggi da assegnare agli attuatori, dovranno essere inseriti in base a come sono state impostate le simulazioni nei modelli *Matlab* per avere la perfetta corrispondenza e poter verificare la correttezza delle equazioni ricavate negli script.

Per poter visualizzare i risultati il software mette a disposizione una finestra di dialogo apposita per il postprocessing, a cui si può accedere tramite il pannello *Result*. In questa sezione si possono effettuare visualizzazioni grafiche del sistema durante la simulazione, oltre che classici plot di grandezze cinematiche o dinamiche sia in funzione del tempo sia in funzione di altri parametri scelti dall'utente. Col fine di validare i modelli *Matlab* si farà ricorso a semplici plot delle grandezze di interesse rispetto al tempo di simulazione.

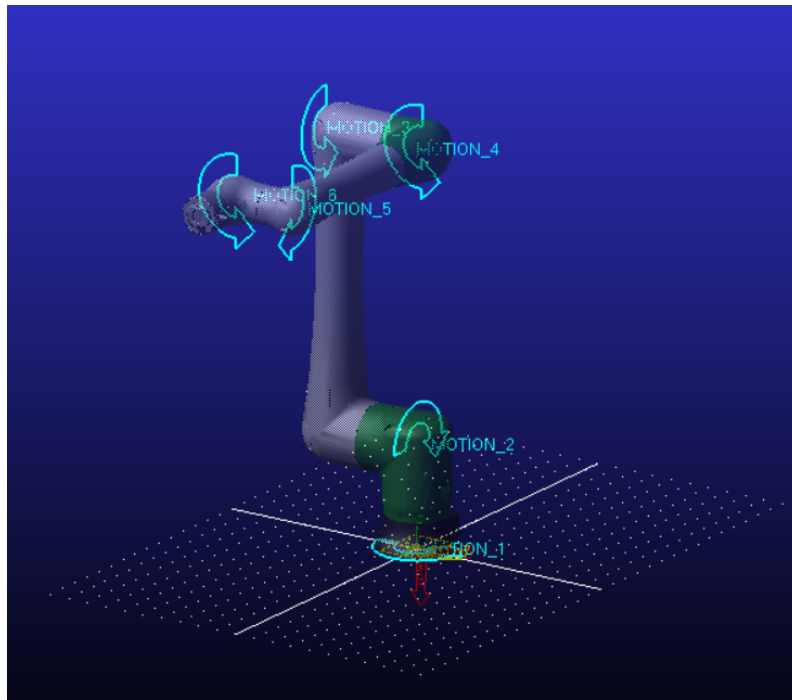


Figura 3.2: Rappresentazione degli attuatori definiti sul robot in *Adams*

3.2 Verifica modello cinematico

Per poter verificare le equazioni della cinematica con il modello *Adams* occorre impostare la simulazione in *Adams* con gli stessi parametri delle simulazioni calcolate tramite gli script per poi confrontare i risultati, avendo la piena corrispondenza delle geometrie modellate. Ai *motion* saranno assegnati valori di velocità angolari costanti, proprio perché il modello cinematico illustrato nella sezione 2.2, si basa su velocità angolari costanti imposte ai giunti. Sarà sufficiente inserire il valore numerico della \dot{q} calcolato in *Matlab* nello spazio dedicato per definire la legge di moto del relativo *motion*, impostando il tipo di controllo su "*Velocity*". Nel lanciare la simulazione si imposta la stessa durata e numero di step utilizzati nel modello matematico, per garantire piena corrispondenza fra le simulazioni.

Per verificare la correttezza delle equazioni cinematiche è necessario confrontare i risultati ottenuti tramite il modello *Matlab* con quelli forniti dal postprocess di *Adams*. I risultati dello script sono forniti tramite plot come quello di figura (2.4), che verrà messo in relazione con la stessa tipologia di plot realizzata in seguito alla simulazione effettuata su *Adams*.

3.2 Verifica modello cinematico

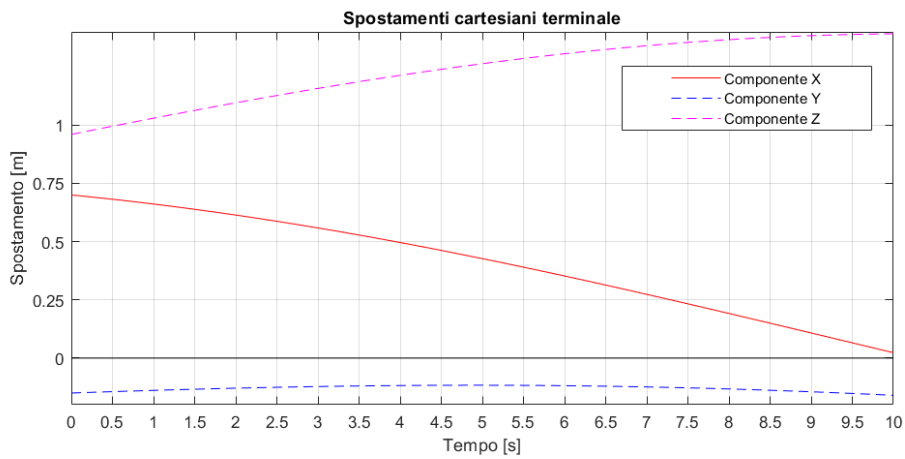


Figura 3.3: Andamento posizione terminale risultante dal modello *Matlab*

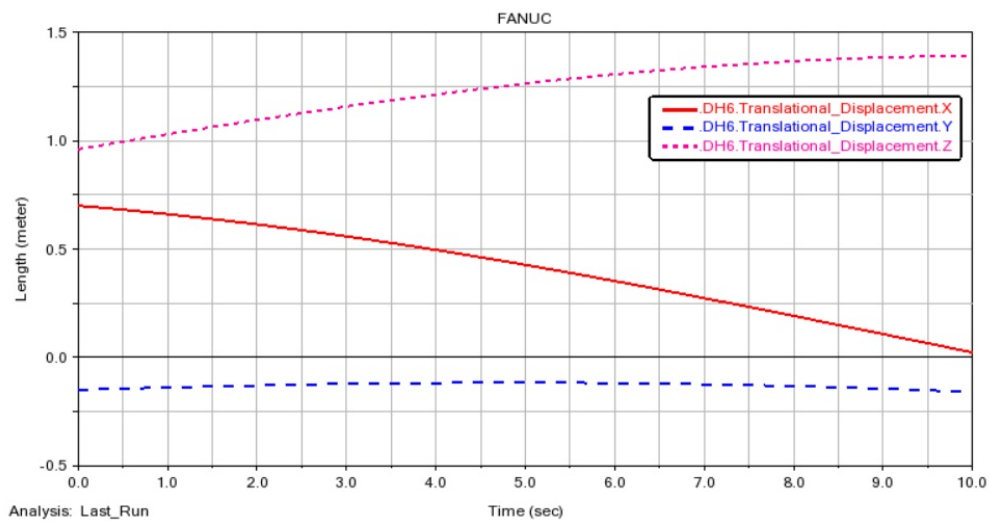


Figura 3.4: Andamento posizione terminale risultante dal modello *Adams*

Oltre a confrontare i valori delle coordinate si paragoneranno anche gli andamenti delle velocità del terminale, per avere un'ulteriore verifica anche tramite le velocità di traslazione.

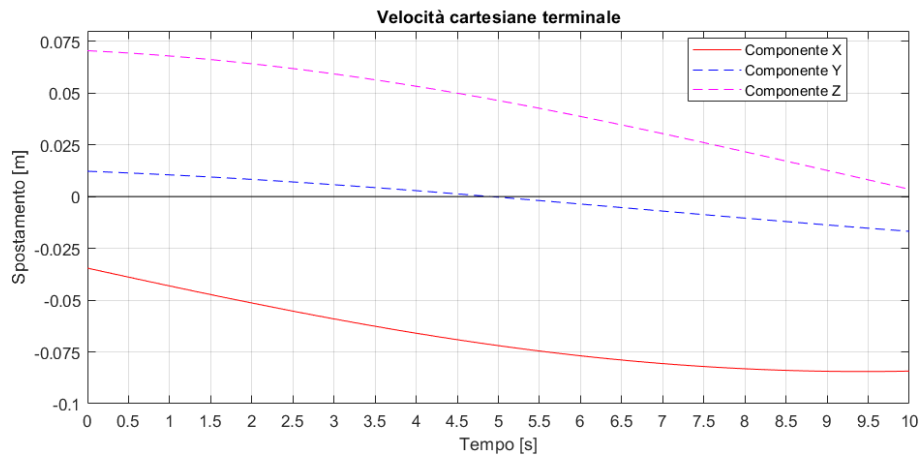


Figura 3.5: Andamento velocità terminale risultante dal modello *Matlab*

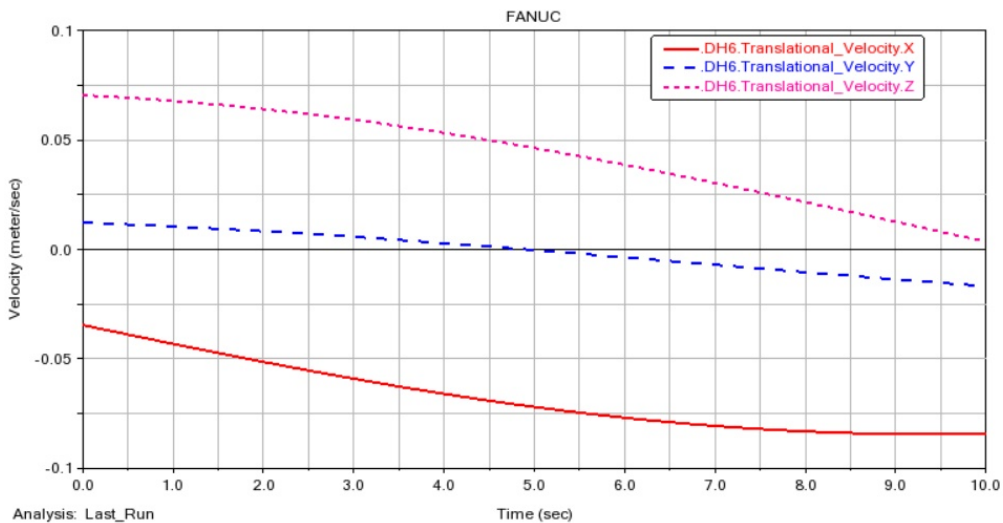


Figura 3.6: Andamento velocità terminale risultante dal modello *Adams*

Tramite il postprocess si realizza il plot dell'andamento delle coordinate (Fig.3.4) e delle velocità di traslazione (Fig.3.6) del terminale al variare del tempo e già effettuato un confronto visivo fra gli andamenti ottenuti con i due modelli è intuibile l'uguaglianza degli andamenti. È anche possibile valutare il valore numerico delle coordinate punto per punto trascinando il cursore sui plot per verificare la perfetta corrispondenza fra i risultati ottenuti. Una prima verifica può essere effettuata osservando le due configurazioni finali del robot nello spazio ottenute con i due modelli senza far riferimento ai plot. Se dovessero apparire simili, come nelle figure (3.7) e (3.8), sarebbe una prima prova della correttezza della cinematica scritta in precedenza nello script.

Configurazione meccanica del robot FanucCRX-10iA/L

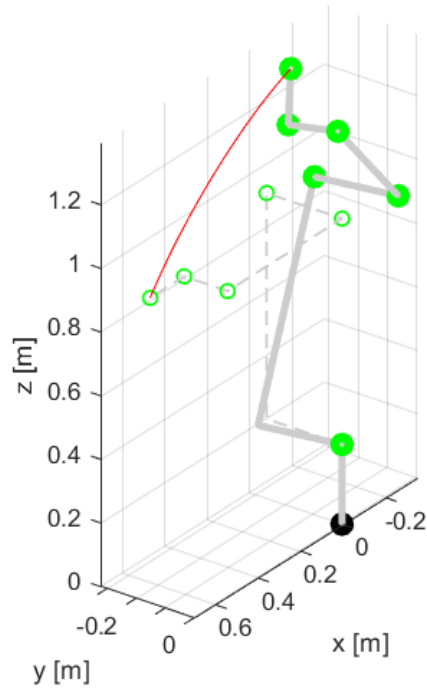


Figura 3.7: Configurazione finale ottenuta in *Matlab*.

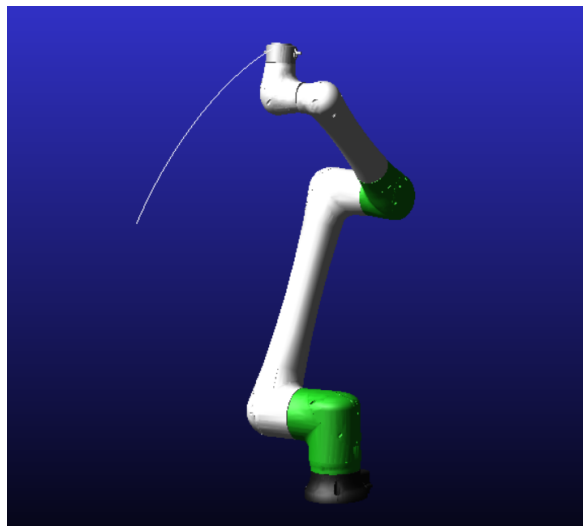


Figura 3.8: Configurazione finale ottenuta in *Adams*.

Tabella 3.1: Valori variabili di giunto simulazione per la validazione.

$q_0[^\circ]$	$q_f[^\circ]$	$\dot{q}[\text{rad/s}]$
0	10	0.0175
90	30	0.0524
0	-15	-0.0262
0	5	0.0087
0	60	0.1047
0	15	0.0262

Nella tabella 3.1 sono raccolti i dati usati nei due modelli per validare i risultati. In particolare i valori di \dot{q} sono quelli che devono essere assegnati ai *motion* in *Adams* per avere la stessa movimentazione prevista dal modello *Matlab*

3.3 Verifica modello dinamico

Come fatto per la cinematica, anche per la dinamica sarà necessario effettuare una simulazione i cui risultati saranno confrontati con quelli del modello illustrato nella sezione 2.3; e anche in questo caso il modello *Adams* richiederà i giusti dati in input per poter confrontare i risultati ed eventualmente validare le equazioni della dinamica. È importante sottolineare che per poter verificare dinamica i dati relativi alle caratteristiche inerziali del robot devono essere uguali, come per la cinematica si richiedeva la stessa geometria dei due modelli; per questo i valori numerici delle masse e inerzie dei membri del manipolatore considerati nello script di *Matlab* sono quelli calcolati direttamente dal software *Adams* caratteristici della geometria importata e del materiale assegnato.

Per la definizione della legge sui giunti è stata utilizzata la legge STEP, definendo 6 leggi di variazione degli spostamenti, una per giunto, con gli stessi parametri chiave x, x_0, x_1, h_0 e h_1 usati nella simulazione *Matlab*. Dovendo verificare le coppie calcolate tramite la formulazione di lagrange (2.6), le coppie in uscita dalla simulazione *Adams* dovranno essere riferite alla terna i -esima, dunque occorre misurare le coppie rispetto alle relative terne locali. Questo è possibile accedendo alla sezione *Measures* del pannello *Design Exploration* e indicando la grandezza da voler misurare e la terna rispetto alla quale esprimerla. Effettuata la simulazione su *Adams* è possibile visualizzare i risultati delle grandezze misurate direttamente dal postprocessor.

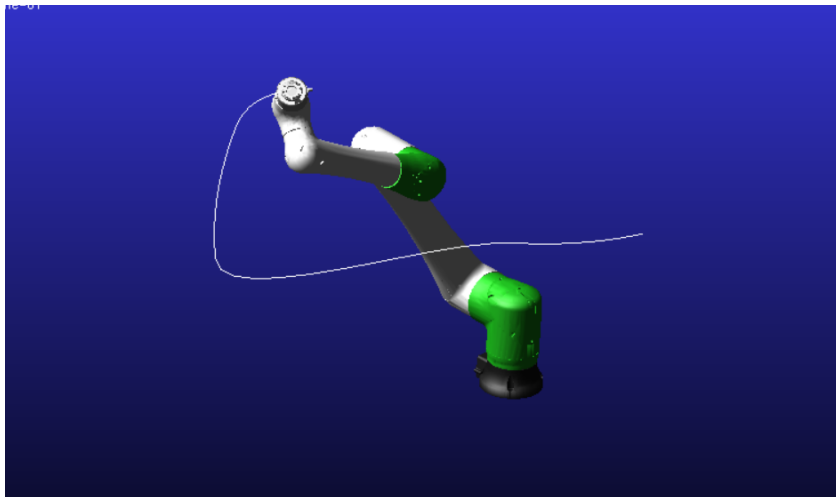


Figura 3.9: Configurazione finale ottenuta in *Adams* per verifica dinamica.

Segue il confronto fra i plot delle coppie ottenute in *Matlab* (Fig.3.10,3.12,3.14,3.16,3.18,3.20) con quelle simulate in *Adams* (Fig.3.11,3.13,3.15,3.17,3.19,3.21).

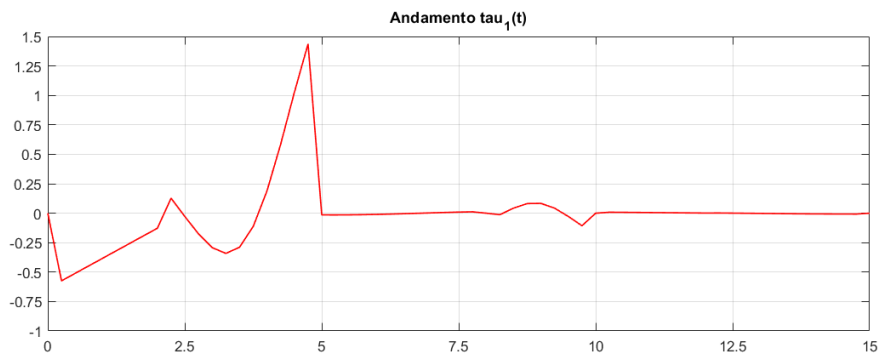


Figura 3.10: Coppia sul giunto 1 calcolata dal modello *Matlab*

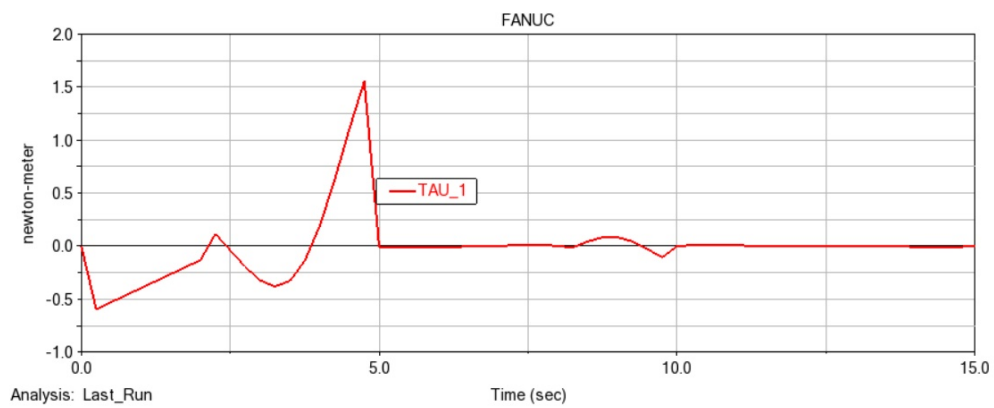


Figura 3.11: Coppia sul giunto 1 ottenuta dal modello *Adams*

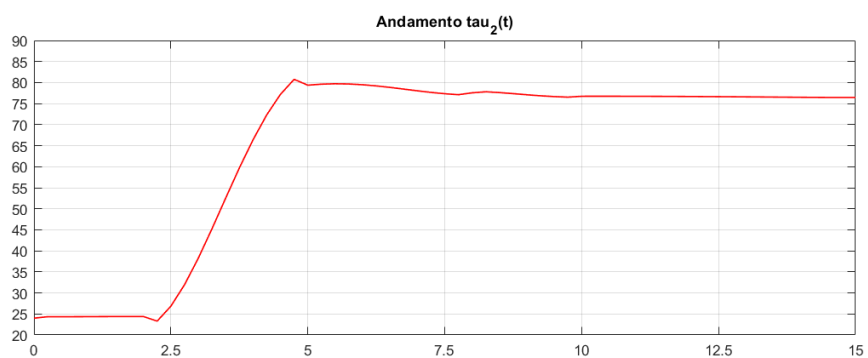


Figura 3.12: Coppia sul giunto 2 calcolata dal modello *Matlab*

3.3 Verifica modello dinamico

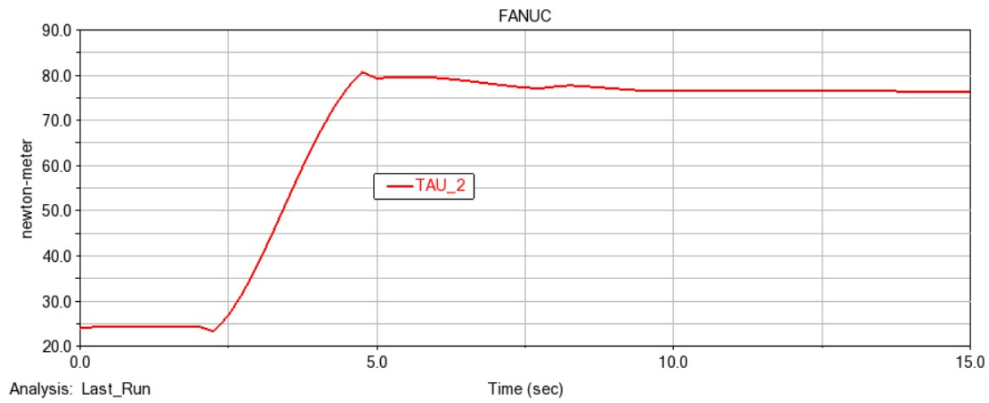


Figura 3.13: Coppia sul giunto 2 ottenuta dal modello *Adams*

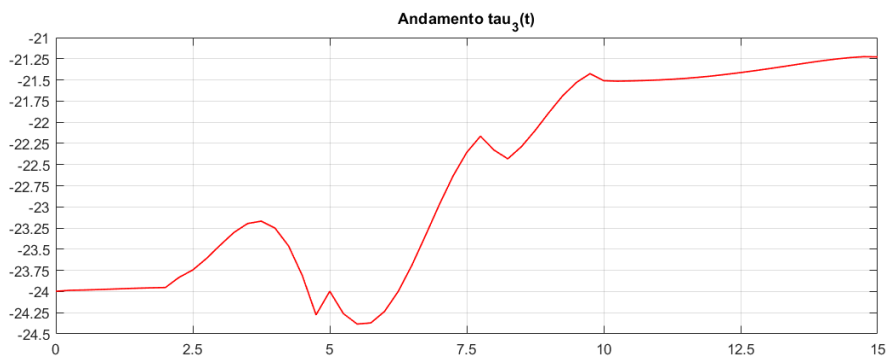


Figura 3.14: Coppia sul giunto 3 calcolata dal modello *Matlab*

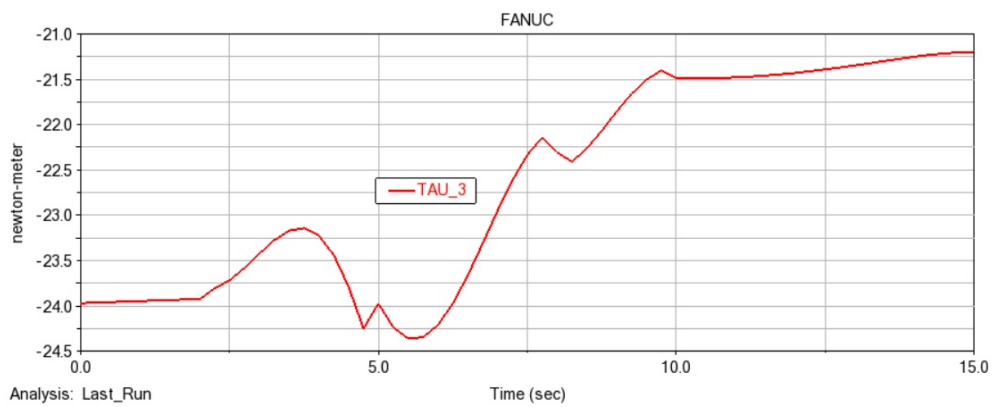


Figura 3.15: Coppia sul giunto 3 ottenuta dal modello *Adams*

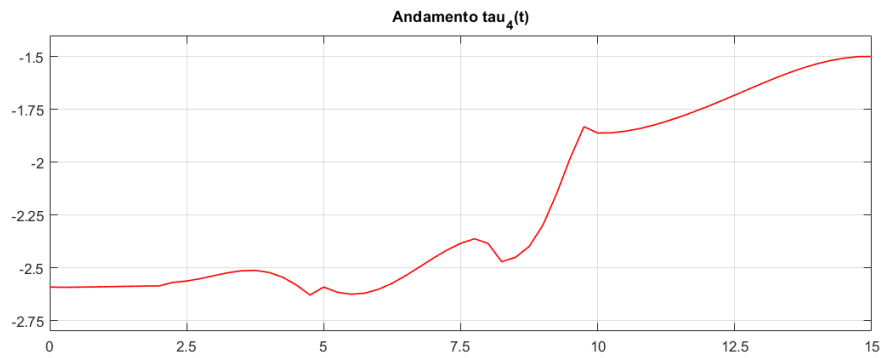


Figura 3.16: Coppia sul giunto 4 calcolata dal modello *Matlab*

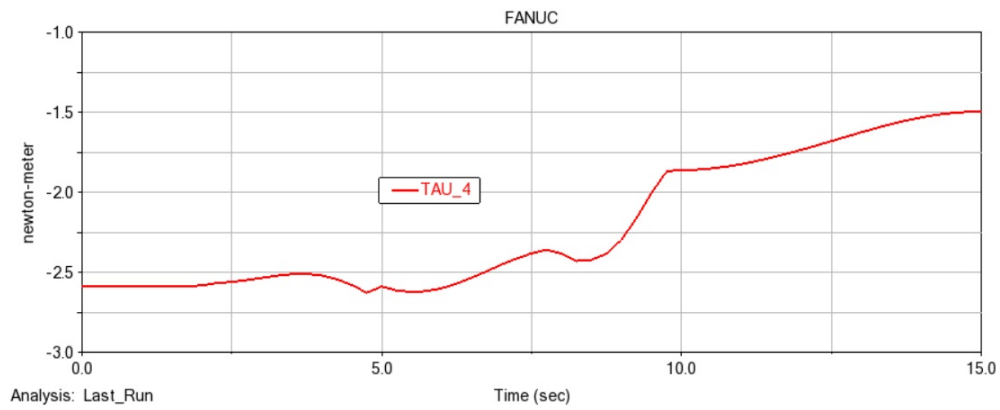


Figura 3.17: Coppia sul giunto 4 ottenuta dal modello *Adams*

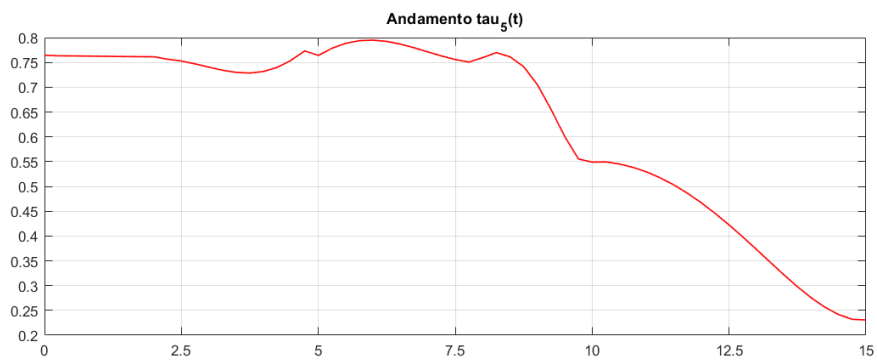


Figura 3.18: Coppia sul giunto 5 calcolata dal modello *Matlab*

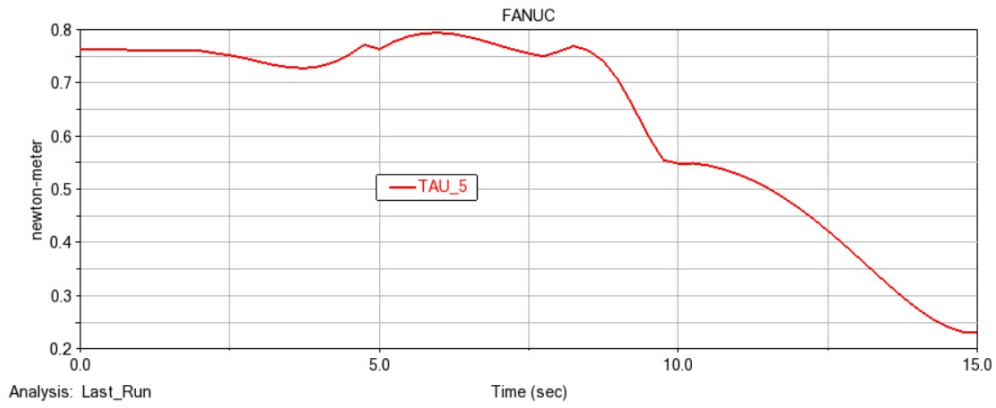


Figura 3.19: Coppia sul giunto 5 ottenuta dal modello *Adams*

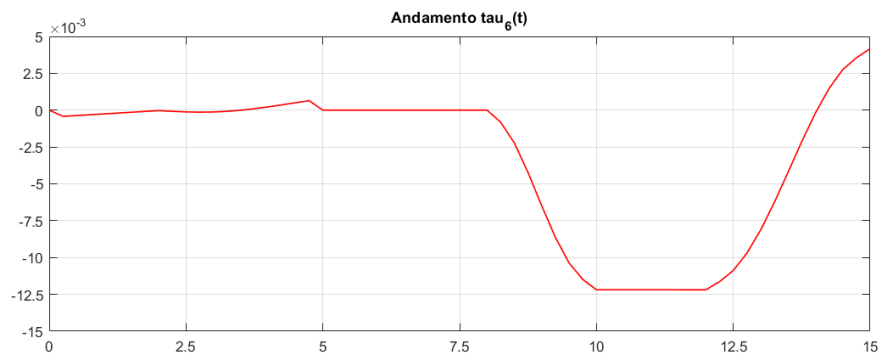


Figura 3.20: Coppia sul giunto 6 calcolata dal modello *Matlab*

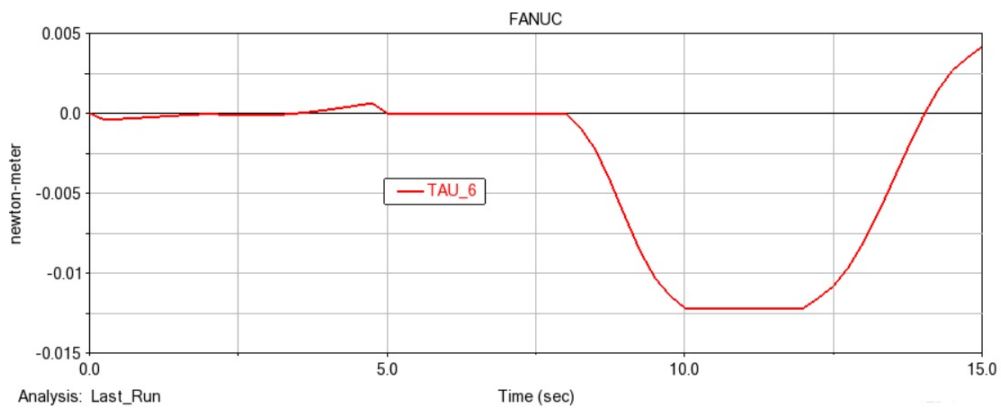


Figura 3.21: Coppia sul giunto 6 ottenuta dal modello *Adams*

Gli andamenti dei grafici risultano perfettamente congruenti fra loro, si possono riscontrare piccole differenze numeriche dovute ai due differenti metodi di approssimazione dei software che comunque rimangono all'interno di un range tollerabile e permette di validare le equazioni ottenute tramite la formulazione di Lagrange. Inoltre, i delta maggiori si riscontrano sulla coppia al giunto 1, questo perché un aumento repentino dei valori di coppia, come riportato nei grafici in figura (3.10) e (3.11) risulta più difficile da calcolare e validare; vista l'elevata corrispondenza delle altre coppie calcolate, eseguendo una simulazione che prevede un andamento di τ_1 più dolce si riscontrerebbe un delta più basso.

Tabella 3.2: Valori variabili di giunto e tempi simulazione per la validazione.

$q_0[^\circ]$	$q_f[^\circ]$	$t_0[s]$	$t_f[s]$
0	-90	0	5
0	-35	2	5
15	-60	2	8
0	-45	8	10
0	40	10	15
0	90	12	15

Nella tabella 3.2 sono raccolti i dati usati nei due modelli per validare i risultati; in particolare i valori riportati sono quelli assegnati ai *motion* in *Adams* per calcolare la funzione STEP, mentre nella figura (3.9) era riportata la configurazione finale assunta dal FANUC CRX-10iA/L nella simulazione *Adams*, con evidenziata la traiettoria percorsa dal terminale.

Capitolo 4

Identificazione

4.1 Formulazione del regressore

Verificate le equazioni della dinamica riguardanti il robot è possibile procedere con la definizione della matrice di regressione, avendo la certezza di basare i nuovi script su equazioni che effettivamente rappresentano la dinamica del sistema. Per poter ottenere la matrice di regressione vengono presi due approcci, entrambi sono algoritmi basati sulla formulazione lagrangiana della dinamica.

Il primo è illustrato nel dettaglio in [4] e prevede la costruzione della matrice combinando le n matrici di regressione $Y^{(i)}$, ognuna relativa al corpo i , ottenute esaminando l'espressione simbolica della formulazione lagrangiana.

Infatti nota la (2.5) e la (2.6) è possibile riscrivere la (4.33) nella seguente formula:

$$\left[\frac{d}{dt} \frac{\partial T^{(i)}}{\partial \dot{q}} - \frac{\partial T^{(i)}}{\partial q} + \frac{\partial U^{(i)}}{\partial q} \right]^T = Y^{(i)} \pi^{(i)}, \quad (4.1)$$

ricordando che $\pi^{(i)} \in \mathbb{R}^{10}$ raggruppa i 10 parametri inerziali del corpo i -esimo, mentre $Y^{(i)} \in \mathbb{R}^{n \times 10}$. A questo punto occorre esplicitare le formule di $T^{(i)}$ e di $U^{(i)}$ e poi derivarle per riscrivere la (4.1) linearizzandola rispetto al vettore pi .

Introducendo le seguenti relazioni cinematiche,

$${}^i \omega_i = {}^0 R_i^T \omega_i \quad (4.2)$$

$${}^0 v_{G_i} = {}^0 v_i + {}^0 \omega_i \times {}^0 p_{G_i} = J_{v_i} \dot{q} + J_{\omega_i} \dot{q} \times {}^0 R_i^0 p_{G_i} \quad (4.3)$$

$${}^0 \omega = J_{\omega_i} \dot{q} \quad (4.4)$$

dove J_{v_i} rappresenta la parte lineare della matrice jacobiana e J_{ω_i} quella angolare, nella (2.7) l'energia cinetica può essere riscritta come:

$$T^{(i)} = \frac{1}{2} m_i (J_{v_i} \dot{q} + J_{\omega_i} \dot{q} \times {}^0 R_i^0 p_{G_i})^T (J_{v_i} \dot{q} + J_{\omega_i} \dot{q} \times {}^0 R_i^0 p_{G_i}) + \frac{1}{2} (J_{\omega_i}^T {}^0 R_i^i I_{G_i}^0 R_i^T J_{\omega_i}) \dot{q} \quad (4.5)$$

Capitolo 4 Identificazione

Introducendo la matrice antisimmetrica $S(x) \in \mathbb{R}^{3 \times 3}$, tale che $S(x)y = S^T(y)x = x \times y$ e che permette di passare alla forma

$$\begin{aligned} T^{(i)} &= \frac{1}{2}m_i\dot{q}^T(J_{v_i}^T J_{v_i})\dot{q} - \frac{1}{2}m_i\dot{q}^T(J_{v_i}^T S(^0R_i)p_{G_i} J_{v_i})\dot{q} \\ &+ \frac{1}{2}m_i\dot{q}^T\{J_{\omega_i}^T S(^0R_i)p_{G_i} J_{\omega_i}\}\dot{q} \\ &+ \frac{1}{2}\dot{q}^T\{J_{\omega_i}^T R_i(I_{G_i} + m_i S^T(p_{G_i})S(p_{G_i}))^0 R_i^T J_{\omega_i} J_{\omega_i}\}\dot{q} \end{aligned} \quad (4.6)$$

da cui poter ricavare la derivata rispetto a \dot{q} e t e quella rispetto a q .

Si può arrivare ad esprimere il primo termine della (4.1) linearmente rispetto al vettore $\pi^{(i)}$.

$$\left[\frac{d}{dt} \frac{\partial T^{(i)}}{\partial \dot{q}} \right]^T = (J_{v_i}^T J_{v_i})\dot{q}m_i + \{J_{v_i}^T S(J_{\omega_i}\dot{q})^0 R_i - J_{\omega_i}^T S(J_{v_i}\dot{q})^0 R_i\}m_i p_{G_i} + J_{\omega_i}^T R_i^i I_i^0 R_i^T J_{\omega_i}\dot{q} \quad (4.7)$$

Definendo il tensore $E \in \mathbb{R}^{3 \times 3 \times 6}$ ed esprimendo la matrice di inerzia del corpo i rispetto alla terna i -esima come,

$${}^i I_i = E J \quad (4.8)$$

con

$$J_i = \begin{bmatrix} J_{ixx} & J_{ixy} & J_{ixz} & J_{iyy} & J_{iyz} & J_{izz} \end{bmatrix}^T$$

$$E = \begin{bmatrix} E_1 & E_2 & E_3 & E_4 & E_5 & E_6 \end{bmatrix}$$

$$E1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad E2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad E3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (4.9)$$

$$E4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad E5 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad E6 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Il terzo termine della (4.7), diventa:

$$J_{\omega_i}^T R_i^i I_i^0 R_i^T J_{\omega_i}\dot{q} = \left[J_{\omega_i}^T R_i E^0 R_i^T J_{\omega_i}\dot{q} \right] J_i = \left[J_{\omega_i}^T R_i E_1^0 R_i^T J_{\omega_i}\dot{q} \mid \dots \mid J_{\omega_i}^T R_i E_6^0 R_i^T J_{\omega_i}\dot{q} \right] J_i \quad (4.10)$$

Ora è possibile estrarre i parametri inerziali dal primo termine della formulazione lagrangiana:

$$\left[\frac{d}{dt} \frac{\partial T^{(i)}}{\partial \dot{q}} \right]^T = \dot{X}_0^i \pi_0^{(i)} + \dot{X}_1^i \pi_1^{(i)} + \dot{X}_2^i \pi_2^{(i)} \quad (4.11)$$

dove

$$\begin{aligned}
 X_0^i &= (J_{v_i}^T J_{v_i}) \dot{q}, \in \mathbb{R} \\
 X_1^i &= \{J_{v_i}^T S(J_{\omega_i} \dot{q})^0 R_i - J_{\omega_i}^T S(J_{v_i} \dot{q})^0 R_i\}, \in \mathbb{R}^3 \\
 X_2^i &= J_{\omega_i}^T R_i \left[E_1 | E_2 | E_3 | E_4 | E_5 | E_6 \right]^0 R_i^T J_{\omega_i} \dot{q} \in \mathbb{R}^6 \\
 \pi_0^{(i)} &= m_i, \in \mathbb{R} \\
 \pi_1^{(i)} &= \left[m_i p_{G_i x} \quad m_i p_{G_i y} \quad m_i p_{G_i z} \right]^T, \in \mathbb{R}^3 \\
 \pi_2^{(i)} &= J_i, \in \mathbb{R}^6
 \end{aligned} \tag{4.12}$$

Per il secondo termine a destra della (4.1), si procede allo stesso modo, derivando l'espressione (4.6) rispetto a q , arrivando all'espressione seguente.

$$\left[\frac{\partial T^{(i)}}{\partial q} \right]^T = W_0^i \pi_0^{(i)} + W_1^i \pi_1^{(i)} + W_2^i \pi_2^{(i)} \tag{4.13}$$

dove

$$\begin{aligned}
 W_0^i &= \frac{1}{2} \dot{q}^T \begin{bmatrix} \frac{\partial}{\partial q_1} (J_{v_i}^T J_{v_i}) \\ \vdots \\ \frac{\partial}{\partial q_n} (J_{v_i}^T J_{v_i}) \end{bmatrix} \dot{q}, \in \mathbb{R} \\
 W_1^i &= \frac{1}{2} \begin{bmatrix} \frac{\partial}{\partial q_1} \left[{}^0 R_i^T S^T(J_{\omega_i} \dot{q}) J_{v_i} \dot{q} - {}^0 R_i^T S^T(J_{v_i} \dot{q}) J_{\omega_i} \dot{q} \right] \\ \vdots \\ \frac{\partial}{\partial q_n} \left[{}^0 R_i^T S^T(J_{\omega_i} \dot{q}) J_{v_i} \dot{q} - {}^0 R_i^T S^T(J_{v_i} \dot{q}) J_{\omega_i} \dot{q} \right] \end{bmatrix} \dot{q}, \in \mathbb{R}^3 \\
 W_2^i &= \frac{1}{2} \dot{q}^T \begin{bmatrix} \frac{\partial}{\partial q_1} (J_{\omega_i}^T R_i E^0 R_i^T J_{\omega_i}) \\ \vdots \\ \frac{\partial}{\partial q_n} (J_{\omega_i}^T R_i E^0 R_i^T J_{\omega_i}) \end{bmatrix} \dot{q}, \in \mathbb{R}^6
 \end{aligned} \tag{4.14}$$

Per l'energia potenziale viene seguito lo stesso approccio partendo dall'espressione generale di U^i scritta come

$$U^i = -m_i g^T ({}^0 p_i + {}^0 R_i p_{G_i}) \quad (4.15)$$

Derivando per q si ottiene il terzo termine a destra della (4.1):

$$\left[\frac{\partial U^{(i)}}{\partial q} \right]^T = -m_i \left\{ g^T \frac{\partial {}^0 p_i}{\partial q} + g^T \frac{\partial {}^0 R_i}{\partial q} p_{G_i} \right\}^T = -J_{v_i}^T g m_i - \left[\frac{\partial (g^T {}^0 R_i)}{\partial q} m_i p_{G_i} \right]^T \quad (4.16)$$

Che a sua volta può essere espresso linearmente rispetto a $\pi^{(i)}$ tramite la formula

$$\left[\frac{\partial U^{(i)}}{\partial q} \right]^T = Z_0^i \pi_0^{(i)} + Z_1^i \pi_1^{(i)} \quad (4.17)$$

dove

$$Z_0^i = -J_{v_i}^T g, \in \mathbb{R}$$

$$Z_1^i = - \begin{bmatrix} \left(\frac{\partial}{\partial q_1} ({}^0 R^T g) \right)^T \\ \vdots \\ \left(\frac{\partial}{\partial q_n} ({}^0 R^T g) \right)^T \end{bmatrix}, \in \mathbb{R}^3 \quad (4.18)$$

Dopo che tutte le manipolazioni sono state effettuate correttamente è possibile riscrivere la formulazione dinamica lagrangiana per il link i come

$$\left[\frac{d}{dt} \frac{\partial T^{(i)}}{\partial \dot{q}} - \frac{\partial T^{(i)}}{\partial q} + \frac{\partial U^{(i)}}{\partial q} \right]^T = Y^{(i)} \pi^{(i)} = \begin{bmatrix} Y_0^{(i)} & Y_1^{(i)} & Y_2^{(i)} \end{bmatrix} \begin{bmatrix} \pi_0^i \\ \pi_1^i \\ \pi_2^i \end{bmatrix} \quad (4.19)$$

dove

$$Y_0^{(i)} = \dot{X}_0^{(i)} - W_0^{(i)} + Z_0^{(i)}, \in \mathbb{R}$$

$$Y_1^{(i)} = \dot{X}_1^{(i)} - W_1^{(i)} + Z_1^{(i)}, \in \mathbb{R}^3 \quad (4.20)$$

$$Y_2^{(i)} = \dot{X}_2^{(i)} - W_2^{(i)}, \in \mathbb{R}^6$$

La matrice di regressione del manipolatore è ottenibile combinando le n matrici ottenute.

$$Y(q, \dot{q}, \ddot{q}) = \begin{bmatrix} Y^1 & \dots & Y^n \end{bmatrix} \quad (4.21)$$

come anche il vettore contenente tutti i parametri del robot

$$\pi = \left[\pi^{(1)T} \dots \pi^{(n)T} \right]^T \quad (4.22)$$

Questo algoritmo risulta molto efficace in quanto può essere applicato in maniera sistematica a qualsiasi sistema sia stato modellato con la formulazione lagrangiana, basterà infatti prendere le definizioni riportate in (4.20) per poi assemblare la matrice di regressione. Dal punto di vista di implementazione in *Matlab* risulta alquanto complesso riportare in uno script simbolico tutte le espressioni necessarie, inoltre, la difficoltà maggiore la si avrebbe nell'ottenere l'espressione simbolica finale del regressore. In generale gli script simbolici risultano molto pesanti e complessi da poter essere calcolati, meglio preferire una formulazioni più semplice e lineare per ottenere la matrice di regressione in forma simbolica.

In [5] viene illustrata una procedura, anch'essa sistematica, per ricavare la matrice di regressione più adatta ad essere riportata in uno script simbolica di *Matlab*. Partendo sempre dalla funzione lagrangiana del generico link, l'obiettivo è sempre quello di ottenere le espressioni dei parametri inerziali in forma lineare per poterli isolare e assemblare poi il regressore.

Come indicato in [5], esprimendo le grandezze cinematiche presenti nella formula dell'cinetica del generico link i rispetto alla relativa terna locale i si ottiene la linearità dei parametri inerziali: infatti sostituendo le relazioni,

$$\begin{aligned} {}^i\omega_i &= R_i^{T0}\omega_i \\ {}^i v_{c_i} &= {}^i v_i + {}^i\omega_i \times {}^i p_{G_i} \end{aligned} \quad (4.23)$$

nell'espressioni di $T^{(i)}$ ed introducendo la matrice antisimmetrica $S(x)$, l'energia cinetica può essere valutata come:

$$T^{(i)} = \frac{1}{2}m_i {}^i v_i^{T^i} v_i + {}^i v_i^T S({}^i\omega_i) m_i p_{G_i} + \frac{1}{2} {}^i\omega_i^{T^i} I_i {}^i\omega_i \quad (4.24)$$

Per l'energia potenziale, sfruttando la relazione cinematica seguente

$${}^0 p_{G_i} = {}^0 p_{0_i} + R_i^i p_{G_i} \quad (4.25)$$

si può scrivere

$$U^{(i)} = -g^T (m_i {}^0 p_{0_i} + R_i^i m_i p_{G_i}) \quad (4.26)$$

Sia nella (4.24) sia nella (4.26) i parametri inerziali appaiono in forma lineare, quindi è possibile scrivere la funzione di Lagrange nella forma

$$L = \sum_{i=1}^n (\beta_{T^{(i)}} - \beta_{U^{(i)}}) \pi \quad (4.27)$$

I termini sono due 10×1 , ricavati dalle espressioni delle energie cinetiche e potenziali, che permettono di scrivere la Lagrangiana come funzione lineare di π . Entrambi sono funzioni solamente delle grandezze cinematiche: $\beta_{T^{(i)}} = \beta_{T^{(i)}}(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n)$ e $\beta_{U^{(i)}} = \beta_{U^{(i)}}(q_1, \dots, q_n)$.

Effettuando le derivare previste sulla (4.27) dalla formulazione di Lagrange è

possibile ricavare la coppia sul giunto i

$$\tau^{(i)} = \sum_{j=1}^n y_j^{(i)T} \pi^{(i)} \quad (4.28)$$

Tenendo in conto che

$$y_j^{(i)} = \frac{d}{dt} \frac{\partial \beta_{T^{(j)}}}{\partial \dot{q}_i} - \frac{\partial \beta_{T^{(j)}}}{\partial q_i} + \frac{\partial \beta_{U^{(j)}}}{\partial q_i} \quad (4.29)$$

La matrice di regressione si può così comporre, dopo aver ricavato tutte le espressioni di $y_j^{(i)}$

$$Y(q, \dot{q}, \ddot{q}) = \begin{bmatrix} y_1^{(1)T} & y_2^{(1)T} & \dots & y_n^{(1)T} \\ 0^T & y_2^{(2)T} & \dots & y_n^{(2)T} \\ \vdots & \vdots & \ddots & \vdots \\ 0^T & 0^T & \dots & y_n^{(n)T} \end{bmatrix} \quad (4.30)$$

Questa formulazione del regressore è particolarmente adatta in quanto le espressioni delle energie cinetiche e potenziali dei corpi sono già state calcolate in forma simbolica per lo studio della dinamica nello script *Formulazione_Symb_Lagrange.mlx*. Occorrerà solamente cambiare i riferimenti rispetto a cui sono espresse le velocità dei baricentri, linearizzando così le espressioni rispetto alle grandezze inerziali e poi sarà possibile ricavare i vettori $\beta_{T^{(i)}}$ e $\beta_{U^{(i)}}$.

Tutto l'algoritmo è racchiuso in un unico script simbolico dove, partendo dalle energie dei membri, vengono individuati e derivati i vettori $\beta_{T^{(i)}}$ e $\beta_{U^{(i)}}$ e poi combinati per ottenere tutte le $y_j^{(i)}$ che comporranno la matrice simbolica di regressione, la quale una volta definita viene associata ad una funzione *Y_function.m* con la quale si valuterà numericamente la matrice stessa.

$$\pi = (\bar{Y}^T \bar{Y})^{-1} \bar{Y} \bar{\tau} \quad (4.31)$$

Ottenuta la matrice $Y(q, \dot{q}, \ddot{q})$ è possibile effettuare l'identificazione vera e propria dei parametri tramite la formula (4.31) già anticipata in (1.2), che permette di calcolare i valori delle grandezze inerziali conoscendo le coppie ed i valori delle variabili di giunto relativi.

4.2 Identificazione dei parametri base

Come già detto nella sezione [1.2] non tutti i parametri dinamici di un sistema ne influenzano il comportamento dinamico, è possibile individuare tre categorie differenti di parametri:

- unicamente identificabili;
- identificabili in combinazione lineare;
- non identificabili;

Per capire quali parametri appartengono a quale categoria è necessario lavorare sulla matrice di regressione. L'importanza di ricavare il set dei parametri base sta nella possibilità di esprimere la dinamica del sistema tramite la formula:

$$Y_B(q, \dot{q}, \ddot{q})\pi_B = \tau \quad (4.32)$$

e quindi di poter snellire molto il carico computazionale necessario per calcolare o controllare la dinamica del manipolatore, poiché le dimensioni del problema vengono ridotte rispetto a quelle caratterizzanti la forma canonica:

$$Y(q, \dot{q}, \ddot{q})\pi = \tau \quad (4.33)$$

Si passa da un vettore $\pi \in \mathbb{R}^r$ a $\pi_B \in \mathbb{R}^b$ e il regressore $Y(q, \dot{q}, \ddot{q}) \in \mathbb{R}^{n \times r}$ viene ridotto a $Y(q, \dot{q}, \ddot{q}) \in \mathbb{R}^{n \times b}$, con $b < r$.

La riduzione del regressore inizia ispezionando le colonne della matrice ed eliminando quelle totalmente nulle: per ogni colonna nulla nel regressore corrisponde un parametro non influente nella dinamica e quindi anche un parametro inerziale sarà rimosso dal vettore π generico. Se la j -esima colonna in Y risulta totalmente nulla allora il j -esimo parametro in π verrà eliminato poiché non ha influenza sulla dinamica del sistema. Dopo questa operazione si ha una matrice $n \times q$ che può essere ulteriormente esaminata per identificare i parametri base. Un approccio differente per l'identificazione di tali parametri è presentato in [6], dove vengono suggerite delle relazioni in forma chiusa, valide per manipolatori seriali, tramite le quali individuare direttamente il set minimo di parametri. L'algoritmo parte con l'esclusione dei parametri non influenzanti la dinamica per poi passare alla determinazione di condizioni per il raggruppamento dei parametri influenti; condizioni che sono basate sulle derivate delle espressioni di energia cinetica e potenziale rispetto ai parametri stessi e ovviamente in base alle tipologie di giunto caratterizzanti il sistema variano le condizioni con cui raggrupparle le grandezze inerziali.

Un metodo numerico per poter ricavare i parametri base è invece presentato in [7], questo algoritmo è basato sulla *QR decomposition* della matrice di regressione e si presta meglio all'analisi del FANUC CRX-10iA/L, perché avendo già ricavato l'espressione del regressore, basta sfruttare la funzione presente in *Matlab* per eseguire la QR decomposition per arrivare in maniera alquanto lineare alla definizione dei parametri base.

Una volta eliminati i parametri ininfluenti, e quindi in possesso di una matrice $n \times q$, questa viene valutata numericamente in M step di calcolo tramite valori randomici delle coordinate di giunto, facendo sì che risulti $M \gg n$. Le M matrici di regressione vengono tutte impilate in un'unica matrice Y_{QR} , operazione necessaria per avere una matrice caratterizzata da più righe che colonne. A questo punto effettuando la QR decomposition della matrice Y_{QR} si ottiene:

$$Q^T Y_{QR} = \begin{bmatrix} R \\ 0_{(Mn-q) \times q} \end{bmatrix} \quad (4.34)$$

dove Q è una matrice $Mn \times Mn$ ortogonale, quindi con inversa uguale alla sua trasposta e R è una matrice $q \times q$ triangolare superiore, mentre $0_{(Mn-q) \times q}$ è una matrice di zeri.

Per identificare quali parametri possono essere esclusi, poiché identificabili in combinazione lineare, occorre analizzare la diagonale di R :

- $|R_{ii}| \approx 0$: i -esimo parametro risulta non essere identificabile ed è associato ad una colonna di Y_{QR} dipendente;
- $|R_{ii}| \neq 0$: i -esima colonna di Y_{QR} risulta essere indipendente ed è associata ad un parametro identificabile;

Si ottiene la relazione

$$Y_{QR}\pi = \begin{bmatrix} Y_1 & Y_2 \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix} \quad (4.35)$$

dove Y_1 è la matrice $Mn \times b$ contenente le colonne indipendenti di Y_{QR} e Y_2 è la matrice che contiene le colonne dipendenti avente dimensione $Mn \times q - b$.

Le due matrici ottenute, Y_1 e Y_2 , sono messe in relazione dalla matrice β .

$$Y_2 = Y_1\beta \quad (4.36)$$

di conseguenza la (4.35) diventa:

$$Y_{QR}\pi = \begin{bmatrix} Y_1 & Y_2 \end{bmatrix} \begin{bmatrix} \pi_b \\ \pi_0 \end{bmatrix} = Y_1\pi_B \quad (4.37)$$

dove π_B è il vettore dei parametri base e vale $\pi_B = \pi_1 + \beta\pi_2$; quindi è necessario identificare la matrice β prima di poter definire il vettore π_B .

4.2 Identificazione dei parametri base

Eseguendo una nuova QR decomposition della matrice $[Y_1 \ Y_2]$ è si ottiene:

$$\begin{bmatrix} Y_1 & Y_2 \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 & R_2 \\ \mathbf{0}_{(Mn-b) \times b} & \mathbf{0}_{(Mn-b) \times (q-b)} \end{bmatrix} = \begin{bmatrix} Q_1 R_1 & Q_2 R_2 \end{bmatrix} \quad (4.38)$$

Da cui è possibile ottenere

$$\begin{aligned} Q_1 &= Y_1 R_1^{-1} \\ Y_2 &= Q_1 R_2 = Y_1 R_1^{-1} R_2 \end{aligned} \quad (4.39)$$

e richiamando l'equazione (4.36) è possibile calcolare la matrice β come segue.

$$\beta = R_1^{-1} R_2 \quad (4.40)$$

Ovviamente la matrice di regressione associata al vettore dei parametri base π_B è la matrice $Y_B = Y_1$. Seguendo questi passaggi è possibile arrivare alla formula (4.32).

Tutto l'algoritmo è riportato in un unico file *Matlab*, inizialmente viene calcolato il regressore usando la funzione simbolica definita seguendo la procedura illustrata nella sezione [4.1] poi, dopo aver eliminato le colonne nulle ed i relativi parametri, è effettuata la prima QR decomposition usando la funzione *qr* che restituisce le due matrici Q e R , lo script conclude poi con il calcolo di β per individuare il vettore π_B

4.3 Verifica dei risultati

Per poter verificare la riuscita dell'applicazione degli algoritmi nel ricavare il regressore e nel calcolo dei parametri base è sufficiente confrontare i risultati ottenuti dalle formule (4.31) e (4.32), con i valori reali ottenuti dal robot, ovvero dal modello *Adams*.

Per verificare il regressore sono stati ricavati i parametri inerziali tramite la (4.31), in *Matlab* è stato usato il comando *pinv* per ricavare la pseudoinversa della matrice Y e moltiplicando per il vettore delle coppie si sono ottenuti i valori numerici dei parametri inerziali. È possibile visualizzare i valori dei parametri identificati rispetto ai valori effettivi nel grafico che segue. Per valutarne l'accuratezza è stato calcolato l'errore percentuale commesso su ogni parametro:

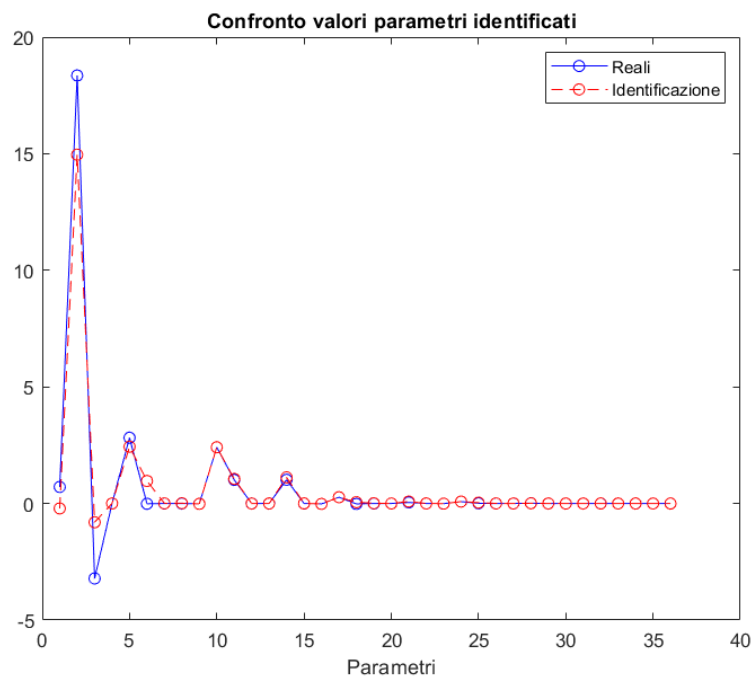


Figura 4.1: Confronto valori grandezze dinamiche identificate

Per verificare la riduzione ed il set di parametri base individuato è stato calcolato l'andamento delle coppie usando la (4.32) e poi sono stati paragonati i grafici con le coppie misurate in *Adams*. Di seguito i grafici in cui sono state messe a confronto: in blu sono riportate le coppie misurate mentre col tratto rosso tratteggiato sono riportate le coppie calcolate con lo script *Matlab*.

Gli andamenti si mostrano praticamente identici, a meno di qualche piccolo discostamento, questo a confermare la bontà della scrittura del regressore, dimostrata già con i risultati in figura (4.1), ma anche quella dell'algoritmo di identificazione dei parametri base.

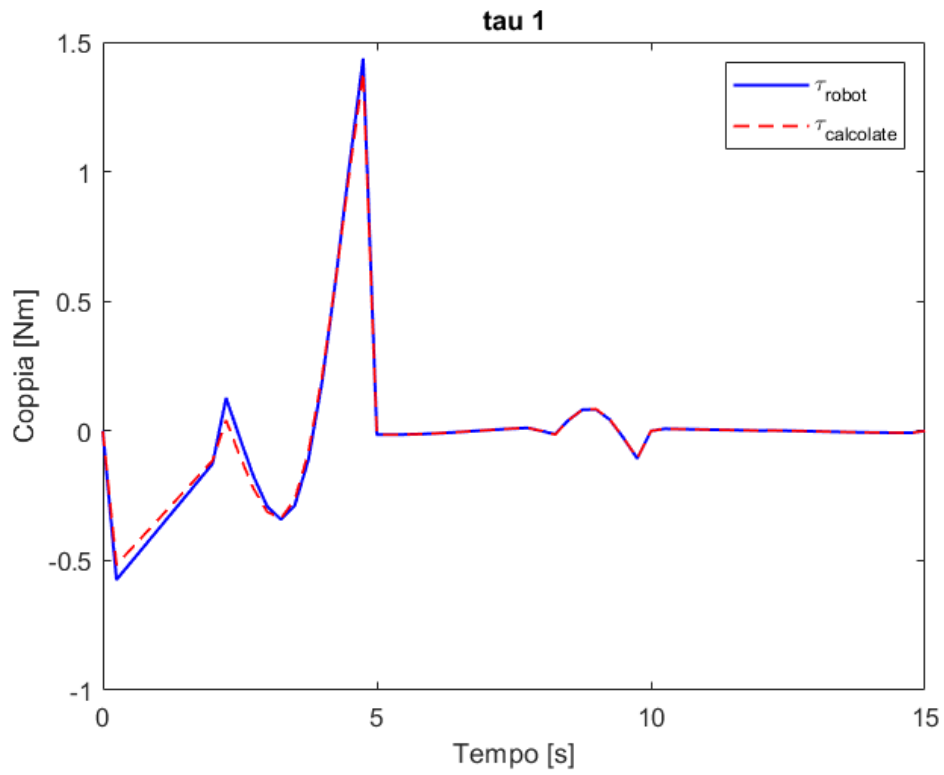


Figura 4.2: Confronto coppia al giunto 1.

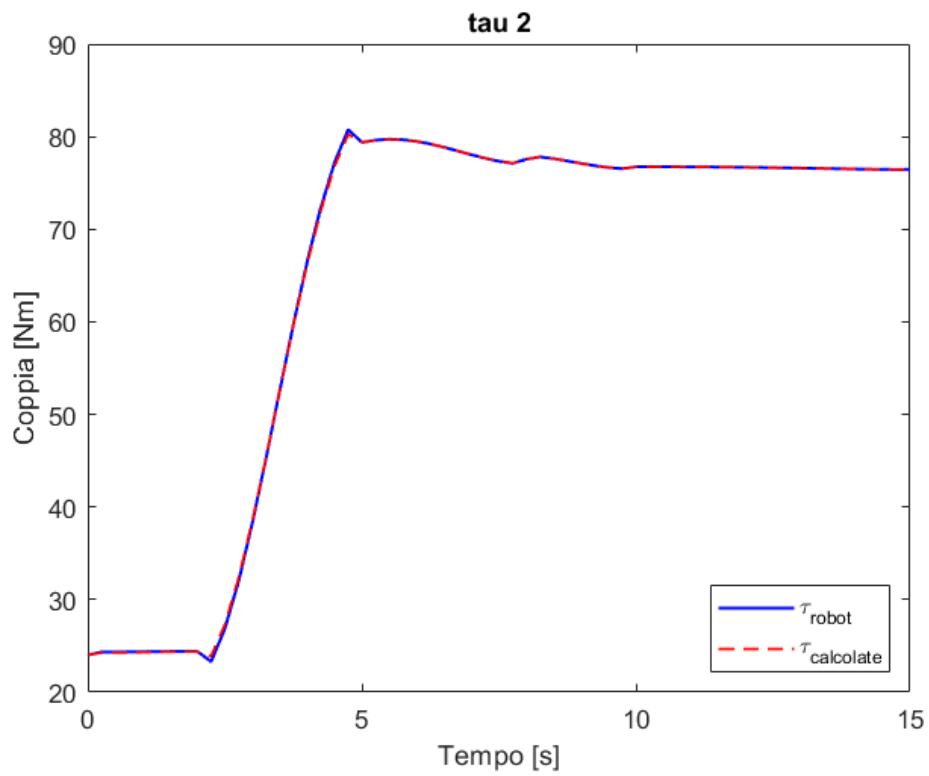


Figura 4.3: Confronto coppia al giunto 2.

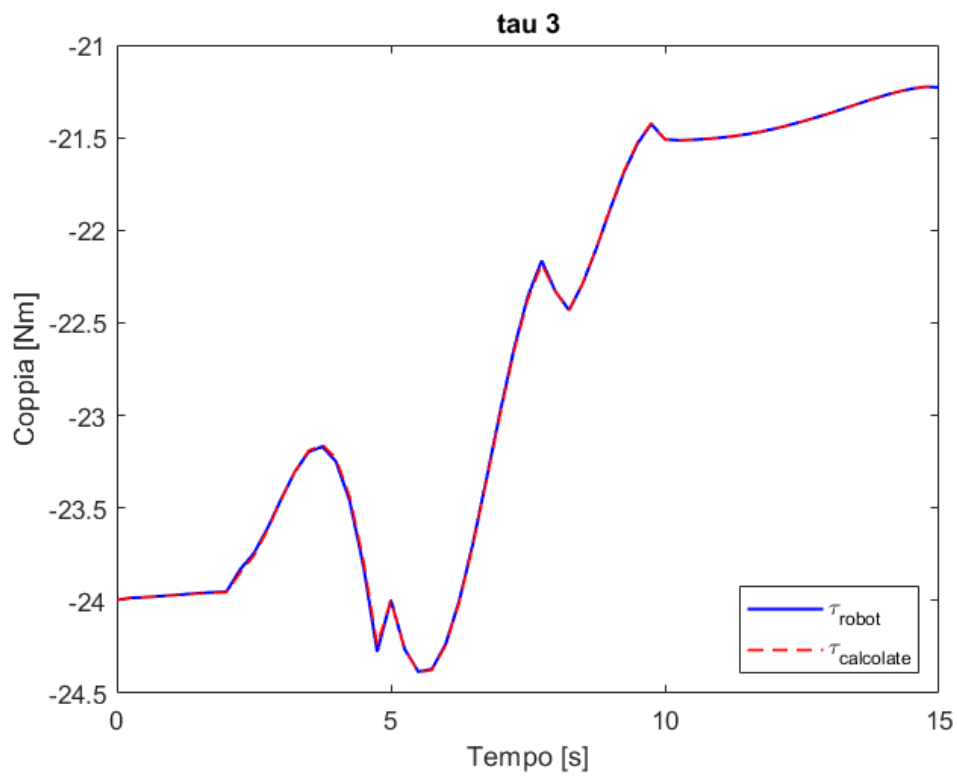


Figura 4.4: Confronto coppia al giunto 3.

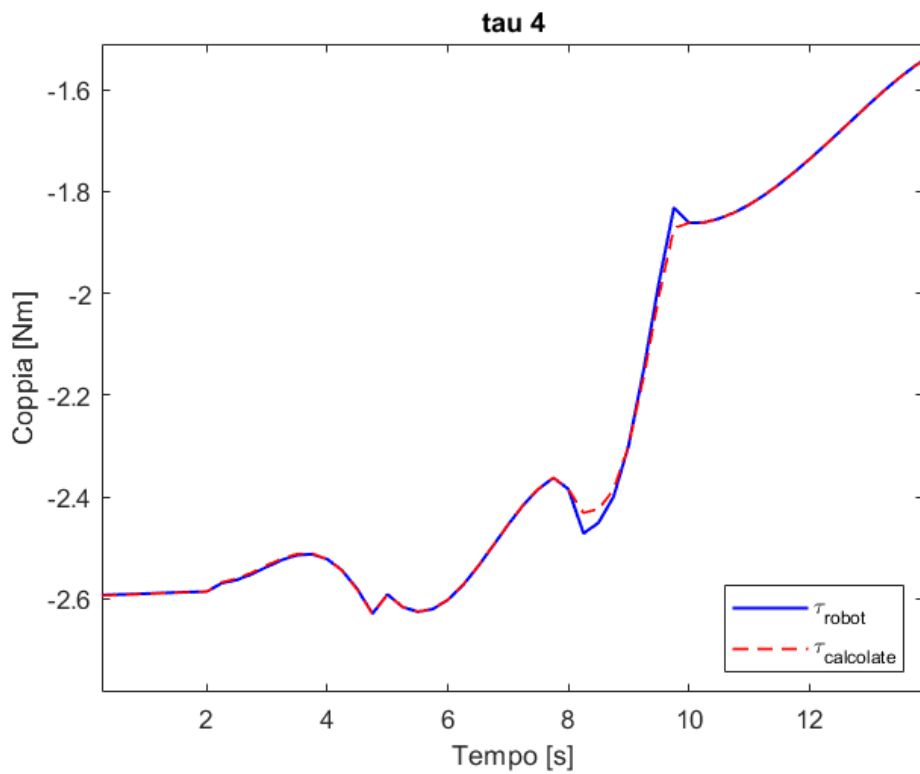


Figura 4.5: Confronto coppia al giunto 4.

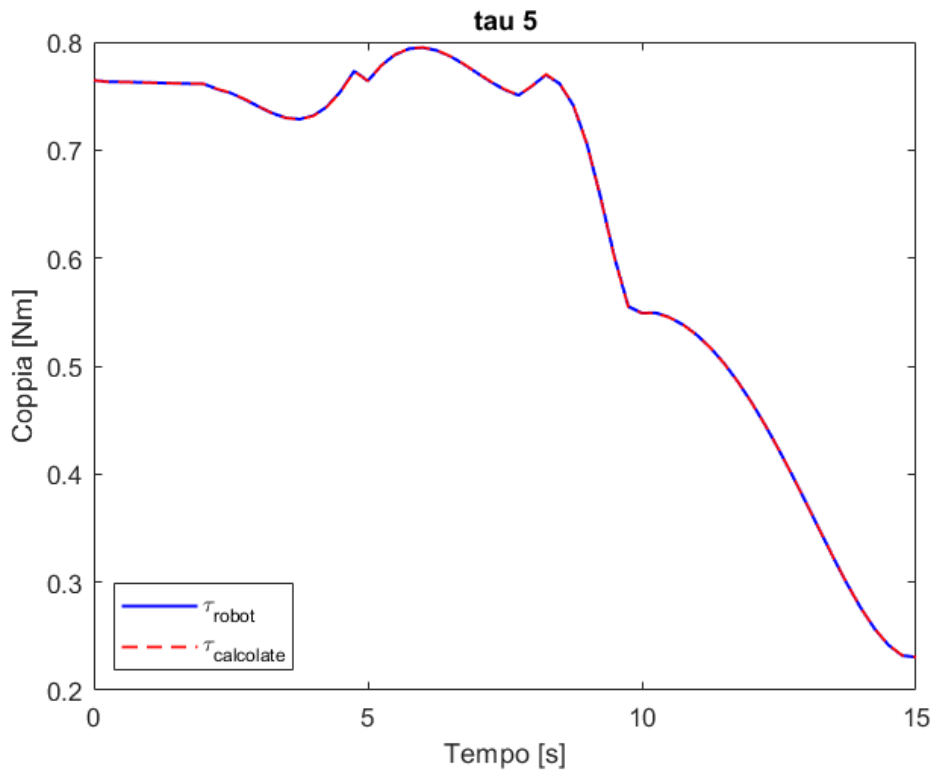


Figura 4.6: Confronto coppia al giunto 5.

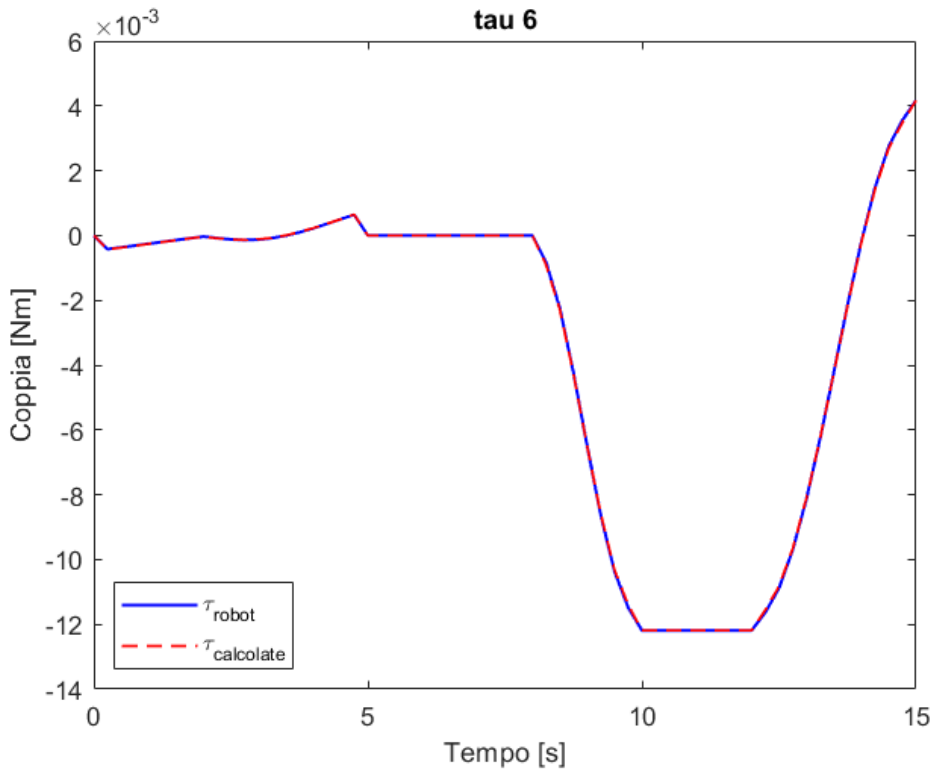


Figura 4.7: Confronto coppia al giunto 6.

Capitolo 5

Conclusioni

In seguito all'applicazione di tutti gli algoritmi illustrati, sul robot FANUC CRX-10iA/L e dopo averne verificato l'affidabilità e la corretta scrittura, sono stati individuati nove parametri che non influenzano il comportamento del robot e sono i seguenti:

$$m_1; m_1c_{1x}; m_1c_{1y}; m_1c_{1z}; J_{xx}; J_{xy}; J_{xz}; J_{yz}; J_{zz};$$

Effettuando la QR decomposition della matrice di regressione vengono individuati 15 parametri:

$$\begin{array}{ccccc} J_{xz_2} & J_{yy_2} & J_{zz_2} & m_3 & m_{3c3y} \\ J_{zz_3} & m_4 & m_{4c4y} & J_{zz_4} & m_5 \\ m_{5c5y} & J_{zz_5} & m_6 & m_{6c6z} & J_{yy_6} \end{array}$$

che saranno raggruppati agli altri 36 indipendenti:

$$\begin{array}{cccccc} J_{yy_1} & m_2 & m_{2c2x} & m_{2c2y} & m_{2c2z} & J_{xx_2} \\ J_{xy_2} & J_{yz_2} & m_{3c3x} & m_{3c3z} & J_{xx_3} & J_{xy_3} \\ J_{xz_3} & J_{yy_3} & J_{yz_3} & m_{4c4x} & m_{4c4z} & J_{xx_4} \\ J_{xy_4} & J_{xz_4} & J_{yy_4} & J_{yz_4} & m_{5c5x} & m_{5c5z} \\ J_{xx_5} & J_{xy_5} & J_{xz_5} & J_{yy_5} & J_{yz_5} & m_{6c6x} \\ m_{6c6y} & J_{xx_6} & J_{xy_6} & J_{xz_6} & J_{yz_6} & J_{zz_6} \end{array}$$

È bene sottolineare che tutti questi risultati sono stati ottenuti sempre facendo riferimento al robot virtuale modellato in *Adams* e non a dati ricavati sperimentalmente dal modello fisico del robot. Per il futuro è previsto rendere sperimentale questo processo di identificazione basato su algoritmi e verificare l'eventuale corrispondenza con i dati realmente estratti dalla macchina. Inoltre andrà approfondito l'andamento della precisione degli algoritmi al variare delle traiettorie con cui si eccita il sistema, poiché sicuramente al variare di esse varia il modo in cui viene sollecitata la dinamica.

Bibliografia

- [1] FANUC. *CRX-10iA/L Data sheet*. 2022.
- [2] B. Siciliano O. Khatib. *Springer Handbook of Robotics*. Springer International Publishing, 2016.
- [3] M-C. Palpacelli e M Callegari L. Carbonari. Inverse kinematics of a class of 6r collaborative robots with non-spherical wrist. *Robotics*, 2023.
- [4] M. Gabbicini e A. Bracci. Explicit lagrangian formulation of the dynamic regressors for serial manipulators. *IEEE ICRA*, 2009.
- [5] B. Siciliano L. Sciavicco L. Villani G. Oriolo. *Robotics Modelling, Planning and Control*. Springer International Publishing, 2009.
- [6] M. Gautier e W. Khalil. Direct calculation of minimum set of inertial parameters of serial robots. *IEE Transactions on robotics and automation*, 6(3), 1990.
- [7] M. Gautier. Numerical calculation of the base inertial parameters of robots. *IEEE International Conference on Robotics and Automation*, 1990.
- [8] MSC Software. *Adams View User's Guide*. 2021.