

Capitolo 1

Matlab ed il toolbox di robotica

1.1 Comandi Matlab più utilizzati

I comandi Matlab possono essere eseguiti direttamente in linea digitando il comando seguito da 'Invio' oppure in modo 'script' inserendo più comandi consecutivi in un unico file ed eseguendo tale file, digitandone il nome (nb. un comando seguito dal carattere ';' non produce output sul monitor).

Il primo comando da conoscere è "help" che seguito dal nome di una funzione fornisce informazioni riguardanti il contenuto della funzione stessa e le sue modalità d'uso; l'altro comando molto utile è "demo" che fornisce una panoramica completa delle principali istruzioni dell'ambiente Matlab.

Per quanto affermato precedentemente nell'introduzione, un comando fondamentale in Matlab è quello che definisce un vettore di punti: " $x = [x_i : \Delta : x_f]^n$ ", con tale comando si crea un vettore x che ha come elemento iniziale x_i , come elemento finale x_f ed ha gli elementi equidistanziati di Δ , per un numero totale di elementi pari a $\lfloor (x_f - x_i) / \Delta \rfloor + 1$ (valore che viene calcolato anche da Matlab stesso con il comando "length(x)"). Ovviamente si può anche definire un vettore formato da valori qualsiasi come ad esempio " $w = [0.5 \ 1 \ 4 \ -3 \ 5]^n$ ".

I vettori possono anche essere di tipo colonna e quindi essere così definiti ad esempio: " $c = [0.5; 1; 4; -3; 5]^n$ " oppure come trasposizione di un vettore riga $c = x'$. In modo del tutto analogo si possono definire le matrici, ad esempio scrivere: " $A = [1 \ 3 \ 5; 0 \ 1 \ 0; 7 \ 8 \ 9]^n$ " è equivalente a porre

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 0 & 1 & 0 \\ 7 & 8 & 9 \end{bmatrix}$$

Le operazioni con vettori e matrici sono di due tipi in Matlab: quelle comunemente conosciute dall'algebra (come ad esempio somma '+', prodotto '*', divisione '/', potenza '^', esponenziale 'exp', inversione 'inv', determinante 'det', rango 'rank', etc.) e quelle che operano singolarmente elemento per elemento (che vengono precedute dal carattere '.' e che sono ad esempio: '.', '.*', './', '.*.', './.', etc.). Per comprendere meglio facciamo un esempio, siano A e B le seguenti matrici

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 0 & 1 & 0 \\ 7 & 8 & 9 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 6 & 0 \\ 2 & 3 & 4 \end{bmatrix}$$

allora $A * B$ è il normale prodotto tra matrici (righe per colonne), mentre $A .* B$ è ottenuto moltiplicando elemento per elemento, con i seguenti risultati

$$A * B = \begin{bmatrix} 11 & 33 & 22 \\ 0 & 6 & 0 \\ 25 & 75 & 50 \end{bmatrix}$$

$$A .* B = \begin{bmatrix} 1 & 0 & 10 \\ 0 & 6 & 0 \\ 14 & 24 & 36 \end{bmatrix}$$

Esercitazioni del Corso di Robotica Industriale

Paolo Arcara

DEIS

Facoltà di Ingegneria

Università di Bologna

e-mail: parcara@deis.unibo.it

Anno Accademico 2000 – 2001

che sono normalmente differenti.

I comandi 'eye', 'zeros', 'ones' e 'rand' permettono di definire particolari matrici.

Gli operatori principali che possono essere applicati ad una matrice sono: 'inv' (inversa), 'pinv' (pseudo-inversa), 'det' (determinante), 'cond' (condizionante), 'svd' (decomposizione a valori singolari), 'eig' (autovalori ed autovettori), 'null' (spazio nullo), 'orth' (spazio raggiungibile) e 'rank' (range).

La definizione di una funzione di una o più variabili reali si traduce perciò in una o più operazioni tra vettori e/o matrici. Di seguito elenchiamo qualche esempio di definizione di funzione in Matlab (dove x rappresenta il vettore 'campionato' contenente le ascisse di interesse): $y = x.^2$, $y = x.^3 + (2 * x)$, $y = \exp(x)$, $y = \sin(x)$, $y = \sqrt{x}$.

$$y = x.^2, \quad y = x.^3 + 2x, \quad y = \exp(x), \quad y = \sin(x), \quad y = \sqrt{x}.$$

Il comando 'plot(x,y)' visualizza nel piano cartesiano i punti con ascissa x ed ordinata corrispondente y ; digitando "help plot" si possono vedere i comandi associati alla visualizzazione dei grafici (come per esempio: l'inserimento del titolo, di nomi, del range dei valori graficati, etc.), alcuni tra i più utilizzati sono: 'subplot', 'title', 'xlabel', 'ylabel', 'axis', 'holdfigure', 'log', 'polar' e 'bar'.

Per selezionare parte di una matrice si può sfruttare il carattere ':'; ad esempio per estrarre dalla seconda alla quarta colonna di una matrice si può usare $b = a(:, 2 : 4)$; si può anche costruire una matrice formata da più matrici combinate tra di loro, purché di ordine congruente, ad esempio $m = [a \ b; 0 \ 0 \ 1]$ è un'istruzione valida se a e b sono matrici con lo stesso numero di righe e con un numero complessivo di colonne pari a 4 (perché l'ultima riga è formata da 4 elementi).

In Matlab possono essere definiti degli 'script' e delle funzioni, mentre i primi sono insiemi di comandi inseriti in un unico file di testo (creato con un editor qualsiasi e con estensione *.M) che possono essere eseguiti digitando il nome del file, le seconde, cioè le funzioni, sono insiemi di istruzioni in un po' particolari in quanto ammettono dei valori di ingresso e possono restituire degli altri come risultato finale. Le funzioni sono dei file di tipo *.M che iniziano con l'espressione chiave

function [y1,y2,...] = nome-funzione(x1,x2,...)

dove y sono le uscite ed x gli ingressi. Per eseguire una funzione bisogna digitare dal prompt di Matlab: $[a1, a2, \dots] = \text{nome-funzione}(i_1, i_2, \dots)$ con i_1, i_2, \dots variabili in cui inserire rispettivamente gli ingressi e le uscite della funzione.

1.2 Toolbox di robotica

Il toolbox di robotica è stato realizzato da Peter I. Corke (vedi [6] per ulteriori dettagli), può essere ottenuto in modo gratuito (vedi [7]) e la sua ultima versione (Release 5) è datata aprile 1999.

Il Robotic Toolbox è stato completamente sviluppato in ambiente Matlab ed è stato progettato scoprendo i vari problemi da affrontare (cinematica, dinamica, modello di un manipolatore, etc.) in sottoproblemi, che vengono risolti da funzioni più semplici da realizzare.

Il toolbox non ha la pretesa di essere esaustivo, esso può essere ulteriormente ampliato con la definizione di nuove funzioni che si appoggiano su quelle già esistenti, come vedremo più avanti nel testo.

Le varie funzioni del toolbox di robotica sono elencate nelle Tabelle 1.1-1.8 suddivise per argomenti, in tali tabelle compare una breve descrizione di ognuna delle funzioni presentate¹. Ulteriori informazioni possono essere ottenute attraverso l'help in linea per ognuna delle funzioni oppure consultando [8].

Le rotazioni e le traslazioni possono essere trattate attraverso la matrice di trasformazione omogenea oppure attraverso gli angoli di Eulero o di Roll, Pitch e Yaw.

È possibile definire manipolatori a proprio piacere specificando i parametri di Denavit-Hartenberg (per la cinematica) e le inerzie e gli attriti (per la dinamica)²; nel toolbox sono già presenti, e possono essere utilizzati, i parametri del robot Puma560. Dopo aver definito un manipolatore se ne può calcolare la cinematica e la dinamica, si possono pianificare delle traiettorie e visualizzare l'andamento temporale del robot nello spazio cartesiano.

Il programma dimostrativo "rdemo" è in grado di dare un'idea completa delle potenzialità del toolbox e può rappresentare un ottimo punto di partenza ai fini della comprensione e dell'utilizzo del toolbox medesimo. Ai fini di una migliore padronanza si consiglia poi di leggere il listato delle singole funzioni, a cominciare dalle più semplici.

¹Il termine trasformazione omogenea che compare nelle tabelle fa riferimento alla matrice di trasformazione omogenea associata ad una rototraslazione.

²Una novità della Release 5 del toolbox è dell'utilizzo delle class in ambiente Matlab (disponibili dalla versione 5.3); per ulteriori informazioni su questo argomento digitare "help class".

Rototraslazioni	
eul2tr	dagli angoli di Eulero zyz alla trasformazione omogenea
oa2tr	dai vettori orientamento ed approccio alla trasformazione omogenea
rotvec	dalla rotazione attorno ad un asse alla trasformazione omogenea
rotx	dalla rotazione attorno ad x alla trasformazione omogenea
roty	dalla rotazione attorno ad y alla trasformazione omogenea
rotz	dalla rotazione attorno ad z alla trasformazione omogenea
rpv2tr	da Roll, Pitch e Yaw (Eulero zyz) alla trasformazione omogenea
tr2eul	dalla trasformazione omogenea agli angoli di Eulero
tr2rot	dalla trasformazione omogenea alla sottomatrice di rotazione
tr2rpy	dalla trasformazione omogenea a Roll, Pitch e Yaw
transl	dalla traslazione alla trasformazione omogenea (o viceversa)

Tabella 1.1: Descrizione delle funzioni del toolbox di robotica: Rototraslazioni

Quaternioni	
plot	disegna un quaternione come rotazione 3D
qinterp	interpola tra 2 quaternioni al variare di un parametro
qnorm	normalizza un quaternione
quaternion	definisce un oggetto della classe QUATERNION

Tabella 1.2: Descrizione delle funzioni del toolbox di robotica: Quaternioni

Cinematica	
diff2tr	da valori differenziali alla trasformazione omogenea
fkine	calcola la cinematica diretta per un manipolatore seriale
ikine	calcola la cinematica inversa
ikine560	calcola la cinematica inversa del Puma560
jacob0	calcola lo jacobiano nel sistema di riferimento di base
jacobn	calcola lo jacobiano nel sistema di riferimento dell'end-effector
tr2diff	dalla trasformazione omogenea ai valori differenziali
tr2jac	dalla trasformazione omogenea allo jacobiano associato

Tabella 1.3: Descrizione delle funzioni del toolbox di robotica: Cinematica

Dinamica	
accel	dalle coppie applicate all'accelerazione ai giunti
cinertia	calcola la matrice di inerzia nello spazio operativo
coriolis	calcola le coppie dovute ai termini di coriolis
fdyn	integra di dinamica diretta
friction	calcola le coppie dovute ai termini di attrito
frans	sposta il vettore delle forze/momenti in un altro riferimento
gravload	calcolo del carico gravitazionale sui giunti
inertia	calcola la matrice di inerzia nello spazio dei giunti
itorque	calcola le coppie dovute ai termini di inerzia
rnc	calcolo della dinamica inversa

Tabella 1.4: Descrizione delle funzioni del toolbox di robotica: Dinamica

Modelli di manipolatori	
dl*	matrice contenente i parametri di Denavit-Hartenberg (cinematica)
dyn*	matrice contenente le inerzie e gli attriti (dinamica)
link	definisce un oggetto della classe LINK
linktran	dai parametri cinematici del giunto alla trasformazione omogenea
nofriction	definisce un giunto senza attrito
puma560	definizione dei parametri per il manipolatore Puma560 (1)
puma560akb	definizione dei parametri per il manipolatore Puma560 (2)
robot	definisce un oggetto della classe ROBOT
stanford	definizione dei parametri per il manipolatore Stanford

Tabella 1.5: Descrizione delle funzioni del toolbox di robotica: Modelli di manipolatori (* non rappresentano delle funzioni ma solo dei modelli di riferimento)

Generazione di traiettorie	
ctrj	traiettoria nello spazio cartesiano tra 2 posizioni estreme
drivepar	calcola parametri guida utili per pianificare una traiettoria
jtraj	traiettoria nello spazio dei giunti tra 2 configurazioni estreme
trinterp	interpola tra 2 trasformazioni omogenee al variare di un parametro

Tabella 1.6: Descrizione delle funzioni del toolbox di robotica: Generazione di traiettorie

Funzioni grafiche	
plot	disegna il movimento del robot nello spazio

Tabella 1.7: Descrizione delle funzioni del toolbox di robotica: Funzioni grafiche

Altre funzioni	
maniply	calcola l'indice di manipolabilità
rtdemo	programma dimostrativo delle funzioni del toolbox
ui	utility per variare la prospettiva nei grafici 3D
unit	calcola il vettore associato ad un vettore

Tabella 1.8: Descrizione delle funzioni del toolbox di robotica: Altre funzioni

Capitolo 2

Trasformazione di coordinate

Le trasformazioni di coordinate considerate in robotica sono quelle che servono per descrivere i le posizioni ed i movimenti di un corpo rigido nello spazio euclideo \mathbb{R}^3 . Come noto, tali trasformazioni omogenee vengono utilizzate per la definizione dei modelli cinematici dei manipolatori industriali che presentano due diversi tipi di giunto: giunto prismatico (di traslazione) e giunto rotoideale (di rotazione).

2.1 Rotazione

Dato un vettore $p \in \mathbb{R}^3$, la notazione 0p_1 esprime la posizione di un punto solidale con il corpo rigido 1 visto dal sistema di riferimento 0.

La matrice ${}^0R_1 \in \mathbb{R}^{3 \times 3}$ esprime l'orientamento relativo tra 2 terne cartesiane e viene detta "matrice di rotazione": R serve, tra le altre cose, per passare dal riferimento 0 al riferimento 1 o viceversa. Le colonne della matrice R sono i versori del riferimento 1 espressi nel riferimento 0; vale il discorso duale per le righe della matrice.

Proprietà: $R * R^T = I_3$, $R^{-1} = R^T$, ${}^1R_0 = {}^0R_1^T$.

Esercizio 1 (Rotazioni elementari) Scrivere le matrici di rotazione associate a rotazioni del sistema di riferimento di un angolo θ lungo gli assi x , y e z (suggerimento: si vedano le funzioni "rotx", "roty" e "rotz" del toolbox di robotica).

Altro significato della matrice R : dato un punto 0p_1 con il prodotto $R {}^0p_1$ si calcolano le coordinate del nuovo punto 0p_2 ottenuto dalla rotazione R .

Esercizio 2 (Rotazione di un vettore) Si consideri un vettore v con coordinate $[3, 1, 4]$ rispetto ad un sistema di riferimento (sdr) 0. Determinare le coordinate del vettore ottenuto ruotando v di 45° rispetto a z_0 . Il vettore così ottenuto lo si esprima rispetto ad un sdr 1 ottenuto ruotando il sdr 0 attorno a x_0 di 45° (soluzione: ${}^0v = Rot(z_0, 45^\circ)v$ e ${}^1v = {}^1R_0 {}^0v = Rot(x_0, 45^\circ) {}^0v$, dove il termine Rot esprime la matrice R associata alla rotazione, rispetto all'asse indicato come primo parametro, di un angolo espresso come secondo parametro).

La rotazione di θ attorno ad un asse generico w (vedi fig.2.1) può essere ottenuta attraverso successive rotazioni elementari:

$$Rot(w, \theta) = Rot(z_0, \alpha) Rot(y_0, \beta) Rot(z_0, \theta) Rot(y_0, -\beta) Rot(z_0, -\alpha)$$

dove α è l'angolo compreso tra l'asse x_0 e la proiezione di w sul piano x_0y_0 (utile a portare il vettore w nel piano x_0z_0) e β è l'angolo compreso tra l'asse z_0 ed il vettore w (utile per allineare i vettori v e z_0).

Esercizio 3 (Matrice di rotazione) Si consideri un sistema di riferimento 0 ed un vettore ${}^0v = [0, 0, 0.7071, 0, 0.7071]^T$. Determinare la matrice di rotazione R che esprime una rotazione di 30° attorno a v (soluzione: valgono in questo caso $\alpha = 90^\circ$ e $\beta = 45^\circ$ e quindi $R = Rot(z_0, 90^\circ) Rot(y_0, 45^\circ) Rot(z_0, 30^\circ) Rot(y_0, -45^\circ) Rot(z_0, -90^\circ)$).

Esercizio 4 (Asse di rotazione equivalente) Data una matrice di rotazione R generica, scrivere una funzione Matlab per determinare l'asse di rotazione equivalente w e l'angolo θ (suggerimento: autovettore w associato all'autovalore unitario $Rw = w$).

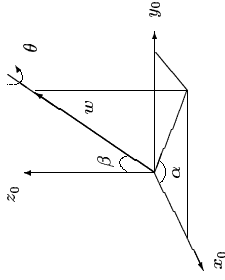


Figura 2.1: Rotazione attorno ad un asse generico w .

2.2 Composizione di rotazioni

La composizione di più rotazioni successive può essere fatta in modo differente a seconda che si operi nel sistema di riferimento di base oppure nel sistema di riferimento corrente; è necessario infatti utilizzare a seconda dei 2 casi delle post-moltiplicazioni oppure delle pre-moltiplicazioni, e siccome il prodotto matriciale non è commutativo si giunge in generale a risultati differenti.

Nel caso che le matrici di rotazione R_i siano espresse tutte nel sistema di riferimento di base (fisso), la composizione delle rotazioni porta a

$${}^0p_n = R_n R_{n-1} \dots R_1 {}^0p,$$

Se invece le singole matrici di rotazione ${}^{i-1}R_i$ sono espresse nel sistema di riferimento corrente, allora si ottiene

$${}^0p = {}^0R_1 {}^1R_2 \dots {}^{n-1}R_n {}^n p.$$

Esercizio 5 (Composizione di rotazioni 1) Dato il sdr 0 (z_0, y_0, z_0) , determinare la matrice di rotazione che permette di passare dal sdr 0 al sdr 1 dove quest'ultimo è ottenuto con una rotazione attorno z_0 di 90° , seguita da una rotazione di -45° attorno ad y_0 (soluzione: ${}^0R_1 = \text{Rot}(y_0, -45^\circ)\text{Rot}(z_0, 90^\circ)$).

Esercizio 6 (Composizione di rotazioni 2) Dato il sdr 0, determinare la matrice di rotazione che permette di passare dal sdr 0 al sdr 1, con 1 ottenuto con una rotazione attorno a z_0 di 90° , seguita da una rotazione di -45° attorno a y_0 e da un'ultima rotazione di 30° attorno a x_1 (soluzione: $R_p = \text{Rot}(y_0, -45^\circ)\text{Rot}(z_0, 90^\circ)$, ${}^0R_1 = R_p \text{Rot}(x_1, 30^\circ)$ e ${}^1R_0 = {}^0R_1$).

2.3 Angoli di Eulero e di Roll, Pitch e Yaw

La matrice R è composta da 9 elementi (e ne servono non meno di 6 per poter ricavare gli altri), in generale però bastano 3 parametri per definire una rotazione.

Gli angoli di Eulero rappresentano 3 rotazioni successive eseguite rispetto ad uno degli assi del sistema di riferimento corrente. Esistono in generale più rappresentazioni degli angoli di Eulero anche se la più comunemente usata è quella con gli assi xyz che porta a

$${}^0R_3 = R_{\text{Euler}}(\phi, \theta, \psi) = \text{Rot}(z_0, \phi)\text{Rot}(y_1, \theta)\text{Rot}(z_3, \psi).$$

che può essere così riscritta¹

$${}^0R_3 = R_{\text{Euler}}(\phi, \theta, \psi) = \begin{bmatrix} C_\phi C_\theta C_\psi - S_\phi S_\psi & -C_\phi C_\theta S_\psi - S_\phi C_\psi & C_\phi S_\theta & C_\phi S_\theta \\ S_\phi C_\theta C_\psi + C_\phi S_\psi & -S_\phi C_\theta S_\psi + C_\phi C_\psi & S_\phi S_\theta & S_\phi S_\theta \\ -S_\theta C_\psi & S_\theta S_\psi & C_\theta & C_\theta \end{bmatrix}.$$

Esercizio 7 (Problema inverso) Data la matrice R di rotazione, scrivere una funzione Matlab per determinare la corrispondente terna di angoli di Eulero xyz composta da ϕ, θ e ψ (suggerimento: si veda la funzione `"tr2eul"`).

Gli angoli di Roll, Pitch e Yaw esprimono invece rotazioni successive rispetto agli assi x, y, z del sistema di riferimento di base

$${}^0R_3 = R_{\text{rollpitchyaw}}(\phi, \theta, \psi) = \text{Rot}(z_0, \phi)\text{Rot}(y_0, \theta)\text{Rot}(x_0, \psi)$$

¹ C ed S abbreviano rispettivamente \cos e \sin .

dove ψ è l'angolo di imbardata (yaw), θ è l'angolo di beccheggio (pitch) e ϕ è l'angolo di rollio (roll); la matrice indicata può essere così riscritta

$${}^0R_3 = R_{\text{rollpitchyaw}}(\phi, \theta, \psi) = \begin{bmatrix} C_\phi C_\theta & -S_\phi C_\psi + C_\phi S_\theta S_\psi & S_\phi S_\psi + C_\phi S_\theta C_\psi \\ S_\phi C_\theta & C_\phi C_\psi + S_\phi S_\theta S_\psi & -C_\phi S_\psi + S_\phi S_\theta C_\psi \\ -S_\theta & C_\phi S_\psi & C_\theta C_\psi \end{bmatrix},$$

taile rappresentazione è del tutto analoga a quella con gli angoli di Eulero zyx . Le funzioni del toolbox di robotica associate sono `"tr2rpy"` e `"rpy2tr"`.

Esercizio 8 (Angoli di Eulero e di Roll, Pitch e Yaw) Si consideri una matrice di rotazione 0R_1 che identifica l'orientamento di un sdr 1 rispetto al sdr base 0. Calcolare le corrispondenti rappresentazioni di Eulero e di Roll, Pitch e Yaw nel caso

$${}^0R_1 = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}$$

(soluzione: Eulero, $\phi, \theta, \psi = (-\pi/2, \pi/2, \pi)$ oppure $(\pi/2, -\pi/2, 0)$ e RPY, $\phi, \theta, \psi = (; -\pi/2, ;)$ infinite soluzioni con $\phi + \psi = \pi/2$).

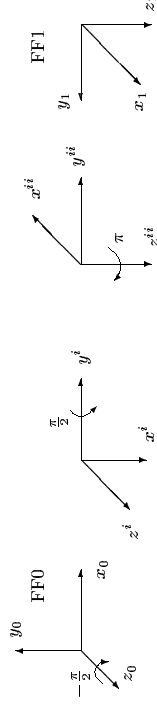


Figura 2.2: Soluzione grafica dell'es.8 (angoli di Eulero - prima soluzione).

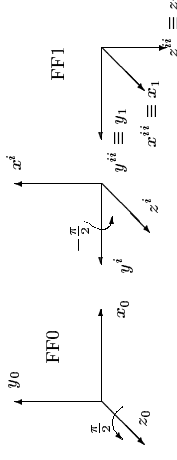


Figura 2.3: Soluzione grafica dell'es.8 (angoli di Eulero - seconda soluzione).

2.4 Traslazione

La traslazione non è un'operazione lineare e non può quindi essere espressa mediante un prodotto matriciale, ma in generale avremo

$${}^0q = {}^0p + {}^0o.$$

2.5 Rototraslazione

La rototraslazione è il moto istantaneo più generale di un corpo rigido: rotazione R attorno ad un asse istantaneo di rotazione e traslazione o lungo un asse

$${}^0p = {}^0R_1 {}^1p + {}^0o_1.$$

Per comodità si può compattare l'espressione precedente introducendo la *matrice di trasformazione omogenea* 0T_1 , così definita

$${}^0T_1 = \begin{bmatrix} {}^0R_1 & \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & 1 \end{bmatrix} = \begin{bmatrix} n_x & s_x & a_x & o_x \\ n_y & s_y & a_y & o_y \\ n_z & s_z & a_z & o_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

e pur di considerare dei nuovi vettori estesi (od omogenei) del tipo $\begin{bmatrix} i p \\ 1 \end{bmatrix}$ si ottiene

$$\begin{bmatrix} {}^0p \\ 1 \end{bmatrix} = {}^0T_1 \begin{bmatrix} {}^1p \\ 1 \end{bmatrix} = \begin{bmatrix} {}^0R_1 & {}^1p + {}^0o_1 \\ & 1 \end{bmatrix}$$

Le rototraslazioni possono a questo punto essere trattate esattamente come le rotazioni attraverso opportuni prodotti matriciali.

Esercizio 9 (Matrice di trasformazione omogenea) Scrivere una funzione Matlab che a partire da un punto 0p , calcoli le coordinate di 1p ottenuto traslando il punto precedente di $[d_x, d_y, d_z]^T$ e ruotandolo di θ attorno all'asse z_0 (suggerimento: si usino le funzioni "transl" e "rotz").

Esercizio 10 (Composizione di rototraslazioni successive) Date 2 matrici di trasformazione omogenea 0T_1 e 1T_2 espresse nel sdr corrente, esprimere la rototraslazione complessiva risultante dalla composizione delle 2 singole rototraslazioni (soluzione: ${}^0T_2 = {}^0T_1 {}^1T_2$).

Inversa di una matrice di trasformazione. Sia data la matrice 0T_1 , definita da

$${}^0T_1 = \begin{bmatrix} R & v \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La matrice inversa ${}^1T_0 = ({}^0T_1)^{-1}$ tale per cui vale ${}^0T_1 {}^1T_0 = I_4$ è data da

$${}^1T_0 = \begin{bmatrix} M & x \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R^T & -R^T v \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Esercizio 11 (Rotazione e traslazione in un sdr) Si determini la matrice di trasformazione omogenea che lega un sdr base 0 con un sdr 1 ottenuto con una rotazione di 45° attorno z_0 seguita da una traslazione di 2 unità lungo z_1 . Determinare inoltre la matrice di trasformazione omogenea inversa 1T_0 (soluzione: $T_{p1} = \text{Rot}(z_0, 45^\circ)$, $T_{p2} = \text{Trans}(0, 0, 2)$, ${}^0T_1 = T_{p1} T_{p2}$ e ${}^1T_0 = {}^0T_1^{-1}$).

Esercizio 12 (Composizione di rototraslazioni) Una stazione di lavoro è dotata di una telecamera la quale inquadra l'origine di un sdr base b, al quale è associato un robot a più giunti; tale telecamera inquadra anche il centro di un oggetto a forma di cubo. Stabiendo un sdr o locale al centro del cubo, questo oggetto, rispetto al sdr c della telecamera (disegnato in fig.2.4), può essere rappresentato da una matrice di trasformazione omogenea cT_b ; inoltre indicando con cT_b la matrice di trasformazione omogenea che esprime posizione e orientamento del sdr base rispetto al sdr della telecamera si ha che

$${}^cT_o = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 10 \\ 0 & 0 & -1 & 9 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^cT_b = \begin{bmatrix} 1 & 0 & 0 & -10 \\ 0 & -1 & 0 & 20 \\ 0 & 0 & -1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Determinare la posizione del centro del cubo rispetto al sistema di coordinate base ovvero la matrice di trasformazione omogenea che permette all'end-effector del robot di posizionarsi sull'oggetto (soluzione: ${}^bT_o = {}^bT_c {}^cT_o = {}^cT_b^{-1} {}^cT_o$).

Esercizio 13 (Disegno di una rototraslazione) Scrivere una funzione Matlab che disegni le 2 terne di un sistema di coordinate, prima e dopo di una rototraslazione con matrice di trasformazione omogenea T.

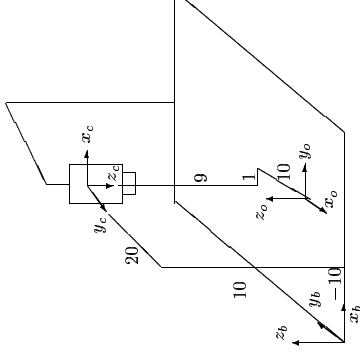


Figura 2.4: I 3 differenti sdr dell'es.12.

2.6 Trasformazione di velocità e forze

Siano v , ω , f ed n rispettivamente i vettori della velocità lineare e di rotazione di un corpo rigido e di forza e coppia applicate ad esso, si definiscono i vettori a 6 componenti $l = \begin{bmatrix} v \\ \omega \\ f \\ n \end{bmatrix}$ (wrench).

La notazione ${}^1l_{01}$ esprime la velocità del punto p_1 rispetto al riferimento 0 espressa nel sistema di riferimento 1.

Sia 0 un sistema di riferimento fisso, mentre siano 1 e 2 solidali con il corpo rigido, le relazioni che legano le velocità di traslazione e rotazione di 1 e 2 rispetto a 0 sono

$$\begin{aligned} {}^0v_{02} &= {}^0v_{01} + {}^0\omega_{01} \times {}^0p_{12} \\ &= {}^0v_{01} + {}^0\omega_{01} \times {}^0R_1 {}^1p_{12} \\ {}^0\omega_{02} &= {}^0\omega_{01} \end{aligned}$$

Manipolando il prodotto vettoriale contenuto nell'espressione precedente si ottiene

$${}^0\omega_{01} \times {}^0R_1 {}^1p_{12} = -{}^0R_1 {}^1p_{12} \times {}^0\omega_{01} = -{}^0p_{12} \times {}^0\omega_{01} = -{}^0P_{12} {}^0\omega_{01}$$

dove se ${}^0p_{12} = [p_x, p_y, p_z]^T$ allora

$${}^0P_{12} = \begin{bmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix}$$

è la matrice emisimmetrica associata², vale infatti

$$Pv = \begin{bmatrix} p_y v_z - p_z v_y \\ p_z v_x - p_x v_z \\ p_x v_y - p_y v_x \end{bmatrix} = p \times v.$$

Si ottiene infine

$$\begin{bmatrix} {}^2v_{02} \\ {}^2\omega_{02} \end{bmatrix} = \begin{bmatrix} {}^2R_1 & -{}^2R_1 {}^1P_{12} \\ 0 & {}^2R_1 \end{bmatrix} \begin{bmatrix} {}^1v_{01} \\ {}^1\omega_{01} \end{bmatrix}$$

cioè

$${}^2l_{02} = {}^2G_1 {}^1l_{01}.$$

Esercizio 14 (Matrice emisimmetrica associata ad un vettore) Scrivere una funzione Matlab che a partire da un vettore p restituisca la matrice emisimmetrica P associata al prodotto vettoriale per tale vettore.

²Matrice che permette di scrivere il prodotto vettoriale tra 2 vettori $(p \times v)$ in forma di prodotto di una matrice per un vettore (Pv) .

Esercizio 15 (Relazione tra velocità in diversi punti di un corpo rigido) Sia

$${}^0T_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 10 \\ 0 & 1 & 0 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

la matrice che esprime posizione/orientamento relativi tra i sdr 0 e 1 associati a 2 punti di uno stesso corpo rigido e sia $t_1 = [-5 \ 0 \ 0 \ 0 \ 1 \ 0]^T$ il twist associato al riferimento 1. Determinare il twist t_0 associato al riferimento 0 (soluzione: $t_0 = [10 \ -5 \ 0 \ 0 \ 0 \ 1]^T$).

Esercizio 16 (Relazione tra velocità in diversi sdr) Sia 0T_1 , uguale all'esercizio precedente, la matrice che esprime la rototraslazione che intercorre tra i sdr 0 e 1 (questa volta associati a punti non appartenenti allo stesso corpo rigido) e sia $t_1 = [-5, 0, 0, 0, 1, 0]^T$ un vettore di twist associato al riferimento 1, determinare il corrispondente vettore di twist t_0 associato al riferimento 0 (soluzione: $t_0 = [0, -5, 0, 0, 0, 1]^T$).

In modo analogo vale per le forze

$$\begin{bmatrix} {}^1f_1 \\ {}^1n_1 \end{bmatrix} = \begin{bmatrix} {}^1R_2 & 0 \\ {}^1P_{12} & {}^1R_2 \end{bmatrix} \begin{bmatrix} {}^2f_2 \\ {}^2n_2 \end{bmatrix}$$

cioè

$${}^1w_1 = {}^2G_1^T {}^2w_2$$

o meglio

$${}^2w_2 = {}^2G_1^{-T} {}^1w_1.$$

Esercizio 17 (Relazione tra forze in diversi sdr) Sia

$${}^0T_1 = \begin{bmatrix} 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 5 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

la matrice che esprime la rototraslazione che intercorre tra i sdr 0 e 1 e sia $w_1 = [0, 10, 0, 2, 0, 0]^T$ il vettore di wrench associato al riferimento 1, determinare il corrispondente vettore di wrench w_0 associato al riferimento 0 (soluzione: $w_0 = [10, 0, 0, 0, 0, -48]^T$).

Esercizio 18 (Conversione di velocità e forze in un corpo rigido) Sia dato un sdr 1 applicato in un punto p_1 di un corpo rigido descritto, rispetto ad un sdr base 0, dalla matrice di trasformazione omogenea 0T_1 . Sia p_2 un altro punto del corpo rigido con associato un sdr 2 descritto, sempre rispetto a 0, da 0T_2 . Si assuma la seguente struttura per 0T_1 e 0T_2

$${}^0T_1 = \begin{bmatrix} -1 & 0 & 0 & -5 \\ 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^0T_2 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & -1 & 0 & -2 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Sia dato il vettore di twist ${}^1t_{01} = [0, 1, -2, 0, 0, 1]^T$ che descrive le velocità lineari e rotazionali del punto p_1 rispetto a 0 espresse nel sdr 1. Determinare il vettore ${}^2t_{02}$. Si consideri ora un vettore di wrench ${}^1w_1 = [1, 0, 1, -2, 0, 0]^T$ che rappresenta forze e coppie agenti su p_1 ed espresse rispetto a 1. Determinare il vettore 2w_2 (soluzione: ${}^2t_{02} = {}^2G_1 {}^1t_{01} = [-5, 2, -1, 0, -1, 0]^T$, dove nella matrice 2G_1 compaiono ${}^2R_1 = {}^2R_0 {}^0R_1 = {}^0R_2^{-1} {}^0R_1$ e ${}^1p_{12} = [-6, -1, -4]^T$, ${}^2w_2 = {}^2G_1^{-T} {}^1w_1 = [0, -1, -1, -2, 1, 1]^T$).

Esercizio 19 (Sensore di forza di un robot) Un robot dotato di un sensore di forza installato sul polso ha afferrato un albrezzo e con questo interagisce con l'ambiente. La situazione è illustrata nella figura 2.6. Tramite il sensore viene misurato il vettore di wrench $w = [-10, 5, 0, -0.75, -1.50, 0.5]^T$. Determinare il vettore delle forze e coppie nel sistema di riferimento posizionato al contatto (soluzione:

$$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.05 \\ 0 & 0 & 1 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ e } {}^eT_s = {}^sT_\alpha^{-1} \text{ e successivamente}$$

$$w_\alpha = {}^aG_s w = [10, 0, 5, 0.25, 0, -0.5]^T).$$

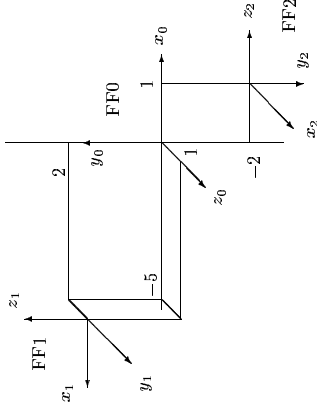


Figura 2.5: Configurazione dei sdr dell'es.18.

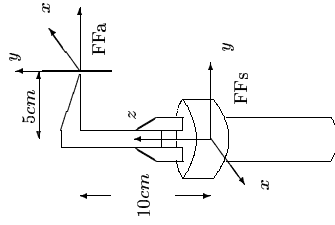


Figura 2.6: Configurazione del sensore di forza dell'es.19.

Esercizio 20 (Calcolo matrici G) Scrivere 3 funzioni Matlab per il calcolo, a partire dalla matrice di trasformazione omogenea T , delle matrici G associate, rispettivamente,

- 1 alla relazione tra i vettori di twist di 2 punti appartenenti allo stesso corpo rigido,
- 2 alla relazione tra i vettori di twist di 2 punti di cui uno è fisso,
- 3 alla relazione tra i vettori di wrench di 2 punti appartenenti allo stesso corpo rigido.

Esercizio 21 (Braccio robotico) Si consideri il braccio robotico disegnato nella figura 2.7. All'estremità del braccio, di lunghezza l , è fissata una pinza in grado di manipolare oggetti; tale pinza mantiene in posizione un oggetto di massa m . Il peso del braccio e quello della pinza possono essere considerati trascurabili rispetto ad m . Il braccio è incernierato alla sua base sull'asse di un motore elettrico che permette il movimento nel piano del foglio. L'angolo θ identifica la posizione del motore, nonché quella del braccio. Oltre al sdr base 0 si supponga di aver fissato un sdr 1 solidale al braccio ed orientato come nella figura ed inoltre sia g l'accelerazione di gravità diretta come indicato nella stessa figura. Determinare la matrice di trasformazione omogenea 0T_1 che descrive posizione ed orientamento del sdr 1 rispetto al sdr 0 in funzione dell'angolo θ . Supponendo inoltre la gravità come unica forza agente sul sistema, determinare l'entità della coppia che deve contrastare il motore per mantenere nella posizione fissata

l'oggetto (soluzione: ${}^0T_1 = \begin{bmatrix} 0 & -S_\theta & C_\theta & lC_\theta \\ 0 & C_\theta & S_\theta & lS_\theta \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ e ${}^1w_1 = [0, -mgC_\theta, -mgS_\theta, 0, 0, 0]^T$ che porta a ${}^0w_0 = {}^0C_1^{-T} {}^1w_1 = [0, -mg, 0, 0, 0, -mg(C_\theta)^T]$ con le prime 3 componenti di reazione vincolare e le altre 3 contenenti la coppia che deve contrastare il motore).

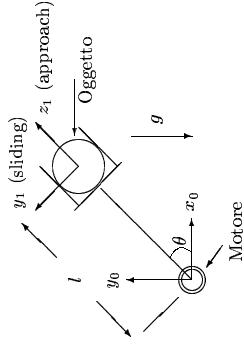


Figura 2.7: Configurazione del braccio del robot dell'es.21.

Capitolo 3

Modello cinematico di un manipolatore

Un manipolatore è formato da una serie di corpi rigidi (link), connessi tra di loro tramite dei giunti prismatici o rotoidali. Ad ogni link è possibile associare un sistema di riferimento e per passare tra i sdr associati ai differenti link che compongono il manipolatore si utilizzano le trasformazioni omogenee.

3.1 Parametri di Denavit-Hartenberg

La notazione di Denavit-Hartenberg viene utilizzata quando si associano i sdr agli $n+1$ link (L_0, \dots, L_n) che compongono il manipolatore¹. I link sono connessi attraverso n giunti (J_1, \dots, J_n) che ne permettono il moto relativo.

I 4 parametri di Denavit-Hartenberg (d, θ, a, α) associati ad ogni giunto descrivono l'insieme di rotazioni e di traslazioni associate al passaggio tra i sdr dei due link collegati dal giunto in considerazione. Per come è definita, questa notazione lascia qualche grado di libertà nel definire i 4 parametri associati ad un giunto.

3.1.1 Convenzioni per la definizione dei parametri

Il giunto J_i collega i due link adiacenti L_{i-1} ed L_i ; come asse z_{i-1} viene scelto l'asse di moto di J_i (z_i sarà quindi allineato con l'asse di moto del giunto successivo); l'origine del sdr i è posta nell'intersezione tra z_i e la normale comune tra z_{i-1} e z_i , se tale normale comune non esiste si pone l'origine sull'asse z_{i-1} (se i due assi si intersecano l'origine è posta nell'intersezione, mentre se i due assi sono coincidenti le origini vengono poste coincidenti); l'asse x_i è diretto lungo il prolungamento della normale comune (qui si introduce qualche grado di libertà perché, come già detto, non sempre tale normale è ben definita); l'asse y_i è scelto in modo da formare una terna destrorsa. Per quanto detto finora, è facile comprendere che non c'è univocità nella scelta del sdr 0 di base (è definito solo z_0 parallelo all'asse di moto J_1) e del sdr n dell'end-effector (è definito solo x_n , perpendicolare a z_{n-1}).

Con le convenzioni appena indicate i 4 parametri di Denavit-Hartenberg associati al passaggio tra due sdr consecutivi ($i-1, i$) si definiscono nel seguente modo:

a_i lunghezza della normale comune tra z_{i-1} e z_i (assi di due giunti consecutivi).

α_i angolo tra z_{i-1} e z_i (con la rotazione che avviene lungo l'asse x_i).

d_i distanza tra l'origine o_{i-1} ed il punto p_i (definito come intersezione tra gli assi z_{i-1} e x_i).

θ_i angolo tra x_{i-1} e x_i (con la rotazione che avviene lungo l'asse z_{i-1}).

3.1.2 Matrice di trasformazione omogenea associata

I parametri a_i ed α_i sono costanti, dipendono dalla geometria del link L_i ed indicano rispettivamente lunghezza ed angolo di rotazione del link. Gli altri due parametri dipendono dal tipo di giunto:

- giunto prismatico: d_i è variabile e θ_i è costante.

¹Il link L_0 rappresenta il link di base del manipolatore.

- giunto rotoidale: θ_i è variabile e d_i è costante.

La matrice di trasformazione omogenea associata ai due sdr $(i-1, i)$ è

$$\begin{aligned}
 {}^{i-1}H_i &= \text{Tras}(z_{i-1}, d_i) \text{Rot}(z_{i-1}, \theta_i) \text{Tras}(x_i, a_i) \text{Rot}(x_i, \alpha_i) \\
 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_{\theta_i} & -S_{\theta_i} & 0 & 0 \\ S_{\theta_i} & C_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_{\alpha_i} & -S_{\alpha_i} & 0 \\ 0 & S_{\alpha_i} & C_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} C_{\theta_i} & -S_{\theta_i} C_{\alpha_i} & S_{\theta_i} S_{\alpha_i} & a_i C_{\theta_i} \\ S_{\theta_i} & C_{\theta_i} C_{\alpha_i} & -C_{\theta_i} S_{\alpha_i} & a_i S_{\theta_i} \\ 0 & S_{\alpha_i} & C_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

e viene detta *trasformazione canonica*.

Ogni matrice ${}^{i-1}H_i$ è funzione della variabile di giunto q_i (che può essere d_i per giunti prismatici oppure θ_i per giunti rotoidali). Complessivamente si può quindi ottenere la matrice di trasformazione omogenea, associata all'intero manipolatore, che trasforma il sdr n nel sdr 0 di base

$${}^0T_n = {}^0H_1(q_1) {}^1H_2(q_2) \dots {}^{n-1}H_n(q_n)$$

questa equazione esprime il *modello cinematico del manipolatore* ed è funzione delle variabili di giunto (q_1, q_2, \dots, q_n) .

Esercizio 22 (Manipolatore planare a tre bracci) Definire attraverso il toolbox di robotica i parametri di Denavit-Hartenberg per il manipolatore planare presentato nella figura 3.1, sapendo che $l_1 = l_2 = l_3 = 10$ (suggerimento: si veda la struttura presente nella funzione "dh").

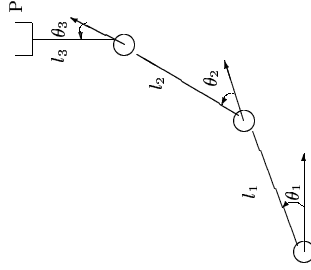


Figura 3.1: Manipolatore planare a tre bracci dell'es.22.

Esercizio 23 (Manipolatore Scara) Definire attraverso il toolbox di robotica i parametri di Denavit-Hartenberg per il manipolatore di tipo Scara presentato nella figura 3.2, sapendo che $l_1 = 2$, $l_2 = 3$ e $d_4 = 4$.

3.2 Cinematica diretta

La cinematica diretta è collegata al modello cinematico del manipolatore e fa riferimento al passaggio dalle variabili di giunto q_i alla configurazione dell'organo terminale del manipolatore espressa nel sdr di base (matrice 0T_n).

Una volta noti i parametri di Denavit-Hartenberg del manipolatore il problema della cinematica diretta è risolto: la matrice 0T_n può essere espressa in funzioni delle matrici ${}^{i-1}H_i$ associate ai singoli giunti e funzioni delle variabili q_i .

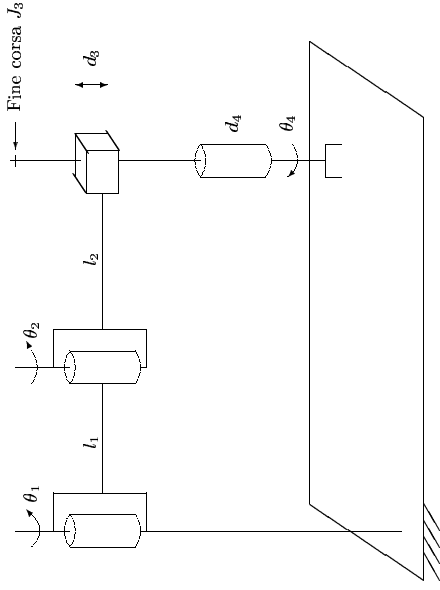


Figura 3.2: Manipolatore Scara dell'es.23.

Esercizio 24 (Calcolo matrice ${}^{i-1}H_i$) Scrivere una funzione Matlab che, a partire dai parametri di Denavit-Hartenberg associati al link di un manipolatore, calcoli la matrice di trasformazione omogenea associata ${}^{i-1}H_i$ (suggerimento: si veda la funzione "linktran").

Esercizio 25 (Manipolatore antropomorfo) Risolvere il problema della cinematica diretta (matrice 0T_n) per il manipolatore antropomorfo presentato in figura 3.3, sapendo che $l_1 = 5$ e $l_2 = 3$ (suggerimento: i parametri di Denavit-Hartenberg non nulli sono $\alpha_1 = \pi/2$, $a_2 = l_1$, $\alpha_3 = l_2$ e le variabili di giunto sono $q_1 = \theta_1$, $q_2 = \theta_2$ e $q_3 = \theta_3$).

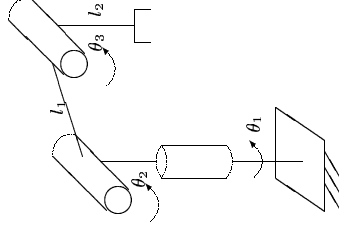


Figura 3.3: Manipolatore antropomorfo dell'es.25.

Esercizio 26 (Manipolatore cilindrico e polso sferico) Risolvere il problema della cinematica diretta (matrice 0T_n) per il manipolatore di Stanford che è composto da un manipolatore cilindrico a 3 gradi di libertà collegato ad un polso sferico; entrambi i componenti sono presentati in figura 3.4, sono noti i parametri $l_1 = 1$ e $l_2 = 3$ (suggerimento: ${}^0T_6 = {}^0T_3 \cdot {}^3T_6$ ed inoltre i parametri di Denavit-Hartenberg non nulli sono $\alpha_1 = -\pi/2$, $\alpha_2 = \pi/2$, $d_2 = l_1$, $\alpha_4 = -\pi/2$, $\alpha_5 = \pi/2$, $d_6 = l_2$ e le variabili di giunto sono $q_1 = \theta_1$, $q_2 = \theta_2$, $q_3 = d_6$, $q_4 = \theta_4$, $q_5 = \theta_5$, $q_6 = \theta_6$).

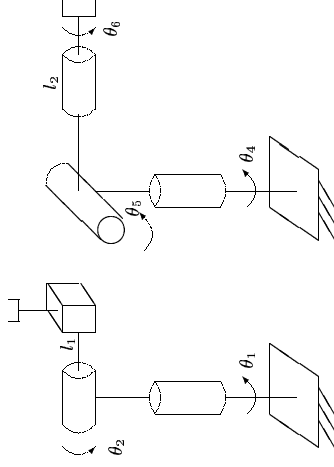


Figura 3.4: Manipolatore cilindrico e polso sferico dell'es.26.

Esercizio 27 (Movimento di manipolatori) Simulare il movimento di uno dei manipolatori analizzati negli esercizi precedenti, definendo i diversi valori delle variabili di giunto nel tempo e disegnando i risultati nello spazio 3D (suggerimento: si veda la funzione "plot" del toolbox di robotica).

3.3 Spazio di lavoro

Lo spazio di lavoro di un manipolatore viene definito come il volume individuato dall'origine del sdr dell'end-effector quando ai giunti del manipolatore si fanno assumere tutte le possibili combinazioni di valori. Esso dipende dalla geometria del manipolatore e dalla variabilità dei singoli giunti.

Esercizio 28 (Disegno dello spazio di lavoro) Scrivere una funzione Matlab che, a partire dai parametri di Denavit-Hartenberg e degli intervalli di variabilità dei singoli giunti di un manipolatore, disegni nello spazio 3D lo spazio di lavoro del manipolatore in esame (suggerimento: una buona approssimazione dello spazio di lavoro si può ottenere fissando tutti i giunti tranne uno ad uno dei loro fine corsa, facendo variare in tutto il suo intervallo ammissibile il giunto libero rimanente e ripetendo tale procedimento per tutte le possibili combinazioni di fine corsa dei diversi giunti).

Esercizio 29 (Spazio di lavoro di alcuni manipolatori) Utilizzare la funzione Matlab dell'esercizio 28 per disegnare lo spazio di lavoro di alcuni manipolatori scelti a proprio piacere.

3.4 Cinematica inversa

Il modello cinematico inverso è quello che consente di passare dalla posizione ed orientamento dell'estremità del manipolatore alle singole variabili di giunto. A differenza della cinematica diretta che ammette sempre un'unica soluzione, il problema della cinematica inversa può avere:

- nessuna soluzione: se la configurazione non appartiene allo spazio di lavoro del manipolatore e non è perciò raggiungibile;
- un numero finito di soluzioni: quando esistono uno o più modi di posizionare i vari link del manipolatore per ottenere la posizione desiderata della parte terminale dello stesso;
- infinite soluzioni: quando esistono infiniti modi di posizionare i vari link per ottenere la posizione desiderata dell'estremità del manipolatore, tipicamente in condizione di singolarità.

Un approccio molto utilizzato per risolvere il problema è quello del "disaccoppiamento cinematico" che consiste nello scomporre la cinematica inversa in due sottoproblemi: 1) trovare la soluzione inversa al problema di posizionamento; 2) trovare la soluzione inversa al problema dell'orientamento dell'estremità del manipolatore.

L'espressione ${}^0T_n = {}^0H_1(q_1) {}^1H_2(q_2) \dots {}^{n-1}H_n(q_n)$ utilizzata per la cinematica diretta fornisce 12 equazioni nelle n incognite q_1, q_2, \dots, q_n che sono le variabili di giunto; a tali equazioni se ne possono aggiungere delle altre ottenute manipolando l'equazione precedente, per esempio per $i < j$ si ottiene:

$$[{}^0H_1(q_1) \dots {}^{i-1}H_i(q_i)]^{-1} {}^0T_n [{}^jH_{j+1}(q_{j+1}) \dots {}^{j-1}H_j(q_j)]$$

e si ricavano altre 12 equazioni per ogni coppia (i, j) . Le più semplici tra le equazioni ottenute possono essere utilizzate per ricavare le n incognite che risolvono il problema della cinematica inversa.

Il problema della cinematica inversa per un manipolatore con polso sferico può essere risolto definendo il posizionamento dell'organo terminale con i primi giunti e l'orientamento con gli ultimi 3 giunti rotoidali che compongono il polso sferico.

Ulteriori considerazioni sulla risoluzione del problema della cinematica inversa verranno fatte dopo aver presentato la cinematica differenziale.

Esercizio 30 (Manipolatore antropomorfo) Risolvere il problema della cinematica inversa per il manipolatore antropomorfo presentato in figura 3.3, sapendo che $l_1 = 5$ e $l_2 = 3$ (suggerimento: l'angolo θ_1 dipende unicamente dai valori (p_x, p_y) della parte terminale del manipolatore; una volta definito θ_1 , a meno di un fattore π , il manipolatore si riconduce ad uno piano con 2 gradi di libertà dati dagli altri 2 giunti; si ottengono per questo caso 4 possibili soluzioni, dette rispettivamente di spalla destra/sinistra e con gomito alto/basso).

Esercizio 31 (Manipolatore cilindrico e polso sferico) Risolvere il problema della cinematica inversa per il manipolatore di Stanford che è composto da un manipolatore cilindrico a 3 gradi di libertà collegato ad un polso sferico; entrambi i componenti sono presentati in figura 3.4, sono noti i parametri $l_1 = 1$ e $l_2 = 3$ (suggerimento: si calcola la posizione del punto di intersezione dei 3 assi del polso sferico $P_p = p - l_2 R_z$, tale punto dipende solo da (q_1, q_2, q_3) , si calcola la matrice ${}^0H_1^{-1} {}^0T_3$ che dipende solo da (q_2, q_3) , si risolve la cinematica inversa per il manipolatore cilindrico, si pone ${}^3R_6 = {}^0P_3^T R$ e si risolve la cinematica inversa del polso sferico per la parte di orientamento).

Esercizio 32 (Robot Puma) Risolvere il problema della cinematica inversa per il robot Puma la cui cinematica diretta è definita nella funzione "Puma560" del toolbox di robotica (suggerimento: si veda la funzione "kin560").

3.5 Cinematica differenziale (Jacobiano)

La cinematica differenziale studia il legame esistente tra le velocità dell'estremità del manipolatore e quelle ai giunti ed il legame esistente tra le forze esercitate sull'ambiente e le coppie applicate ai giunti. Tali relazioni si basano su un operatore lineare detto Jacobiano.

3.5.1 Velocità

La velocità di traslazione è definita come $\dot{p} = \frac{dp}{dt}$, mentre la velocità di rotazione può essere definita in uno dei seguenti modi: 1) si definisce una terna γ che esprime l'orientamento (angoli di Eulero, RPY, etc.) e si utilizza la sua derivata $\dot{\gamma} = \frac{d\gamma}{dt}$; 2) si definisce un vettore di velocità angolare ω la cui direzione coincide con l'asse di istantanea rotazione; a seconda della definizione adottata si perviene allo Jacobiano analitico oppure allo Jacobiano geometrico.

Si può passare da $\dot{\gamma}$ ad ω e viceversa attraverso un opportuna matrice; esistono però velocità di rotazione non esprimibili con il vettore $\dot{\gamma}$ che vengono dette singolarità di rappresentazione di γ . In generale: $\omega = T(\gamma)\dot{\gamma}$; con $T(\cdot)$ che potrebbe essere una matrice singolare, che porta a

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & T(\gamma) \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{\gamma} \end{bmatrix}$$

e quindi

$$J_G = \begin{bmatrix} I & 0 \\ 0 & T(\gamma) \end{bmatrix} J_A$$

dove J_G e J_A sono rispettivamente lo Jacobiano geometrico ed analitico.

Un'espressione analitica dello Jacobiano può essere ottenuta differenziando l'equazione che esprime la cinematica diretta $x = f(q)$, dove nel vettore x sono espressi sia la posizione che l'orientamento secondo

qualche convenzione; da questa si ottiene $dx = \frac{\partial f(q)}{\partial q} dq = J(q) \dot{q}$; lo Jacobiano è quindi funzione non lineare delle variabili di giunto q .

L'espressione geometrica dello Jacobiano può essere ottenuta definendo una velocità di rotazione ω opportuna. Si considerano gli effetti dei singoli giunti del manipolatore sulla velocità complessiva dell'estremità del manipolatore.

Considerando la derivata di una matrice $R(t)$ di rotazione si definisce $S(t) = \dot{R}(t)R^T(t)$ che è della forma

$$S(t) = \begin{bmatrix} 0 & -\omega_x & \omega_y \\ \omega_x & 0 & -\omega_z \\ -\omega_y & \omega_z & 0 \end{bmatrix}$$

e valgono inoltre $\dot{R}(t) = S(t)R(t)$ e $RS(\omega)R^T = S(R\omega)$. La matrice S è una funzione del vettore ω che rappresenta la velocità angolare di $R(t)$.

Si possono definire $v = J_{v1}\dot{q}_1 + J_{v2}\dot{q}_2 + \dots + J_{vn}\dot{q}_n$ e $\omega = J_{\omega1}\dot{q}_1 + J_{\omega2}\dot{q}_2 + \dots + J_{\omega n}\dot{q}_n$, dove per un giunto di rotazione si ha

$$\begin{bmatrix} J_{vi} \\ J_{\omega i} \end{bmatrix} = \begin{bmatrix} {}^0z_{i-1} \times ({}^0p_n - p_{i-1}) \\ 0 \quad z_{i-1} \end{bmatrix}$$

mentre per un giunto prismatico si ottiene

$$\begin{bmatrix} J_{vi} \\ J_{\omega i} \end{bmatrix} = \begin{bmatrix} 0 \quad z_{i-1} \\ 0 \end{bmatrix}$$

dove 0p_n è l'estremità del manipolatore, ${}^0p_{i-1}$ è l'origine del sdr $i - 1$, ${}^0z_{i-1}$ è l'asse z del sdr $i - 1$.

Lo Jacobiano può essere espresso in sistemi di riferimento differenti applicando dei cambi di coordinate.

Esercizio 33 (Calcolo Jacobiano per alcuni manipolatori) Determinare lo Jacobiano geometrico per il manipolatore planare a due gradi di libertà, per quello antropomorfo, per quello cilindrico e per il polso sferico (suggerimento: si veda [1]).

Esercizio 34 (Calcolo Jacobiano) Scrivere una funzione Matlab che, a partire dai parametri di Denavit-Hartenberg di un manipolatore e dalle sue variabili di giunto, calcoli lo Jacobiano geometrico J_C (suggerimento: si vedano le funzioni "jacobn" e "jacob0").

3.5.2 Forze

La relazione tra le forze/coppie può essere determinata facilmente uguagliando i lavori nello spazio operativo ed in quello di giunto: $w^T dx = \tau^T dq$ e ricordandosi che $dx = J(q) \dot{q}$ si ottiene

$$\tau = J^T(q)w$$

che esprime il legame tra la forza applicata dal manipolatore w e le coppie applicate ai giunti τ .

Esercizio 35 (Calcolo coppie ai giunti) Dato lo Jacobiano

$$J = \begin{bmatrix} -a_1 S_1 - a_2 S_{12} & -a_2 S_{12} \\ a_1 C_1 + a_2 C_{12} & a_2 C_{12} \\ 0 & 0 \end{bmatrix}$$

si determini il vettore di coppie da applicare ai giunti per ottenere $f = [10, 0, 0]^T$ (suggerimento: vale l'equazione $\tau = J^T f$; per esempio con $\theta_1 = \theta_2 = 0$ si ottiene $\tau = [0, 0]^T$ che significa che in questa particolare configurazione le coppie applicate ai giunti non hanno influenza nella direzione x dello spazio operativo).

3.5.3 Configurazioni singolari

Le singolarità o configurazioni singolari si hanno quando la matrice che esprime lo Jacobiano perde di rango; in tali configurazioni del manipolatore può accadere che: alcune direzioni di moto non siano realizzabili, velocità limitate dello spazio operativo corrispondono a velocità infinite ai giunti, non si ha una ben definita soluzione al problema cinematico inverso.

3.5.4 Cinematica differenziale inversa

L'equazione della cinematica differenziale inversa può essere ricavata invertendo il legame diretto:

$$\dot{q} = J^T(q)\dot{x}$$

con $J^T = J^T(JJ^T)^{-1}$ oppure $J^T = (J^T J)^{-1} J^T$ a seconda delle dimensioni della matrice.

A partire dalla cinematica differenziale inversa si possono costruire degli algoritmi per l'inversione cinematica. Per esempio si può considerare $q_{k+1} = q_k + J^{-1}(q_k)v_k T$ dove v_k rappresenta la velocità nello spazio operativo e T è il passo di campionamento, tale operazione iterata potrebbe non garantire una convergenza ad un errore nullo; si usano perciò degli schemi in retroazione come nel caso dell'algoritmo con inversa dello Jacobiano (vedi figura 3.5) e di quello con trasposta dello Jacobiano (vedi figura 3.6).

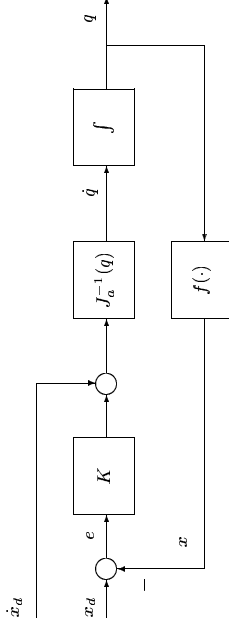


Figura 3.5: Algoritmo con la matrice inversa dello Jacobiano.

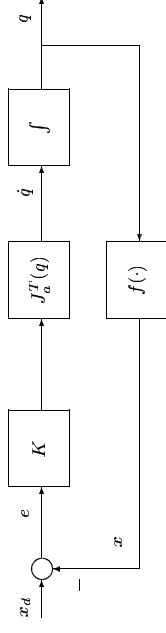


Figura 3.6: Algoritmo con la matrice trasposta dello Jacobiano.

Esercizio 36 (Algoritmo dello Jacobiano inverso) Scrivere una funzione Matlab che implementi l'algoritmo dello Jacobiano inverso illustrato nella figura 3.5 (suggerimento: si veda la funzione "ikine").

3.5.5 Prestazioni dei manipolatori

Si possono confrontare le prestazioni tra manipolatori differenti in termini di velocità e forze. Attraverso lo Jacobiano si può passare dallo spazio di giunto a quello operativo ed osservare le direzioni privilegiate di un manipolatore per quanto riguarda gli spostamenti e gli scambi di forza.

definendo poi $h_{ijk} = \frac{\partial M_{ij}}{\partial q_k} - \frac{1}{2} \frac{\partial M_{ki}}{\partial q_j}$ e $g_i = \frac{\partial P}{\partial q_i}$ si ottiene

$$\sum_{j=1}^n M_{ij}(q) \ddot{q}_j + \sum_{k=1}^n \left(\sum_{j=1}^n h_{ijk}(q) \dot{q}_j \right) \dot{q}_k + g_i(q) = \psi_i.$$

Ponendo in forma matriciale l'equazione precedente si ottiene il seguente *modello dinamico del manipolatore*:

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + g(q) = \psi = \tau + J^T(q) F_o - D \dot{q}.$$

L'espressione precedente può anche essere riscritta, invertendo l'ordine della sommatoria,

$$\sum_{k=1}^n \sum_{j=1}^n \frac{\partial M_{ij}}{\partial q_k} = \frac{1}{2} \sum_{k=1}^n \sum_{j=1}^n \left[\frac{\partial M_{ij}}{\partial q_k} + \frac{\partial M_{ik}}{\partial q_j} \right]$$

e, definendo i *simboli di Christoffel*¹

$$c_{kji} = \frac{1}{2} \left[\frac{\partial M_{ij}}{\partial q_k} + \frac{\partial M_{ik}}{\partial q_j} - \frac{\partial M_{kj}}{\partial q_i} \right],$$

si ottiene una matrice $C(q, \dot{q})$ i cui elementi sono dati da

$$C_{ik} = \sum_{j=1}^n c_{jki} \dot{q}_j$$

e tale per cui la matrice $N(q, \dot{q}) = \dot{M}(q, \dot{q}) - 2C(q, \dot{q})$ è di tipo emisimmetrico.

Le principali proprietà del modello dinamico ottenuto sono le seguenti: la matrice d'inerzia M è simmetrica, definita positiva e limitata inferiormente e superiormente; il termine $C\dot{q}$ è quadratico in \dot{q} e limitato in norma da $v\|\dot{q}\|^2$; il termine di attrito $D\dot{q}$ è limitato da $d\|\dot{q}\|$ ed anche il termine di gravità g è limitato in modulo. Un'ulteriore importante proprietà è quella della linearità del modello nei parametri caratteristici dei link (masse, inerzie, attriti) $Y(q, \dot{q}, \ddot{q})\alpha = \tau$, dove α rappresenta il vettore dei parametri.

Esercizio 37 (Dinamica del manipolatore Scara) Scrivere le equazioni dinamiche associate al manipolatore di tipo Scara presentato nella figura 3.2 utilizzando la formulazione di Eulero-Lagrange (suggerimento: le energie cinetiche associate ai 4 link valgono $K_1 = \frac{1}{2} m_1 \omega_0^2 \dot{q}_1^2 + \frac{1}{2} I_1 \dot{q}_1^2$, $K_2 = \frac{1}{2} m_2 (\alpha_1^2 \dot{q}_1^2 + \dot{q}_2^2 + 2\alpha_1 \alpha_2 \cos(q_2)(\dot{q}_1^2 + \dot{q}_2^2)) + \frac{1}{2} I_2 (\dot{q}_1^2 + \dot{q}_2^2)$, $K_3 = \frac{1}{2} m_3 (\alpha_1^2 \dot{q}_1^2 + \alpha_2^2 (\dot{q}_1 + \dot{q}_2)^2 + 2\alpha_1 \alpha_2 \cos(q_2)(\dot{q}_1^2 + \dot{q}_2^2) + \dot{q}_3^2) + \frac{1}{2} I_3 (\dot{q}_1 + \dot{q}_2)^2$ e $K_4 = \frac{1}{2} m_4 (\alpha_1^2 \dot{q}_1^2 + \alpha_2^2 (\dot{q}_1 + \dot{q}_2)^2 + 2\alpha_1 \alpha_2 \cos(q_2)(\dot{q}_1^2 + \dot{q}_2^2) + \dot{q}_4^2) + \frac{1}{2} I_4 (\dot{q}_1 + \dot{q}_2 - \dot{q}_4)^2$, mentre le energie potenziali valgono $P_1 = 0$, $P_2 = 0$, $P_3 = m_3 g(-l_3)$ e $P_4 = m_4 g(-l_3 - d_4)$).

4.2 Formulazione di Newton-Eulero

Il modello dinamico di Newton-Eulero si basa sul bilancio delle forze e delle coppie agenti sui singoli link del manipolatore, secondo i due fondamentali *teoremi della quantità di moto* e del *momento della quantità di moto*

$$f = \frac{d(mv)}{dt} \\ n = -\frac{d(I\omega)}{dt}$$

che esprimono le variazioni delle sopracitate grandezze in funzione delle forze e delle coppie applicate.

Scrivendo tali equazioni di bilancio per il generico link i -esimo si ottengono:

$$f_i - f_{i+1} + m_i g = m_i \ddot{p}_i C_i \\ v_i + f_i \times r_{i-1, C_i} - v_{i+1} - f_{i+1} \times r_i C_i = \frac{d}{dt} (I_i \omega_i)$$

dove f e v esprimono le forze e le coppie scambiate con i link adiacenti a quello in considerazione, C_i rappresenta il baricentro del link, m_i e I_i sono rispettivamente la massa ed il tensore d'inerzia del link.

¹Vale la relazione $c_{kji} = c_{jki}$ a motivo della simmetria di M .

Capitolo 4

Modello dinamico di un manipolatore

Il modello dinamico di un manipolatore può essere scritto seguendo due approcci differenti, quello di *Eulero-Lagrange* e quello di *Newton-Eulero*, il primo sfrutta l'energia cinetica e potenziale del sistema, il secondo si basa sulle forze e coppie agenti sul generico link del manipolatore.

4.1 Formulazione di Eulero-Lagrange

Il modello dinamico Lagrangiano di un generico sistema materiale può essere espresso con l'equazione

$$\psi_i = \frac{d}{dt} \left(\frac{\partial K}{\partial \dot{q}_i} \right) - \frac{\partial K}{\partial q_i} + \frac{\partial P}{\partial q_i}$$

dove K , P e ψ_i rappresentano rispettivamente l'energia cinetica totale, l'energia potenziale complessiva e le altre forze, applicate al singolo giunto, che non sono espresse attraverso l'energia potenziale

$$\psi_i = \tau_i + [J^T F_{\omega}]_i - d_i \dot{q}_i$$

comprendenti le forze/coppie applicate dagli attuatori, le forze applicate dall'ambiente esterno ed eventuali forze dovute ad attrito viscoso.

L'energia cinetica può essere espressa come

$$K = \sum_{i=1}^n K_i = \sum_{i=1}^n \frac{1}{2} m_i v_i^T v_i + \frac{1}{2} \omega_i^T R_i I_i R_i^T \omega_i$$

dove, considerando il link i -esimo, m_i , v_i , ω_i , R e I rappresentano rispettivamente la massa, la velocità del baricentro, la velocità angolare di rotazione, la matrice di trasformazione rispetto al sistema di coordinate di base e la matrice d'inerzia rispetto ad un sistema di riferimento baricentrico. Ricordando che le velocità nello spazio operativo può essere espressa in funzione delle variabili di giunto si ottiene una nuova espressione per l'energia cinetica:

$$K = \frac{1}{2} \dot{q}^T \sum_{i=1}^n \left[m_i J_{v_i}^T(q) J_{v_i}^T(q) + J_{\omega_i}^T(q) R_i I_i R_i^T J_{\omega_i}^T(q) \right] \dot{q} = \frac{1}{2} \dot{q}^T M(q) \dot{q} = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n M_{ij}(q) \dot{q}_i \dot{q}_j$$

con la matrice $M(q)$ che è detta *matrice d'inerzia*.

L'energia potenziale dipende di solito dalla sola forza di gravità e vale

$$P = \sum_{i=1}^n g_i^T p_i m_i$$

dove g_i , p_i ed m_i sono rispettivamente il vettore forza di gravità, la posizione del baricentro e la massa del link i -esimo.

Le equazioni di Lagrange assumono quindi la seguente struttura:

$$\sum_{j=1}^n M_{ij} \ddot{q}_j + \sum_{k=1}^n \sum_{j=1}^n \left[\frac{\partial M_{ij}}{\partial q_k} - \frac{1}{2} \frac{\partial M_{kj}}{\partial q_i} \right] \dot{q}_k \dot{q}_j + \frac{\partial P}{\partial q_i} = \psi_i,$$

Vale inoltre che il termine legato al tensore d'inerzia può essere così riscritto

$$\frac{d}{dt}(I_i \omega_i) = I_i \dot{\omega}_i + \omega_i \times (I_i \omega_i)$$

mentre le forze generalizzate applicate dal link $i - 1$ al giunto i valgono²

$$\tau_i = \begin{cases} v_i^T z_{i-1} + \left(F_{vi} \dot{\theta}_i + f_{si} \right) & \text{per giunto rotoidale} \\ f_i^T z_{i-1} + \left(F_{vi} \dot{d}_i + f_{si} \right) & \text{per giunto prismatico} \end{cases}$$

e le espressioni legate alle accelerazioni del link possono essere ricavate dalle corrispondenti espressioni delle velocità

$$\dot{p}_i = \begin{cases} \dot{p}_{i-1} + z_{i-1} \dot{d} + 2\omega_i \times z_{i-1} \dot{d} + \dot{\omega}_i \times r_{i-1,i} + \omega_i \times (r_{i-1,i} + \omega_i \times r_{i-1,i}) & \text{per giunto rotoidale} \\ \dot{p}_{i-1} + z_{i-1} \dot{\theta}_i + \omega_{i-1} \times z_{i-1} \dot{\theta}_i & \text{per giunto prismatico} \end{cases}$$

$$\dot{\omega}_i = \begin{cases} \dot{\omega}_{i-1} + z_{i-1} \dot{\theta}_i & \text{per giunto rotoidale} \\ \dot{\omega}_{i-1} & \text{per giunto prismatico} \end{cases}$$

con l'accelerazione del baricentro che vale

$$\ddot{p}_{C_i} = \ddot{p}_i + \dot{\omega}_i \times r_{i,C_i} + \omega_i \times (\omega_i \times r_{i,C_i}).$$

Per determinare il moto complessivo del manipolatore è necessario un algoritmo ricorsivo che propaghi in avanti tra i vari link le velocità e le accelerazioni, all'indietro le forze e le coppie scambiate tra i link e calcoli successivamente le forze generalizzate ai singoli giunti.

Esercizio 38 (Dinamica del manipolatore Scara) Scrivere le equazioni dinamiche associate al manipolatore di tipo Scara presentato nella figura 3.2 utilizzando la formulazione di Newton-Eulero (suggerimento: siano $\dot{p}_0 = 0$, $\omega_0 = \dot{\omega}_0 = 0$, $f_5 = 0$ e $\nu_5 = 0$ le condizioni al contorno, dalla ricorsione in avanti si ottengono $\omega_{1,z} = \dot{q}_1$, $\omega_{2,z} = -(\dot{q}_1 + \dot{q}_2)$, $\omega_{3,z} = -(\dot{q}_1 + \dot{q}_2)$, $\omega_{4,z} = \dot{q}_4 - (\dot{q}_1 + \dot{q}_2)$, $\dot{p}_1 = [-a_1 \dot{q}_1^2 \ a_1 \dot{q}_1 \ 0]^T$, $\dot{p}_{C1} = [-a_{C1} \dot{q}_1^2 \ a_{C1} \dot{q}_1 \ 0]^T$, $\dot{p}_2 = [-a_1 \cos(\dot{q}_2) \dot{q}_1^2 + a_1 \sin(\dot{q}_2) \dot{q}_1 - a_2 \dot{q}_1 - a_1 \cos(\dot{q}_2) \dot{q}_1 - a_2 \dot{q}_1 - a_2 \dot{q}_1 + \dot{q}_2] \ 0]^T$, $\dot{p}_{C2} = [-a_1 \cos(\dot{q}_2) \dot{q}_1^2 + a_1 \sin(\dot{q}_2) \dot{q}_1 - a_{C2}(\dot{q}_1 + \dot{q}_2)^2 - a_1 \sin(\dot{q}_2) \dot{q}_1^2 - a_1 \cos(\dot{q}_2) \dot{q}_1 - a_{C2}(\dot{q}_1 + \dot{q}_2) \ 0]^T$, $\dot{p}_3 = \dot{p}_{C3} = [-a_1 \cos(\dot{q}_2) \dot{q}_1^2 + a_1 \sin(\dot{q}_2) \dot{q}_1 - a_2(\dot{q}_1 + \dot{q}_2)^2 - a_1 \sin(\dot{q}_2) \dot{q}_1^2 - a_1 \cos(\dot{q}_2) \dot{q}_1 - a_2(\dot{q}_2) \dot{q}_1 - a_2(\dot{q}_1 + \dot{q}_2) \ 0]^T$, $\dot{p}_4 = \dot{p}_{C4} = [-a_1 \cos(\dot{q}_2) \cos(\dot{q}_4) \dot{q}_1^2 + a_1 \sin(\dot{q}_2) \cos(\dot{q}_4) \dot{q}_1 - a_2 \cos(\dot{q}_4) (\dot{q}_1 + \dot{q}_2)^2 - a_1 \sin(\dot{q}_2) \dot{q}_1^2 - a_1 \cos(\dot{q}_2) \sin(\dot{q}_4) \dot{q}_1 - a_2 \cos(\dot{q}_2) \sin(\dot{q}_4) \dot{q}_1 - a_2 \sin(\dot{q}_4) (\dot{q}_1 + \dot{q}_2)^2 - a_1 \sin(\dot{q}_2) \cos(\dot{q}_4) \dot{q}_1^2 - a_1 \cos(\dot{q}_2) \cos(\dot{q}_4) (\dot{q}_1 + \dot{q}_2) \ \dot{q}_4]^T$ e da esse si può procedere alla ricorsione all'indietro determinando le equazioni dinamiche per i 4 link del manipolatore).

4.3 Modello dinamico nello spazio di lavoro

Il modello dinamico nello spazio operativo può essere calcolato a partire da quello ottenuto nello spazio di giunto, vale infatti la relazione $\dot{x} = J(q)\dot{q}$ da cui si ricava che $\ddot{x} = J(q)\ddot{q} + \dot{J}(q)\dot{q}$; sostituendo l'espressione di \dot{q} si ottiene

$$\ddot{x} = JM^{-1}(-C\dot{q} - g + \tau + J^T F_a) + \dot{J}\dot{q}$$

e considerando che $\tau = J^T F$ e $\dot{q} = J^{-1} \dot{x}$ si ottiene³ infine

$$\ddot{x} = (-JM^{-1}CJ^{-1} + \dot{J}J^{-1})\dot{x} - JM^{-1}g + JM^{-1}J^T(F + F_a).$$

Ponendo $\hat{M} = J^{-T}MJ^{-1}$, $\hat{C} = J^{-T}CJ^{-1} - \dot{M}J^{-1}$ e $\hat{g} = J^{-T}g$ si ottiene il modello complessivo

$$\hat{M}\ddot{x} + \hat{C}(x, \dot{x})\dot{x} + \hat{g}(x) = F + F_a.$$

Capitolo 5

Pianificazione delle traiettorie

Pianificare una traiettoria significa generare il movimento che si desidera che il manipolatore compia con il suo end-effector, tale movimento si compone di un cammino geometrico e di una legge oraria per percorrere tale percorso in un determinato tempo.

Il percorso può essere definito nello spazio operativo o nello spazio di giunto ed è di solito espresso in forma parametrica, la legge oraria viene definita facendo variare nel tempo il parametro che descrive il cammino da seguire.

5.1 Traiettorie nello spazio di giunto

Consideriamo per semplicità in questa parte il movimento di un singolo giunto.

5.1.1 Traiettorie polinomiali

Di una traiettoria si possono definire posizione e velocità, iniziale e finale (q_i , \dot{q}_i , q_f e \dot{q}_f), che forniscono quattro condizioni al contorno da rispettare; ciò può essere fatto con un polinomio almeno del terzo ordine, che abbia cioè 4 parametri,

$$q(t) = a_0 + a_1(t - t_i) + a_2(t - t_i)^2 + a_3(t - t_i)^3$$

con condizioni al contorno

$$\begin{aligned} q(t_i) &= a_0 &= q_i \\ \dot{q}(t_i) &= a_1 &= \dot{q}_i \\ q(t_f) &= a_0 + a_1(t_f - t_i) + a_2(t_f - t_i)^2 + a_3(t_f - t_i)^3 &= q_f \\ \dot{q}(t_f) &= a_1 + 2a_2(t_f - t_i) + 3a_3(t_f - t_i)^2 &= \dot{q}_f \end{aligned}$$

che conducono a

$$\begin{aligned} a_0 &= q_i \\ a_1 &= \dot{q}_i \\ a_2 &= \frac{-3(q_f - q_i) - (2\dot{q}_i + \dot{q}_f)(t_f - t_i)}{(t_f - t_i)^2} \\ a_3 &= \frac{2(q_f - q_i) + (\dot{q}_i + \dot{q}_f)(t_f - t_i)}{(t_f - t_i)^3} \end{aligned}$$

In questo modo si può suddividere la traiettoria in più tratti che hanno punti iniziali e finali che coincidono a coppie tra di loro, con la definizione di istanti di tempo e velocità iniziali e finali per ogni singolo tratto.

Se si vuole fissare anche l'accelerazione iniziale e finale è necessario ricorrere ad un polinomio del quinto ordine.

Esercizio 39 (Traiettoria polinomiale del quinto ordine) Determinare i coefficienti del polinomio del quinto ordine contenente la traiettoria $q(t)$ che soddisfa le seguenti condizioni al contorno: $t_i = 1$, $t_f = 3$, $q_i = 10$, $q_f = -5$, $\dot{q}_i = 1$, $\dot{q}_f = -1$, $\ddot{q}_i = 0$ e $\ddot{q}_f = 0$.

² I termini tra parentesi nell'espressione indicano eventuale presenza di attrito viscoso o statico al giunto.

³ Bisogna notare che nell'espressione $\dot{x} = J(q)\dot{q}$ si utilizza lo Jacobiano analitico J_A , mentre in $\tau = J^T F$ si utilizza quello geometrico $J = T_A(\tau)J_A$.

5.1.2 Traiettorie lineari con giunzioni paraboliche

Le traiettorie tipicamente progettate nello spazio di giunto sono formate da segmenti lineari uniti da raccordi parabolici, nel tratto lineare la velocità è costante nel tempo, mentre nel tratto parabolico è l'accelerazione ad essere costante e la velocità varia linearmente. Si ottengono così degli andamenti trapezoidali per la velocità, con accelerazione costante nel tratto iniziale, accelerazione nulla nel tratto intermedio e decelerazione costante nel tratto finale.

Solitamente la durata della fase di accelerazione (t_a) è uguale a quella della fase di decelerazione (t_d), mentre i vincoli sono la posizione e la velocità iniziale e finale del manipolatore.

Esistono dei vincoli di realizzabilità di una traiettoria di questo tipo, infatti la durata complessiva delle due fasi di accelerazione e di decelerazione ($t_a + t_d$) non deve essere superiore all'intervallo temporale complessivo ($t_f - t_i$) e la velocità costante (\dot{q}_{int}) determinata nel tratto intermedio non deve superare un certo valore massimo ammissibile (\dot{q}_{max}). Altrimenti si possono imporre una accelerazione ed una decelerazione massime consentite.

Esercizio 40 (Traiettoria lineare con giunzione parabolica) Determinare la traiettoria lineare con giunzione parabolica $q(t)$ che soddisfa le seguenti condizioni: $t_i = 1$, $t_f = 5$, $t_a = t_d = 1$, $\dot{q}_i = -10$, $\dot{q}_f = 10$, $\ddot{q}_i = 2$ e $\ddot{q}_f = 2$.

5.1.3 Traiettorie spline

Una traiettoria spline passa per n punti desiderati attraverso l'unione di $n - 1$ traiettorie polinomiali del terzo ordine; tali traiettorie si raccordano con condizioni sulla continuità delle velocità e delle accelerazioni.

Esercizio 41 (Traiettoria spline) Determinare la traiettoria spline passante per $q(0) = -5$, $q(1) = -2$, $q(3) = 0$, $q(7) = -2$ e $q(10) = 5$, imponendo le condizioni $\dot{q}(0) = 1$ e $\dot{q}(10) = 2$ sulle velocità agli estremi (suggerimento: si veda la funzione "spline" di Matlab).

5.2 Traiettorie nello spazio di lavoro

Le traiettorie possono anche essere generate nello spazio operativo e convertite successivamente nello spazio di giunto utilizzando l'espressione della cinematica inversa del manipolatore.

Bisogna controllare eventuali configurazioni di singolarità lungo la traiettoria descritta, perché potrebbero dar luogo ad eccessive sollecitazioni ai giunti del manipolatore.

5.3 Traiettorie per l'orientamento

Si preferisce definire l'orientamento intermedio tra due posizioni successive di una traiettoria in termini di angoli di Eulero o di Roll, Pitch e Yaw; l'utilizzo della matrice di rotazione R crea invece problemi nella definizione degli orientamenti intermedi, all'interno della traiettoria, non facilmente esprimibili in termini di R .

5.4 Scalatura delle traiettorie

Le traiettorie generate possono essere riscalate nel tempo in modo da non sollecitare eccessivamente il manipolatore e da richiedere coppie di valore contenute.

Sia $\sigma(t)$ la legge oraria di percorrenza della traiettoria, sostituendo $x = kt$ al posto di t si ottiene una nuova legge oraria $\hat{\sigma}(x)$ con $x \in [0, kT]$ ($t \in [0, T]$), tale per cui

$$\begin{aligned}\hat{\sigma}(x) &= \sigma(t) \\ \dot{\hat{\sigma}}(x) &= \frac{\sigma(t)}{\sigma'(t)} \\ \ddot{\hat{\sigma}}(x) &= \frac{\ddot{\sigma}(t)}{k^2}\end{aligned}$$

se $k > 1$ la traiettoria è più lenta, viceversa se $k < 1$.

Capitolo 6

Controllo di un manipolatore

Il controllo di un manipolatore consiste nel determinare le coppie da applicare ai vari giunti per ottenere un certo comportamento desiderato.

Il controllo può essere dei movimenti (controllo di posizione) e/o dell'interazione (controllo di forza) con l'ambiente, può essere nello spazio di giunto o nello spazio operativo ed inoltre può essere centralizzato o decentralizzato (indipendente per ogni giunto).

6.1 Controllo di posizione

In questa sezione analizzeremo il controllo del moto del manipolatore o controllo di posizione.

6.1.1 Controllo nello spazio dei giunti

Il modello dinamico di riferimento di un manipolatore è il seguente

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + Dq + g(q) = \tau + J^T(q)F_a$$

ed il problema del controllo consiste nel determinare le coppie τ da applicare ai giunti per ottenere un moto $q_d(t)$ desiderato.

Siano $q_m(t)$ e $\tau_m(t)$ le posizioni e le coppie degli attuatori e K_r le costanti di riduzione

$$K_r q = q_m \quad \tau_m = K_r \tau,$$

la matrice $M(q)$ può essere scomposta in un termine diagonale costante (inerzie medie viste ai giunti) ed un termine dipendente dalla configurazione

$$M(q) = \hat{M} + \Delta M(q),$$

si ottiene così il nuovo modello dinamico

$$\tau_m = (K_r^{-1} \hat{M} K_r^{-1}) \ddot{q}_m + D_m \dot{q}_m + d$$

dove $D_m = K_r^{-1} D K_r^{-1}$ ed inoltre $d = (K_r^{-1} \Delta M(q) K_r^{-1}) \ddot{q}_m + (K_r^{-1} C(q, \dot{q}) K_r^{-1}) \dot{q}_m + K_r^{-1} g(q)$ che può essere considerato come un disturbo¹.

Si ottiene quindi un sistema dinamico (lineare, disaccoppiato e con un termine di disturbo) con equazione

$$\ddot{q}_m = K_r \hat{M}^{-1} K_r (\tau_m - D_m \dot{q}_m - d)$$

il cui schema è rappresentato in figura 6.1.

6.1.2 Controllo decentralizzato

Negli schemi di controllo decentralizzato ogni giunto è trattato indipendentemente e riceve un opportuno set-point di posizione da inseguire.

Il termine d (vedi fig.6.1) viene considerato come un disturbo esterno e si possono creare fino a tre anelli di retroazione con posizione, velocità ed accelerazione del singolo giunto; spesso si aggiunge anche

¹È stato trascurato in questa formulazione dinamica il termine di interazione con l'ambiente esterno.

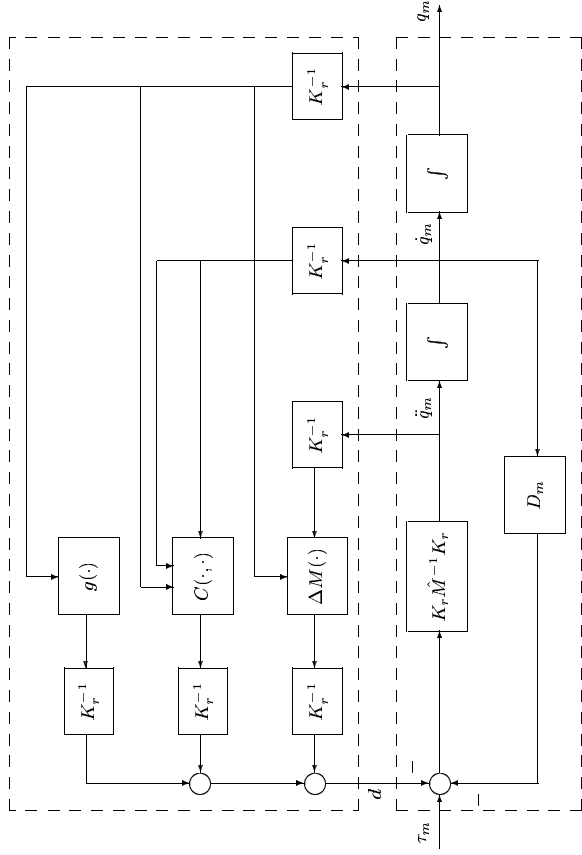


Figura 6.1: Schema della dinamica del manipolatore per il controllo di posizione.

una parte detta di *compensazione in avanti* che tiene conto dei valori di velocità ed accelerazione del riferimento imposto.

Il termine di disturbo d può essere stimato a partire dai riferimenti imposti ai singoli giunti e si può creare perciò un'azione *in avanti centralizzata* per compensare in parte gli effetti del disturbo esterno.

6.1.3 Controllo centralizzato

Il modello dinamico di riferimento per il controllo centralizzato è

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D\dot{q} + g(q) = u$$

dove u rappresenta l'ingresso manipolabile del sistema dinamico. L'obiettivo del controllo è quello di determinare i valori dell'ingresso $u(t)$ in modo tale da avere un andamento delle variabili $q(t)$ il più possibile simile a quello desiderato $q_d(t)$.

Controllo PD con compensazione di gravità

L'ingresso con controllo di tipo PD con compensazione di gravità è così determinato

$$u = g(q) + K_P \dot{q} - K_D \dot{q}$$

dove $\bar{q} = q_d - q$ rappresenta lo scostamento dalla posizione desiderata e K_P , K_D sono matrici definite positive.

Considerando la seguente funzione di Lyapunov

$$V(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} + \frac{1}{2} q^T K_P \bar{q}$$

si può mostrare che l'ingresso precedente garantisce $q = q_d$ all'equilibrio.

Controllo a dinamica inversa

Il modello dinamico può essere riscritto come

$$M(q)\ddot{q} + n(q, \dot{q}) = u$$

e scegliendo

$$u = M(q)y + n(q, \dot{q})$$

si ottiene $\ddot{q} = y$ dove y è il nuovo ingresso del sistema. In questo modo il sistema risulta lineare e disaccoppiato rispetto ad y e scegliendo come legge di controllo

$$y = -K_P \bar{q} - K_D \dot{\bar{q}} + \ddot{q}_d + r$$

si ottiene $\ddot{q} + K_D \dot{\bar{q}} + K_P \bar{q} = r$ che risulta asintoticamente stabile per K_P , K_D definite positive. Imponendo come riferimento $r = \ddot{q}_d + K_D \dot{\bar{q}}_d + K_P \bar{q}_d$ si ottiene la seguente dinamica per l'errore di inseguimento:

$$\ddot{\bar{q}} + K_D \dot{\bar{q}} + K_P \bar{q} = 0$$

6.1.4 Controllo nello spazio operativo

Le traiettorie desiderate del manipolatore possono essere espresse nello spazio operativo x_d , anziché in quello di giunto q_d , ed esistono degli schemi di controllo apposti per questa situazione.

Controllo con trasposta dello Jacobiano

Il controllo PD con compensazione di gravità applicato nello spazio operativo conduce alla legge di controllo

$$u = g(q) + J_A^T(q)K_P \bar{x} - J_A^T(q)K_D J_A(q)\dot{\bar{q}}$$

e considerando la seguente funzione di Lyapunov

$$V(q, \bar{x}) = \frac{1}{2} \dot{\bar{x}}^T M(q) \dot{\bar{x}} + \frac{1}{2} \bar{x}^T K_P \bar{x}$$

si può mostrare che l'ingresso precedente garantisce $x = x_d$ all'equilibrio, purché il termine $K_P \bar{x}$ non appartenga al sottospazio nullo della matrice J_A^T .

Controllo con inversa dello Jacobiano

Il controllo a dinamica inversa applicato nello spazio operativo conduce ad una legge di controllo che sfrutta l'inversa² della matrice Jacobiana. Imponendo

$$y = J_A^{-1}(q)(\ddot{x}_d + K_D \bar{x} + K_P \bar{x} - \dot{J}_A(q)\dot{\bar{q}})$$

e considerando l'uguaglianza $\ddot{x} = J_A(q)\ddot{q} + \dot{J}_A \dot{q}$ si ottiene

$$\ddot{\bar{x}} + K_D \dot{\bar{x}} + K_P \bar{x} = 0$$

che definisce la dinamica dell'errore nello spazio operativo.

²Il manipolatore non deve passare per configurazioni di singolarità, tali configurazioni rendono infatti non invertibile la matrice $J_A(q)$.

6.1.5 Controllo in presenza di incertezze

In presenza di incertezze, dovute a modellizzazione non precisa della dinamica, conoscenza imperfetta dei parametri e presenza di disturbi esterni, si utilizzano algoritmi di controllo robusto o di controllo adattativo per far meglio fronte a tali incertezze.

Esercizio 42 (Controllo di posizione di un manipolatore) Realizzare diversi controllori di posizione per il manipolatore planare verticale a due bracci le cui matrici sono così definite:

$$M(q) = \begin{bmatrix} 20 + 12 \cos(q_2) & 4 + 6 \cos(q_2) \\ 4 + 6 \cos(q_2) & 4 \end{bmatrix}$$

$$C(q, \dot{q}) = \begin{bmatrix} -6 \sin(q_2) \dot{q}_2 & -6 \sin(q_2) (\dot{q}_1 + \dot{q}_2) \\ 6 \sin(q_2) \dot{q}_1 & 0 \end{bmatrix}$$

$$g(q) = \begin{bmatrix} 18g \cos(q_1) + 6g \cos(q_1 + q_2) \\ 6g \cos(q_1 + q_2) \end{bmatrix}$$

$$K_P = \begin{bmatrix} 650 & 0 \\ 0 & 250 \end{bmatrix}; \quad K_D = \begin{bmatrix} 50 & 0 \\ 0 & 30 \end{bmatrix}$$

Sia $x_d(t) = 1.5 - 0.5t$, $y_d(t) = 0.5t$ con $t \in [0; 3]$ la traiettoria rettilinea desiderata espressa nello spazio operativo (i due bracci hanno entrambi lunghezza unitaria); si esprimano i 2 algoritmi di controllo: "PD con compensazione di gravità" e "dinamica inversa", nello spazio operativo e nello spazio di giunto, tenendo conto dei seguenti parametri:

Si evidenzino graficamente in ambiente Matlab i risultati ottenuti partendo da configurazioni differenti del manipolatore.

6.2 Controllo di forza

In questa sezione analizzeremo il controllo dell'interazione con l'ambiente da parte del manipolatore o controllo di forza.

6.2.1 Controllo di rigidità

Il controllo di forza può essere ottenuto mediante un controllo di posizione se si modella l'ambiente come elastico con una certa rigidità k_e .

La forza esercitata dall'ambiente diventa dunque

$$f_e = k_e(x - x_e)$$

dove x_e rappresenta la posizione dell'ostacolo, tale forza è direttamente proporzionale allo sprofondamento nella superficie cedevole.

Imponendo un controllo di tipo

$$f = k_p(x_d - x) - k_v \dot{x}$$

$$\ddot{x} = \frac{k_p x_d + k_v \dot{x}_e}{k_p + k_e}$$

e quindi

$$\ddot{f}_e = k_e(\ddot{x} - \ddot{x}_e) = \frac{k_p k_e}{k_p + k_e}(x_d - x_e)$$

dove la forza di interazione con l'ambiente può essere regolata attraverso la posizione x_d desiderata imposta nel controllo.

Il modello dinamico del manipolatore diventa perciò

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D\dot{q} + g(q) = u + J^T(q)w_e$$

dove w_e rappresenta il vettore di wrench dovuto all'interazione con l'ambiente e

$$u = g(q) + J_A^T(q)K_P \ddot{x} - J_A^T(q)K_D J_A(q)\dot{q}$$

rappresenta il controllo di rigidità.

³Ottenuto imponendo la condizione $f = f_e$.

6.2.2 Controllo di impedenza

Il controllo di impedenza cerca di imporre una relazione statica (o dinamica) tra la forza e la velocità all'end-effector del manipolatore che rappresenta il comportamento desiderato. Ad esempio si può imporre

$$\frac{W(s)}{\dot{X}(s)} = Z(s) = \frac{-(M_d s^2 + D_d s + K_d)}{s}$$

dove $Z(s)$ rappresenta l'impedenza caratteristica.

Il modello dinamico scritto nello spazio operativo diventa

$$\dot{M}(x)\ddot{x} + \dot{C}(x, \dot{x})\dot{x} + \hat{g}(x) = F + F_a$$

e ponendo

$$F = \dot{M}(x)y + \dot{C}(x, \dot{x})\dot{x} + \hat{g}(x)F_a$$

si ottiene il modello lineare $\ddot{x} = y$.

Scegliendo

$$y = \ddot{x}_d + M_d^{-1} [D_d(\dot{x}_d - \dot{x}) + K_d(x_d - x) + F_a]$$

si ottiene

$$M_d(\ddot{x}_d - \ddot{x}) + D_d(\dot{x}_d - \dot{x}) + K_d(x_d - x) = -F_a$$

che rappresenta proprio l'impedenza caratteristica desiderata.

6.2.3 Controllo ibrido (posizione/forza)

Nel controllo ibrido vengono scelte alcune direzioni in cui controllare i movimenti del manipolatore ed altre in cui controllare le forze.

La scelta dipende tipicamente dal tipo di compito che il manipolatore deve svolgere. Si creano due matrici di selezione Σ_v e Σ_f per estrarre separatamente le componenti da controllare in posizione ed in forza e si creano due differenti controllori che operano sui valori calcolati attraverso tali matrici.

Bibliografia

- [1] C. Melchiorri, "Dispense del corso di Robotica Industriale," anno accademico 1999-2000.
- [2] C. Melchiorri, "Traiettorie per azionamenti elettrici," Progetto Leonardo, *Esculapio Ed.*, Bologna, Feb. 2000.
- [3] L. Sciavicco e B. Siciliano, "Modellistica e controllo di robot manipolatori," *McGraw-Hill Italia*, 1995.
- [4] M.W. Spong e M. Vidyasagar, "Robot dynamics and control," *John Wiley & Sons, Inc.*, 1989.
- [5] M.P. Groover, M. Weiss, R.N. Nagel e N.G. Odrey, "Industrial Robotics: Technology, Programming and Applications," *McGraw-Hill*, 1986.
- [6] P.I. Corke, "A Robotics Toolbox for MATLAB," *IEEE Robotics and Automation Magazine*, n. 1, vol. 3, march 1996, pp. 24-32.
- [7] <http://www.cat.cstro.au/cmst/staff/pic/robot.tb.html>, indirizzo web del toolbox di robotica.
- [8] *ROBOT.PS*, file associato al toolbox di robotica.