



UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTÀ DI INGEGNERIA

Corso di Laurea Magistrale in Ingegneria Aerospaziale

**Studio ed analisi
di un manipolatore spaziale per
l'identificazione dei parametri dinamici
attraverso misure di forza**

Relatore:

Prof. Debei Stefano

Correlatore:

Dott. Cocuzza Silvio

Laureando:

Destro Saul

A.A. 2012-2013

Indice

1	Introduzione	1
2	Modelli cinematici e dinamici	3
2.1	Notazione modelli	3
2.2	Equazioni della cinematica	4
2.3	Equazioni della dinamica	7
3	Modelli	11
3.1	Schemi a blocchi	13
3.1.1	Schema a blocchi con dinamometro infinitamente rigido	13
3.1.2	Schema a blocchi con dinamometro cedevole	15
4	Procedura d'identificazione	19
4.1	Procedura di simulazione	19
4.2	Traiettoria utilizzata	19
4.3	Equazioni d'identificazione	20
5	Risultati simulazioni	25
5.1	Modello con dinamometro infinitamente rigido	25
5.2	Modello con dinamometro cedevole	27
5.3	Modello combinato	28
5.4	Conclusioni	29
A	Modello con flessibilità di giunto	31
B	File MatLab®	35
B.1	Generazione traiettoria	35
B.2	Funzioni per il calcolo delle equazioni della forza alla base	38
B.2.1	Equazioni modello dinamometro infinitamente rigido	38
B.2.2	Equazioni modello dinamometro cedevole	40
B.2.3	Equazioni modello ad un link con giunto flessibile	42
B.3	Funzioni per l'identificazione dei parametri di base	44
B.3.1	Identificazione modello dinamometro infinitamente rigido	44
B.3.2	Identificazione modello dinamometro cedevole	48
B.3.3	Identificazione modello ad un link con giunto flessibile	52

Elenco delle figure

2.1	Esempio di notazione	3
2.2	Sistemi di coordinate	5
3.1	Modello con dinamometro infinitamente rigido	12
3.2	Modello con dinamometro cedevole	12
3.3	Schema a blocchi con dinamometro infinitamente rigido	14
3.4	Schema a blocchi con ipotesi di cedevolezza del dinamometro	16
4.1	Traiettoria	21
5.1	Dinamometro non cedevole, errore $\% \hat{\beta}_i$ con $-1\% \cdot X_{max} < err < 1\% \cdot X_{max}$	26
5.2	Dinamometro non cedevole, errore $\% \hat{\beta}_i$ con $-2\% \cdot X_{max} < err < 2\% \cdot X_{max}$	26
5.3	Dinamometro non cedevole, errore $\% \hat{\beta}_i$ con $-5\% \cdot X_{max} < err < 5\% \cdot X_{max}$	26
5.4	Dinamometro cedevole, errore $\% \hat{\beta}_i$ con $-1\% \cdot X_{max} < err < 1\% \cdot X_{max}$	27
5.5	Dinamometro cedevole, errore $\% \hat{\beta}_i$ con $-2\% \cdot X_{max} < err < 2\% \cdot X_{max}$	27
5.6	Dinamometro cedevole, errore $\% \hat{\beta}_i$ con $-5\% \cdot X_{max} < err < 5\% \cdot X_{max}$	27
5.7	Modello combinato, senza introduzione di errori sulle misure	29
5.8	Modello combinato, errore $\% \hat{\beta}_i$ con $-1\% \cdot X_{max} < err < 1\% \cdot X_{max}$	29
5.9	Modello combinato, errore $\% \hat{\beta}_i$ con $-2\% \cdot X_{max} < err < 2\% \cdot X_{max}$	29
5.10	Modello combinato, errore $\% \hat{\beta}_i$ con $-5\% \cdot X_{max} < err < 5\% \cdot X_{max}$	29
A.1	Modello ad un link con giunto flessibile e dinamometro fisso	31
A.2	Schema a blocchi modello ad un link con giunto flessibile e dinamometro fisso	31
A.3	Dinamometro giunto flessibile	33
A.4	Dinamometro giunto flessibile, errore $\% \hat{\beta}_i$ con $-1\% \cdot X_{max} < err < 1\% \cdot X_{max}$	33
A.5	Dinamometro giunto flessibile, errore $\% \hat{\beta}_i$ con $-2\% \cdot X_{max} < err < 2\% \cdot X_{max}$	33
A.6	Dinamometro giunto flessibile, errore $\% \hat{\beta}_i$ con $-5\% \cdot X_{max} < err < 5\% \cdot X_{max}$	33

Capitolo 1

Introduzione

I più avanzati schemi di controllo utilizzati in robotica per applicazioni di tracciamento della traiettoria richiedono modelli dinamici. La precisione, le performance, la stabilità e la robustezza di tali schemi dipendono quindi dall'accuratezza con la quale si conoscono i parametri dinamici dei manipolatori. In vista di ciò la determinazione dei parametri dinamici di un manipolatore risulta essere utile se non necessaria al controllo dello stesso.

I parametri di un robot possono essere suddivisi in categorie:

- i parametri geometrici (lunghezza dei link, posizione dei giunti, ...), considerati costanti e possono essere assunti come valori noti con un elevato grado di precisione poiché derivano direttamente dal disegno e dalla costruzione del robot;
- i parametri inerziali dei link, caratterizzanti la distribuzione di massa di ogni link. I valori numerici di tali parametri sono poco conosciuti e pertanto devono essere identificati. Essi sono costanti per tutta la vita del manipolatore e quindi necessitano di una singola procedura d'identificazione;
- i parametri di un eventuale payload all'end-effector del robot;
- i parametri di attrito dei giunti, considerati variabili per tutta la vita del manipolatore.

Il lavoro di tale tesi si prefigge l'obiettivo di simulare l'identificazione dei parametri dinamici di un robot spaziale planare di cui al documento [3] della bibliografia. Ci sono diversi metodi per misurare i valori dei parametri dinamici. Il primo consiste nel disassemblare il manipolatore ed effettuare le misure dei parametri inerziali e delle sue componenti risultando essere, però, un'opzione laboriosa e sconveniente. Un'alternativa per la stima di tali parametri è quella di usare un pacchetto software di Computer Aided Design che per robot con forme irregolari, numerose componenti e di diverso materiale come il robot in esame è di difficile realizzazione. L'ultima opzione e scelta più conveniente è quella di effettuare delle misure di forza alla base del manipolatore tramite il dinamometro ad essa collegato e risalire attraverso la conoscenza delle equazioni della dinamica e dei valori di input che generano il moto dei link del robot ai parametri dinamici che li caratterizzano.

Dal documento riguardante il manipolatore sotto esame si utilizzeranno tutti i dati ad esso riguardanti per la costruzione di un modello che lo rappresenti il più fedelmente possibile mentre nelle procedure di identificazione gli unici dati che si riterranno noti sono i parametri geometrici poiché è possibile conoscerne il valore in ogni istante mediante procedimenti di misura veloci, accurati e non invasivi.

Capitolo 2

Modelli cinematici e dinamici

2.1 Notazione modelli

In questo elaborato la notazione usata per la descrizione dei modelli del manipolatore planare è mostrata in figura 2.1 e riassunta qui di seguito.

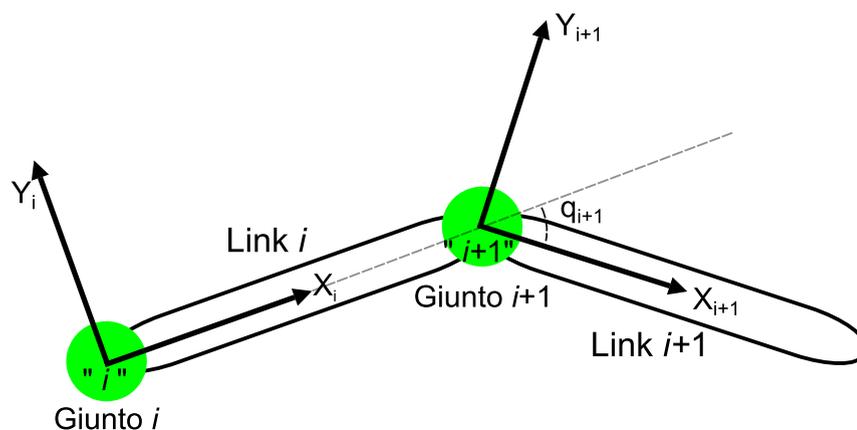


Figura 2.1: Esempio di notazione

La numerazione dei corpi e dei giunti del manipolatore si accorda alla seguente convenzione:

- i link del meccanismo robotico sono numerati in ordine crescente;
- i giunti rotazionali del meccanismo robotico sono anch'essi numerati in ordine crescente con il giunto $i+1$ localizzato tra i link i e $i+1$;
- il giunto i è considerato parte del link i quindi le velocità del giunto e del link coincidono;
- per ogni giunto i è definito un corrispondente sistema di coordinate cartesiane " i " avente l'origine ad esso fissato, l'asse \vec{Z}_i parallelo alla direzione dell'asse di rotazione del giunto e verso uscente dal piano e l'asse \vec{X}_i è posto lungo la normale comune tra gli assi \vec{Z}_i e \vec{Z}_{i+1} .

Si definiscono inoltre le seguenti grandezze:

- ${}^i\vec{l}_i$ è la lunghezza del link cioè la distanza tra \vec{Z}_i e \vec{Z}_{i+1} lungo \vec{X}_i espressa nel sistema “ i ”;
- ${}^i\vec{q}_i$ è l'angolo tra \vec{X}_{i-1} e \vec{X}_i lungo \vec{Z}_i espressa nel sistema “ i ”.

2.2 Equazioni della cinematica

Tale sezione esaminerà la descrizione del moto di un corpo rigido. La semplificazione che si adotta in questa sezione è quella di fissare un sistema di coordinate ad ogni corpo di cui si intende descriverne il moto cosicché quest'ultimo possa essere equivalentemente studiato come il moto del relativo sistema di coordinate rispetto ad un altro.

Descrizione posizione spaziale

Si considerino i due sistemi di riferimento di figura 2.1. Si desidera descrivere la posizione di un punto P rispetto al sistema di coordinate “ i ” (figura 2.2).

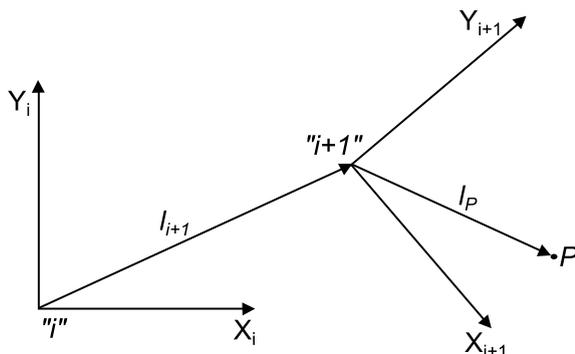


Figura 2.2: Sistemi di coordinate

La posizione sarà descritta da un vettore $[3 \times 1]$ contenente le coordinate lungo gli assi \vec{X}_i , \vec{Y}_i e \vec{Z}_i . La posizione dell'origine del sistema di coordinate “ $i+1$ ” rispetto ad “ i ” è definita dal vettore ${}^i\vec{l}_{i+1}$. Il vettore di posizione di P rispetto ad “ i ” è dato da:

$${}^i\vec{l}_P = {}^i\vec{l}_{i+1} + {}^{i+1}\vec{l}_P, \quad (2.1)$$

dove ${}^i\vec{l}_P$ è la distanza di P da “ i ” e ${}^{i+1}\vec{l}_P$ è la distanza di P da “ $i+1$ ”. Introducendo l'utilizzo delle matrici di rotazione che descrivono l'orientazione di un sistema di riferimento rispetto ad un altro si può riscrivere l'equazione 2.1 come segue:

$${}^i\vec{l}_P = {}^i\vec{l}_{i+1} + {}^i_{i+1}R \cdot {}^{i+1}\vec{l}_P, \quad (2.2)$$

nella quale ${}^i_{i+1}R$ è una matrice $[3 \times 3]$ le cui colonne sono i versori del sistema di riferimento “ $i+1$ ” rispetto al sistema di coordinate “ i ” mentre le righe sono i versori del sistema “ i ” rispetto al sistema “ $i+1$ ”, difatti:

$${}^i_{i+1}R = {}^{i+1}_iR^T. \quad (2.3)$$

La matrice di rotazione è costruita utilizzando il metodo degli angoli di Eulero Z-Y-X nel quale partendo da un orientazione di “ $i+1$ ” coincidente con “ i ”, sistema di riferimento noto, si ruota in maniera successiva il sistema “ $i+1$ ” lungo l’asse \vec{Z}_{i+1} di un angolo ψ poi lungo \vec{Y}_{i+1} di un angolo θ ed infine lungo \vec{X}_{i+1} di un angolo ϕ . Si noti che ogni rotazione avviene lungo un asse la cui posizione dipende dalle rotazioni precedenti. La matrice di rotazione che permette di esprimere un vettore dal sistema “ $i+1$ ” al sistema “ i ” è:

$$\begin{aligned} {}^i_{i+1}R &= R_Z(\psi)R_Y(\theta)R_X(\phi) \\ &= \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}. \end{aligned} \quad (2.4)$$

Velocità

Nel caso i due sistemi di riferimento considerati mantengano la mutua orientazione costante nel tempo la velocità assoluta del punto P , ${}^i\vec{v}_P$, è:

$$\begin{aligned} {}^i\vec{v}_P &= \frac{d}{dt} ({}^i\vec{l}_P) \\ &= \frac{d}{dt} ({}^i\vec{l}_{i+1}) + {}^i_{i+1}R \cdot \frac{d}{dt} ({}^{i+1}\vec{l}_P) \\ &= {}^i\vec{v}_{i+1} + {}^i_{i+1}R \cdot {}^{i+1}\vec{v}_P, \end{aligned} \quad (2.5)$$

somma della velocità del sistema “ $i+1$ ” rispetto a “ i ” nota anche come velocità di trascinamento ${}^i\vec{v}_{i+1}$, e della velocità relativa, ${}^{i+1}\vec{v}_P$, del punto P rispetto al sistema di riferimento “ $i+1$ ”.

Si consideri ora il caso in cui i due sistemi “ i ” e “ $i+1$ ” abbiano origini coincidenti. Si ipotizzi inoltre che il punto P sia privo di moto rispetto al sistema “ $i+1$ ” e che i due sistemi abbiano mutua orientazione variabile nel tempo; si definisce allora la velocità angolare di “ $i+1$ ” rispetto al sistema inerziale “ i ” come ${}^i\vec{\omega}_{i+1}$ $[3 \times 1]$. La velocità ${}^i\vec{v}_P$ diviene:

$$\begin{aligned}
{}^i\vec{v}_P &= {}^i\vec{\omega}_{i+1} \times {}^{i+1}\vec{l}_P \\
&= {}^i\vec{\omega}_{i+1} \times \left({}^i_{i+1}R \cdot {}^{i+1}\vec{l}_P \right), \tag{2.6}
\end{aligned}$$

e se il punto P possedesse una velocità lineare relativamente al sistema “ $i+1$ ” allora la 2.6 assumerebbe tale forma:

$${}^i\vec{v}_P = {}^i_{i+1}R \cdot {}^{i+1}\vec{v}_P + {}^i\vec{\omega}_{i+1} \times \left({}^i_{i+1}R \cdot {}^{i+1}\vec{l}_P \right). \tag{2.7}$$

Nell’ipotesi che le origini dei due sistemi di coordinate non siano coincidenti ed esista moto relativo tra di essi (${}^i\vec{v}_{i+1} \neq 0$) il vettore ${}^i\vec{v}_P$ assume la seguente forma:

$${}^i\vec{v}_P = {}^i\vec{v}_{i+1} + {}^i_{i+1}R \cdot {}^{i+1}\vec{v}_P + {}^i\vec{\omega}_{i+1} \times \left({}^i_{i+1}R \cdot {}^{i+1}\vec{l}_P \right). \tag{2.8}$$

Accelerazione lineare e angolare

Partendo dall’equazione 2.7 nelle ipotesi di origini coincidenti, si calcola l’accelerazione lineare assoluta di P vista da “ i ”:

$${}^i\dot{\vec{v}}_P = \frac{d}{dt} \left({}^i_{i+1}R \cdot {}^{i+1}\vec{v}_P \right) + {}^i\dot{\vec{\omega}}_{i+1} \times \left({}^i_{i+1}R \cdot {}^{i+1}\vec{l}_P \right) + {}^i\vec{\omega}_{i+1} \times \frac{d}{dt} \left({}^i_{i+1}R \cdot {}^{i+1}\vec{l}_P \right). \tag{2.9}$$

Sviluppando la precedente equazione si ottiene:

$$\begin{aligned}
{}^i\dot{\vec{v}}_P &= {}^i_{i+1}R \cdot {}^{i+1}\dot{\vec{v}}_P + 2 \cdot {}^i\vec{\omega}_{i+1} \times \left({}^i_{i+1}R \cdot {}^{i+1}\vec{v}_P \right) \\
&\quad + {}^i\dot{\vec{\omega}}_{i+1} \times \left({}^i_{i+1}R \cdot {}^{i+1}\vec{l}_P \right) + {}^i\vec{\omega}_{i+1} \times \left({}^i\vec{\omega}_{i+1} \times \left({}^i_{i+1}R \cdot {}^{i+1}\vec{l}_P \right) \right)
\end{aligned} \tag{2.10}$$

Generalizzando al caso in cui le due origini non siano coincidenti ed il sistema “ $i+1$ ” sia in moto linearmente accelerato rispetto al sistema inerziale si aggiunge all’equazione 2.10 la componente dell’accelerazione di trascinamento ottenendo:

$$\begin{aligned}
{}^i\dot{\vec{v}}_P &= {}^i\dot{\vec{v}}_{i+1} + {}^i_{i+1}R \cdot {}^{i+1}\dot{\vec{v}}_P + 2 \cdot {}^i\vec{\omega}_{i+1} \times \left({}^i_{i+1}R \cdot {}^{i+1}\vec{v}_P \right) \\
&\quad + {}^i\dot{\vec{\omega}}_{i+1} \times \left({}^i_{i+1}R \cdot {}^{i+1}\vec{l}_P \right) + {}^i\vec{\omega}_{i+1} \times \left({}^i\vec{\omega}_{i+1} \times \left({}^i_{i+1}R \cdot {}^{i+1}\vec{l}_P \right) \right)
\end{aligned} \tag{2.11}$$

Nel caso particolare in cui P sia un punto fisso nel sistema “ $i+1$ ”, caso in cui appartenga al link $i+1$, (${}^A\vec{v}_P = {}^A\vec{v}_{i+1} = 0$), l’espressione 2.11 diviene:

$${}^i\dot{\vec{v}}_P = {}^i\dot{\vec{v}}_{i+1} + {}^i\vec{\omega}_{i+1} \times \left({}^i\vec{\omega}_{i+1} \times \left({}^i_{i+1}R \cdot {}^{i+1}\vec{l}_P \right) \right) + {}^i\dot{\vec{\omega}}_{i+1} \times \left({}^i_{i+1}R \cdot {}^{i+1}\vec{l}_P \right). \tag{2.12}$$

Per ricavare l'equazione dell'accelerazione angolare assoluta si aggiunga un terzo sistema di coordinate “ $i+2$ ”. Se la velocità angolare di trascinamento é ${}^i\vec{\omega}_{i+1}$, e quella relativa tra “ $i+2$ ” e “ $i+1$ ” é ${}^i\vec{\omega}_{i+2}$ allora la velocità assoluta ${}^i\vec{\omega}_{i+2}$ é:

$${}^i\vec{\omega}_{i+2} = {}^i\vec{\omega}_{i+1} + {}^i_{i+1}R \cdot {}^{i+1}\vec{\omega}_{i+2}. \quad (2.13)$$

Derivando quest'ultima si ottiene l'equazione dell'accelerazione angolare assoluta:

$$\begin{aligned} {}^i\dot{\vec{\omega}}_{i+2} &= {}^i\dot{\vec{\omega}}_{i+1} + \frac{d}{dt} ({}^i_{i+1}R \cdot {}^i\vec{\omega}_{i+2}) \\ &= {}^i\dot{\vec{\omega}}_{i+1} + {}^i_{i+1}R \cdot {}^{i+1}\dot{\vec{\omega}}_{i+2} + {}^i\vec{\omega}_{i+1} \times ({}^i_{i+1}R \cdot {}^{i+1}\vec{\omega}_{i+2}). \end{aligned} \quad (2.14)$$

2.3 Equazioni della dinamica

Si consideri ora il problema del calcolo della coppia e delle forze alla base corrispondenti ad una data traiettoria del manipolatore. Sono pertanto necessarie le equazioni della dinamica per entrambi i modelli del robot. L'algoritmo di Newton-Eulero é uno utile strumento che consente di trovare tali equazioni in maniera semplice e veloce grazie all'ausilio del Symbolic Math Toolbox di MatLab®. L'algoritmo usa le equazioni della cinematica descritte nella sezione 2.2 per il calcolo delle accelerazioni lineari dei giunti e dei baricentri dei link ed utilizza le equazioni di Newton ed Eulero per le forze e coppie inerziali. Le accelerazioni lineari ed angolari si propagano da un link al successivo ed é quindi possibile determinarle in maniera progressiva. La velocità angolare assoluta del link $i+1$ é fornita dalla seguente equazione:

$${}^{i+1}\vec{\omega}_{i+1} = {}^i_{i+1}R \cdot {}^i\vec{\omega}_i + \dot{\vec{q}}_{i+1}, \quad (2.15)$$

nella quale \dot{q}_{i+1} é la velocità angolare esplicita al giunto $i+1$ e quindi la velocità angolare del link $i+1$ relativamente al link i . Dalla 2.14 si ottiene l'equazione per il calcolo dell'accelerazione angolare da un braccio al successivo:

$${}^{i+1}\dot{\vec{\omega}}_{i+1} = {}^i_{i+1}R \cdot {}^i\dot{\vec{\omega}}_i + ({}^i_{i+1}R \cdot {}^i\vec{\omega}_i) \times \dot{\vec{q}}_{i+1} + \ddot{\vec{q}}_{i+1}. \quad (2.16)$$

La propagazione dell'accelerazione lineare tra i link (origine sistema di riferimento) si ottiene applicando la 2.12:

$${}^{i+1}\dot{\vec{v}}_{i+1} = {}^i_{i+1}R \cdot \left[{}^i\dot{\vec{v}}_i + {}^i\vec{\omega}_i \times ({}^i\vec{\omega}_i \times {}^i\vec{l}_{i+1}) + {}^i\dot{\vec{\omega}}_i \times {}^i\vec{l}_{i+1} \right]. \quad (2.17)$$

Le accelerazioni dei centri di massa si ottengono sostituendo nella 2.17 i vettori posizione dei link con i vettori posizione dei baricentri. Per il computo delle forze e coppie inerziali si definiscono le equazioni di Newton ed

Eulero.

Si consideri un corpo rigido, di massa m , il cui centro di massa sia soggetto ad una accelerazione $\dot{\vec{v}}_C$. La forza agente al centro di massa e causa di tale accelerazione è data dall'equazione di Newton:

$$\vec{F} = m \cdot \dot{\vec{v}}_C. \quad (2.18)$$

Si consideri ora lo stesso corpo rigido soggetto ad una rotazione con velocità angolare $\vec{\omega}$ e accelerazione angolare $\dot{\vec{\omega}}$. Il momento angolare agente sul moto e causa di tale rotazione è dato dall'equazione di Eulero:

$$\vec{N} = {}^C I \cdot \dot{\vec{\omega}} + \vec{\omega} \times ({}^C I \cdot \vec{\omega}), \quad (2.19)$$

con ${}^C I$ tensore di inerzia del corpo rigido scritto in un sistema di coordinate con origine nel centro di massa.

Le forze e le coppie inerziali agenti al centro di massa di ogni link sono:

$$\vec{F}_i = m_i \cdot \dot{\vec{v}}_{C_i} \quad (2.20)$$

$$\vec{N}_i = {}^{C_i} I \cdot \dot{\vec{\omega}} + \vec{\omega} \times ({}^{C_i} I \cdot \vec{\omega}), \quad (2.21)$$

nella quale ${}^{C_i} I$ è il tensore di inerzia scritto rispetto ad un sistema di riferimento baricentrico che ha la stessa orientazione di sistema di coordinate del link, "i".

Ora che sono note le equazioni che permettono di ricavare le accelerazioni lineari ed angolari dei giunti, le coppie e le forze inerziali è necessario presentare le equazioni che consentono di trovare le forze e le coppie ai giunti.

Ogni generico link i è soggetto alle forze ed alle coppie dei link adiacenti $i-1$ e $i+1$. Se si esegue un bilancio delle forze agenti su tale link si avrà:

$${}^i \vec{F}_i = {}^i \vec{f}_i - {}^i_{i+1} R \cdot {}^{i+1} \vec{f}_{i+1} \quad (2.22)$$

nella quale ${}^i \vec{f}_i$ e ${}^{i+1} \vec{f}_{i+1}$ sono rispettivamente le forze esercitate dai link $i-1$ e $i+1$ sul link i .

Eseguendo invece un bilancio dei momenti agenti sul link i si ottiene:

$${}^i \vec{N}_i = {}^i \vec{n}_i - {}^i_{i+1} R \cdot {}^{i+1} \vec{n}_{i+1} - {}^i \vec{l}_{C_i} \times {}^i \vec{F}_i - {}^i \vec{l}_{i+1} \times ({}^i_{i+1} R \cdot {}^{i+1} \vec{f}_{i+1}). \quad (2.23)$$

Esplicitando la forza, ${}^i \vec{f}_i$, ed il momento, ${}^i \vec{n}_i$, al giunto i :

$${}^i \vec{f}_i = {}^i_{i+1} R \cdot {}^{i+1} \vec{f}_{i+1} + {}^i \vec{F}_i, \quad (2.24)$$

$${}^i \vec{n}_i = {}^i \vec{N}_i + {}^i_{i+1} R \cdot {}^{i+1} \vec{n}_{i+1} + {}^i \vec{l}_{C_i} \times {}^i \vec{F}_i + {}^i \vec{l}_{i+1} \times ({}^i_{i+1} R \cdot {}^{i+1} \vec{f}_{i+1}). \quad (2.25)$$

L'algoritmo è composto di due parti. La prima, eseguendo delle iterazioni crescenti a partire dalla base fino a giungere all'end-effector, permette il computo delle accelerazioni e delle coppie e forze d'inerzia. La seconda parte, con iterazioni decrescenti dall'end-effector alla base, consente di calcolare le forze e le coppie agenti ad ogni giunto.

ALGORITMO ITERATIVO DI NEWTON-EULERO

Iterazioni verso l'esterno: $i : 1^\circ \text{ link} \rightarrow \text{end-effector}$

$${}^{i+1}\vec{\omega}_{i+1} = {}^i R \cdot {}^i \dot{\vec{\omega}}_i + {}^{i+1} \dot{\vec{q}}_{i+1}, \quad (2.26a)$$

$${}^{i+1}\ddot{\vec{\omega}}_{i+1} = {}^i R \cdot {}^i \ddot{\vec{\omega}}_i + ({}^i R \cdot {}^i \dot{\vec{\omega}}_i) \times {}^{i+1} \dot{\vec{q}}_{i+1} + {}^{i+1} \ddot{\vec{q}}_{i+1}, \quad (2.26b)$$

$${}^{i+1}\dot{\vec{v}}_{i+1} = {}^i R \cdot \left({}^i \dot{\vec{\omega}}_i \times {}^i \vec{l}_{i+1} + {}^i \vec{\omega}_i \times ({}^i \vec{\omega}_i \times {}^i \vec{l}_{i+1}) + {}^i \dot{\vec{v}}_i \right), \quad (2.26c)$$

$${}^{i+1}\dot{\vec{v}}_{C_{i+1}} = {}^{i+1}\dot{\vec{\omega}}_{i+1} \times {}^{i+1}\vec{l}_{C_{i+1}} + {}^{i+1}\vec{\omega}_{i+1} \times ({}^{i+1}\vec{\omega}_{i+1} \times {}^{i+1}\vec{l}_{C_{i+1}}) + {}^{i+1}\dot{\vec{v}}_{i+1}, \quad (2.26d)$$

$${}^{i+1}\vec{F}_{i+1} = m_{i+1} \cdot {}^{i+1}\dot{\vec{v}}_{C_{i+1}}, \quad (2.26e)$$

$${}^{i+1}\vec{N}_{i+1} = {}^{C_{i+1}} I_{i+1} \cdot {}^{i+1}\dot{\vec{\omega}}_{i+1} + {}^{i+1}\vec{\omega}_{i+1} \times ({}^{C_{i+1}} I_{i+1} \cdot {}^{i+1}\vec{\omega}_{i+1}). \quad (2.26f)$$

Iterazioni verso l'interno: $i : \text{end-effector} \rightarrow 1^\circ \text{ link}$

$${}^i \vec{f}_i = {}^i R \cdot {}^{i+1} \vec{f}_{i+1} + {}^i \vec{F}_i, \quad (2.27a)$$

$${}^i \vec{n}_i = {}^i \vec{N}_i + {}^i R \cdot {}^{i+1} \vec{n}_{i+1} + {}^i \vec{l}_{C_i} \times {}^i \vec{F}_i + {}^i \vec{l}_{i+1} \times ({}^i R \cdot {}^{i+1} \vec{f}_{i+1}). \quad (2.27b)$$

dove:

- ${}^i R$ è la matrice di rotazione dal sistema di riferimento $i+1$ al sistema i e la matrice ${}^{i+1} R$ il viceversa, ${}^i R = {}^{i+1} R^T$;
- la generica rotazione \vec{q}_i del link i rispetto al link precedente può essere esplicitata solamente lungo all'asse \vec{Z}_i , la matrice di rotazione tra il "i" ed il sistema "i-1" è dalla equazione 2.4:

$${}^{i-1} R = R_Z(\psi) = \begin{bmatrix} \cos(q_i) & -\sin(q_i) & 0 \\ \sin(q_i) & \cos(q_i) & 0 \\ 0 & 0 & 1 \end{bmatrix};$$

- il pedice e l'apice indicano rispettivamente il link cui la grandezza si riferisce e il sistema di riferimento nel quale è espressa;
- $\vec{\omega}$ e $\dot{\vec{\omega}}$ sono rispettivamente la velocità e l'accelerazione angolare del link;

- $\dot{\vec{v}}$ e $\dot{\vec{v}}_C$ sono l'accelerazione lineare dell'origine del sistema di riferimento e del baricentro del link;
- \vec{F} e \vec{N} sono rispettivamente la forza e la coppia inerziali agenti al centro di massa del link;
- \vec{f}_i e \vec{n}_i sono rispettivamente la forza e la coppia esercitate dal link $i - 1$ sul link i ;
- \vec{l}_C e m sono il vettore posizione del baricentro e la massa del link;
- ${}^{C_i}I$ ha lo stesso significato descritto nelle equazioni 2.20. Ogni link i ha forma cilindrica con asse di simmetria coincidente con \vec{X}_i quindi il tensore di inerzia riferito ad un sistema di coordinate baricentriche e con orientazione coincidente con il sistema di riferimento del link è diagonale:

$${}^{C_i}I_i = \begin{bmatrix} I_{xx_i} & 0 & 0 \\ 0 & I_{yy_i} & 0 \\ 0 & 0 & I_{zz_i} \end{bmatrix}.$$

Capitolo 3

Modelli

I modelli costruiti per rappresentare il manipolatore planare sono due:

- modello con dinamometro infinitamente rigido,
- modello con dinamometro cedevole.

In entrambi i modelli il sistema di misura, dinamometro, e la base del manipolatore sono stati considerati come una struttura unica. Nel primo modello tale struttura è stata rappresentata come un corpo rigido al quale non sono consentiti spostamenti e ad essa sono collegati gli altri tre link del manipolatore. Il dinamometro-base del manipolatore ha la sola funzione di un sensore che fornisce informazioni sulle forze alla base del manipolatore.

Nel secondo modello la struttura dinamometro-base del manipolatore è stata schematizzata come una struttura collegata ad un riferimento inerziale tramite tre molle e tre smorzatori: due molle e due smorzatori lineari agenti lungo gli assi \vec{X}_G e \vec{Y}_G e una molla ed uno smorzatore torsionali agenti in \vec{Z}_G . I valori di tali elementi ne determinano l'ampiezza degli spostamenti e della rotazione della struttura cui sono collegati. La struttura del dinamometro costruito le robot reale (riferimento biblio) ha una rigidezza lungo \vec{Y}_G molto maggiore alla rigidezza in \vec{X}_G perciò si è ipotizzato, come varrà mostrato in seguito di utilizzare per il calcolo dei parametri dinamici le forze alla base sull'asse \vec{X}_G e della coppia alla base in \vec{Z}_G . Di conseguenza i valori dati alle molle e smorzatori di tali due assi sono quelli che sono stati calcolati negli esperimenti precedenti (riferimento biblio), mentre il valore della rigidità e dello smorzamento lungo l'asse \vec{Y}_G sono stati scelti tali da rappresentare in maniera opportuna il dinamometro. I valori delle rigidezze (k) e degli smorzamenti (c) sono:

Tabella 3.1: Valori molle e smorzatori

	\vec{X}_G	\vec{Y}_G	\vec{Z}_G
k	390100 [N/m]	$40 \cdot 10^6$ [N/m]	6078 [N · m/rad]
c	17.57 [N · s/m]	50 [N · s/m]	1.722 [N · m · s/rad]

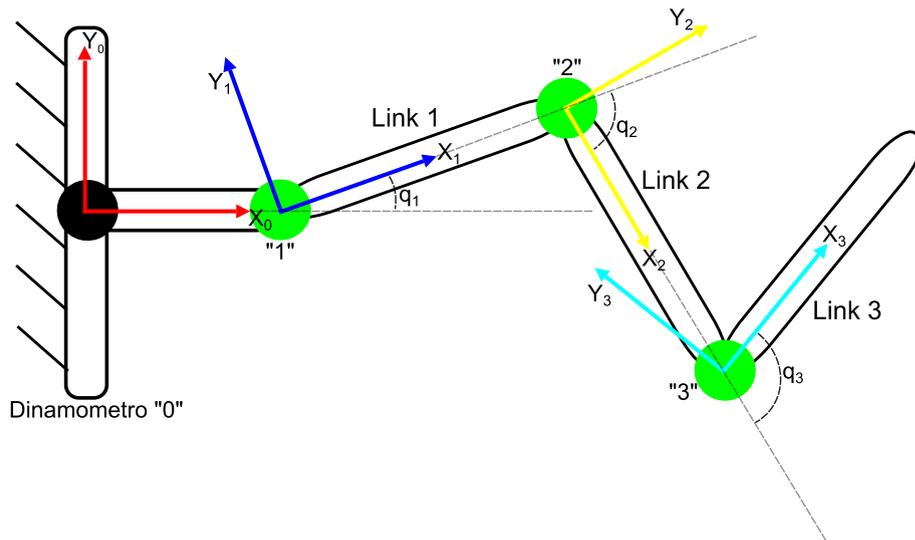


Figura 3.1: Modello con dinamometro infinitamente rigido

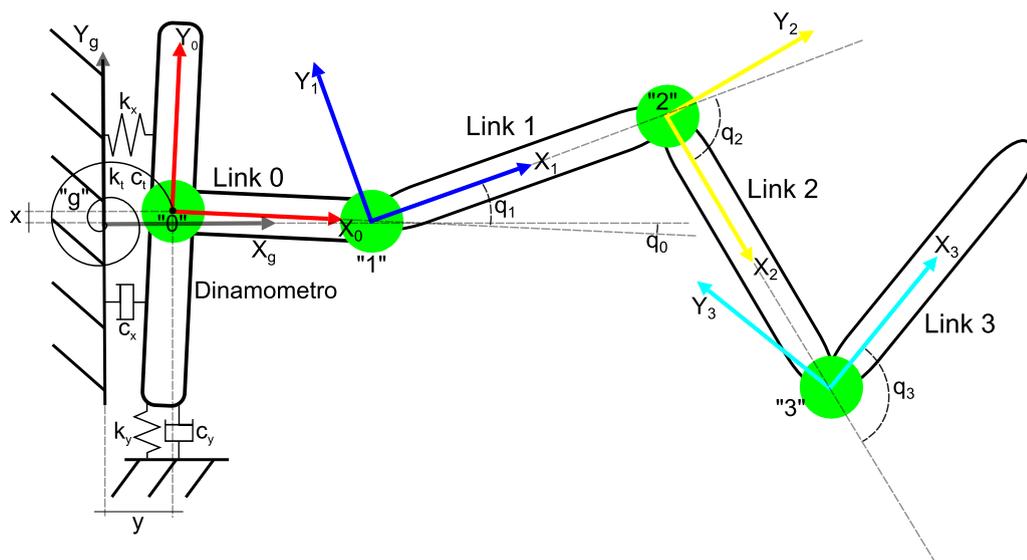


Figura 3.2: Modello con dinamometro cedevole

La figura 3.1 illustra il modello con dinamometro infinitamente rigido mentre la figura 3.2 rappresenta il modello con dinamometro cedevole. Come si vede dalla figura 3.1 la numerazione dei link, dei giunti e dei sistemi di riferimento ad essi ancorati va da 1 a 3. Il dinamometro ed il sistema di coordinate ad esso vincolato e, in questo caso, coincidente con l'inerziale sono identificati dal numero 0. Nel modello nel quale si considera la cedevolezza del dinamometro (figura 3.2), la notazione descritta precedentemente subisce un'unica variazione: il braccio del dinamometro è considerato il primo link (*Link 0*) ed il sistema di riferimento inerziale, "G" non coincide più con il sistema "0". Lo spostamento e la rotazione del sistema di riferimento "0" rispetto a "G" sono descritti dall'angolo q_0 attorno all'asse \vec{Z}_0 e dalle coordinate x lungo \vec{X}_G e y lungo \vec{Y}_G .

3.1 Schemi a blocchi

Qui di seguito sarà illustrata la realizzazione degli schemi a blocchi dei due modelli del manipolatore. Il toolbox utilizzato per la costruzione dei modelli è SimMechanics di Simulink/MatLab[®]. Dai modelli delle figure 3.1 e 3.2 sono stati costruiti, rispettivamente gli schemi a blocchi delle figure 3.3 e 3.4.

3.1.1 Schema a blocchi con dinamometro infinitamente rigido

Lo schema 3.3 è caratterizzato dai blocchi:

- **Machine environment** è un blocco nel quale si dichiara che il modello non è soggetto alla forza gravitazionale;
- **Ground** identifica il sistema inerziale rispetto al quale si definiscono le forze e coppie di reazione "G";
- **Dinamometro** è un blocco *Weld* senza alcun grado di libertà e rappresenta lo strumento di misura per il computo delle forze e delle coppie alla base e il sistema di riferimento "0" coincidente in questo caso con "G";
- **Braccio dinamometro** è un blocco *Body* e descrive la base del manipolatore, in esso sono definite proprietà di massa, inerzia, lunghezza baricentrica e lunghezza del link, l'unica che in questo modello influenzi le equazioni 4.2;
- **Giunto 1, Giunto 2 e Giunto 3** sono blocchi *Revolute* che identificano i tre giunti rotazionali del robot e rispettivamente i sistemi di riferimento "1", "2" e "3", in essi è stato dichiarato l'asse \vec{Z}_i per il quale è consentita la rotazione;
- **Link 1, Link 2 e Link 3** sono blocchi *Body* rappresentanti i tre link del robot; in essi sono stati inseriti i dati dei link del robot dalla tabella 3.2;
- **Attuatore giunto 1, Attuatore giunto 2 e Attuatore giunto 3** sono tre blocchi *Joint Actuator* che rappresentano i micromotori alloggiati rispettivamente nei *Link 0, Link 1 e Link 2*;

- **PVA giunto 1**, **PVA giunto 2** e **PVA giunto 3** ciascuno di essi è un insieme di tre blocchi *From Workspace* che prelevano i valori di posizione velocità ed accelerazione dal Workspace di MatLab[®] per fornirli ai blocchi degli attuatori; tali valori sono generati da una opportuna funzione;
- **SD**, **S1**, **S2** e **S3** sono blocchi *Joint Sensor* nei quali sono state definite le grandezze, in uscita dal sistema di misura e dai giunti, di cui si necessita conoscere il valore, per il successivo procedimento di identificazione dei parametri:
 - **SD** , in esso sono state definite come uscite i valori delle forza e coppia di reazione che in questo caso coincidono con l'opposto delle forze e delle coppie alla base,
 - **S1**, **S2** e **S3** forniscono rispettivamente i valori di posizione velocità ed accelerazione;
- **PVA misurate** è un blocco *To Workspace* che raccoglie tutte i valori in uscita dai blocchi S1, S2 e S3 e li rende disponibili nel Workspace di MatLab[®];
- **Base Force** e **Base Torque** sono due blocchi *To Workspace* che rendono disponibili nel workspace di MatLab[®] i valori in uscita dal blocco **SD** .

3.1.2 Schema a blocchi con dinamometro cedevole

Lo schema a blocchi del modello con ipotesi di cedevolezza del dinamometro segue la costruzione del precedente. Le principali differenze sono qui elencate:

- **Dinamometro o Giunto 0** è un blocco *Planar* e rappresenta lo strumento di misura come un giunto planare nel quale sono consentiti gli spostamenti lungo gli assi \vec{X}_G e \vec{Y}_G e la rotazione attorno l'asse \vec{Z}_G ;
- **Braccio dinamometro o link 0** è un blocco *Body* che descrive la struttura dinamometro-base del manipolatore e nel quale sono state definiti i valori di massa, inerzie e posizione baricentrica; poiché l'unico valore noto è la lunghezza l_{X_0} la scelta degli altri valori è stata presa in maniera realistica (tabella 3.2);
- **Molle-Smorzatori** è un blocco *Joint Spring & Damper* nel quale sono stati definiti i valori delle rigidzze k e delle costanti di smorzamento c attribuiti al dinamometro (tabella 3.1);
- **SDx**, **SDy** e **SDq** sono blocchi *Joint Sensor* nei quali sono state definite le grandezze, in uscita dal dinamometro-base del manipolatore, di cui si deve conoscere il valore per affrontare il procedimento di identificazione dei parametri. I blocchi **SDx** e **SDy** forniscono rispettivamente i valori di accelerazione lungo gli assi \vec{X}_G e \vec{Y}_G . Il blocco **SDq** fornisce anche i valori posizione, velocità ed accelerazione angolari e i valori della forza di reazione di valore opposto alla forza alla base e il valore della coppia alla base.

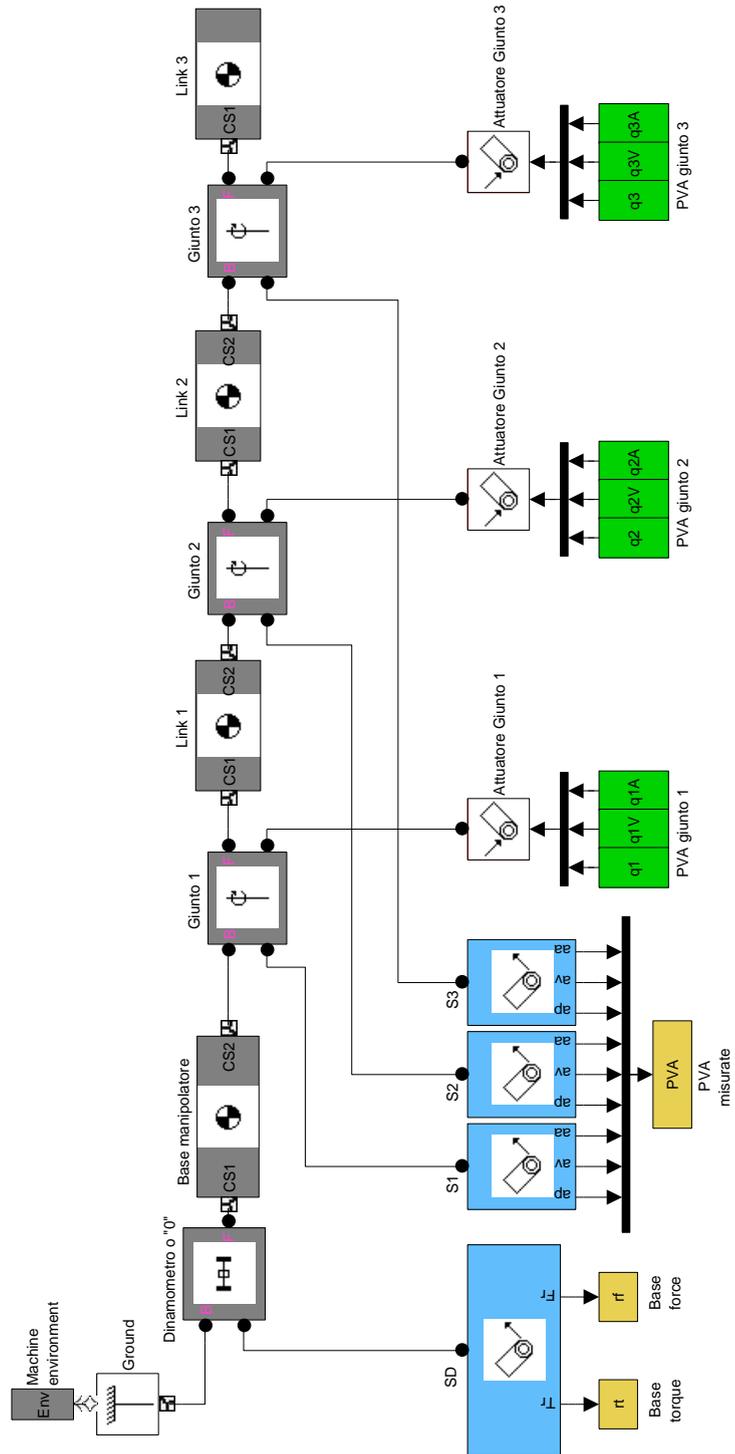


Figura 3.3: Schema a blocchi con dinamometro infinitamente rigido

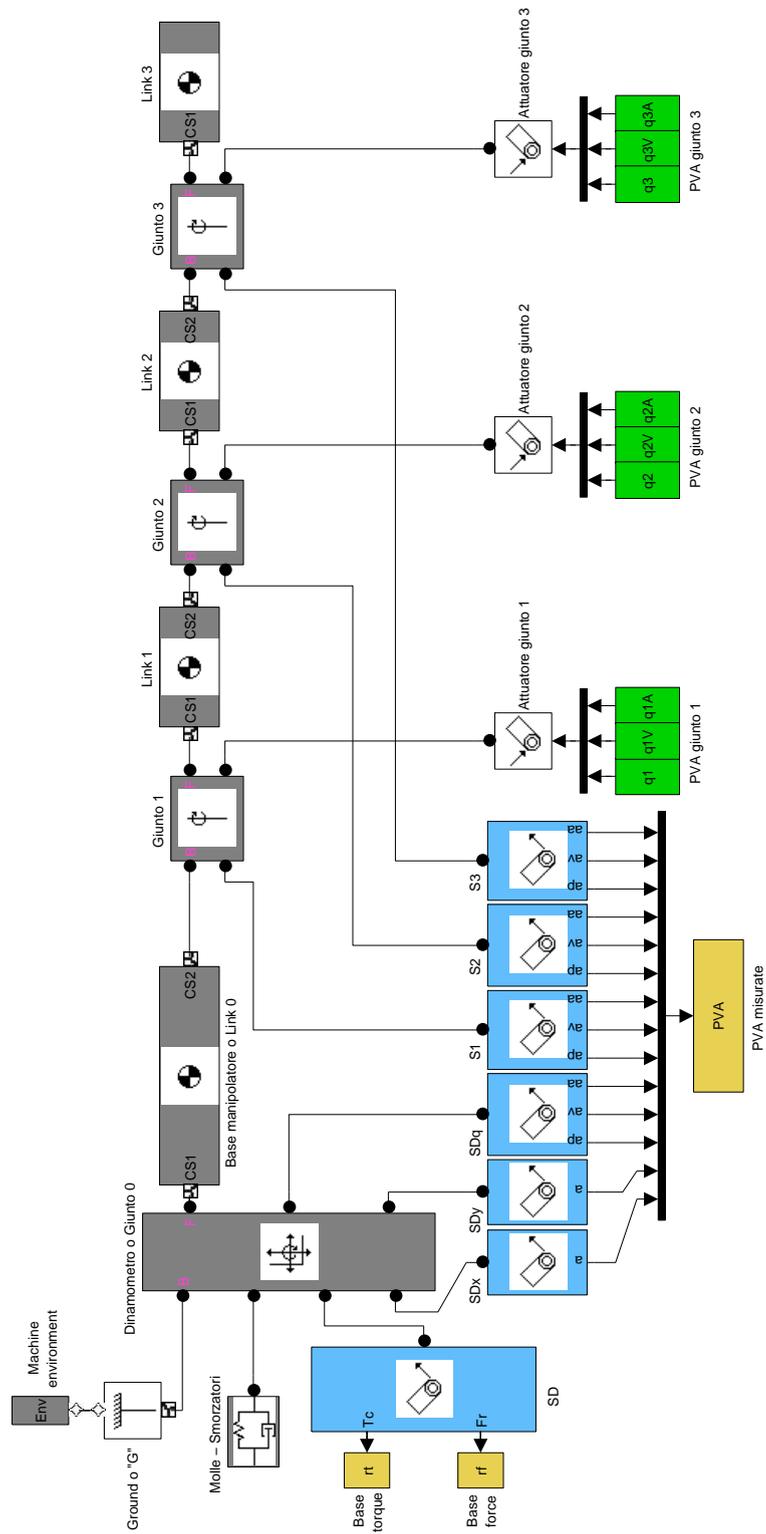


Figura 3.4: Schema a blocchi con ipotesi di cedevolezza del dinamometro

Tabella 3.2: **Parametri link**

	Link 1	Link 2	Link 3	Link 0
m [kg]	0.615	0.615	0.200	0.4
l_x [m]	0.176	0.176	0.135	0.087
l_{C_x} [m]	0.115	0.115	0.056	0.050
I_{xx} [kg · m ²]	$2.43 \cdot 10^{-4}$	$2.43 \cdot 10^{-4}$	$1.21 \cdot 10^{-4}$	$1.50 \cdot 10^{-4}$
I_{yy} [kg · m ²]	$2.73 \cdot 10^{-3}$	$2.73 \cdot 10^{-3}$	$1.24 \cdot 10^{-3}$	$2.00 \cdot 10^{-3}$
I_{zz} [kg · m ²]	$2.73 \cdot 10^{-3}$	$2.73 \cdot 10^{-3}$	$1.24 \cdot 10^{-3}$	$2.00 \cdot 10^{-3}$

Si rende noto che per entrambi gli schemi i blocchi **PVA misurate** restituiscono valori identicamente uguali ai valori ricevuti in input dalla funzione che genera la traiettoria poiché nella simulazione non sono stati modellati fenomeni di disturbo. L'inserimento dei blocchi è stato pensato per meglio rappresentare la realtà fisica del manipolatore. Le posizioni, velocità ed accelerazioni dei giunti e quindi dei link sono in generale afflitte da disturbi che comportano errori. Si necessita quindi l'uso di misuratori esterni di tali grandezze, come rappresentato nei due modelli in questione.

Capitolo 4

Procedura d'identificazione

4.1 Procedura di simulazione

Per effettuare le simulazioni sono stati eseguiti i seguenti passi:

- generazione di una determinata traiettoria per ognuno dei tre giunti;
- simulazione dello schema a blocchi nel quale gli attuatori ricevono in input i valori generati al passo precedente, in output sono restituiti i valori di forze e coppie alla base e i valori di posizione, velocità ed accelerazione di ciascuno giunto;
- costruzione delle equazioni delle forze e delle coppie con i parametri di base ed identificazione dei parametri di base.

Le sezioni seguenti chiariranno i punti appena elencati.

4.2 Traiettoria utilizzata

Come anticipato nel capitolo precedente i giunti degli schemi a blocchi costruiti sono collegati a degli attuatori che ricevono informazioni di posizione, velocità ed accelerazione. Tali valori sono calcolati tramite una funzione dedicata.

La funzione utilizzata per definire la traiettoria è la seguente:

$$q_i(t) = \left[10 \left(\frac{t}{t_{fin_i}} \right)^3 - 15 \left(\frac{t}{t_{fin_i}} \right)^4 + 6 \left(\frac{t}{t_{fin_i}} \right)^5 \right] \cdot (q_{fin_i} - q_{in_i}) + q_{in_i}. \quad (4.1)$$

Tale funzione, molto usata in ambito robotico, permette di avere velocità ed accelerazioni iniziali e finali gradualmente crescenti e decrescenti. Per la definizione della traiettoria è necessario scegliere q_{in_i} angolo iniziale, q_{fin_i} angolo finale e t_{fin_i} tempo al quale si raggiunge l'angolo finale. Le traiettorie sono costruite tramite interpolazione con $0 < t < t_{fin_i}$ e un passo temporale di $\Delta t = 0.0001[s]$.

4.3 Equazioni d'identificazione

Il sistema di misura del manipolatore consente di conoscere i valori delle coppie e forze alla base. Le equazioni di tali valori derivano direttamente dalle equazioni 2.27 dell'algoritmo di Newton-Eulero ed esse sono diverse per i due modelli di robot considerati:

- modello con dinamometro infinitamente rigido,

$$\vec{R}f = -({}^0\vec{f}_0) \quad (4.2a)$$

$$\vec{R}t = -({}^0\vec{n}_0); \quad (4.2b)$$

- modello con ipotesi di cedevolezza del dinamometro,

$$\vec{R}f = -({}^G\vec{f}_0) \quad (4.3a)$$

$$\vec{R}t = -({}^G\vec{n}_0). \quad (4.3b)$$

Parametri di base

Nel caso più generale per ogni link i del manipolatore ci sono dieci parametri inerziali standard: la massa (m_i), i tre primi momenti di inerzia definiti come prodotto tra la massa i e le tre distanze baricentriche rispetto il frame i ($m_i \cdot l_{C_{x_i}}$, $m_i \cdot l_{C_{y_i}}$, $m_i \cdot l_{C_{z_i}}$) e le sei componenti del tensore di inerzia, (I_{xx_i} , I_{yy_i} , I_{zz_i} , I_{xy_i} , I_{xz_i} , I_{yz_i}). A causa dei vincoli cinematici non tutti i parametri influenzano la risposta dinamica del manipolatore. Tali parametri standard possono essere suddivisi in tre categorie: unicamente identificabili, identificabili in combinazione lineare e non identificabili. Il numero minimo di parametri inerziali o loro combinazione che determinano unicamente la risposta dinamica del manipolatore e che possono essere identificati è chiamato set dei parametri di base.

I parametri di base, obiettivo dell'identificazione, sono stati determinati in modo tale che le equazioni 4.2 possano essere riscritte come equazioni lineari e che quindi si possa scriverle come prodotto matrice per vettore.

Le equazioni 4.2 e 4.3 possono essere espresse in maniera semplificata come:

$$\vec{F} = W(\ddot{q}, \dot{q}, q) \cdot \vec{\beta} + \vec{\rho} \quad (4.4)$$

dove \vec{F} è il vettore $[n \times 1]$ contenente le forze (Rf_x, Rf_y, Rf_z), e le coppie (Rt_x, Rt_y, Rt_z), W è la matrice di osservazione o regressore $[n \times r]$ dipendente dagli angoli, dalle velocità angolari e dalle accelerazioni angolari dei giunti, $\vec{\beta}$ è il vettore $[r \times 1]$ dei parametri di base e $\vec{\rho}$ è il vettore degli errori residui $[n \times 1]$. La dimensione n indica i gradi di libertà del robot, la dimensione r indica il numero di parametri di base.

I modelli considerati in questa tesi, rappresentando un robot planare, possiedono tre gradi di libertà. Il vettore F ha dimensione $[3 \times 1]$; esso è $F = [Rf_x, Rf_y, Rt_z]^T$. I parametri inerziali standard sono quattro per ogni link i : la massa m_i , i primi

momenti di inerzia $m_i \cdot l_{C_{x_i}}$, $m_i \cdot l_{C_{y_i}}$ ed il solo secondo momento di inerzia I_{zz_i} . Si noti nella tabella 3.2 non sono presenti i valori della distanza baricentrica lungo \vec{Y}_i , cioè il baricentro dei link del i -esimo link giace sul asse \vec{X}_i , e ciò comporta la riduzione del numero dei parametri dinamici standard a tre: $m_i, m_i \cdot l_{C_{x_i}}, I_{zz_i}$ per ogni singolo link.

Valori traiettoria

I valori q_{in_i} , q_{fin_i} e t_{fin_i} utilizzati per definire la traiettoria per le simulazioni di entrambi i modelli sono stati scelti tali da rendere la matrice W dell'equazione 4.4 a rango pieno (tabella 4.1).

Tabella 4.1: Valori utilizzati per definire la traiettoria

	Giunto 1	Giunto 2	Giunto 3
q_{in_i} [deg]	0	0	0
q_{fin_i} [deg]	90	90	90
t_{fin_i} [s]	4	3	2

Gli andamenti delle posizioni, velocità ed accelerazioni angolari ottenute con una funzione che calcola i valori da inviare agli attuatori dei giunti dei modelli 3.3 e 3.4. In figura 4.1 sono mostrate le andature di posizione, velocità e accelerazione dei giunti.

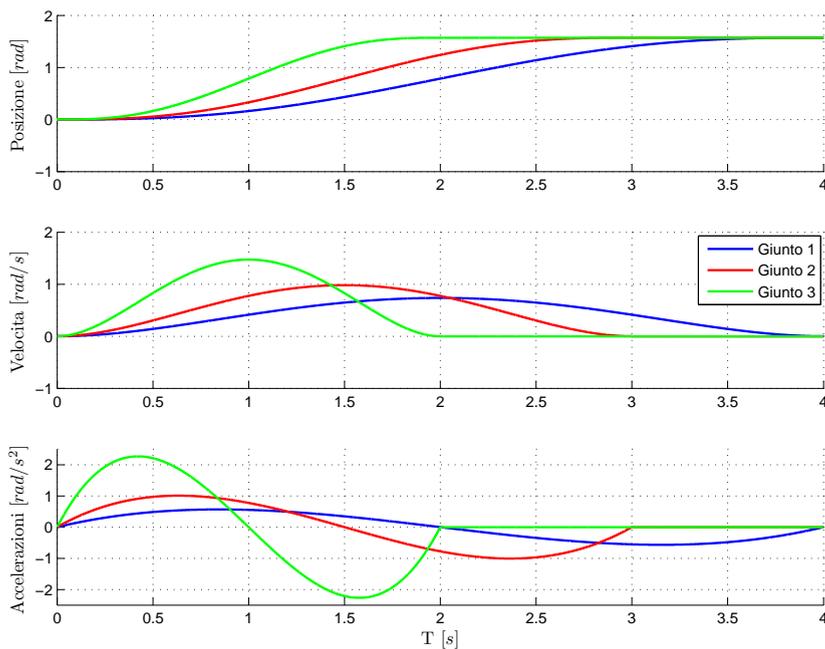


Figura 4.1: Traiettoria

Identificazione dei parametri di base

Per avere il maggior numero di dati disponibili ogni simulazione utilizza come tempo di raccolta dati il massimo dei valori t_{fin_i} , cioè $t_{sim} = t_{fin_1} = 4[s]$. Per ogni simulazione è stata effettuata una campionatura costante nel tempo. Tali campionamenti avvengono ad ogni istante di tempo δt , scelto tale che sia multiplo di Δt e che $t_{sim}/\delta t = n$ con $n \in \mathbb{N}$. Ad ogni campionamento sono stati prelevati dagli schemi a blocchi posizioni, velocità ed accelerazioni angolari di ciascuno giunto, le forze e le coppie alla base.

Le equazioni utilizzate per la costruzione della matrice W sono quelle della componente della forza \vec{Rf} lungo l'asse \vec{X}_G perché verosimilmente ha un valore assoluto maggiore della componente lungo \vec{Y}_G e quindi più facilmente misurabile e la componente della coppia \vec{Rt} lungo \vec{Z}_G , peraltro unica componente della coppia non nulla.

Si ricorre quindi all'uso di due equazioni per ogni campionamento per $t_{sim}/\delta t$ campionamenti. Il termine n che fino ad ora assumeva il valore dei gradi di libertà del manipolatore ora invece adotta il valore $n = 2 \cdot t_{sim}/\delta t$.

La 4.4 assume perciò tale forma:

$$\begin{pmatrix} Rf_{X_{\delta t}} \\ Rf_{X_{\delta t + \delta t}} \\ \dots \\ Rf_{X_{t_{fin}/\delta t}} \\ Rt_{Z_{\delta t}} \\ Rt_{Z_{\delta t + \delta t}} \\ \dots \\ Rt_{Z_{t_{fin}/\delta t}} \end{pmatrix} = \begin{bmatrix} W, Rf_{X_{\delta t,1}} & W, Rf_{X_{\delta t,2}} & \dots & W, Rf_{X_{\delta t,r}} \\ W, Rf_{X_{\delta t + \delta t,1}} & W, Rf_{X_{\delta t + \delta t,2}} & \dots & W, Rf_{X_{\delta t + \delta t,r}} \\ \vdots & \vdots & \vdots & \vdots \\ W, Rf_{X_{t_{fin}/\delta t,1}} & W, Rf_{X_{t_{fin}/\delta t,2}} & \dots & W, Rf_{X_{t_{fin}/\delta t,r}} \\ W, Rt_{Z_{\delta t,1}} & W, Rt_{Z_{\delta t,2}} & \dots & W, Rt_{Z_{\delta t,r}} \\ W, Rt_{Z_{\delta t + \delta t,1}} & W, Rt_{Z_{\delta t + \delta t,2}} & \dots & W, Rt_{Z_{\delta t + \delta t,r}} \\ \vdots & \vdots & \vdots & \vdots \\ W, Rt_{Z_{t_{fin}/\delta t,1}} & W, Rt_{Z_{t_{fin}/\delta t,2}} & \dots & W, Rt_{Z_{t_{fin}/\delta t,r}} \end{bmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_r \end{pmatrix} + \begin{pmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_r \end{pmatrix}$$

in cui i primi $n/2$ elementi sono i valori le forze alla base rilevate ad ogni δt mentre gli elementi da $n/2 + 1$ a n sono i valori delle coppie rilevati ad ogni campionatura.

Per ricavare i valori dei parametri di base, vettore $\vec{\beta}$, si effettua l'inversione dell'equazione 4.4 dopo aver sostituito a ciascuna variabile il proprio valore. Per eseguire l'inversione della matrice W si ricorre all'utilizzo della pseudoinversa di Moore-Penrose ($W^\dagger [r \times n]$). La matrice pseudoinversa di Moore-Penrose (W^\dagger) e la soluzione dell'equazione 4.5 (vettore $\vec{\beta}$) godono delle seguenti proprietà:

- se $n = r$, $W^\dagger = W^{-1}$ se W è a rango pieno;
- se $n > r$, che corrisponde al caso in cui ci sia un numero di equazioni superiore alle variabili da ricercare, la matrice pseudoinversa restituisce la soluzione tale che la norma del vettore residuo sia minima:

$$\hat{\beta} = \beta (\text{Min} \|\vec{\rho}\|^2).$$

$\hat{\beta} = W^\dagger \cdot F$ risulta essere la soluzione più vicina possibile, nel senso dei minimi quadrati, al vettore β .

- se $n < r$, in questo caso le equazioni a disposizione sono in numero inferiore alle variabili da identificare quindi il problema ha generalmente infinite soluzioni e la soluzione ottenuta con la matrice pseudoinversa W^\dagger è la soluzione particolare nella quale la norma di $\vec{\beta}$: $\|\vec{\beta}\|$ è minima.

Se la matrice W è a rango pieno allora per i tre casi appena presentati il calcolo della matrice pseudoinversa di Moore-Penrose è:

- caso $n = r$, come già detto $W^\dagger = W^{-1}$;
- caso $n > r$, $W^\dagger = (W^T \cdot W)^{-1} \cdot W^T$;
- caso $n < r$, $W^\dagger = W^T \cdot (W \cdot W^T)^{-1}$.

In tutte le simulazioni il tempo di campionamento è scelto tale da fornire un numero di equazioni n maggiore del numero delle incognite r . Così facendo la matrice W rientra nel caso di un sistema sovradeterminato in cui il numero di righe è maggiore del numero delle colonne, $W^\dagger = (W^T \cdot W)^{-1} \cdot W^T$. Quindi l'equazione di risoluzione che consentirà l'identificazione dei parametri di base è:

$$\hat{\beta} = W^\dagger \cdot F. \tag{4.5}$$

Capitolo 5

Risultati simulazioni

5.1 Modello con dinamometro infinitamente rigido

Come anticipato i parametri di standard di ogni link di questo modello sono la massa m_i , il primo momento di inerzia $m_i \cdot l_{C_{x_i}}$, ed il secondo momento di inerzia I_{zz_i} per un totale di nove elementi. Esaminando le equazioni 4.2 sono state calcolati i parametri di base che possono essere identificati in tale modello. Tali parametri sono combinazioni lineari dei parametri standard e consentono di riscrivere le equazioni 4.2 in forma lineare.

I parametri di base individuati sono sei e sono mostrati in tabella 5.1

Tabella 5.1: Set parametri di base modello con dinamometro non cedevole

Parametro	Valore
$\beta_1 = I_{zz_3} + m_3 \cdot l_{C_{x_3}}^2$	$= 18.6720 \cdot 10^{-4} [kg \cdot m^2]$
$\beta_2 = I_{zz_2} + m_2 \cdot l_{C_{x_2}}^2 + m_3 \cdot l_{x_2}^2$	$= 170.586 \cdot 10^{-4} [kg \cdot m^2]$
$\beta_3 = I_{zz_1} + m_1 \cdot l_{C_{x_1}}^2 + (m_3 + m_2) \cdot l_{x_1}^2$	$= 361.088 \cdot 10^{-4} [kg \cdot m^2]$
$\beta_4 = m_3 \cdot l_{C_{x_3}}$	$= 112.000 \cdot 10^{-4} [kg \cdot m]$
$\beta_5 = m_2 \cdot l_{C_{x_2}} + m_3 \cdot l_{x_2}$	$= 1059.25 \cdot 10^{-4} [kg \cdot m]$
$\beta_6 = m_1 \cdot l_{C_{x_1}} + (m_3 + m_2) \cdot l_{x_1}$	$= 2141.65 \cdot 10^{-4} [kg \cdot m]$

I parametri β_1 , β_2 e β_3 rappresentano i secondi momenti di inerzia dei link misurati lungo gli assi verticali passanti per i relativi giunti. I parametri β_4 , β_5 e β_6 rappresentano i primi momenti di inerzia dei link misurati lungo gli assi \vec{X}_i .

Le prime simulazioni effettuate sono state svolte con dati di output dallo schema a blocchi in assenza di errori. I valori di output (posizione, velocità, accelerazione, forze e coppie) hanno consentito la costruzione della matrice d'osservazione W e l'identificazione dei parametri di base si è risolta con l'inversione dell'equazione 4.4 ma

con vettore $\vec{\rho}$ nullo. Si è avuto in questo caso $\hat{\beta} = \beta$ e quindi i valori dei parametri di base sono stati identici a quelli di tabella 5.1.

In seguito si sono alterati i valori in uscita dallo schema a blocchi con l'aggiunta di errori casuali. In questo caso si è risolta l'identificazione dei parametri con l'equazione 4.5 quindi con la minimizzazione di $\vec{\rho}$. I valori degli elementi del vettore $\hat{\beta}$ ottenuti sono perciò una approssimazione della reale soluzione. La verifica della precisione di tali soluzioni e quindi sull'affidabilità del metodo di risoluzione è stata testata con simulazioni a diversi δt . I tempi di campionamento usati sono:

- $\delta t = 1[s]$, 4 campionamenti;
- $\delta t = 0.1[s]$, 40 campionamenti;
- $\delta t = 0.01[s]$, 40 campionamenti;
- $\delta t = 0.001[s]$, 4000 campionamenti;

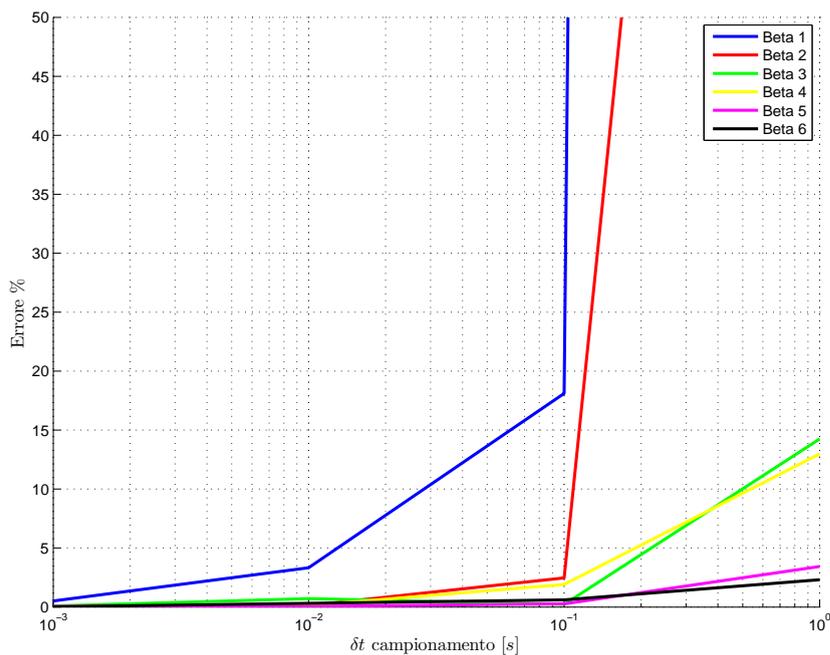


Figura 5.1: Dinamometro non cedevole, errore% $\hat{\beta}_i$ con $-1\% \cdot X_{max} < err < 1\% \cdot X_{max}$

L'errore introdotto sui dati di output err è un errore di tipo casuale proporzionale al valore massimo della corrispondente grandezza in uscita dallo schema a blocchi:

$$-p\% \cdot X_{max} < err < p\% \cdot X_{max} \quad (5.1)$$

dove $p\%$ è il valore percentuale scelto tra 1, 2 e 5 e X_{max} è il valore massimo della grandezza considerata.

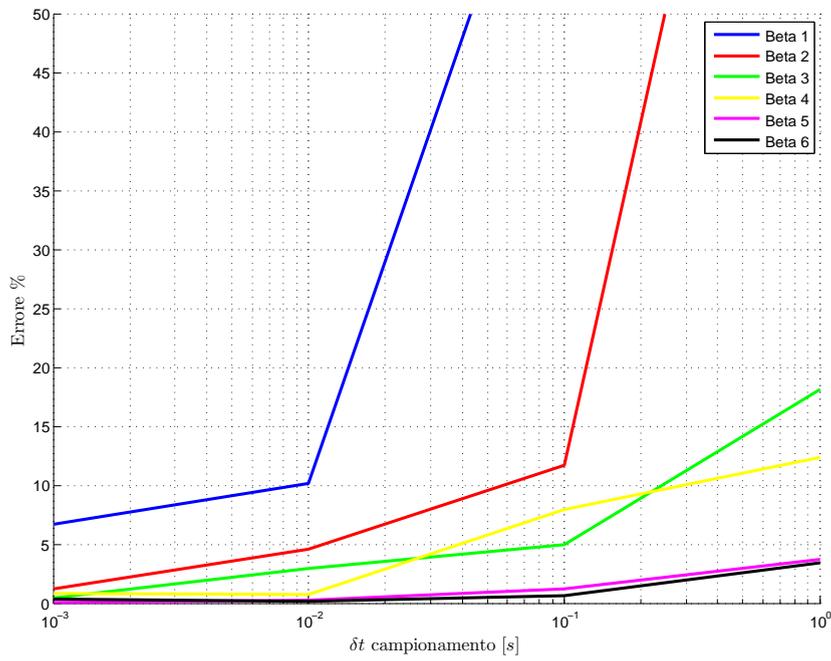


Figura 5.2: Dinamometro non cedevole, errore% $\hat{\beta}_i$ con $-2\% \cdot X_{max} < err < 2\% \cdot X_{max}$

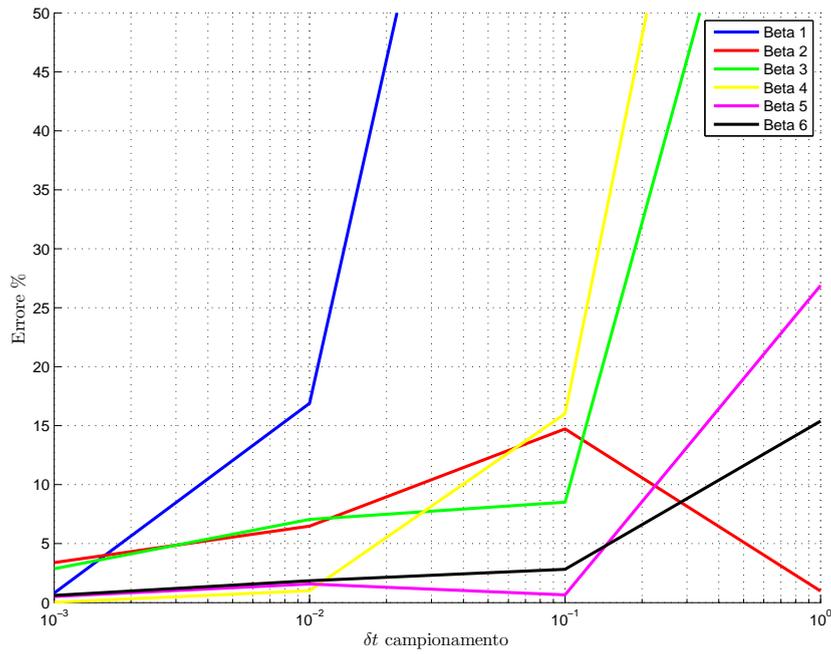


Figura 5.3: Dinamometro non cedevole, errore% $\hat{\beta}_i$ con $-5\% \cdot X_{max} < err < 5\% \cdot X_{max}$

Nelle figure 5.1, 5.2 e 5.3 sono mostrati gli andamenti degli errori percentuali ottenuti con le identificazioni a diversi δt per ogni elemento $\hat{\beta}_i$.

Gli errori inseriti nel calcolo del set di parametri di base sono errori casuali ciò comporta che ogni simulazione svolta sia diversa dalla precedente e che talune volte a piccoli valori di δt ci siano errori più elevati rispetto a δt maggiori. In linea generale, però, si nota un deciso abbassamento del valore percentuale degli errori sul set di parametri di base ottenuti $\hat{\beta}$ all'aumentare del numero dei campionamenti. In questo modello i sei parametri di base ottenuti si discostano poco dai valori reali poiché la minimizzazione della norma del vettore residuo risulta efficiente.

5.2 Modello con dinamometro cedevole

Tale modello risulta maggiormente realistico rispetto al modello precedente. In esso è stata introdotta la cedevolezza del dinamometro e l'introduzione di tale caratteristica comporta equazioni delle forze e delle coppie alla base del manipolatore diverse da quelle del modello precedente. Difatti la cedevolezza del dinamometro consente un moto planare dello stesso dovuto alla dinamica dei link 1, 2 e 3. Si ottiene perciò un manipolatore a quattro link le cui equazioni delle forze e coppie alla base sono le equazioni 4.3. Dall'esame delle equazioni 4.3 si è trovato il nuovo set di parametri di base (tabella 5.2). Si fa presente che lo schema a blocchi utilizzato per le simulazioni è quello di figura 3.4, nel quale in output si hanno le misure delle accelerazioni in \vec{X}_G e \vec{Y}_G e delle posizioni, velocità ed accelerazioni angolari del giunto θ .

Tabella 5.2: Set parametri di base modello con dinamometro cedevole

Parametri	Valore
$\beta_1 = I_{zz3} + m_3 \cdot l_{C_{x_3}}^2$	= 18.6720 $\cdot 10^{-4}$ [kg \cdot m ²]
$\beta_2 = I_{zz2} + m_2 \cdot l_{C_{x_2}}^2 + m_3 \cdot l_{x_2}^2$	= 170.586 $\cdot 10^{-4}$ [kg \cdot m ²]
$\beta_3 = I_{zz1} + m_1 \cdot l_{C_{x_1}}^2 + (m_3 + m_2) \cdot l_{x_1}^2$	= 361.088 $\cdot 10^{-4}$ [kg \cdot m ²]
$\beta_4 = I_{zz0} + m_0 \cdot l_{C_{x_0}}^2 + (m_3 + m_2 + m_1) \cdot l_{x_0}^2$	= 123.341 $\cdot 10^{-4}$ [kg \cdot m ²]
$\beta_5 = m_3 \cdot l_{C_{x_3}}$	= 112.000 $\cdot 10^{-4}$ [kg \cdot m]
$\beta_6 = m_2 \cdot l_{C_{x_2}} + m_3 \cdot l_{x_2}$	= 1059.25 $\cdot 10^{-4}$ [kg \cdot m]
$\beta_7 = m_1 \cdot l_{C_{x_1}} + (m_3 + m_2) \cdot l_{x_1}$	= 2141.65 $\cdot 10^{-4}$ [kg \cdot m]
$\beta_8 = m_0 \cdot l_{C_{x_0}} + (m_3 + m_2 + m_1) \cdot l_{x_0}$	= 1328.10 $\cdot 10^{-4}$ [kg \cdot m]
$\beta_9 = m_3 + m_2 + m_1 + m_0$	= 1.58000 [kg]

I parametri β_1 , β_2 , β_3 , β_5 , β_6 e β_7 sono rispettivamente uguali ai parametri del set di parametri di base del modello con dinamometro infinitamente rigido. I tre nuovi elementi del set sono β_4 , β_8 e β_9 dove i primi due descrivono rispettivamente il secondo ed il primo momento di inerzia della struttura dinamometro-base del manipolatore

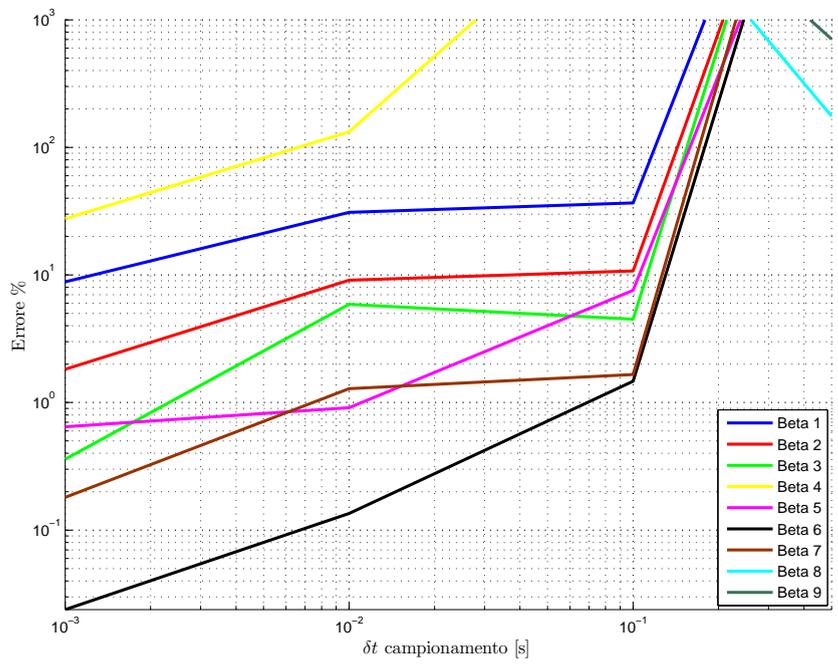


Figura 5.4: Dinamometro cedevole, errore % $\hat{\beta}_i$ con $-1\% \cdot X_{max} < err < 1\% \cdot X_{max}$

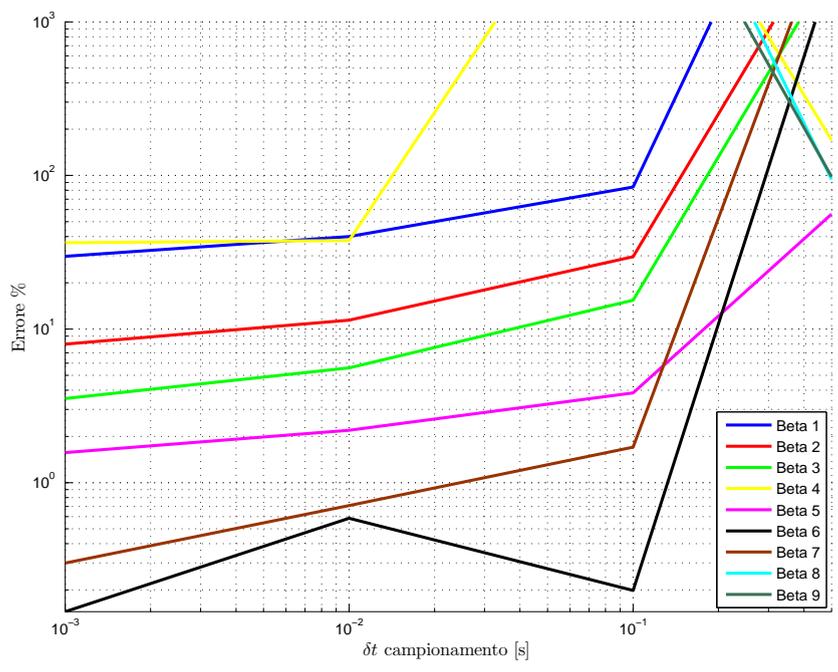


Figura 5.5: Dinamometro cedevole, errore % $\hat{\beta}_i$ con $-2\% \cdot X_{max} < err < 2\% \cdot X_{max}$

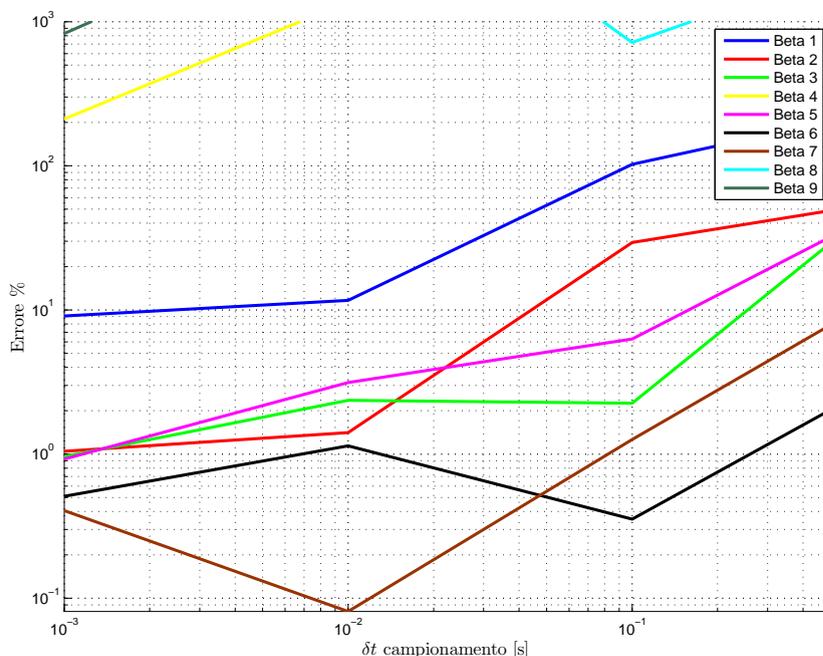


Figura 5.6: Dinamometro cedevole, errore % $\hat{\beta}_i$ con $-5\% \cdot X_{max} < err < 5\% \cdot X_{max}$

misurati il primo lungo l'asse \vec{Z}_0 ed il secondo in \vec{X}_0 mentre β_9 rappresenta la massa del manipolatore.

Anche per questo modello le simulazioni sono state effettuate prima senza l'introduzione di errori e poi con l'aggiunta di un errore percentuale sulle misure di output dallo schema a blocchi. Il tipo di errore inserito è lo stesso del modello precedente (5.1). I δt usati per verificare la minimizzazione della norma del vettore residuo sono gli stessi utilizzati per le simulazioni del modello precedente, a meno del primo preso pari a 0.5[s] per poter avere W con $n = 16 > 9 = r$. Le figure 5.4, 5.5 e 5.6 riportano gli errori ottenuti dalle simulazione. Anche da questi grafici si individuano gli elementi del vettore $\hat{\beta}$ soggetti ad una minimizzazione degli errori poco efficiente. Gli elementi che più risentono dell'introduzione degli errori nei dati di output dello schema a blocchi sono l'elemento che rappresenta la massa del manipolatore $\hat{\beta}_9$, l'elemento che descrive il primo momento d'inerzia relativo al link θ , $\hat{\beta}_8$ e l'elemento $\hat{\beta}_4$ che rappresenta il secondo momento di inerzia del link θ . Dall'esame delle equazioni della forza e della coppia alla base di si è notata una relazione tra questi tre parametri e la presenza di sole variabili relative alla rotazione e agli spostamenti del dinamometro negli elementi della matrice W ad essi corrispondenti.

5.3 Modello combinato

Nel modello con dinamometro è stata eseguita l'identificazione dei parametri di base di tabella 5.2 utilizzando le equazioni 4.3. Per effettuare tale identificazioni si sono dovute utilizzare delle misure sul giunto θ . Tali grandezze sono tuttavia molto piccole, dovute alla rigidità delle molle con cui è stato modellato il dinamometro,

perciò la loro misura nella realtà risulta difficilmente affrontabile, inoltre come osservato precedentemente esiste una relazione tra tali grandezze e una poco efficiente identificazione dei parametri ad essi univocamente abbinati. Si è pensato quindi di trascurare gli effetti del moto indotto cui è soggetto il corpo dinamometro-base del manipolatore e quindi eseguire delle simulazioni del modello con dinamometro cedevole 3.2 identificando poi il set di parametri di base derivante dalle equazioni 4.2 relative al modello con dinamometro non cedevole. Di seguito si riassumono gli aspetti più importanti di prossime simulazioni:

- utilizzo dello schema a blocchi di figura 3.4;
- per la costruzione dell'equazione 4.4 si sono utilizzate le equazioni 4.2;
- utilizzo delle equazioni 4.2 comporta il trascurare tutte le grandezze relative al moto del dinamometro-base del manipolatore;
- il set di parametri di base che può essere identificato con questa simulazione è il set di parametri di base di tabella 5.1, quindi nessuna informazione sui parametri dinamici del corpo dinamometro-base del manipolatore sarà disponibile.

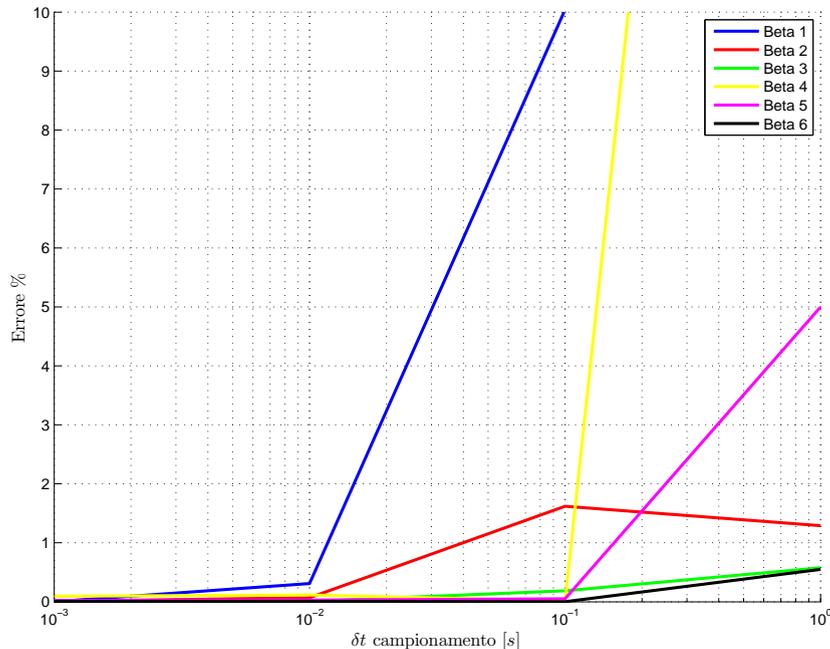


Figura 5.7: Modello combinato, senza introduzione di errori sulle misure

Sono state effettuate quattro simulazioni. La prima simulazione è stata svolta senza l'introduzione di errori per verificare quanto i valori del vettore $\hat{\beta}$ ottenuti si discostino dai valori reali.

Nelle tre simulazioni successive invece sono stati introdotti gli errori nella forma 5.1.

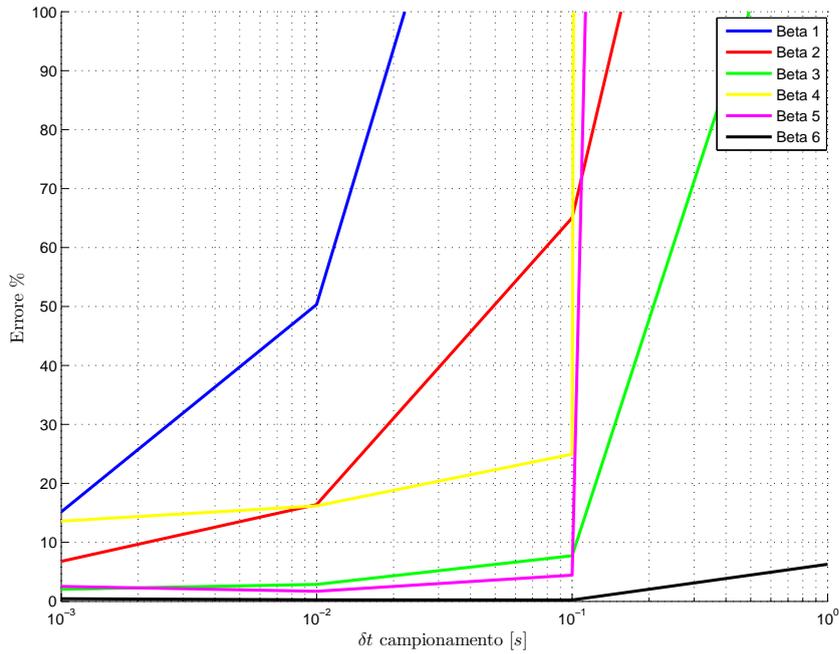


Figura 5.8: Modello combinato, errore % $\hat{\beta}_i$ con $-1\% \cdot X_{max} < err < 1\% \cdot X_{max}$

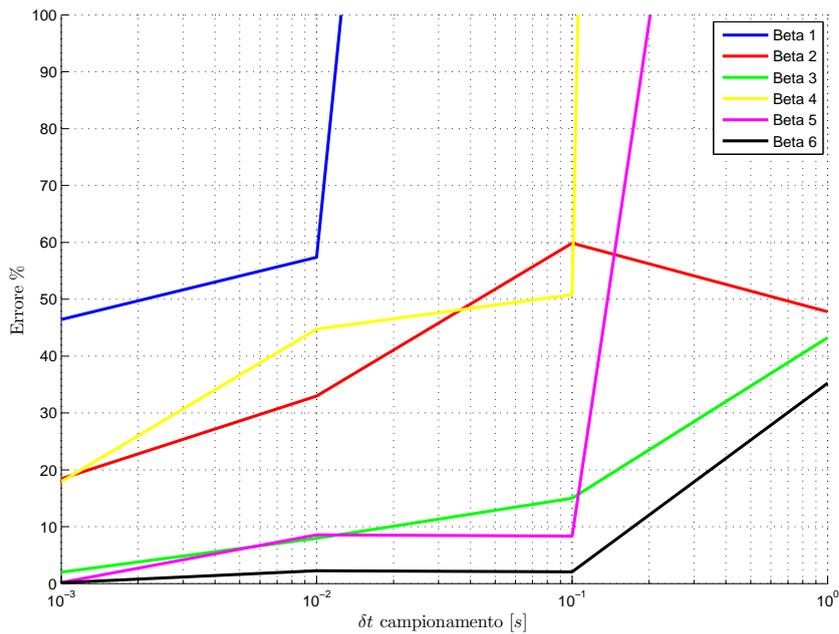


Figura 5.9: Modello combinato, errore % $\hat{\beta}_i$ con $-2\% \cdot X_{max} < err < 2\% \cdot X_{max}$

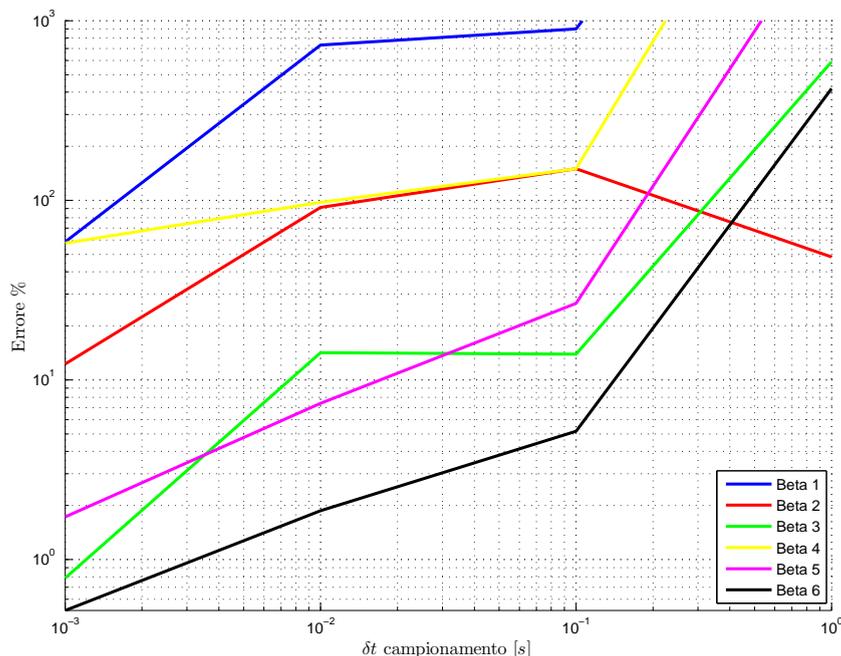


Figura 5.10: Modello combinato, errore % $\hat{\beta}_i$ con $-5\% \cdot X_{max} < err < 5\% \cdot X_{max}$

Tutte le simulazioni sono state fatte agli stessi δt di campionamento delle simulazioni del modello precedente. I grafici degli errori ottenuti sono riportati nelle figure 5.7, 5.8, 5.9 e 5.10. Dall'osservazione dei grafici si evince che in assenza di errori si hanno ottimi risultati con tempi di campionamento pari a un centesimo di secondo. Con l'introduzione di errori nelle variabili in output dagli schemi a blocchi i parametri di base $\hat{\beta}_1$ e $\hat{\beta}_4$ sono affetti da errori non trascurabili anche a tempi di campionamento piccoli.

5.4 Conclusioni

La ricerca del set di parametri di base dei due modelli ha portato in evidenza che la determinazione dei β_i segue un preciso sviluppo matematico.

Considerando i set di parametri trovati per i due modelli (tabelle 5.1 e 5.2) e che il modello con dinamometro infinitamente rigido ed il modello con dinamometro cedevole sono riconducibile rispettivamente ad un manipolatore planare a tre link con base fissa e ad un manipolatore planare a quattro link con base mobile si può affermare che il set di parametri di base segue tale schema:

- il numero di parametri di base è pari al doppio del numero dei link del manipolatore i cui parametri dinamici rientrano nella dinamica del moto N , se la base è mobile l'ultimo elemento β_{2N+1} è la somma delle masse di tali link:

per $i = 1 \rightarrow N$:

$$\beta_{2N+1} = \sum_{k=i}^N m_k;$$

- i parametri che includono i primi momenti di inerzia dei link sono in numero pari al numero dei link e costruiti a partite dal primo link come segue:

per $i = 1 \rightarrow N$ e $j = N \rightarrow 1$:

$$\beta_j = m_i \cdot l_{C_{x_i}} + \left(\sum_{k=i+1}^N m_k \right) \cdot l_{x_i};$$

- i parametri che includono i secondi momenti di inerzia dei link sono anch'essi in numero pari al numero dei link e costruiti a partite dal primo link come segue:

per $i = 1 \rightarrow N$ e $j = 2N \rightarrow N + 1$:

$$\beta_j = I_{zz_i} + m_i \cdot l_{C_{x_i}}^2 + \left(\sum_{k=i+1}^N m_k \right) \cdot l_{x_i}^2.$$

Si evidenzia il fatto che i parametri di tabella 5.2 sono stati identificati con tali criteri ma con una numerazione diversa. Il numero di link che rientrano nella equazioni della dinamica sono quattro $N = 4$ ma sono stati numerati a partire da 0, perciò i cicli saranno svolti per $i = 0 \rightarrow N - 1$ e le sommatorie delle masse avranno come limite superiore $N - 1$.

Le simulazioni hanno evidenziato che con tempi di campionamento bassi si è in grado di ottenere buoni risultati del set dei parametri di base del individuati con le equazioni 4.2. Dalle simulazioni effettuate sui modelli si è inoltre notato che l'individuazione dei tre parametri di base introdotti dall'utilizzo delle equazioni 4.3 con l'uso delle misure delle grandezze relative alla struttura dinamometro-base del manipolatore è soggetta ad altissimi errori e perciò non efficiente. Le simulazione effettuate sul modello con dinamometro cedevole trascurando le misure del moto dello stesso hanno evidenziato che è possibile identificare i parametri di base con elevata accuratezza solamente se si utilizzano sensori con ottime prestazioni metrologiche.

Appendice A

Modello con flessibilità di giunto

Nella realtà i giunti del manipolatore sono caratterizzati da una certa flessibilità. Tale flessibilità, dovuta alla trasmissione del moto attraverso elementi che non garantiscono rigidità elevate, introduce una variazione temporale tra i valori di posizione, velocità ed accelerazione angolari inviati come input ai servomotori dei giunti e quelli assunti dai link.

Nelle simulazioni precedenti si è ipotizzata l'assenza di tale caratteristica. In questo capitolo si vuole studiare l'influenza di tale ipotesi sui risultati ricercati.

Il lavoro ivi svolto è ristretto al caso di un manipolatore planare ad un unico link con dinamometro-base del manipolatore fisso. La flessibilità del giunto è stata schematizzata mediante l'uso di una molla e di uno smorzatore torsionali agenti lungo l'asse \vec{Z}_1 come si può vedere in figura A.1.

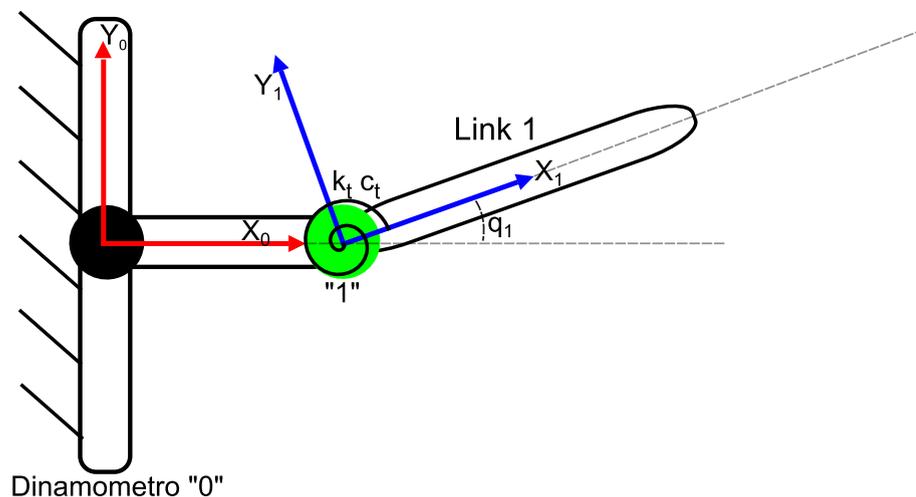


Figura A.1: Modello ad un link con giunto flessibile e dinamometro fisso

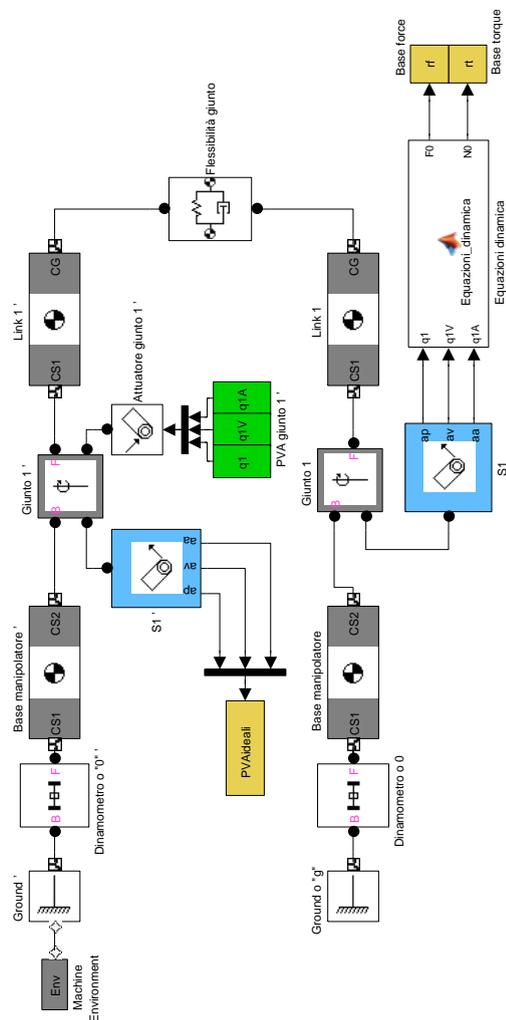


Figura A.2: Schema a blocchi modello ad un link con giunto flessibile e dinamometro fisso

Lo schema a blocchi utilizzato per svolgere la simulazione è quello di figura A.2. L'idea di base di tale schema a blocchi è quella di due robot ognuno con un unico giunto e link. Il primo robot ha la funzione di un robot guida al cui giunto è collegato un attuatore che riceve le informazioni di posizione velocità ed accelerazione angolari. Il baricentro del link del robot guida è collegato al baricentro del link dell'altro robot, robot reale, tramite una molla ed uno smorzatore lineari. Grazie a questo collegamento il moto del robot guida è trasferito al robot reale.

I valori della molla e dello smorzatore torsionale del manipolatore sono noti [3]. I loro valori sono:

$$k_j = 6.207 [N \cdot m / rad], \quad c_j = 1.795 \cdot 10^{-2} [N \cdot m \cdot s / rad]. \quad (A.1)$$

Poiché la molla e lo smorzatore utilizzati nello schema a blocchi sono di tipo lineare e non torsionale i valori appena dati sono stati trasformati. Per fare ciò si sono eguagliati il momento dovuto all'azione di una molla ed uno smorzatore torsionali e quelli dovuti all'azione di una molla ed uno smorzatore lineari.

$$M = k_j \cdot \Delta\theta + c_j \cdot \dot{\theta} = (k \cdot \Delta r + c \cdot \dot{r}) \cdot l_{C_{x_1}},$$

se si considerano piccoli angoli, ipotesi realistica, si ha:

$$\Delta r = \Delta\theta \cdot l_{C_{x_1}}, \quad \dot{r} = \dot{\theta} \cdot l_{C_{x_1}},$$

perciò si può affermare che:

$$k = k_j / (l_{C_{x_1}})^2 = 469.3 [N/m] \quad c = c_j / (l_{C_{x_1}})^2 = 1.357 [N \cdot s/m]. \quad (A.2)$$

Dallo schema a blocchi si nota la presenza di un blocco chiamato **Equazioni dinamica**, blocco *MatLab Function*, al cui interno è stato inserito l'algoritmo di Newton-Eulero per un link con lo scopo di calcolare le forze e le coppie alla base a partire dai dati di posizione, velocità ed accelerazione del giunto del robot reale. Tale operazione è stata fatta poiché nello schema a blocchi non è possibile usare un blocco *Joint Sensor* per conoscere il valore di tali grandezze. Difatti un blocco *Joint Sensor* collegato al blocco **Dinamometro** del robot reale, per come è costruito, presenterebbe l'inconveniente di fornire le forze e le coppie alla base dovuto al moto sinergico dei link dei due manipolatori. Il blocco *MatLab Function* permette invece di utilizzare i valori di posizione, velocità ed accelerazione angolare del giunto del manipolatore reale come valori di input per il calcolo delle grandezze cercate. I valori delle forze e delle coppie alla base sono poi inviate come output nel workspace di MatLab® per essere utilizzati nella procedura di identificazione.

I parametri di base individuati per il modello studiato sono due (tabella A.1).

Tabella A.1: Set parametri di base modello ad un link con flessibilità di giunto

Parametro	Valore
$\beta_1 = I_{zz_1} + m_1 \cdot l_{C_{x_1}}^2$	$= 108.633 \cdot 10^{-4} [kg \cdot m^2]$
$\beta_2 = m_1 \cdot l_{C_{x_1}}$	$= 707.250 \cdot 10^{-4} [kg \cdot m]$

La procedura d'identificazione è identica a quella presentata per le simulazioni dei modelli precedenti: la formula utilizzata è la 4.4 nella quale la dimensione r è 2 e la dimensione n è determinata dal tempo di campionamento. I valori **PVA Giunto 1'** in ingresso all'attuatore del giunto del modello reale sono gli stessi descritti in figura 4.1 relativi al primo giunto.

I δt di campionamento usati per le simulazioni del modello in esame sono:

- $\delta t = 1[s]$, 4 campionamenti;
- $\delta t = 0.1[s]$, 40 campionamenti;
- $\delta t = 0.01[s]$, 400 campionamenti;
- $\delta t = 0.001[s]$, 4000 campionamenti.

Anche per questo modello sono state effettuate procedure di identificazione con l'introduzione nelle misure di posizione, velocità ed accelerazione angolari relative al giunto di un errore percentuale sulla misura massima della relativa grandezza (equazione 5.1).

Le figure A.3, A.4, A.5 e A.6 riassumono gli errori percentuali sui valori degli elementi $\hat{\beta}_1$ e $\hat{\beta}_2$ trovati ai diversi tempi di campionamento e in assenza di errori e per errori percentuali pari a 1,2 e 5 del valore massimo delle grandezze.

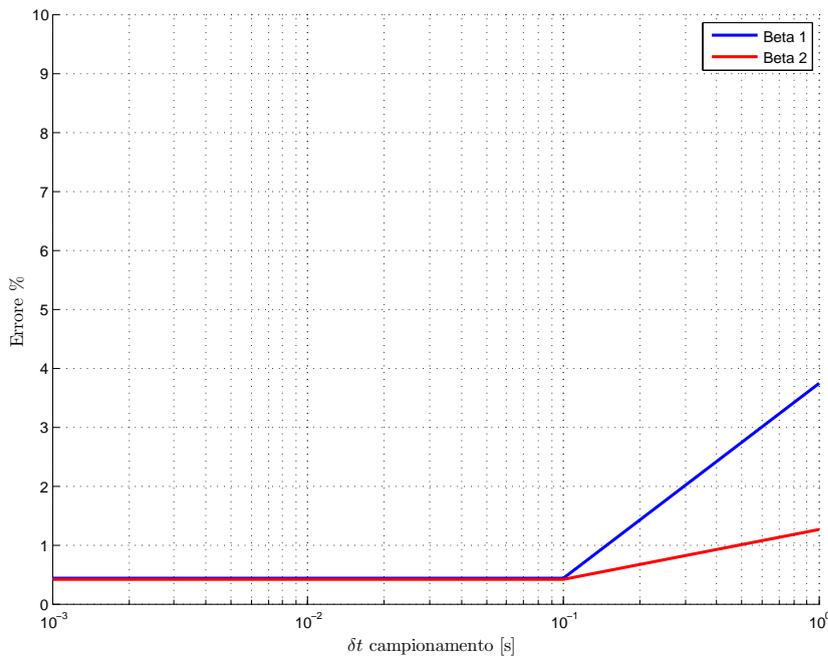


Figura A.3: Dinamometro giunto flessibile

Dall'esame dei risultati ottenuti si evince che la mancata considerazione della flessibilità del giunto del manipolatore per tempi di campionamento sufficientemente bassi ($0.01 [s]$) non comporta un'influenza marcata sui valori ottenuti nella procedura di identificazione. Si può pertanto ritenere che l'ipotesi di giunto rigido è ammissibile per il computo dei valori del set dei parametri di base.

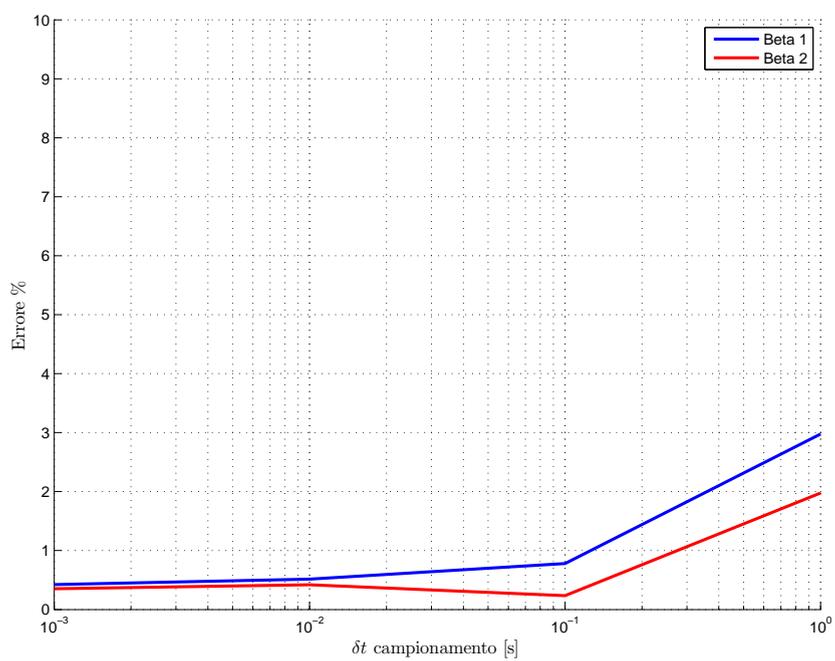


Figura A.4: Dinamometro giunto flessibile, errore $\% \hat{\beta}_i$ con $-1\% \cdot X_{max} < err < 1\% \cdot X_{max}$

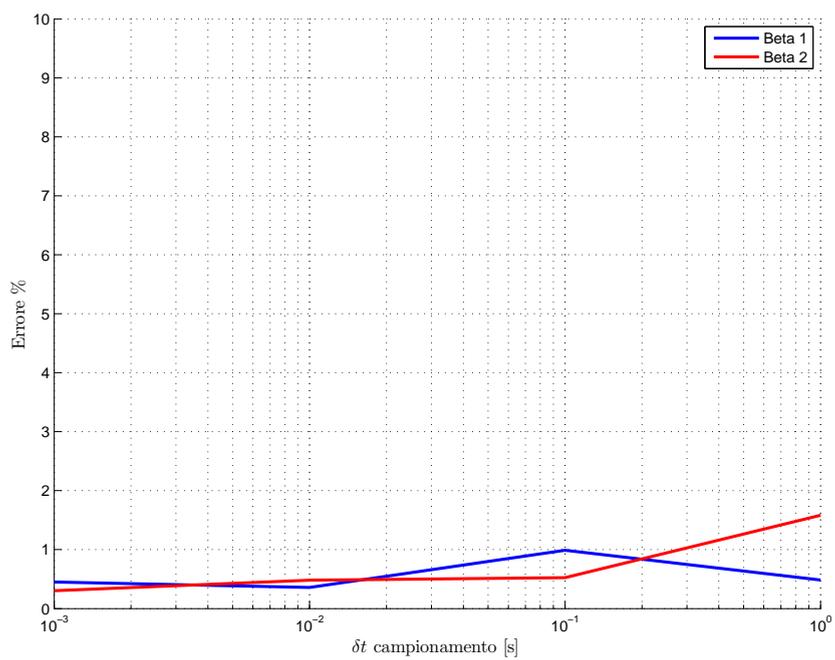


Figura A.5: Dinamometro giunto flessibile, errore $\% \hat{\beta}_i$ con $-2\% \cdot X_{max} < err < 2\% \cdot X_{max}$

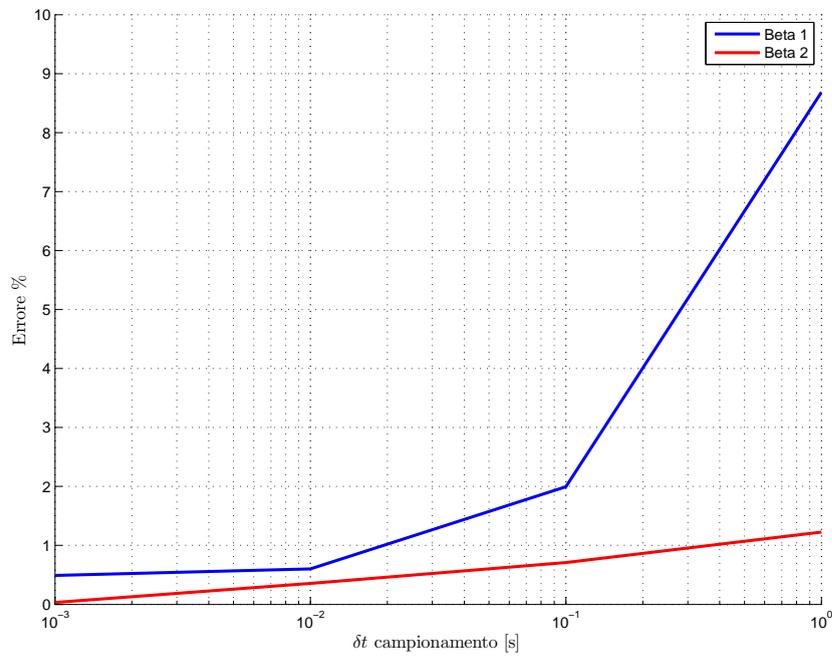


Figura A.6: Dinamometro giunto flessibile, errore % $\hat{\beta}_i$ con $-5\% \cdot X_{max} < err < 5\% \cdot X_{max}$

Appendice B

File MatLab[®]

B.1 Generazione traiettoria

```
function[q1,q1V,q1A,q2,q2V,q2A,q3,q3V,q3A,tf,sample]...
    =Ingresso_modello_3.link

sample=0.0001;

%% q1
tf1=3;
count=tf1/sample+1;
qli=0;
qlf=90*pi/180;

% Posizione
q1=ones(count,2);           % Preallocazione matrice posizioni–tempi
q1(1:count,1)=0:sample:tf1; % Colonna tempi
t=0;
for i=1:count;              % Colonna posizioni
    q1(i,2)=(10*(t/tf1).^3-15*(t/tf1).^4+6*(t/tf1).^5)*(qlf-qli)+qli;
    t=t+sample;
end

% Velocita'
q1V=ones(count,2);         % Preallocazione matrice velocita'–tempi
q1V(1:count,1)=0:sample:tf1; % Colonna tempi
q1V(count,2) = 0;
t=0;
for i=1:count-1;           % Colonna velocita'
    q1V(i,2)=(q1(i+1,2)-q1(i,2))/sample;
    t=t+sample;
end

% Accelerazione
q1A=ones(count,2);         % Preallocazione matrice accelerazioni–tempi
q1A(1:count,1)=0:sample:tf1; % Colonna tempi
q1A(count,2)=0;
t=0;
for i=1:count-1;           % Colonna accelerazioni
```

```

    q1A(i,2)=(q1V(i+1,2)-q1V(i,2))/sample;
    t=t+sample;
end

%% q2
tf2=4;
count=tf2/sample+1;
q2i=0;
q2f=-60*pi/180;

% Posizione
q2=ones(count,2);           % Preallocazione matrice posizioni-tempi
q2(1:count,1)=0:sample:tf2; % Colonna tempi
t=0;
for i=1:count;              % Colonna posizioni
    q2(i,2)=(10*(t/tf2).^3-15*(t/tf2).^4+6*(t/tf2).^5)*(q2f-q2i)+q2i;
    t=t+sample;
end

% Velocita'
q2V=ones(count,2);          % Preallocazione matrice velocita'-tempi
q2V(1:count,1)=0:sample:tf2; % Colonna tempi
q2V(count,2)=0;
t=0;
for i=1:count-1;            % Colonna velocita'
    q2V(i,2)=(q2(i+1,2)-q2(i,2))/sample;
    t=t+sample;
end

% Accelerazione
q2A=ones(count,2);          % Preallocazione matrice accelerazioni-tempi
q2A(1:count,1)=0:sample:tf2; % Colonna tempi
q2A(count,2)=0;
t=0;
for i=1:count-1;            % Colonna accelerazioni
    q2A(i,2)=(q2V(i+1,2)-q2V(i,2))/sample;
    t=t+sample;
end

%% q3
tf3=5;
count=tf3/sample+1;
q3i=0;
q3f=-45*pi/180;

% Posizione
q3=ones(count,2);           % Preallocazione matrice posizioni-tempi
q3(1:count,1)=0:sample:tf3; % Colonna tempi
t=0;
for i=1:count;              % Colonna posizioni
    q3(i,2)=(10*(t/tf3).^3-15*(t/tf3).^4 + 6*(t/tf3).^5)*(q3f-q3i)+q3i;
    t=t+sample;
end

```

```
% Velocita'
q3V=ones(count,2); % Preallocavione matrice velocita'-tempi
q3V(1:count,1)=0:sample:tf3; % Colonna tempi
q3V(count,2)=0;
t=0;
for i=1:count-1; % Colonna velocita'
    q3V(i,2)=(q3(i+1,2)-q3(i,2))/sample;
    t=t+sample;
end
q3V(count,2) = 0; % Ultima derivazione

% Accelerazione
q3A=ones(count,2); % Preallocavione Matrice accelerazioni-tempi
q3A(1:count,1)=0:sample:tf3; % Colonna tempi
q3A(count,2)=0;
t=0;
for i=1:count-1; % Colonna accelerazioni
    q3A(i,2)=(q3V(i+1,2)-q3V(i,2))/sample;
    t=t+sample;
end

tf=[tf1, tf2, tf3];
```

B.2 Funzioni per il calcolo delle equazioni della forza alla base

B.2.1 Equazioni modello dinamometro infinitamente rigido

```

% Dichiarazione variabili simboliche
syms q1 q1V q1A q2 q2V q2A q3 q3V q3A...
      Ixx1 Iyy1 Izz1 Ixx2 Iyy2 Izz2 Ixx3 Iyy3 Izz3...
      m1 m2 m3 l0 b1 l1 b2 l2 b3 l3      real positive;

% Tensori d'inerzia
I1 = [Ixx1 0 0; 0 Iyy1 0; 0 0 Izz1];
I2 = [Ixx2 0 0; 0 Iyy2 0; 0 0 Izz2];
I3 = [Ixx3 0 0; 0 Iyy3 0; 0 0 Izz3];

% Lunghezze relative ai link
bar1 = [b1 0 0];           % distanza baricentrica link 1 lungo x
link1 = [l1 0 0];         % lunghezza link 1 lungo x
bar2 = [b2 0 0];           % distanza baricentrica link 2 lungo x
link2 = [l2 0 0];         % lunghezza link 2 lungo x
bar3 = [b3 0 0];           % distanza baricentrica link 3 lungo x
link3 = [l3 0 0];         % lunghezza link 3 lungo x
rb    = [l0 0 0];         % distanza dal punto di misura (braccio)...
                                % (componente in y = 0)

% Vettori velocita' e accelerazioni dei giunti definiti...
% nel sistema di riferimento del giunto stesso
wV1 = [0 0 q1V];
wA1 = [0 0 q1A];
wV2 = [0 0 q2V];
wA2 = [0 0 q2A];
wV3 = [0 0 q3V];
wA3 = [0 0 q3A];

% Matrice rotazioni
R01 = [cos(q1) sin(q1) 0; -sin(q1) cos(q1) 0; 0 0 1];
R12 = [cos(q2) sin(q2) 0; -sin(q2) cos(q2) 0; 0 0 1];
R23 = [cos(q3) sin(q3) 0; -sin(q3) cos(q3) 0; 0 0 1];

%% Algoritmo Newton-Eulero: Dinamica diretta per robot planare
% 1) Propagazione velocita' angolare ai link successivi
% 2) Propagazione accelerazione angolare ai link successivi
% 3) Accelerazione lineare baricentrica
% 4) Accelerazione lineare del giunto successivo
% 5) Forza inerziale del link
% 6) Coppia inerziale del link

% Accelerazione dinamometro-base manipolatore (bloccato)
A10 = [0 0 0]';

% Link 1
wV11=wV1;
wA11=wA1;

```

B.2. FUNZIONI PER IL CALCOLO DELLE EQUAZIONI DELLA FORZA ALLA BASE45

```
Acl=R01*A10+cross(wA11,bar1)'+cross(wV11,cross(wV11,bar1))';
A21=R01*A10+cross(wA11,link1)'+cross(wV11,cross(wV11,link1))';
Fc1=m1*Ac1;
Nc1=I1*wA11'+cross(wV11,(I1*wV11))';

% Link 2
wV22=(R12*wV11)'+wV2;
wA22=(R12*wA11)'+cross((R12*wV11),wV2)+wA2;
Ac2=R12*A21+cross(wA22,bar2)'+cross(wV22,cross(wV22,bar2))';
A32=R12*A21+cross(wA22,link2)'+cross(wV22,cross(wV22,link2))';
Fc2=m2*Ac2;
Nc2=I2*wA22'+cross(wV22,(I2*wV22))';

% Link 3
wV33=(R23*wV22)'+wV3;
wA33=(R23*wA22)'+cross((R23*wV22),wV3)+wA3;
Ac3=R23*A32+cross(wA33,bar3)'+cross(wV33,cross(wV33,bar3))';
A43=R23*A32+cross(wA33,link3)'+cross(wV33,cross(wV33,link3))';
Fc3=m3*Ac3;
Nc3=I3*wA33'+cross(wV33,(I3*wV33))';

%% Dinamica inversa per robot planare
% 1) Propagazione forze ai giunti
% 2) Propagazione coppie ai giunti

% Link 3
F33=Fc3;
N33=Nc3+cross(bar3,Fc3)';

% Link 2
F22=(R23'*F33)+Fc2;
N22=Nc2+(R23'*N33)+cross(bar2,Fc2)'+cross(link2,(R23'*F33))';

% Link 1
F11=(R12'*F22)+Fc1;
N11=Nc1+(R12'*N22)+cross(bar1,Fc1)'+cross(link1,(R12'*F22))';

%% Forze e coppie alla base
F0=R01'*F11;
N0=R01'*N11+cross(rb,(R01'*F11))';

Fx = simple (F0(1));    Fy = simple (F0(2));    Rt = simple (N0(3));
```

B.2.2 Equazioni modello dinamometro cedevole

```

% Dichiarazione variabili simboliche
syms q0 q0V q0A q1 q1V q1A q2 q2V q2A q3 q3V q3A x xA y yA...
      Ixx0 Iyy0 Izz0 Ixx1 Iyy1 Izz1 Ixx2 Iyy2 Izz2 Ixx3 Iyy3 Izz3...
      m0 m1 m2 m3 b0 l0 b1 l1 b2 l2 b3 l3    real positive;

% Tensori d'inerzia
I0 = [Ixx0 0 0; 0 Iyy0 0; 0 0 Izz0];
I1 = [Ixx1 0 0; 0 Iyy1 0; 0 0 Izz1];
I2 = [Ixx2 0 0; 0 Iyy2 0; 0 0 Izz2];
I3 = [Ixx3 0 0; 0 Iyy3 0; 0 0 Izz3];

% Lunghezze relative ai link
bar0 = [b0 0 0];           % distanza baricentrica braccio lungo x
link0 = [l0 0 0];         % lunghezza braccio lungo x
bar1 = [b1 0 0];           % distanza baricentrica link 1 lungo x
link1 = [l1 0 0];         % lunghezza link 1 lungo x
bar2 = [b2 0 0];           % distanza baricentrica link 2 lungo x
link2 = [l2 0 0];         % lunghezza link 2 lungo x
bar3 = [b3 0 0];           % distanza baricentrica link 3 lungo x
link3 = [l3 0 0];         % lunghezza link 3 lungo x

% Vettori velocita' e accelerazioni dei giunti definiti
% nel sistema di riferimento del giunto stesso
wV0 = [0 0 q0V];
wA0 = [0 0 q0A];
wV1 = [0 0 q1V];
wA1 = [0 0 q1A];
wV2 = [0 0 q2V];
wA2 = [0 0 q2A];
wV3 = [0 0 q3V];
wA3 = [0 0 q3A];

% Matrice rotazioni
RG0 = [cos(q0) sin(q0) 0; -sin(q0) cos(q0) 0; 0 0 1];
R01 = [cos(q1) sin(q1) 0; -sin(q1) cos(q1) 0; 0 0 1];
R12 = [cos(q2) sin(q2) 0; -sin(q2) cos(q2) 0; 0 0 1];
R23 = [cos(q3) sin(q3) 0; -sin(q3) cos(q3) 0; 0 0 1];

%% Algoritmo Newton-Eulero: Dinamica diretta per robot planare
% 1) Propagazione velocita' angolare ai link successivi
% 2) Propagazione accelerazione angolare ai link successivi
% 3) Accelerazione lineare baricentrica
% 4) Accelerazione lineare del giunto successivo
% 5) Forza inerziale del link
% 6) Coppia inerziale del link

% Accelerazione dinamometro-base manipolatore
A0G=[xA yA 0]';

% Link 0
wV00=(RG0*wV0)';
wA00=(RG0*wA0)';

```

B.2. FUNZIONI PER IL CALCOLO DELLE EQUAZIONI DELLA FORZA ALLA BASE47

```
Ac0=RG0*A0G+cross(wA00,bar0)'+cross(wV00,cross(wV00,bar0))';
A10=RG0*A0G+cross(wA00,link0)'+cross(wV00,cross(wV00,link0))';
Fc0=m0*Ac0;
Nc0=I0*wA00'+cross(wV00,(I0*wV00))';

% Link 1
wV11=(R01*wV00)'+wV1;
wA11=(R01*wA00)'+cross((R01 * wV00'),wV1)+wA1;
Ac1=R01*A10+cross(wA11,bar1)'+cross(wV11,cross(wV11,bar1))';
A21=R01*A10+cross(wA11,link1)'+cross(wV11,cross(wV11,link1))';
Fc1=m1*Ac1;
Nc1=I1*wA11'+cross(wV11,(I1*wV11))';

% Link 2
wV22=(R12*wV11)'+wV2;
wA22=(R12*wA11)'+cross((R12*wV11'),wV2)+wA2;
Ac2=R12*A21+cross(wA22,bar2)'+cross(wV22,cross(wV22,bar2))';
A32=R12*A21+cross(wA22,link2)'+cross(wV22,cross(wV22,link2))';
Fc2=m2*Ac2;
Nc2=I2*wA22'+cross(wV22,(I2*wV22))';

% Link 3
wV33=(R23*wV22)'+wV3;
wA33=(R23*wA22)'+cross((R23*wV22'),wV3)+wA3;
Ac3=R23*A32+cross(wA33,bar3)'+cross(wV33,cross(wV33,bar3))';
A43=R23*A32+cross(wA33,link3)'+cross(wV33,cross(wV33,link3))';
Fc3=m3*Ac3;
Nc3=I3*wA33'+cross(wV33,(I3*wV33))';

%% Dinamica inversa per robot planare
% 1) Propagazione forze ai giunti
% 2) Propagazione coppie ai giunti

% Link 3
F33=Fc3;
N33=Nc3+cross(bar3,Fc3)';

% Link 2
F22=(R23'*F33)+Fc2;
N22=Nc2+(R23'*N33)+cross(bar2,Fc2)'+cross(link2,(R23'*F33))';

% Link 1
F11=(R12'*F22)+Fc1;
N11=Nc1+(R12'*N22)+cross(bar1,Fc1)'+cross(link1,(R12'*F22))';

% Link 0
F00=(R01'*F11)+Fc0;
N00=Nc0+(R01'*N11)+cross(bar0,Fc0)'+cross(link0,(R01'*F11))';

%% Forze e coppie alla base
F0G = RG0' * F00;
N0G = RG0' * N00;

Fx = simple (F0G(1));    Fy = simple (F0G(2));    Rt = simple (N0G(3));
```

B.2.3 Equazioni modello ad un link con giunto flessibile

```

% Dichiarazione variabili simboliche
syms q1 q1V q1A...
      Ixx1 Iyy1 Izz1...
      m1 l0 l1 b1 real positive;

% Tensori d'inerzia
I1 = [Ixx1 0 0; 0 Iyy1 0; 0 0 Izz1];

% Lunghezze relative ai link
bar1 = [b1 0 0]; % distanza baricentrica link 1 lungo x
link1 = [l1 0 0]; % lunghezza link 1 lungo x
rb = [l0 0 0]; % distanza dal punto di misura (braccio)...
      % (componente in y = 0)

% Vettori velocita' e accelerazioni dei giunti definiti...
% nel sistema di riferimento del giunto stesso
wV1 = [0 0 q1V];
wA1 = [0 0 q1A];

% Matrice rotazioni
R01 = [cos(q1) sin(q1) 0; -sin(q1) cos(q1) 0; 0 0 1];

%% Algoritmo Newton-Eulero: Dinamica diretta per robot planare
% 1) Propagazione velocita' angolare ai link successivi
% 2) Propagazione accelerazione angolare ai link successivi
% 3) Accelerazione lineare baricentrica
% 4) Accelerazione lineare del giunto successivo
% 5) Forza inerziale del link
% 6) Coppia inerziale del link

% Accelerazione dinamometro-base manipolatore (bloccato)
A10 = [0 0 0]';

% Link 1
wV11=wV1;
wA11=wA1;
Ac1=R01*A10+cross(wA11,bar1)'+cross(wV11,cross(wV11,bar1))';
A21=R01*A10+cross(wA11,link1)'+cross(wV11,cross(wV11,link1))';
Fc1=m1*Ac1;
Nc1=I1*wA11'+cross(wV11,(I1*wV11'))';

%% Dinamica inversa per robot planare
% 1) Propagazione forze ai giunti
% 2) Propagazione coppie ai giunti

% Link 1
F11=Fc1;
N11=Nc1+cross(bar1,Fc1)';

%% Forze e coppie alla base
F0=R01'*F11;

```

B.2. FUNZIONI PER IL CALCOLO DELLE EQUAZIONI DELLA FORZA ALLA BASE49

```
N0=R01'*N11+cross(rb,(R01'*F11))';
```

```
Fx = simple (F0(1));    Fy = simple (F0(2));    Rt = simple (N0(3));
```

B.3 Funzioni per l'identificazione dei parametri di base

B.3.1 Identificazione modello dinamometro infinitamente rigido

```

%% Richiamo funzione traiettoria
[q1, q1V, q1A, q2, q2V, q2A, q3, q3V, q3A, sample, time]=Ingresso;

%% Simulazione
% Valori parametrici dinamici per lo schema a blocchi
m0=0.15;
m1=0.615;
m2=0.615;
m3=0.2;

Ixx0=1.04e-4; Iyy0=1.04e-3; Izz0=1.04e-3;
Ixx1=2.43e-4; Iyy1=2.73e-3; Izz1=2.73e-3;
Ixx2=2.43e-4; Iyy2=2.73e-3; Izz2=2.73e-3;
Ixx3=1.21e-4; Iyy3=1.24e-3; Izz3=1.24e-3;

b0=0.056; l0=0.087;
b1=0.115; l1=0.176;
b2=0.115; l2=0.176;
b3=0.056; l3=0.135;

% Lancio simulazione
Modellodinamometrofisso=sim('Modellodinamometrofisso','StopTime','time');

PVA=Modellodinamometrofisso.find('PVA');
rf=Modellodinamometrofisso.find('rf');
rt=Modellodinamometrofisso.find('rt');

%% Calcolo set parametri di base %%
% Definizione incognite e sostituzione nelle equazioni
% A = (Izz3 + m3 * b3^2) A
% B = (Izz2 + m2 * b2^2) + m3 * l2^2 B
% C = (Izz1 + m1 * b1^2) + (m2 + m3) * l1^2 C
% D = m3 * b3 D
% E = m2 * b2 + m3 * l2 E
% F = m1 * b1 + (m2 + m3) * l1 F

% Dichiarazione variabili simboliche
syms q1 q1V q1A q2 q2V q2A q3 q3V q3A...
      l0 l1 l2...
      A B C D E F...
      real positive;

% Fx
Fx1= D * (-sin(q1+q2+q3) * (q1A+q2A+q3A) - cos(q1+q2+q3) * (q1V+q2V+q3V)^2);
Fx2= E * (-sin(q1+q2) * (q1A+q2A) - cos(q1+q2) * (q1V+q2V)^2);
Fx3= F * (-q1V^2*cos(q1) - q1A*sin(q1));

```

```

% Fy
Fy1= D * (cos (q1+q2+q3) * (q1A+q2A+q3A)-sin (q1+q2+q3) * (q1V+q2V+q3V) ^2);
Fy2= E * (cos (q1+q2) * (q1A+q2A)-sin (q1+q2) * (q1V+q2V) ^2);
Fy3= F * (q1A*cos (q1)-q1V^2*sin (q1));

% Ct
Ctz1= A * (q1A+q2A+q3A) ...
      +B * (q1A+q2A) ...
      +C *q1A;
Ctz2= D * ((10*(cos (q1+q2+q3) * (q1A+q2A+q3A) ...
            -sin (q1+q2+q3) * (q1V+q2V+q3V) ^2)) ...
            +11*(cos (q2+q3) * (2*q1A+q2A+q3A) ...
            -sin (q2+q3) * (2*(q1V*q2V+q1V*q3V+q2V*q3V) + (q2V^2+q3V^2))) ...
            +12*(cos (q3) * (2*q1A+2*q2A+q3A)-sin (q3) *q3V*(2*q1V+2*q2V+q3V)));
Ctz3= E * ((10*(cos (q1+q2) * (q1A+q2A)-sin (q1+q2) * (q1V+q2V) ^2)) ...
            +11*(cos (q2) * (2*q1A+q2A)-sin (q2) *q2V*(2*q1V+q2V)));
Ctz4= F * (10*(q1A*cos (q1)-q1V^2*sin (q1)));

% Equazioni forze e coppie alla base: Fx Fy Ctz
Fx=-(Fx1+Fx2+Fx3);
Fy=-(Fy1+Fy2+Fy3);
Ctz=-(Ctz1+Ctz2+Ctz3+Ctz4);

%% errore casuale (n% * max < e < n% * max) %%
n = 0;

% Definizione dei massimi valori delle posizioni, velocita'...
% ed accelerazioni misurate e delle forze e delle coppie alla base
X1=max(abs(PVA.signals.values), [],1);
maxq1=X1(1); maxq1V=X1(2); maxq1A=X1(3); maxq2=X1(4);
maxq2V=X1(5); maxq2A=X1(6); maxq3=X1(7); maxq3V=X1(8);
maxq3A=X1(9);

X2=max(abs(rf.signals.values), [],1);
maxrfx=X2(1); maxrfy=X2(2);

X3=max(abs(rt.signals.values), [],1);
maxrtz=X3(3);

%% passo in secondi
stepsec=1;

it_sim=time/sample; % dati simulink disponibili
step=stepsec/sample; % passo
num_react=2; % numero reazioni
parameter=6; % parametri da individuare
nrows=((it_sim)/step)*num_react; % numero righe matrice
matrix=ones(nrows,parameter); % preallocazione matrice
reaction=ones(nrows,1); % vettore reazioni
vector=ones((it_sim)/step,1); % vettore punti

i=1;
for l = 2:step:it_sim; % vettore punti per...
    vector(i,l)=1; % il prelevamento dei dati in...

```

```

        i=i+1;                                % uscita dallo schema a blocchi
    end

k=1;
for i=1:1:nrows/2;
    RX=subs(Fx, [q1, q1V, q1A, q2, q2V, q2A, q3, q3V, q3A], ...
[PVA.signals.values(vector(k,1),1)+(rand(1)*n*2-n)/100*maxq1, ...
PVA.signals.values(vector(k,1),2)+(rand(1)*n*2-n)/100*maxq1V, ...
PVA.signals.values(vector(k,1),3)+(rand(1)*n*2-n)/100*maxq1A, ...
PVA.signals.values(vector(k,1),4)+(rand(1)*n*2-n)/100*maxq2, ...
PVA.signals.values(vector(k,1),5)+(rand(1)*n*2-n)/100*maxq2V, ...
PVA.signals.values(vector(k,1),6)+(rand(1)*n*2-n)/100*maxq2A, ...
PVA.signals.values(vector(k,1),7)+(rand(1)*n*2-n)/100*maxq3, ...
PVA.signals.values(vector(k,1),8)+(rand(1)*n*2-n)/100*maxq3V, ...
PVA.signals.values(vector(k,1),9)+(rand(1)*n*2-n)/100*maxq3A]);

matrix(i,1)=subs(RX, [A,B,C,D,E,F], [1,0,0,0,0,0]);
matrix(i,2)=subs(RX, [A,B,C,D,E,F], [0,1,0,0,0,0]);
matrix(i,3)=subs(RX, [A,B,C,D,E,F], [0,0,1,0,0,0]);
matrix(i,4)=subs(RX, [A,B,C,D,E,F], [0,0,0,1,0,0]);
matrix(i,5)=subs(RX, [A,B,C,D,E,F], [0,0,0,0,1,0]);
matrix(i,6)=subs(RX, [A,B,C,D,E,F], [0,0,0,0,0,1]);

reaction(i,1)=rf.signals.values(vector(k,1),1)+(rand(1)*n*2-n)/100*maxrfx;

k=k+1;
end

k=1;
for i=nrows/2+1:1:nrows;
RTZ=subs(Ctz, [q1, q1V, q1A, q2, q2V, q2A, q3, q3V, q3A], ...
[PVA.signals.values(vector(k,1),1)+(rand(1)*n*2-n)/100*maxq1, ...
PVA.signals.values(vector(k,1),2)+(rand(1)*n*2-n)/100*maxq1V, ...
PVA.signals.values(vector(k,1),3)+(rand(1)*n*2-n)/100*maxq1A, ...
PVA.signals.values(vector(k,1),4)+(rand(1)*n*2-n)/100*maxq2, ...
PVA.signals.values(vector(k,1),5)+(rand(1)*n*2-n)/100*maxq2V, ...
PVA.signals.values(vector(k,1),6)+(rand(1)*n*2-n)/100*maxq2A, ...
PVA.signals.values(vector(k,1),7)+(rand(1)*n*2-n)/100*maxq3, ...
PVA.signals.values(vector(k,1),8)+(rand(1)*n*2-n)/100*maxq3V, ...
PVA.signals.values(vector(k,1),9)+(rand(1)*n*2-n)/100*maxq3A, ...
0.087, 0.176, 0.176]);

matrix(i,1)=subs(RTZ, [A,B,C,D,E,F], [1,0,0,0,0,0]);
matrix(i,2)=subs(RTZ, [A,B,C,D,E,F], [0,1,0,0,0,0]);
matrix(i,3)=subs(RTZ, [A,B,C,D,E,F], [0,0,1,0,0,0]);
matrix(i,4)=subs(RTZ, [A,B,C,D,E,F], [0,0,0,1,0,0]);
matrix(i,5)=subs(RTZ, [A,B,C,D,E,F], [0,0,0,0,1,0]);
matrix(i,6)=subs(RTZ, [A,B,C,D,E,F], [0,0,0,0,0,1]);

reaction(i,1)=rt.signals.values(vector(k,1),3)+(rand(1)*n*2-n)/100*maxrtz;

k=k+1;
end

pseudomatrix=pinv(matrix);

```

```
sol=pseudomatrix*reaction;
```

B.3.2 Identificazione modello dinamometro cedevole

```

%% Richiamo funzione traiettoria
[q1, q1V, q1A, q2, q2V, q2A, q3, q3V, q3A, sample, time]=Ingresso;

%% Simulazione
% Valori parametrici dinamici per lo schema a blocchi
m0=0.15;
m1=0.615;
m2=0.615;
m3=0.2;

Ixx0=1.04e-4; Iyy0=1.04e-3; Izz0=1.04e-3;
Ixx1=2.43e-4; Iyy1=2.73e-3; Izz1=2.73e-3;
Ixx2=2.43e-4; Iyy2=2.73e-3; Izz2=2.73e-3;
Ixx3=1.21e-4; Iyy3=1.24e-3; Izz3=1.24e-3;

b0=0.056; l0=0.087;
b1=0.115; l1=0.176;
b2=0.115; l2=0.176;
b3=0.056; l3=0.135;

% Lancio simulazione
Modellodinamometrolabile=sim('Modellodinamometrolabile','StopTime','time');

PVA=Modellodinamometrolabile.find('PVA');
rf=Modellodinamometrolabile.find('rf');
rt=Modellodinamometrolabile.find('rt');

%% Calcolo set parametri di base
% Definizione incognite e sostituzione nelle equazioni
% A   = (Izz3 + m3 * b3^2)                                A
% B   = (Izz2 + m2 * b2^2) + m3 * l2^2                    B
% C   = (Izz1 + m1 * b1^2) + (m2 + m3) * l1^2            C
% D   = (Izz0 + m0 * b0^2) + (m1 + m2 + m3) * l0^2      D
% E   = m3 * b3                                           E
% F   = m2 * b2 + m3 * l2                                  F
% G   = m1 * b1 + (m2 + m3) * l1                          G
% H   = m0 * b0 + (m1 + m2 + m3) * l0                    H
% I   = m0 + m1 + m2 + m3                                 I

% Dichiarazione variabili simboliche
syms q0 q0V q0A q1 q1V q1A q2 q2V q2A q3 q3V q3A x xA y yA...
      l0 l1 l2...
      A B C D E F G H I...
      real positive;

% Fx
Fx1= -E * (sin(q0+q1+q2+q3) * (q0A+q1A+q2A+q3A) ...
        +cos(q0+q1+q2+q3) * (q0V+q1V+q2V+q3V)^2);
Fx2= -F * (sin(q0+q1+q2) * (q0A+q1A+q2A) ...
        +cos(q0+q1+q2) * (q0V+q1V+q2V)^2);
Fx3= -G * (sin(q0+q1) * (q0A+q1A) +cos(q0+q1) * (q0V+q1V)^2);
Fx4= -H * (q0V^2*cos(q0)+q0A*sin(q0));

```

```

Fx5= I *xA;

% Fy
Fy1= E * (sin(q0+q1+q2+q3) * (q0A+q1A+q2A+q3A) ...
        -cos(q0+q1+q2+q3) * (q0V+q1V+q2V+q3V) ^2);
Fy2= F * (cos(q0+q1+q2) * (q0A+q1A+q2A) ...
        -sin(q0+q1+q2) * (q0V+q1V+q2V) ^2);
Fy3= G * (cos(q0+q1) * (q0A+q1A) -sin(q0+q1) * (q0V+q1V) ^2);
Fy4= H * (q0A*cos(q0) -q0V^2*sin(q0));
Fy5= I *yA;

% Rt
Ctz1= A * (q0A+q1A+q2A+q3A) ...
      +B * (q0A+q1A+q2A) ...
      +C * (q0A+q1A) ...
      +D *q0A;
Ctz2= E * (yA*cos(q0+q1+q2+q3) -xA*sin(q0+q1+q2+q3) ...
        -(10*sin(q1+q2+q3) * ( (q1V+q2V+q3V) ^2+2*q0V*(q1V+q2V+q3V) ) ) ...
        +(10*(2*q0A+q1A+q2A+q3A) *cos(q1+q2+q3) ) ...
        -(11*(q2V*(2*q0V+2*q1V+q2V) ...
            +q3V*(2*q0V+2*q1V+2*q2V+q3V) ) *sin(q2+q3) ) ...
        +(11*(2*q0A+2*q1A+q2A+q3A) *cos(q2+q3) ) ...
        -(12*q3V*(2*q0V+2*q1V+2*q2V+q3V) *sin(q3) ) ...
        +(12*(2*q0A+2*q1A+2*q2A+q3A) *cos(q3) ) );
Ctz3= F * (yA*cos(q0+q1+q2) -xA*sin(q0+q1+q2) ...
        -(10*sin(q1+q2) * ( (q1V+q2V) ^2+2*q0V*(q1V+q2V) ) ) ...
        +(10*(2*q0A+q1A+q2A) *cos(q1+q2) ) ...
        -(11*q2V*(2*q0V+2*q1V+q2V) *sin(q2) ) ...
        +(11*(2*q0A+2*q1A+q2A) *cos(q2) ) );
Ctz4= G * (yA*cos(q0+q1) -xA*sin(q0+q1) ...
        -(10*q1V*(2*q0V+q1V) *sin(q1) ) +(10*(2*q0A+q1A) *cos(q1) ) );
Ctz5= H * (yA*cos(q0) -xA*sin(q0) );

% Equazioni forze e coppie alla base: Fx Fy Ctz
Fx=-(Fx1+Fx2+Fx3+Fx4+Fx5);
Fy=-(Fy1+Fy2+Fy3+Fy4+Fy5);
Ctz=Ctz1+Ctz2+Ctz3+Ctz4+Ctz5;

%% errore casuale (n% * max < e < n% * max) %%
n = 0;

% Definizione dei massimi valori delle posizioni, velocita'...
% ed accelerazioni misurate e delle forze e delle coppie di reazione
X1=max(abs(PVA.signals.values), [],1);
maxxA=X1(1); maxyA=X1(2); maxq0=X1(3); maxq0V=X1(4); maxq0A=X1(5);
maxq1=X1(6); maxq1V=X1(7); maxq1A=X1(8); maxq2=X1(9); maxq2V=X1(10);
maxq2A=X1(11); maxq3=X1(12); maxq3V=X1(13); maxq3A=X1(14);

X2 = max(abs(rf.signals.values), [],1);
maxrfx=X2(1); maxrfy=X2(2);

X3=max(abs(rt.signals.values), [],1);
maxrtz=X3(1);

%% passo in secondi

```

```

stepsec=0.5;

it_sim=time/sample;           % dati simulink disponibili
step=stepsec/sample;         % passo
num_react=2;                 % numero reazioni
parameter=9;                 % parametri da individuare
nrows=((it_sim)/step)*num_react; % numero righe matrice
matrix=ones(nrows,parameter); % preallocazione matrice
reaction=ones(nrows,1);     % vettore reazioni
vector=ones((it_sim)/step,1); % vettore punti

i=1;
for l=2:step:it_sim;         % vettore punti per...
    vector(i,l)=1;          % il prelevamento dei dati in...
    i=i+1;                  % uscita dallo schema a blocchi
end

k=1;
for i=1:1:nrows/2;
    RX=subs(Fx, [xA, yA, q0, q0V, q0A, q1, q1V, q1A, q2, q2V, q2A, q3, q3V, q3A], ...
[PVA.signals.values(vector(k,1),1)+(rand(1)*n*2-n)/100*maxxA, ...
PVA.signals.values(vector(k,1),2)+(rand(1)*n*2-n)/100*maxyA, ...
PVA.signals.values(vector(k,1),3)+(rand(1)*n*2-n)/100*maxq0, ...
PVA.signals.values(vector(k,1),4)+(rand(1)*n*2-n)/100*maxq0V, ...
PVA.signals.values(vector(k,1),5)+(rand(1)*n*2-n)/100*maxq0A, ...
PVA.signals.values(vector(k,1),6)+(rand(1)*n*2-n)/100*maxq1, ...
PVA.signals.values(vector(k,1),7)+(rand(1)*n*2-n)/100*maxq1V, ...
PVA.signals.values(vector(k,1),8)+(rand(1)*n*2-n)/100*maxq1A, ...
PVA.signals.values(vector(k,1),9)+(rand(1)*n*2-n)/100*maxq2, ...
PVA.signals.values(vector(k,1),10)+(rand(1)*n*2-n)/100*maxq2V, ...
PVA.signals.values(vector(k,1),11)+(rand(1)*n*2-n)/100*maxq2A, ...
PVA.signals.values(vector(k,1),12)+(rand(1)*n*2-n)/100*maxq3, ...
PVA.signals.values(vector(k,1),13)+(rand(1)*n*2-n)/100*maxq3V...
PVA.signals.values(vector(k,1),14)+(rand(1)*n*2-n)/100*maxq3A]);

matrix(i,1)=subs(RX, [A, B, C, D, E, F, G, H, I], [1, 0, 0, 0, 0, 0, 0, 0, 0]);
matrix(i,2)=subs(RX, [A, B, C, D, E, F, G, H, I], [0, 1, 0, 0, 0, 0, 0, 0, 0]);
matrix(i,3)=subs(RX, [A, B, C, D, E, F, G, H, I], [0, 0, 1, 0, 0, 0, 0, 0, 0]);
matrix(i,4)=subs(RX, [A, B, C, D, E, F, G, H, I], [0, 0, 0, 1, 0, 0, 0, 0, 0]);
matrix(i,5)=subs(RX, [A, B, C, D, E, F, G, H, I], [0, 0, 0, 0, 1, 0, 0, 0, 0]);
matrix(i,6)=subs(RX, [A, B, C, D, E, F, G, H, I], [0, 0, 0, 0, 0, 1, 0, 0, 0]);
matrix(i,7)=subs(RX, [A, B, C, D, E, F, G, H, I], [0, 0, 0, 0, 0, 0, 1, 0, 0]);
matrix(i,8)=subs(RX, [A, B, C, D, E, F, G, H, I], [0, 0, 0, 0, 0, 0, 0, 1, 0]);
matrix(i,9)=subs(RX, [A, B, C, D, E, F, G, H, I], [0, 0, 0, 0, 0, 0, 0, 0, 1]);

reaction(i,1)=rf.signals.values(vector(k,1),1)+(rand(1)*n*2-n)/100*maxrfx;

k=k+1;
end

k=1;
for i=nrows/2+1:1:nrows;
RTZ=subs(Ctz, [xA, yA, q0, q0V, q0A, q1, q1V, q1A, q2, q2V, q2A, q3, q3V, q3A, ...
10, l1, l2], ...
[PVA.signals.values(vector(k,1),1)+(rand(1)*n*2-n)/100*maxxA, ...

```

```

PVA.signals.values(vector(k,1),2)+(rand(1)*n*2-n)/100*maxyA,...
PVA.signals.values(vector(k,1),3)+(rand(1)*n*2-n)/100*maxq0,...
PVA.signals.values(vector(k,1),4)+(rand(1)*n*2-n)/100*maxq0V,...
PVA.signals.values(vector(k,1),5)+(rand(1)*n*2-n)/100*maxq0A,...
PVA.signals.values(vector(k,1),6)+(rand(1)*n*2-n)/100*maxq1,...
PVA.signals.values(vector(k,1),7)+(rand(1)*n*2-n)/100*maxq1V,...
PVA.signals.values(vector(k,1),8)+(rand(1)*n*2-n)/100*maxq1A,...
PVA.signals.values(vector(k,1),9)+(rand(1)*n*2-n)/100*maxq2,...
PVA.signals.values(vector(k,1),10)+(rand(1)*n*2-n)/100*maxq2V,...
PVA.signals.values(vector(k,1),11)+(rand(1)*n*2-n)/100*maxq2A,...
PVA.signals.values(vector(k,1),12)+(rand(1)*n*2-n)/100*maxq3,...
PVA.signals.values(vector(k,1),13)+(rand(1)*n*2-n)/100*maxq3V...
PVA.signals.values(vector(k,1),14)+(rand(1)*n*2-n)/100*maxq3A...
0.087,0.176,0.176]);

matrix(i,1)=subs(RTZ,[A,B,C,D,E,F,G,H,I],[1,0,0,0,0,0,0,0,0]);
matrix(i,2)=subs(RTZ,[A,B,C,D,E,F,G,H,I],[0,1,0,0,0,0,0,0,0]);
matrix(i,3)=subs(RTZ,[A,B,C,D,E,F,G,H,I],[0,0,1,0,0,0,0,0,0]);
matrix(i,4)=subs(RTZ,[A,B,C,D,E,F,G,H,I],[0,0,0,1,0,0,0,0,0]);
matrix(i,5)=subs(RTZ,[A,B,C,D,E,F,G,H,I],[0,0,0,0,1,0,0,0,0]);
matrix(i,6)=subs(RTZ,[A,B,C,D,E,F,G,H,I],[0,0,0,0,0,1,0,0,0]);
matrix(i,7)=subs(RTZ,[A,B,C,D,E,F,G,H,I],[0,0,0,0,0,0,1,0,0]);
matrix(i,8)=subs(RTZ,[A,B,C,D,E,F,G,H,I],[0,0,0,0,0,0,0,1,0]);
matrix(i,9)=subs(RTZ,[A,B,C,D,E,F,G,H,I],[0,0,0,0,0,0,0,0,1]);

reaction(i,1)=rt.signals.values(vector(k,1),1)+(rand(1)*n*2-n)/100*maxrtz;

k=k+1;

end

pseudomatrix=pinv(matrix);
sol=pseudomatrix*reaction;

```

B.3.3 Identificazione modello ad un link con giunto flessibile

```

%% Richiamo funzione traiettoria %%
[q1, q1V, q1A, sample, time]=Ingresso;

%% Simulazione %%
% Valori parametrici dinamici per lo schema a blocchi
m0=0.15;
m1=0.615;

Ixx0=1.04e-4; Iyy0=1.04e-3; Izz0=1.04e-3;
Ixx1=2.43e-4; Iyy1=2.73e-3; Izz1=2.73e-3;

b0=0.056; l0=0.087;
b1=0.115; l1=0.176;

% Lancio simulazione
Modellolgiuntoflex=sim('Modellolgiuntoflex','StopTime','time');

PVAideali=Modellolgiuntoflex.find('PVAideali');
rf=Modellolgiuntoflex.find('rf');
rt=Modellolgiuntoflex.find('rt');

%% Calcolo set parametri di base %%
% Definizione incognite e sostituzione nelle equazioni
% A   =   Izz1 + m1 * b1^2           A
% B   =   m1 * b1                   B

% Dichiarazione variabili simboliche
syms q1 q1V q1A...
      l0 l1...
      A B C D E F...
      real positive;

% Fx
Fx= -B * (cos(q1)*q1V^2+q1A*sin(q1));

% Fy
Fy=  B * (q1A*cos(q1)-q1V^2*sin(q1));

% Ctz
Ctz1= A *q1A;
Ctz2= B * (l0*(q1A*cos(q1)-q1V^2*sin(q1)));

% Dallo schema a blocchi ho come output le forze e le coppie alla base
% Fx e Fy non cambiano, Ctz  somma delle 2 componenti
Ctz=Ctz1+Ctz2;

%% errore casuale   (n% * max < e < n% * max) %%
n = 5;

% Definizione dei massimi valori delle posizioni, velocita'...
% ed accelerazioni misurate e delle forze e delle coppie alla base

```

```

X1=max(abs(PVAideali.signals.values), [], 1);
maxq1=X1(1); maxq1V=X1(2); maxq1A=X1(3);

X2=max(abs(rf.signals.values), [], 1);
maxrfx=X2(1); maxrfy=X2(2);

X3=max(abs(rt.signals.values), [], 1);
maxrtz=X3(3);

%% passo in secondi
stepsec=1;

it_sim=time/sample; % dati simulink disponibili
step=stepsec/sample; % passo
num_react=2; % numero reazioni
parameter=2; % parametri da individuare
nrows=((it_sim)/step)*num_react; % numero righe matrice
matrix=ones(nrows,parameter); % preallocazione matrice
reaction=ones(nrows,1); % vettore reazioni
vector=ones((it_sim)/step,1); % vettore punti

i=1;
for l = 2:step:it_sim; % vettore punti per...
    vector(i,1) = l; % il prelevamento dei dati in...
    i = i+1; % uscita dallo schema a blocchi
end

k=1;
for i=1:1:nrows/2;
    FX=subs(Fx, [q1, q1V, q1A], ...
[PVAideali.signals.values(vector(k,1),1)+(rand(1)*n*2-n)/100*maxq1, ...
PVAideali.signals.values(vector(k,1),2)+(rand(1)*n*2-n)/100*maxq1V, ...
PVAideali.signals.values(vector(k,1),3)+(rand(1)*n*2-n)/100*maxq1A]);

matrix(i,1)=subs(FX, [A,B], [1,0]);
matrix(i,2)=subs(FX, [A,B], [0,1]);

reaction(i,1)=rf.signals.values(vector(k,1),1)+(rand(1)*n*2-n)/100*maxrfx;

k=k+1;
end

k=1;
for i=nrows/2+1:1:nrows;
    RT=subs(Ctz, [q1, q1V, q1A, l0], ...
[PVAideali.signals.values(vector(k,1),1)+(rand(1)*n*2-n)/100*maxq1, ...
PVAideali.signals.values(vector(k,1),2)+(rand(1)*n*2-n)/100*maxq1V, ...
PVAideali.signals.values(vector(k,1),3)+(rand(1)*n*2-n)/100*maxq1A...
0.087]);

matrix(i,1)=subs(RT, [A,B], [1,0]);
matrix(i,2)=subs(RT, [A,B], [0,1]);

reaction(i,1)=rt.signals.values(vector(k,1),3)+(rand(1)*n*2-n)/100*maxrtz;

```

```
k=k+1;  
end
```

```
pseudomatrix=pinv(matrix);  
sol=pseudomatrix*reaction;
```

Bibliografia

- [1] John J. Craig, 2005. **Introduction to Robotics - Mechanics and Control (Third Edition)**. *Pearson Education International*.
- [2] Bruno Siciliano and Oussama Khatib (Eds.) 2008. **Springer Handbook of Robotics**. *Springer*.
- [3] Silvio Cocuzza, Isacco Pretto and Stefano Debei. **Least-Squares-Based Reaction Control of Space Manipulators**. In: *Journal of Guidance, Control and Dynamics, Vol. 35; No. 3, May-June 2012*.
- [4] Eftychios G. Christoforou. **On-line parameter identification and adaptive control of rigid robots using base reaction forces/torques**. In: *2007 IEEE International Conference on Robotics and Automation Roma, Italy 10-14 April 2007*.
- [5] B. Raucent, G. Campion, G. Bastin, J. C. Samin and P.Y. Willems. **Identification of the barycentric parameters of robot manipulators from external measurements**. In: *Automatica, Vol. 28, No.5, pp. 1011-1016, 1992*.
- [6] W. Khalil and E. Dombre. 2004. **Modeling, Identification and Control of Robots**. *Kogan Page Science*.
- [7] C. M. Pham and M. Gautier. **Essential parameters robots**. In: *Proceedings of the 30th Conference on Decision and Control - Brighton, England, December 1991*.