



Esercitazioni di
Tecnologie per sistemi robotici

Corso di Laurea in Ingegneria Informatica e Robotica – A.A. 2022-2023

DIPARTIMENTO DI INGEGNERIA

docente

Prof. Francesco FERRANTE

Prof. Francesco CROCETTI

Esercitazioni Tecnologie per Sistemi Robotici

studenti

Alex Ardelean alexnicolae.ardelean@studenti.unipg.it

0. Indice

1 Esercitazione 1: problema asse-angolo diretto	4
1.1 Descrizione del problema	4
1.1.1 Assi coordinati	7
1.2 Soluzione	7
1.2.1 Verifica per assi coordinati	8
2 Esercitazione 2: singolarità di rappresentazione	13
2.1 Descrizione del problema	13
2.1.1 Rotazione ZXZ	14
2.1.2 Rotazione ZXY	15
2.1.3 Singolarità	16
2.2 Soluzione	17
3 Esercitazione 3: cinematica diretta	20
3.1 Descrizione del problema	20
3.1.1 Convenzione DH	20
3.2 Soluzione	23
4 Esercitazione 4: cinematica inversa e spazi operativi	39
4.1 Descrizione del problema	39
4.2 Soluzione: cinematica inversa	40
4.2.1 Soluzione: spazio operativo	45
5 Esercitazione 5: cinematica inversa tramite disaccoppiamento	53
5.1 Descrizione del problema	53
5.1.1 Proprietà di disaccoppiamento polso-struttura portante	53
5.1.2 Manipolatore di stanford: cinematica diretta	54
5.2 Soluzione	59
5.2.1 Cinematica inversa per il manipolatore sferico RRP	59
5.2.2 Cinematica inversa per il polso sferico	61

6 Esercitazione 6: vettore velocità angolare	68
6.1 Descrizione del problema	68
6.1.1 Relazione fondamentale della cinematica	68
6.1.2 Rotazioni infinitesimali	70
6.1.3 Derivata di una matrice di rotazione	70
6.2 Soluzione	72
7 Esercitazione 7: mobilità e singolarità robot RRR	77
7.1 Descrizione del problema	77
7.2 Soluzione	78
8 Esercitazione 8: analisi polso sferico	93
8.1 Descrizione del problema	93
8.1.1 Jacobiano geometrico	93
8.1.2 Velocità di un braccio	94
8.1.3 Metodo sistematico per un giunto prismatico	95
8.1.4 Metodo sistematico per un giunto rotoidale	96
8.1.5 Metodo sistematico per il calcolo dello Jacobiano geometrico	97
8.2 Soluzione	98
9 Esercitazione 9: algoritmo di Newton e del gradiente	104
9.1 Descrizione del problema	104
9.1.1 Algoritmo di Newton	104
9.1.2 Algoritmo del gradiente	105
9.2 Soluzione	106
10 Esercitazione 10: cinematica inversa analitica e numerica robot RRP	131
10.1 Descrizione del problema	131
10.2 Soluzione	132
10.2.1 Soluzione analitica	132
10.2.2 Soluzione numerica	134
11 Esercitazione 11: manipolabilità RR	140
11.1 Descrizione del problema	140
11.2 Soluzione	141
12 Esercitazione 12: analisi leggi di controllo	146
12.1 Descrizione del problema	146
12.1.1 Controllo P.D.	147
12.1.2 Controllo P.D. con gravità	148
12.1.3 Controllo P.I.D. per la compensazione della gravità	149

12.2 Soluzione	150
12.2.1 Controllo PID	150
12.2.2 Controllo PD	152
12.2.3 Controllo PD + compensazione della gravità	152
13 Esercitazione 13: modello dinamico robot PR	154
13.1 Descrizione del problema	154
13.1.1 Metodo di Lagrange	154
13.1.2 Energia di corpi rigidi in moto nello spazio	155
13.2 Soluzione	161

1. Esercitazione 1: problema asse-angolo diretto

(1) Codice per il problema diretto asse-angolo. Verifica con rotazioni intorno agli assi coordinati.

1.1 Descrizione del problema

Il problema asse-angolo diretto può essere espresso nel seguente modo: trovare la matrice $\mathbf{R}(\theta, \underline{r})$ che effettua la rotazione dell'angolo θ attorno all'asse \underline{r} .

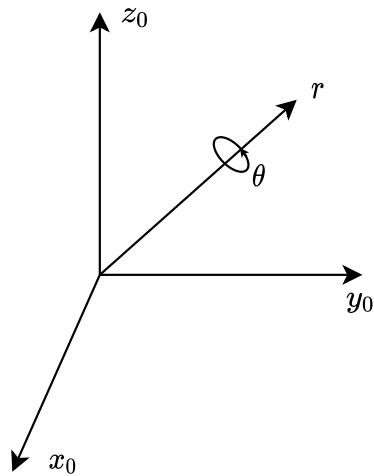


Figura 1.1: Asse-Angolo

Per risolvere tale problema si considera una sequenza di tre rotazioni:

1. Si porta l'asse \underline{z}_0 lungo \underline{r} tramite la matrice \mathbf{C} .
2. Si ruota di θ lungo il nuovo asse \underline{z} tramite la matrice di rotazione elementare $\mathbf{R}_z(\theta)$.
3. Si effettua la rotazione inversa rispetto al punto 1.

La matrice di rotazione complessiva può quindi essere ricavata come composizione delle tre rotazioni precedentemente descritte:

$$\mathbf{R}(\theta, \underline{r}) = \underbrace{\mathbf{C}}_1 \underbrace{\mathbf{R}_z(\theta)}_2 \underbrace{\mathbf{C}^T}_3 \quad (1.1)$$

La matrice \mathbf{C} porta l'asse \underline{z} lungo \underline{r} . Dato che, in generale, le colonne di una matrice di rotazione rappresentano le coordinate del nuovo sistema di riferimento nel vecchio, e il nuovo sistema di riferimento ha \underline{r} come asse \underline{z} , la matrice \mathbf{C} ha \underline{r} come terza colonna, mentre gli altri due vettori sono arbitrari. Quindi la matrice \mathbf{C} è

$$\mathbf{C} = [\underline{n} \quad \underline{s} \quad \underline{r}] \quad (1.2)$$

tali che \underline{n} e \underline{s} sono ortogonali, hanno norma unitaria e $\underline{n} \times \underline{s} = \underline{r}$ in modo da formare una terna levogira.

Ricordando che la matrice di rotazione di un angolo θ intorno all'asse \underline{z} è

$$\mathbf{R}_z(\theta) = \begin{bmatrix} c(\theta) & -s(\theta) & 0 \\ s(\theta) & c(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

è possibile riscrivere la 1.1 come segue:

$$\begin{aligned} \mathbf{R}(\theta, \underline{r}) &= \mathbf{C} \mathbf{R}_z(\theta) \mathbf{C}^T = [\underline{n} \quad \underline{s} \quad \underline{r}] \begin{bmatrix} c(\theta) & -s(\theta) & 0 \\ s(\theta) & c(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \underline{n}^T \\ \underline{s}^T \\ \underline{r}^T \end{bmatrix} = \\ &= \underline{r}\underline{r}^T + (\underline{n}\underline{n}^T + \underline{s}\underline{s}^T)c(\theta) + (\underline{s}\underline{n}^T - \underline{n}\underline{s}^T)s(\theta) \end{aligned}$$

In realtà è possibile esprimere $\mathbf{R}(\theta, \underline{r})$ in funzione dei soli θ e \underline{r} , eliminando perciò la dipendenza da \underline{n} e \underline{s} . È infatti possibile considerare i seguenti vincoli:

- $\mathbf{C}\mathbf{C}^T$ è di rotazione $\mathbf{I} = \underline{n}\underline{n}^T + \underline{s}\underline{s}^T + \underline{r}\underline{r}^T$ e quindi $\underline{n}\underline{n}^T + \underline{s}\underline{s}^T = \mathbf{I} - \underline{r}\underline{r}^T$
- $\underline{n} \times \underline{s} = \det \begin{bmatrix} \underline{i} & \underline{j} & \underline{k} \\ n_x & n_y & n_z \\ s_x & s_y & s_z \end{bmatrix} = \begin{bmatrix} n_y s_z - s_y n_z \\ n_x s_x - s_z n_x \\ n_x s_y - s_x n_y \end{bmatrix} = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \underline{r}$.

Quindi sapendo che:

$$r_x = n_y s_z - s_y n_z$$

$$r_y = n_x s_x - s_z n_x$$

$$r_z = n_x s_y - s_x n_y$$

è possibile dire che:

$$\begin{aligned} \underline{s}\underline{n}^T - \underline{n}\underline{s}^T &= \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix} \begin{bmatrix} n_x & n_y & n_z \end{bmatrix} - \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \begin{bmatrix} s_x & s_y & s_z \end{bmatrix} = \\ &= \begin{bmatrix} s_x n_x & s_x n_y & s_x n_z \\ s_y n_x & s_y n_y & s_y n_z \\ s_z n_x & s_z n_y & s_z n_z \end{bmatrix} - \begin{bmatrix} n_x s_x & n_x s_y & n_x s_z \\ n_y s_x & n_y s_y & n_y s_z \\ n_z s_x & n_z s_y & n_z s_z \end{bmatrix} = \\ &= \begin{bmatrix} 0 & -(n_x s_y - s_x n_y) & n_x s_x - s_z n_x \\ n_x s_y - s_x n_y & 0 & -(n_y s_z - s_y n_z) \\ -(n_x s_x - s_z n_x) & n_y s_z - s_y n_z & 0 \end{bmatrix} = \\ &= \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} = \mathbf{S}(\underline{r}) \end{aligned}$$

dove $\mathbf{S}(\underline{r})$ è chiamato vee operator e mappa un vettore in una matrice antisimmetrica ($A^T = -A$).

È quindi possibile dire che

$$\mathbf{R}(\theta, \underline{r}) = \underline{r}\underline{r}^T + (\mathbf{I} - \underline{r}\underline{r}^T)c(\theta) + \mathbf{S}(\underline{r})s(\theta) \quad (1.3)$$

1.1.1 Assi coordinati

Se \underline{r} corrisponde a uno degli assi si ricade nel caso di matrici di rotazione elementari.

Ad esempio se \underline{r} corrisponde all'asse \underline{z} , ovvero $\underline{r} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$, si ha:

$$\underline{rr}^T = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{I} - \underline{rr}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{S}(\underline{r}) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

e quindi:

$$\mathbf{R}(\theta, \underline{r}) = \begin{bmatrix} c(\theta) & -s(\theta) & 0 \\ s(\theta) & c(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \stackrel{\text{def}}{=} \mathbf{R}_z(\theta)$$

Analogamente nei casi $\underline{r} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ e $\underline{r} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ si ottengono rispettivamente:

$$\mathbf{R}(\theta, \underline{r}) = \begin{bmatrix} c(\theta) & 0 & s(\theta) \\ 0 & 1 & 0 \\ -s(\theta) & 0 & c(\theta) \end{bmatrix} \stackrel{\text{def}}{=} \mathbf{R}_y(\theta) \quad \mathbf{R}(\theta, \underline{r}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\theta) & -s(\theta) \\ 0 & s(\theta) & c(\theta) \end{bmatrix} \stackrel{\text{def}}{=} \mathbf{R}_x(\theta)$$

1.2 Soluzione

```
function [R] = matRot(r,theta)
    % Calcola la matrice di rotazione dato l'asse e l'angolo.

    % Controlla che il vettore r abbia 3 componenti
    assert(numel(r) == 3, 'Il vettore r deve avere 3 componenti');

    % Controlla che theta sia uno scalare
    assert(isscalar(theta), 'L''angolo theta deve essere uno scalare');

    % Converte il vettore r in un vettore colonna a 3 componenti
```

```

r = reshape(r, [3,1]);

% Normalizza il vettore r
% Se r e' nullo si pone theta=0 in modo da avere R=I
rNorm = norm(r);
if(rNorm > 0)
    r = r/rNorm;
else
    theta = 0;
end

rx = r(1);
ry = r(2);
rz = r(3);
S = [0, -rz, ry; rz, 0, -rx; -ry, rx, 0];

R = r*(r.') + (eye(3)-r*(r.'))*cos(theta) + S*sin(theta);
end

```

Code 1.1: Funzione MATLAB per il calcolo della matrice di rotazione dato l'asse e l'angolo

1.2.1 Verifica per assi coordinati

Test della funzione matRot

Test sugli assi coordinati

Si vuole testare la funzione matRot per i tre assi coordinati.

Per fare ciò si definisce una variabile simbolica che rappresenta l'angolo di rotazione.

```
syms theta real
```

Asse coordinato X

```
x = [1; 0; 0];
Rx = matRot(x, theta)
```

```
Rx =

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix}$$

```

Asse coordinato Y

```
y = [0; 1; 0];
Ry = matRot(y, theta)
```

```
Ry =

$$\begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix}$$

```

Asse coordinato Z

```
z = [0; 0; 1];
Rz = matRot(z, theta)
```

```
Rz =

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```

Test grafico

```
% Assi
x = [1 0 0];
y = [0 1 0];
z = [0 0 1];

% Plot degli assi
quiver3(0, 0, 0, x(1), x(2), x(3), 'r', 'LineWidth', 2);
hold on;
quiver3(0, 0, 0, y(1), y(2), y(3), 'g', 'LineWidth', 2);
```

```

quiver3(0, 0, 0, z(1), z(2), z(3), 'b', 'LineWidth', 2);

text(x(1), x(2), x(3), 'X', 'FontSize', 14, 'Color', 'r');
text(y(1), y(2), y(3), 'Y', 'FontSize', 14, 'Color', 'g');
text(z(1), z(2), z(3), 'Z', 'FontSize', 14, 'Color', 'b');

% Rotazione di un vettore
vec1 = [1; 0.5; 0];
angle = pi/4;
R = matRot([0 0 1], angle);
vec2 = R*vec1;

% Plot del vettore ruotato
quiver3(0, 0, 0, vec1(1), vec1(2), vec1(3), 'k');
quiver3(0, 0, 0, vec2(1), vec2(2), vec2(3), 'k');

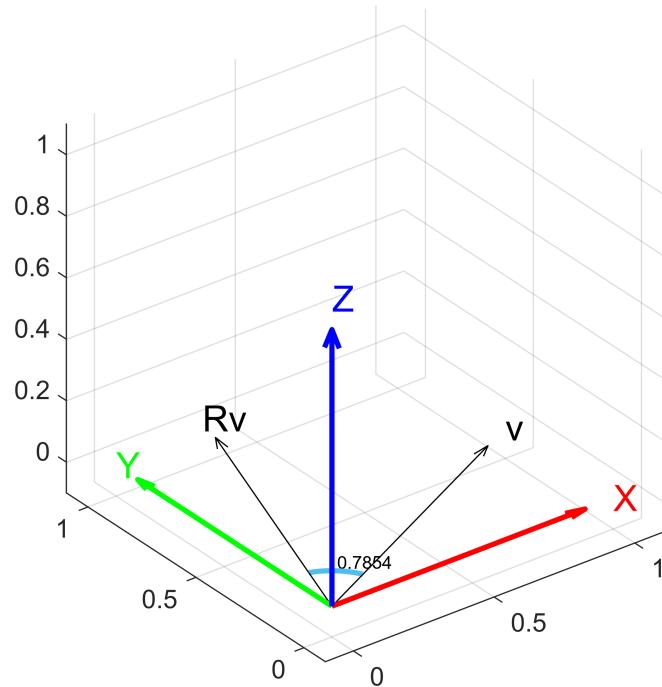
text(vec1(1), vec1(2), vec1(3), 'v', 'FontSize', 14, 'Color', 'k');
text(vec2(1), vec2(2), vec2(3), 'Rv', 'FontSize', 14, 'Color', 'k');

% Plot dell'angolo
drawAngle(vec1, vec2)

% limiti assi e label
grid on
axis equal;
xlim([-0.1 1.1]);
ylim([-0.1 1.1]);
zlim([-0.1 1.1]);
title('Rotazione intorno all''asse z');

```

Rotazione intorno all'asse z



```

function [] = drawAngle(vec1, vec2)
    % Disegna l'angolo tra due vettori

    raggio = 0.2;
    P1 = vec1/norm(vec1)*raggio;
    P2 = vec2/norm(vec2)*raggio;
    nrPunti = 100;
    angle = acos(dot(P1,P2) / (norm(P1) * norm(P2)));
    cross_product = cross(P1, P2);
    segno = 1;
    if cross_product(3) < 0
        angle = 2*pi - angle;
        segno = -1;
    end
    theta = linspace(0, angle, nrPunti);
    asse = cross(P1, P2);
    x = [];
    y = [];
    z = [];
    for i = 1:length(theta)
        R = matRot(asse, segno*theta(i));
        p = R*P1;
        x = [x p(1)];
        y = [y p(2)];
        z = [z p(3)];
    end
    c = segno*(P1+P2)/1.5;
    plot3(x, y, z, 'LineWidth', 2);
    text(c(1), c(2), c(3), string(angle), 'FontSize', 7);
end

```

Code 1.2: Funzione MATLAB per disegnare l'angolo tra due vettori

2. Esercitazione 2: singolarità di rappresentazione

(2) *Angoli di Eulero, singolarità di rappresentazione per due terne di angoli a scelta.*

2.1 Descrizione del problema

Le matrici di rotazione forniscono una descrizione ridondante dell'orientamento di una terna; esse sono infatti caratterizzate da nove elementi che non sono indipendenti ma legati tra di loro dai sei vincoli dovuti alle condizioni di ortogonalità espresse dalla relazione $\mathbf{R}\mathbf{R}^T = \mathbf{I}$.

Ciò porta a riconoscere che i parametri effettivi per la descrizione dell'orientamento sono in numero di tre. Una rappresentazione dell'orientamento in termini di tre parametri indipendenti costituisce una rappresentazione minima. Infatti, una rappresentazione minima del gruppo speciale ortonormale $SO(m)$ richiede $\frac{m(m-1)}{2}$ parametri; quindi, per parametrizzare $SO(3)$ sono necessari tre parametri, laddove nel caso di una rotazione nel piano $SO(2)$ è sufficiente un solo parametro.

Si è visto che una rappresentazione non minima dell'orientamento può ottenersi ricorrendo anche a quattro parametri che esprimono la rotazione di un certo angolo intorno a un asse dello spazio.

Una rappresentazione minima dell'orientamento può essere invece ottenuta utilizzando un insieme di tre parametri $\Phi = [\phi \ \theta \ \psi]^T$. Una matrice di rotazione che esprime la rotazione elementare intorno a uno degli assi coordinati è funzione di un singolo angolo. Pertanto, una generica matrice di rotazione può essere ricavata per composizione di tre rotazioni elementari secondo opportune sequenze in ma-

niera tale da garantire che due rotazioni successive non avvengano intorno ad assi paralleli. Questo comporta che solo 12 insiemi di tre angoli tra le 27 combinazioni possibili sono fattibili; ciascun insieme rappresenta una terna di angoli di Eulero. Questo tipo di rappresentazione è nota anche come rappresentazione ad assi mobili, ovvero gli angoli di rotazione rispetto al sistema di riferimento corrente, quindi una volta applicata la prima rotazione le successive sono applicate rispetto al nuovo sistema di riferimento (ruotato). Tuttavia esiste un altro tipo di rappresentazione nota come rappresentazione ad assi fissi, ovvero gli angoli di rotazione specificati si riferiscono tutti alla terna iniziale. Questo tipo di rappresentazione è molto usata in aeronautica e in robotica e in questo caso gli angoli prendono il nome roll, pitch e yaw (RPY).

2.1.1 Rotazione ZXZ

La rotazione descritta dagli angoli ZXZ viene costruita con la composizione delle seguenti rotazioni elementari:

1. si ruota la terna origine dell'angolo ϕ intorno all'asse z : tale rotazione è descritta dalla matrice di rotazione $\mathbf{R}_z(\phi)$;
2. si ruota la terna origine dell'angolo θ intorno all'asse x' corrente: tale rotazione è descritta dalla matrice di rotazione $\mathbf{R}_{x'}(\theta)$.
3. si ruota la terna origine dell'angolo ψ intorno all'asse z'' corrente: tale rotazione è descritta dalla matrice di rotazione $\mathbf{R}_{z''}(\psi)$.

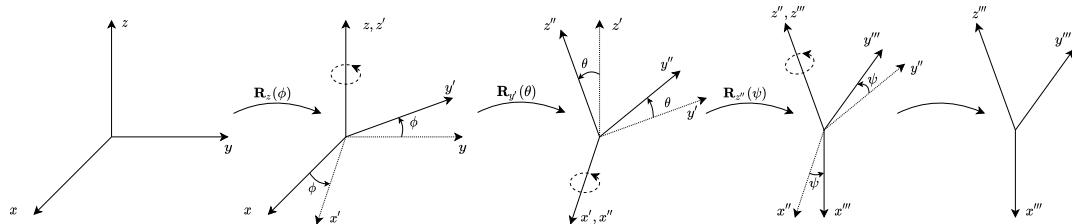


Figura 2.1: Rotazione ZXZ (assi mobili)

Problema diretto

Dati ϕ , θ e ψ , il problema diretto consiste nel determinare la matrice di rotazione $\mathbf{R}_{zx'z''}(\phi, \theta, \psi)$ che realizza tale rotazione. Per risolvere il problema quindi basta

applicare in maniera sequenziale le tre rotazioni, ovvero effettuare:

$$\begin{aligned} \mathbf{R}_{zx'z''}(\phi, \theta, \psi) &= \mathbf{R}_z(\phi)\mathbf{R}_{x'}(\theta)\mathbf{R}_{z''}(\psi) = \\ &= \begin{bmatrix} c(\phi)c(\psi) - s(\phi)s(\theta)s(\psi) & -c(\phi)s(\psi) - s(\phi)c(\theta)c(\psi) & s(\phi)s(\theta) \\ s(\phi)c(\psi) + c(\phi)c(\theta)s(\psi) & -s(\phi)s(\psi) + c(\phi)c(\theta)c(\psi) & -c(\phi)s(\theta) \\ s(\theta)s(\psi) & s(\theta)c(\psi) & c(\theta) \end{bmatrix} \quad (2.1) \end{aligned}$$

Problema inverso

Il problema inverso consiste nel ricavare ϕ , θ e ψ data la matrice di rotazione $\mathbf{R}_{zx'z''}(\phi, \theta, \psi)$.

Per ricavare θ è possibile osservare che l'elemento R_{33} della matrice corrisponde a $c(\theta)$, inoltre $R_{13}^2 + R_{23}^2 = (s(\phi)s(\theta))^2 + (-c(\phi)s(\theta))^2 = s^2(\phi)s^2(\theta) + c^2(\phi)s^2(\theta) = s^2(\theta)(s^2(\theta) + c^2(\phi)) = s^2(\theta)$ e quindi:

$$\theta = \text{atan2}\left(\pm\sqrt{R_{13}^2 + R_{23}^2}, R_{33}\right)$$

Per ricavare ψ è possibile osservare che $s(\psi) = \frac{R_{31}}{s(\theta)}$ e $c(\psi) = \frac{R_{32}}{s(\theta)}$ e quindi:

$$\psi = \text{atan2}\left(\frac{R_{31}}{s(\theta)}, \frac{R_{32}}{s(\theta)}\right)$$

Per ricavare ϕ è possibile osservare che $s(\phi) = \frac{R_{13}}{s(\theta)}$ e $c(\phi) = -\frac{R_{23}}{s(\theta)}$ e quindi:

$$\phi = \text{atan2}\left(\frac{R_{13}}{s(\theta)}, -\frac{R_{23}}{s(\theta)}\right)$$

2.1.2 Rotazione ZXY

La rotazione descritta dagli angoli ZXY viene costruita con la composizione delle seguenti rotazioni elementari:

1. si ruota la terna origine dell'angolo ϕ intorno all'asse z : tale rotazione è descritta dalla matrice di rotazione $\mathbf{R}_z(\phi)$;
2. si ruota la terna origine dell'angolo θ intorno all'asse x' corrente: tale rotazione è descritta dalla matrice di rotazione $\mathbf{R}_{x'}(\theta)$.
3. si ruota la terna origine dell'angolo ψ intorno all'asse z'' corrente: tale rotazione è descritta dalla matrice di rotazione $\mathbf{R}_{y''}(\psi)$.

Problema diretto

Dati ϕ , θ e ψ , il problema diretto consiste nel determinare la matrice di rotazione $\mathbf{R}_{zx'y''}(\phi, \theta, \psi)$ che realizza tale rotazione. Per risolvere il problema quindi basta applicare in maniera sequenziale le tre rotazioni, ovvero effettuare:

$$\begin{aligned}\mathbf{R}_{zx'y''}(\phi, \theta, \psi) &= \mathbf{R}_z(\phi)\mathbf{R}_{x'}(\theta)\mathbf{R}_{y''}(\psi) = \\ &= \begin{bmatrix} c(\phi)c(\psi) - s(\phi)s(\psi)s(\theta) & -c(\theta)s(\phi) & c(\phi)s(\psi) + c(\psi)s(\phi)s(\theta) \\ c(\psi)s(\phi) + c(\phi)s(\psi)s(\theta) & c(\phi)c(\theta) & s(\phi)s(\psi) - c(\phi)c(\psi)s(\theta) \\ -c(\theta)s(\psi) & s(\theta) & c(\psi)c(\theta) \end{bmatrix} \quad (2.2)\end{aligned}$$

Problema inverso

Il problema inverso consiste nel ricavare ϕ , θ e ψ data la matrice di rotazione $\mathbf{R}_{zx'y''}(\phi, \theta, \psi)$.

Per ricavare θ è possibile osservare che l'elemento R_{32} della matrice corrisponde a $s(\theta)$, inoltre $R_{12}^2 + R_{22}^2 = (-c(\theta)s(\phi))^2 + (c(\phi)c(\theta))^2 = c^2(\theta)s^2(\phi) + c^2(\phi)c^2(\theta) = c^2(\theta)(s^2(\phi) + c^2(\phi)) = c^2(\theta)$ e quindi:

$$\theta = \text{atan2}\left(R_{32}, \pm\sqrt{R_{12}^2 + R_{22}^2}\right)$$

Per ricavare ψ è possibile osservare che $s(\psi) = -\frac{R_{31}}{c(\theta)}$ e $c(\psi) = \frac{R_{33}}{c(\theta)}$ e quindi:

$$\psi = \text{atan2}\left(-\frac{R_{31}}{c(\theta)}, \frac{R_{33}}{c(\theta)}\right)$$

Per ricavare ϕ è possibile osservare che $s(\phi) = -\frac{R_{12}}{c(\theta)}$ e $c(\phi) = \frac{R_{22}}{c(\theta)}$ e quindi:

$$\phi = \text{atan2}\left(-\frac{R_{12}}{c(\theta)}, \frac{R_{22}}{c(\theta)}\right)$$

2.1.3 Singolarità

Angoli ZXZ

Si è visto che la soluzione al problema inverso per gli angoli ZXZ è:

$$\begin{aligned}\theta &= \text{atan2}\left(\pm\sqrt{R_{13}^2 + R_{23}^2}, R_{33}\right) \\ \psi &= \text{atan2}\left(\frac{R_{31}}{s(\theta)}, \frac{R_{32}}{s(\theta)}\right) \\ \phi &= \text{atan2}\left(\frac{R_{13}}{s(\theta)}, -\frac{R_{23}}{s(\theta)}\right)\end{aligned}$$

Tale soluzione degenera quando $s(\theta) = 0$, in questo caso è infatti possibile determinare soltanto la somma o la differenza di ψ e ϕ . Infatti, se $\theta = 0, \pi$ le rotazioni successive di ϕ e ψ sono effettuate intorno ad assi della terna corrente paralleli fra loro, fornendo così effetti di rotazione equivalenti.

Angoli ZX \bar{Y}

Si è visto che la soluzione al problema inverso per gli angoli ZXZ è:

$$\begin{aligned}\theta &= \text{atan}2\left(R_{32}, \pm \sqrt{R_{12}^2 + R_{22}^2}\right) \\ \psi &= \text{atan}2\left(-\frac{R_{31}}{c(\theta)}, \frac{R_{33}}{c(\theta)}\right) \\ \phi &= \text{atan}2\left(-\frac{R_{12}}{c(\theta)}, \frac{R_{22}}{c(\theta)}\right)\end{aligned}$$

Tale soluzione degenera quando $c(\theta) = 0$, in questo caso è infatti possibile determinare soltanto la somma o la differenza di ψ e ϕ . Infatti, se $\theta = \frac{\pi}{2}, \frac{3\pi}{2}$ le rotazioni successive di ϕ e ψ sono effettuate intorno ad assi della terna corrente paralleli fra loro, fornendo così effetti di rotazione equivalenti.

2.2 Soluzione

Singolarità di rappresentazione

Singolarità per angoli ZXZ

Si imposta l'angolo $\theta = 0$ in modo che causi il fenomeno di singolarità.

```
theta = 0;
```

Si impostano ora gli angoli ψ, ϕ come variabili simboliche.

```
syms psi phi real
```

Si calcola la matrice di rotazione ZXZ come $R_{zx'z''}(\phi, \theta, \psi) = R_z(\phi)R_x'(\theta)R_z''(\psi)$.

```
z = [0 0 1];
x = [1 0 0];
Rz_psi = matRot(z, psi);
Rx_theta = matRot(x, theta);
Rz_phi = matRot(z, phi);
Rzxz = simplify(Rz_phi*Rx_theta*Rz_psi)
```

```
Rzxz =

$$\begin{pmatrix} \cos(\phi + \psi) & -\sin(\phi + \psi) & 0 \\ \sin(\phi + \psi) & \cos(\phi + \psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```

E' possibile notare come gli angoli ψ, ϕ compaiano insieme, il che implica la non possibilità di determinarli indipendentemente. E' tuttavia possibile ricavare la loro somma.

Singolarità per angoli ZXY

Si imposta l'angolo $\theta = \frac{\pi}{2}$ in modo che causi il fenomeno di singolarità.

```
theta = pi/2;
```

Si impostano ora gli angoli ψ, ϕ come variabili simboliche.

```
syms psi phi real
```

Si calcola la matrice di rotazione RPY come $R_{zx'y''}(\phi, \theta, \psi) = R_z(\phi)R_x'(\theta)R_y''(\psi)$.

```
z = [0 0 1];
x = [1 0 0];
y = [0 1 0];
Rz_phi = matRot(z, phi);
Rx_theta = round(matRot(x, theta));
Ry_psi = matRot(y, psi);
```

```
Rzxy = simplify(Rz_phi*Rx_theta*Ry_psi)
```

Rzxy =

$$\begin{pmatrix} \cos(\phi + \psi) & 0 & \sin(\phi + \psi) \\ \sin(\phi + \psi) & 0 & -\cos(\phi + \psi) \\ 0 & 1 & 0 \end{pmatrix}$$

E' possibile notare come l'angoli ψ compare esclusivamente insieme a ϕ , il che implica la non possibilità di determinarli indipendentemente. E' tuttavia possibile ricavare la loro somma.

3. Esercitazione 3: cinematica diretta

(3) Tabella di DH e cinematica diretta completa per le seguenti strutture robotiche: Manipolatore sferico, manipolatore SCARA, manipolatore antropomorfo, manipolatore di Standford (solo struttura portante), polso sferico.

3.1 Descrizione del problema

3.1.1 Convenzione DH

Per poter esprimere, rispetto ad una terna di riferimento, la posa dell'organo terminale in funzione delle variabili di giunto della struttura meccanica in maniera sistematica si utilizza la convenzione Denavit-Hartenberg. Tale convenzione viene utilizzata per definire le terne solidali ai bracci. Notando che il giunto i connette il braccio $i - 1$ con il braccio i la convenzione prevede i seguenti passaggi:

1. Scegliere z_i lungo l'asse del giunto $i + 1$.
2. Individuare O_i all'intersezione dell'asse z_i con la normale comune agli assi z_{i-1} e z_i .
3. Scegliere l'asse x_i diretto lungo la normale comune agli assi z_{i-1} e z_i . Da notare che il verso positivo dell'asse x_i va dal giunto i al giunto $i + 1$.
4. Scegliere l'asse y_i in modo da completare la terna levogira. La terna i è solidale al braccio i .

Tale convenzione presenta però delle ambiguità:

- Per la terna 0 l'origine e l'asse x_0 sono arbitrari (dato che non esiste l'asse -1).
- Per la terna n l'origine e l'asse z_n sono arbitrari (dato che non esiste l'asse $n+1$).
- Quando due assi consecutivi sono paralleli (ci sono infinite possibili normali comuni tra i 2 assi). L'origine O_i può essere definita in modo arbitrario lungo z_i .
- Quando due assi consecutivi sono incidenti il verso di x_i è arbitrario.

Una volta specificate le terne solidali ai bracci, la posa della terna i rispetto alla terna $i-1$ è completamente specificata dai seguenti parametri:

1. a_i rappresenta lo spostamento tra O_{i-1} e O_i lungo x_i . In generale a_i è costante e dipende dalla struttura del manipolatore.
2. d_i rappresenta lo spostamento tra O_{i-1} e O_i lungo z_{i-1} . Se il giunto è prismatico il parametro d_i è la variabile di giunto.
3. α_i rappresenta l'angolo tra gli assi z_{i-1} e z_i intorno all'asse x_i . In generale α_i è costante e dipende dalla struttura del manipolatore ed è valutato positivo in senso antiorario. Esso viene chiamato twist angle.
4. θ_i rappresenta l'angolo tra gli assi x_{i-1} e x_i intorno all'asse z_{i-1} . Se il giunto è rotoidale il parametro θ_i è la variabile di giunto ed è valutata positiva in senso antiorario.

I parametri DH (a_i , d_i , α_i e θ_i) fanno sovrapporre con rotazioni elementari e traslazioni il frame $i-1$ con il frame i . Per portare la terna $i-1$ sulla terna i si parte dalla terna $i-1$ e:

- Si ricava la terna i' tramite una roto-traslazione intorno e lungo z_{i-1} di θ_i e d_i .

La matrice omogenea ${}^{i-1}\mathbf{A}_{i'}(q_i)$ che effettua tale roto-traslazione è ottenibile

come composizione di una rotazione e una traslazione:

$$\begin{aligned}
{}^{i-1}\mathbf{A}_{i'}(q_i) &= \underbrace{\begin{bmatrix} \mathbf{R}_z(\theta_i) & 0_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix}}_{\text{rotazione}} \underbrace{\begin{bmatrix} \mathbf{I}_{3 \times 3} & 0 \\ 0 & d_i \\ 0_{1 \times 3} & 1 \end{bmatrix}}_{\text{traslazione}} = \\
&= \begin{bmatrix} c(\theta_i) & -s(\theta_i) & 0 & 0 \\ s(\theta_i) & c(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
&= \begin{bmatrix} c(\theta_i) & -s(\theta_i) & 0 & 0 \\ s(\theta_i) & c(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

dove

$$q_i = \begin{cases} \theta_i & \text{per giunti rotoidali} \\ d_i & \text{per giunti prismatici} \end{cases}$$

Da notare che la matrice omogenea è il risultato del prodotto di due matrici omogenee, una che fa la rotazione e una che fa la traslazione, ma, siccome queste avvengono intorno e lungo lo stesso asse, si possono moltiplicare in qualsiasi ordine e il risultato è invariante.

- Si arriva alla terna i tramite una roto-traslazione intorno e lungo x_i di α_i e a_i .

Analogamente a prima la matrice ${}^{i'}\mathbf{A}_i$ che effettua la roto-traslazione è ricavabile come composizione di una rotazione e una traslazione. Il risultato è:

$${}^{i'}\mathbf{A}_i = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & c(\alpha_i) & -s(\alpha_i) & 0 \\ 0 & s(\alpha_i) & c(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Tale matrice è costante.

In sintesi per portare la terna $i - 1$ sulla terna i è possibile comporre le precedenti roto-traslazioni nel seguente modo:

$${}^{i-1}\mathbf{A}_i(q_i) = {}^{i-1}\mathbf{A}_{i'}(q_i){}^{i'}\mathbf{A}_i = \begin{bmatrix} c(\theta_i) & -s(\theta_i)c(\alpha_i) & s(\theta_i)s(\alpha_i) & a_i c(\theta_i) \\ s(\theta_i) & c(\theta_i)c(\alpha_i) & -c(\theta_i)s(\alpha_i) & a_i s(\theta_i) \\ 0 & s(\alpha_i) & c(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Da notare che tale trasformazione omogenea è funzione solo di q_i e

$$q_i = \begin{cases} \theta_i & \text{per giunti rotoidali} \\ d_i & \text{per giunti prismatici} \end{cases}$$

mentre a_i e α_i sono costanti e dipendono dalla struttura del manipolatore.

Il calcolo della funzione cinematica diretta è effettuabile come:

$${}^0\mathbf{T}_{n-1}(\underline{q}) = {}^0\mathbf{A}_1 {}^1\mathbf{A}_2 \dots {}^{n-1}\mathbf{A}_n$$

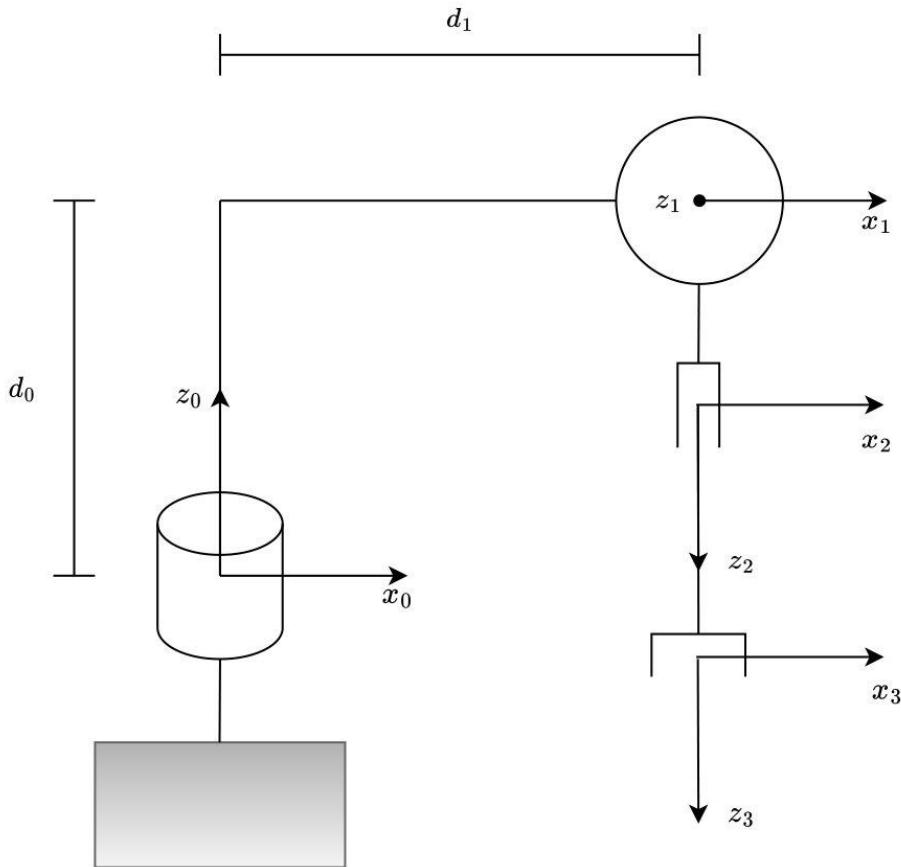
e quindi la posizione dell'E-E è ovviamente

$${}^0\underline{P}_{EE} = {}^0\mathbf{T}_{n-1}(\underline{q}) \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

3.2 Soluzione

Cinematica diretta completa

Manipolatore sferico di primo tipo



```
syms q1 q2 q3 d0 d1 real
DHsferico1tipo = [d1, pi/2, d0, q1;
                   0, pi/2, 0, q2;
                   0, 0, q3, 0]
```

```
DHsferico1tipo =
```

$$\begin{pmatrix} d_1 & \frac{\pi}{2} & d_0 & q_1 \\ 0 & \frac{\pi}{2} & 0 & q_2 \\ 0 & 0 & q_3 & 0 \end{pmatrix}$$

```
TsfericoList = cinDirDH(DHsferico1tipo);
T01 = TsfericoList{1}
```

```
T01 =
```

$$\begin{pmatrix} \cos(q_1) & 0 & \sin(q_1) & d_1 \cos(q_1) \\ \sin(q_1) & 0 & -\cos(q_1) & d_1 \sin(q_1) \\ 0 & 1 & 0 & d_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T12 = TsfericoList{2}
```

$$T12 = \begin{pmatrix} \cos(q_2) & 0 & \sin(q_2) & 0 \\ \sin(q_2) & 0 & -\cos(q_2) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T23 = TsfericoList{3}
```

$$T23 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

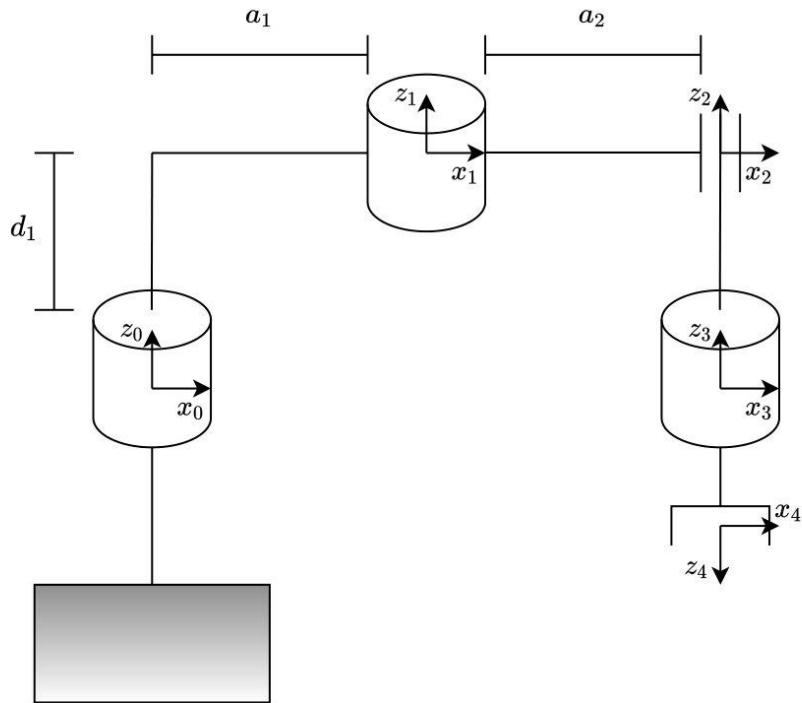
```
T03 = TsfericoList{4}
```

$$T03 = \begin{pmatrix} \cos(q_1) \cos(q_2) & \sin(q_1) & \cos(q_1) \sin(q_2) & \cos(q_1) \sigma_1 \\ \cos(q_2) \sin(q_1) & -\cos(q_1) & \sin(q_1) \sin(q_2) & \sin(q_1) \sigma_1 \\ \sin(q_2) & 0 & -\cos(q_2) & d_0 - q_3 \cos(q_2) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where

$$\sigma_1 = d_1 + q_3 \sin(q_2)$$

Manipolatore SCARA



```

syms q1 q2 q3 q4 d1 d4 a1 a2 real
DHscara = [a1, 0, d1, q1;
            a2, 0, 0, q2;
            0, 0, q3, 0;
            0, pi, d4, q4]

```

DHscara =

$$\begin{pmatrix} a_1 & 0 & d_1 & q_1 \\ a_2 & 0 & 0 & q_2 \\ 0 & 0 & q_3 & 0 \\ 0 & \pi & d_4 & q_4 \end{pmatrix}$$

```

TscaraList = cinDirDH(DHscara);
T01 = TscaraList{1}

```

T01 =

$$\begin{pmatrix} \cos(q_1) & -\sin(q_1) & 0 & a_1 \cos(q_1) \\ \sin(q_1) & \cos(q_1) & 0 & a_1 \sin(q_1) \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```

T12 = TscaraList{2}

```

T12 =

$$\begin{pmatrix} \cos(q_2) & -\sin(q_2) & 0 & a_2 \cos(q_2) \\ \sin(q_2) & \cos(q_2) & 0 & a_2 \sin(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T23 = TscaraList{3}
```

T23 =

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T34 = TscaraList{4}
```

T34 =

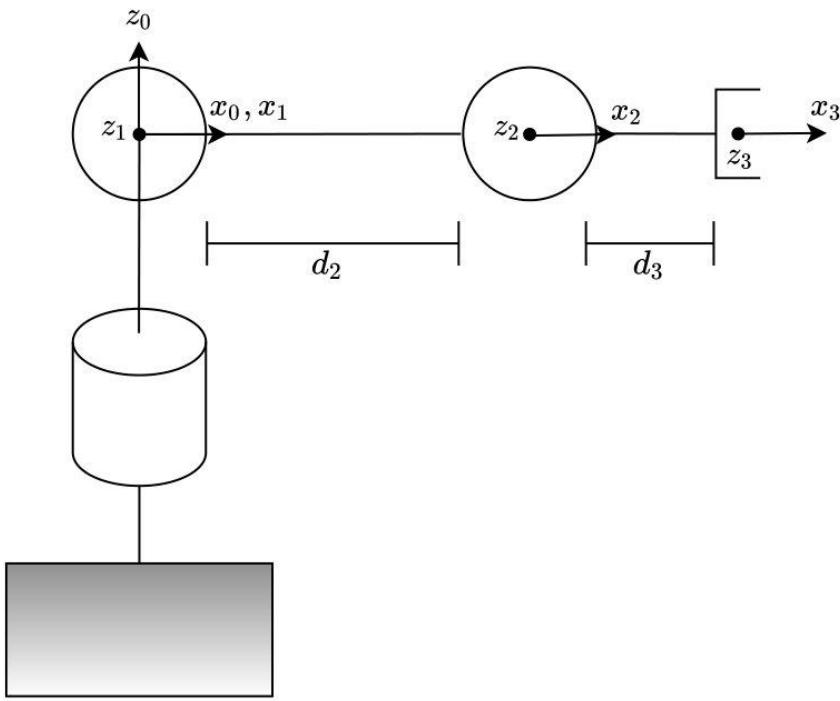
$$\begin{pmatrix} \cos(q_4) & \sin(q_4) & 0 & 0 \\ \sin(q_4) & -\cos(q_4) & 0 & 0 \\ 0 & 0 & -1 & d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T04 = TscaraList{5}
```

T04 =

$$\begin{pmatrix} \cos(q_1 + q_2 + q_4) & \sin(q_1 + q_2 + q_4) & 0 & a_2 \cos(q_1 + q_2) + a_1 \cos(q_1) \\ \sin(q_1 + q_2 + q_4) & -\cos(q_1 + q_2 + q_4) & 0 & a_2 \sin(q_1 + q_2) + a_1 \sin(q_1) \\ 0 & 0 & -1 & d_1 + d_4 + q_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Manipolatore antropomorfo



```
syms q1 q2 q3 d2 d3 real
DHant = [0, pi/2, 0, q1;
          d2, 0, 0, q2;
          d3, 0, 0, q3]
```

DHant =

$$\begin{pmatrix} 0 & \frac{\pi}{2} & 0 & q_1 \\ d_2 & 0 & 0 & q_2 \\ d_3 & 0 & 0 & q_3 \end{pmatrix}$$

```
TantList = cinDirDH(DHant);
T01 = TantList{1}
```

T01 =

$$\begin{pmatrix} \cos(q_1) & 0 & \sin(q_1) & 0 \\ \sin(q_1) & 0 & -\cos(q_1) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T12 = TantList{2}
```

T12 =

$$\begin{pmatrix} \cos(q_2) & -\sin(q_2) & 0 & d_2 \cos(q_2) \\ \sin(q_2) & \cos(q_2) & 0 & d_2 \sin(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T23 = TantList{3}
```

```
T23 =
```

$$\begin{pmatrix} \cos(q_3) & -\sin(q_3) & 0 & d_3 \cos(q_3) \\ \sin(q_3) & \cos(q_3) & 0 & d_3 \sin(q_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T03 = TantList{4}
```

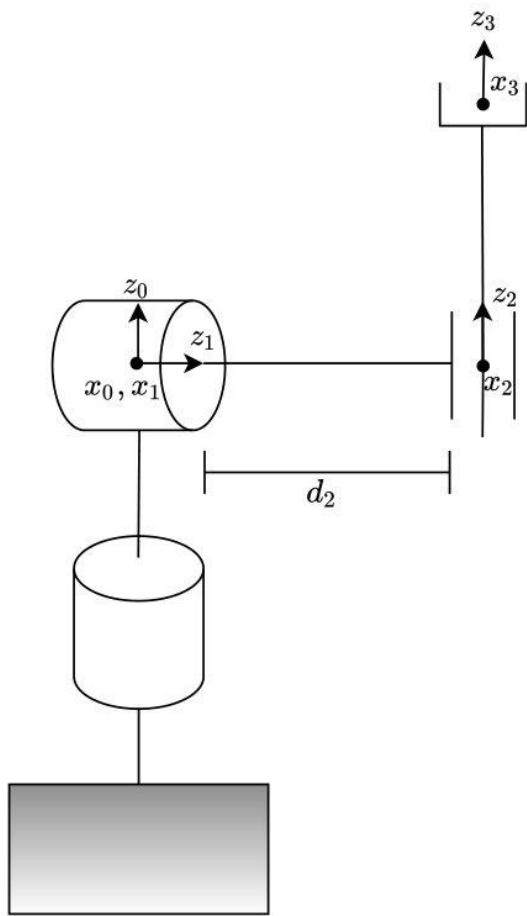
```
T03 =
```

$$\begin{pmatrix} \cos(q_2 + q_3) \cos(q_1) & -\sin(q_2 + q_3) \cos(q_1) & \sin(q_1) & \cos(q_1) \sigma_1 \\ \cos(q_2 + q_3) \sin(q_1) & -\sin(q_2 + q_3) \sin(q_1) & -\cos(q_1) & \sin(q_1) \sigma_1 \\ \sin(q_2 + q_3) & \cos(q_2 + q_3) & 0 & d_3 \sin(q_2 + q_3) + d_2 \sin(q_2) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where

$$\sigma_1 = d_3 \cos(q_2 + q_3) + d_2 \cos(q_2)$$

Manipolatore di Stanford (solo struttura portante = manipolatore sferico di II tipo)



```

syms q1 q2 q3 d2 real
DHsferico = [0, -pi/2, 0, q1;
              0, pi/2, d2, q2;
              0, 0, q3, 0]

```

```

DHsferico =

$$\begin{pmatrix} 0 & -\frac{\pi}{2} & 0 & q_1 \\ 0 & \frac{\pi}{2} & d_2 & q_2 \\ 0 & 0 & q_3 & 0 \end{pmatrix}$$


```

```

TsfericoList = cinDirDH(DHsferico);
T01 = TsfericoList{1}

```

```
T01 =
```

$$\begin{pmatrix} \cos(q_1) & 0 & -\sin(q_1) & 0 \\ \sin(q_1) & 0 & \cos(q_1) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T12 = TsfericoList{2}
```

T12 =

$$\begin{pmatrix} \cos(q_2) & 0 & \sin(q_2) & 0 \\ \sin(q_2) & 0 & -\cos(q_2) & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T23 = TsfericoList{3}
```

T23 =

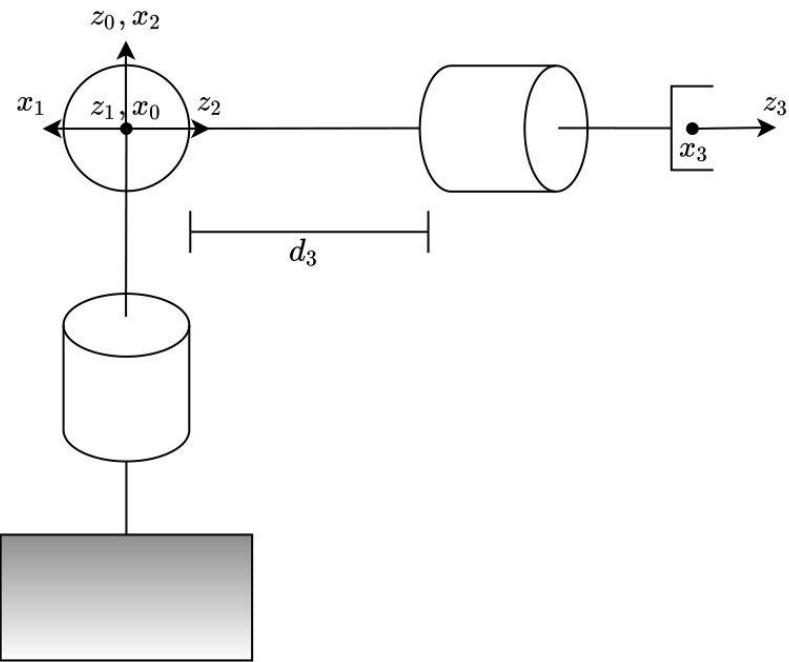
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T03 = TsfericoList{4}
```

T03 =

$$\begin{pmatrix} \cos(q_1) \cos(q_2) & -\sin(q_1) & \cos(q_1) \sin(q_2) & q_3 \cos(q_1) \sin(q_2) - d_2 \sin(q_1) \\ \cos(q_2) \sin(q_1) & \cos(q_1) & \sin(q_1) \sin(q_2) & d_2 \cos(q_1) + q_3 \sin(q_1) \sin(q_2) \\ -\sin(q_2) & 0 & \cos(q_2) & q_3 \cos(q_2) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Polso sferico



```

syms q1 q2 q3 d3 real
DHpolso = [0, -pi/2, 0, q1;
            0, pi/2, 0, q2;
            0, 0, d3, q3]

```

$$\text{DHpolso} = \begin{pmatrix} 0 & -\frac{\pi}{2} & 0 & q_1 \\ 0 & \frac{\pi}{2} & 0 & q_2 \\ 0 & 0 & d_3 & q_3 \end{pmatrix}$$

```

TpolsoList = cinDirDH(DHpolso);
T01 = TpolsoList{1}

```

$$T01 = \begin{pmatrix} \cos(q_1) & 0 & -\sin(q_1) & 0 \\ \sin(q_1) & 0 & \cos(q_1) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T12 = TpolsoList{2}
```

$$T12 = \begin{pmatrix} \cos(q_2) & 0 & \sin(q_2) & 0 \\ \sin(q_2) & 0 & -\cos(q_2) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T23 = TpolsoList{3}
```

T23 =

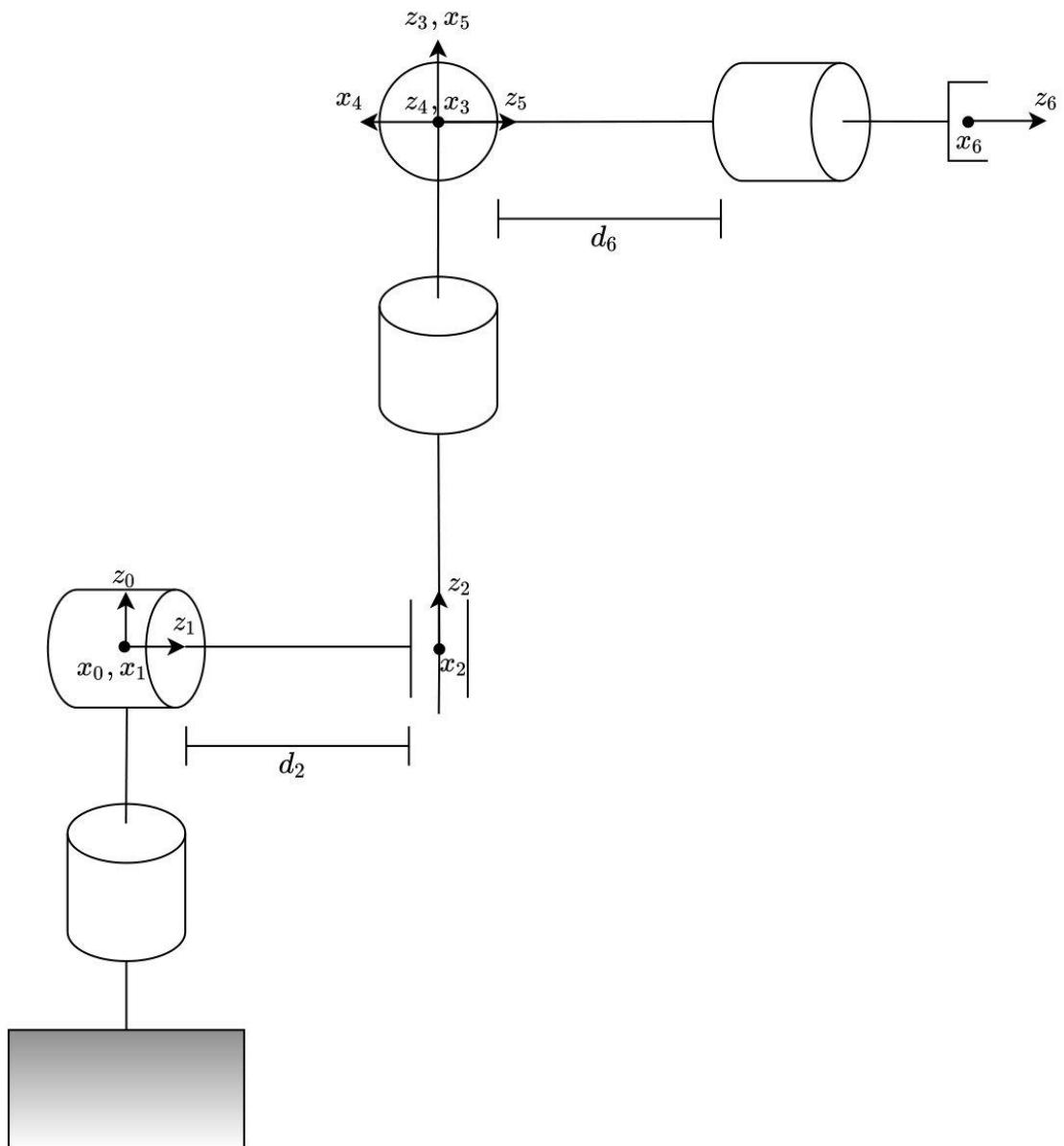
$$\begin{pmatrix} \cos(q_3) & -\sin(q_3) & 0 & 0 \\ \sin(q_3) & \cos(q_3) & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T03 = TpolsoList{4}
```

T03 =

$$\begin{pmatrix} \cos(q_1) \cos(q_2) \cos(q_3) - \sin(q_1) \sin(q_3) & -\cos(q_3) \sin(q_1) - \cos(q_1) \cos(q_2) \sin(q_3) & \cos(q_1) \sin(q_2) & d_3 \cos(q_2) \\ \cos(q_1) \sin(q_3) + \cos(q_2) \cos(q_3) \sin(q_1) & \cos(q_1) \cos(q_3) - \cos(q_2) \sin(q_1) \sin(q_3) & \sin(q_1) \sin(q_2) & d_3 \sin(q_2) \\ -\cos(q_3) \sin(q_2) & \sin(q_2) \sin(q_3) & \cos(q_2) & d \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Manipolatore di Stanford



```

syms q1 q2 q3 q4 q5 q6 d2 d6 real
DHstanford = [0, -pi/2, 0, q1;
              0, pi/2, d2, q2;
              0, 0, q3, 0;
              0, -pi/2, 0, q4;
              0, pi/2, 0, q5;
              0, 0, d6, q6]

```

DHstanford =

$$\begin{pmatrix} 0 & -\frac{\pi}{2} & 0 & q_1 \\ 0 & \frac{\pi}{2} & d_2 & q_2 \\ 0 & 0 & q_3 & 0 \\ 0 & -\frac{\pi}{2} & 0 & q_4 \\ 0 & \frac{\pi}{2} & 0 & q_5 \\ 0 & 0 & d_6 & q_6 \end{pmatrix}$$

```
TstanfordList = cinDirDH(DHstanford);
T01 = TstanfordList{1}
```

T01 =

$$\begin{pmatrix} \cos(q_1) & 0 & -\sin(q_1) & 0 \\ \sin(q_1) & 0 & \cos(q_1) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T12 = TstanfordList{2}
```

T12 =

$$\begin{pmatrix} \cos(q_2) & 0 & \sin(q_2) & 0 \\ \sin(q_2) & 0 & -\cos(q_2) & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T23 = TstanfordList{3}
```

T23 =

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T34 = TstanfordList{4}
```

T34 =

$$\begin{pmatrix} \cos(q_4) & 0 & -\sin(q_4) & 0 \\ \sin(q_4) & 0 & \cos(q_4) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T45 = TstanfordList{5}
```

T45 =

$$\begin{pmatrix} \cos(q_5) & 0 & \sin(q_5) & 0 \\ \sin(q_5) & 0 & -\cos(q_5) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

T56 = TstanfordList{6}

T56 =

$$\begin{pmatrix} \cos(q_6) & -\sin(q_6) & 0 & 0 \\ \sin(q_6) & \cos(q_6) & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

T06 = TstanfordList{7}

T06 =

$$\begin{pmatrix} -\cos(q_6) \sigma_8 - \sin(q_6) \sigma_4 & \sin(q_6) \sigma_8 - \cos(q_6) \sigma_4 & \sigma_5 - \sin(q_5) \sigma_{10} & q_3 \cos(q_1) \sin(q_2) \\ \cos(q_6) \sigma_6 + \sin(q_6) \sigma_3 & \cos(q_6) \sigma_3 - \sin(q_6) \sigma_6 & \sigma_7 & d_2 \cos(q_1) \\ \sin(q_2) \sin(q_4) \sin(q_6) - \cos(q_6) \sigma_2 & \sin(q_6) \sigma_2 + \cos(q_6) \sin(q_2) \sin(q_4) & \sigma_1 & \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

where

$$\sigma_1 = \cos(q_2) \cos(q_5) - \cos(q_4) \sin(q_2) \sin(q_5)$$

$$\sigma_2 = \cos(q_2) \sin(q_5) + \cos(q_4) \cos(q_5) \sin(q_2)$$

$$\sigma_3 = \cos(q_1) \cos(q_4) - \cos(q_2) \sin(q_1) \sin(q_4)$$

$$\sigma_4 = \cos(q_4) \sin(q_1) + \cos(q_1) \cos(q_2) \sin(q_4)$$

$$\sigma_5 = \cos(q_1) \cos(q_5) \sin(q_2)$$

$$\sigma_6 = \cos(q_5) \sigma_9 - \sin(q_1) \sin(q_2) \sin(q_5)$$

$$\sigma_7 = \sin(q_5) \sigma_9 + \cos(q_5) \sin(q_1) \sin(q_2)$$

$$\sigma_8 = \cos(q_5) \sigma_{10} + \cos(q_1) \sin(q_2) \sin(q_5)$$

$$\sigma_9 = \cos(q_1) \sin(q_4) + \cos(q_2) \cos(q_4) \sin(q_1)$$

$$\sigma_{10} = \sin(q_1) \sin(q_4) - \cos(q_1) \cos(q_2) \cos(q_4)$$

Lo stesso risultato poteva essere ottenuto concatenando la cinematica del manipolatore sferico con la cinematica del polso sferico ${}^0\mathbf{T}_6 = \underbrace{{}^0\mathbf{T}_3}_{\text{man. sferico}} \quad \underbrace{{}^0\mathbf{T}_3}_{\text{polso sferico}}$.

```
DHsferico = [0, -pi/2, 0, q1;
             0, pi/2, d2, q2;
             0, 0, q3, 0];
DHpolso = [0, -pi/2, 0, q4;
            0, pi/2, 0, q5;
            0, 0, d6, q6];

TsfericoList = cinDirDH(DHsferico);
TpolsoList = cinDirDH(DHpolso);

T03_manSferico = TsfericoList[4];
T03_polso = TpolsoList[4];
T06_2 = simplify(T03_manSferico*T03_polso);
isequal(expand(T06), expand(T06_2))

ans = Logical
1
```

```

function [mList] = cinDirDH(par)
    % cinDirDH - Calcola le matrici di trasformazione (i-1)-(i) e (0)-(n)
    %             data la matrice dei parametri DH.
    % L'ordine dei parametri e': a alpha d theta

    matDH = eye(4);
    N = size(par,1);
    for k = 1:N
        a = par(k, 1);
        alpha = par(k, 2);
        d = par(k, 3);
        theta = par(k, 4);

        cth = cos(theta);
        sth = sin(theta);
        cal = cos(alpha);
        sal = sin(alpha);

        A = [cth, -sth*cal, sth*sal, a*cth;
              sth, cth*cal, -cth*sal, a*sth;
              0, sal, cal, d;
              0, 0, 0, 1];

        matDH = matDH*A;
        mList{k} = simplify(A);
    end
    mList{k+1} = simplify(matDH);
end

```

Code 3.1: Funzione MATLAB per il calcolo della cinematica diretta

4. Esercitazione 4: cinematica inversa e spazi operativi

(4) *Cinematica inversa e spazi operativi primario e secondario per il robot planare RRR.*

4.1 Descrizione del problema

Il problema della cinematica inversa può essere espresso nel seguente modo: data la posa $\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{P} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}$ dell'organo terminale, si vuole trovare la configurazione dei giunti \underline{q} tale che ${}^0\mathbf{A}_n(\underline{q}) = \mathbf{T}$ realizza la posa assegnata.

Definition 4.1.1 (Spazio operativo primario). Si chiama spazio operativo primario l'insieme $WS_1 = \{p \in \mathbb{R}^3 : \exists q : f(q) = p \text{ per un qualche orientamento}\}$

Tale insieme rappresenta tutti i punti raggiungibili dall'organo terminale senza badare all'orientamento.

Property 4.1.1. Se $p \notin WS_1 \Rightarrow \nexists q$ che risolve il problema inverso.

Property 4.1.2. Se $p \in WS_1 \Rightarrow \exists q$ che risolve il problema inverso per un certo orientamento (descritto da una matrice \mathbf{R}).

Definition 4.1.2 (Spazio operativo secondario (o destro)). Si chiama spazio operativo secondario (o destro) l'insieme $WS_2 = \left\{ p \in \mathbb{R}^3 : \forall \mathbf{R} \in SO(3) \exists q : {}^n\mathbf{A}_0(q) = \begin{bmatrix} \mathbf{R} & \mathbf{P} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \right\}$.

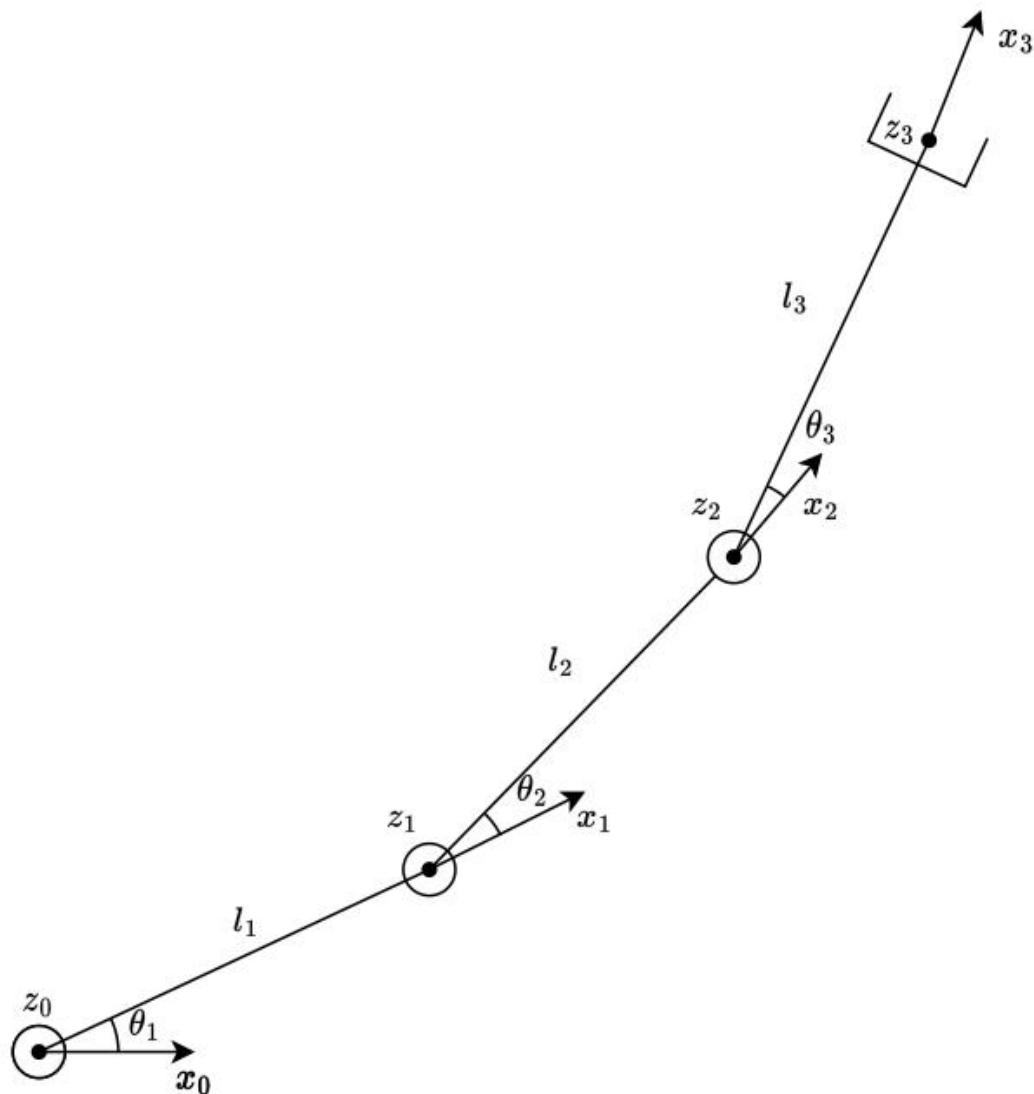
Tale insieme rappresenta tutti i punti raggiungibili dall'organo terminale con qualsiasi orientamento.

Property 4.1.3. Lo spazio operativo destro è contenuto nello spazio operativo primario: $WS_2 \subseteq WS_1$.

4.2 Soluzione: cinematica inversa

Conviene prima di tutto calcolare la cinematica diretta.

Cinematica diretta robot RRR planare



```

syms q1 q2 q3 l1 l2 l3 real
DHRRRplanare = [l1, 0, 0, q1;
                 l2, 0, 0, q2;
                 l3, 0, 0, q3]

```

DHRRRplanare =

$$\begin{pmatrix} l_1 & 0 & 0 & q_1 \\ l_2 & 0 & 0 & q_2 \\ l_3 & 0 & 0 & q_3 \end{pmatrix}$$

```

TRRRplanareList = cinDirDH(DHRRRplanare);
T01 = TRRRplanareList{1}

```

T01 =

$$\begin{pmatrix} \cos(q_1) & -\sin(q_1) & 0 & l_1 \cos(q_1) \\ \sin(q_1) & \cos(q_1) & 0 & l_1 \sin(q_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T12 = TRRRplanareList{2}
```

T12 =

$$\begin{pmatrix} \cos(q_2) & -\sin(q_2) & 0 & l_2 \cos(q_2) \\ \sin(q_2) & \cos(q_2) & 0 & l_2 \sin(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T23 = TRRRplanareList{3}
```

T23 =

$$\begin{pmatrix} \cos(q_3) & -\sin(q_3) & 0 & l_3 \cos(q_3) \\ \sin(q_3) & \cos(q_3) & 0 & l_3 \sin(q_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T03 = TRRRplanareList{4}
```

T03 =

$$\begin{pmatrix} \sigma_2 & -\sigma_1 & 0 & l_2 \cos(q_1 + q_2) + l_1 \cos(q_1) + l_3 \sigma_2 \\ \sigma_1 & \sigma_2 & 0 & l_2 \sin(q_1 + q_2) + l_1 \sin(q_1) + l_3 \sigma_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where

$$\sigma_1 = \sin(q_1 + q_2 + q_3)$$

$$\sigma_2 = \cos(q_1 + q_2 + q_3)$$

La cinematica diretta del robot RRR planare è quindi

$${}^0\mathbf{T}_3 = \begin{bmatrix} c_{123} & -s_{123} & 0 & l_1c_1 + l_2c_{12} + l_3c_{123} \\ s_{123} & c_{123} & 0 & l_1s_1 + l_2s_{12} + l_3s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Al fine di individuare le variabili di giunto q_1, q_2 e q_3 conviene però specificare la posizione e l'orientamento in termini di un numero minimo di parametri. La scelta più ovvia risulta essere

$$\begin{bmatrix} x \\ y \\ \phi \end{bmatrix} = \begin{bmatrix} l_1c_1 + l_2c_{12} + l_3c_{123} \\ l_1s_1 + l_2s_{12} + l_3s_{123} \\ q_1 + q_2 + q_3 \end{bmatrix}$$

È ora possibile notare che

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{R}_1(q_1 + q_2 + q_3) \begin{bmatrix} l_3 \\ 0 \end{bmatrix} + \mathbf{R}_2(q_1 + q_2) \begin{bmatrix} l_2 \\ 0 \end{bmatrix} + \mathbf{R}_3(q_1) \begin{bmatrix} l_1 \\ 0 \end{bmatrix}$$

dove $\mathbf{R}_1(q_1 + q_2 + q_3) = \begin{bmatrix} c_{123} & -s_{123} \\ s_{123} & c_{123} \end{bmatrix}$, $\mathbf{R}_2(q_1 + q_2) = \begin{bmatrix} c_{12} & -s_{12} \\ s_{12} & c_{12} \end{bmatrix}$ e $\mathbf{R}_2(q_1) = \begin{bmatrix} c_1 & -s_1 \\ s_1 & c_1 \end{bmatrix}$.

Variabile q_2

Sfruttando la precedente relazione e ricordando che la cinematica diretta del robot RR planare è $\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \mathbf{R}_2(q_1 + q_2) \begin{bmatrix} l_2 \\ 0 \end{bmatrix} + \mathbf{R}_3(q_1) \begin{bmatrix} l_1 \\ 0 \end{bmatrix}$ sotto la condizione $|l_1 - l_2| \leq \sqrt{\hat{x}^2 + \hat{y}^2} \leq l_1 + l_2$ è possibile dire che

$$\begin{aligned} \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} &= \begin{bmatrix} x \\ y \end{bmatrix} - \mathbf{R}_1(q_1 + q_2 + q_3) \begin{bmatrix} l_3 \\ 0 \end{bmatrix} = \mathbf{R}_2(q_1 + q_2) \begin{bmatrix} l_2 \\ 0 \end{bmatrix} + \mathbf{R}_3(q_1) \begin{bmatrix} l_1 \\ 0 \end{bmatrix} = \\ &= \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} l_3c_{123} \\ l_3s_{123} \end{bmatrix} = \begin{bmatrix} l_1c_1 + l_2c_{12} \\ l_1s_1 + l_2s_{12} \end{bmatrix} \end{aligned} \tag{4.1}$$

e quindi quadrando e sommando \hat{x} e \hat{y} si ottiene

$$\begin{aligned} \hat{x}^2 + \hat{y}^2 &= (l_1c_1 + l_2c_{12})^2 + (l_1s_1 + l_2s_{12})^2 = \\ &= l_1^2c_1^2 + 2l_1l_2c_1c_{12} + l_2^2c_{12}^2 + l_1^2s_1^2 + 2l_1l_2s_1s_{12} + l_2^2s_{12}^2 = \\ &= l_1^2(c_1^2 + s_1^2) + l_2^2(c_{12}^2 + s_{12}^2) + 2l_1l_2(c_1c_{12} + s_1s_{12}) = \\ &= l_1^2 + l_2^2 + 2l_1l_2 \cos(q_1 - q_1 + q_2) = \\ &= l_1^2 + l_2^2 + 2l_1l_2c_2 \end{aligned}$$

da cui è possibile dire che

$$c_2 = \frac{\hat{x}^2 + \hat{y}^2 - l_1^2 - l_2^2}{2l_1l_2} \in [-1, 1]$$

È quindi possibile porre

$$s_2 = \pm \sqrt{1 - c_2^2}$$

dove il segno positivo individua la postura del gomito (+ per il gomito “basso” e - per il gomito “alto”).

Si ha quindi che

$$\begin{cases} c_2 = \frac{\hat{x}^2 + \hat{y}^2 - l_1^2 - l_2^2}{2l_1l_2} \\ s_2 = \pm \sqrt{1 - c_2^2} \end{cases} \Rightarrow q_2 = \text{atan2}(s_2, c_2)$$

Variabile q_1

Una volta ricavato q_2 è possibile sostituirlo nella 4.1 per ricavare q_1 :

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} l_1c_1 + l_2c_{12} \\ l_1s_1 + l_2s_{12} \end{bmatrix} = \begin{bmatrix} l_1c_1 + l_2(c_1c_2 - s_1s_2) \\ l_1s_1 + l_2(s_1c_2 + c_1s_2) \end{bmatrix} = \underbrace{\begin{bmatrix} l_1 + l_2c_2 & -l_2s_2 \\ l_2s_2 & l_1 + l_2c_2 \end{bmatrix}}_{\boldsymbol{\pi}(q_2)} \begin{bmatrix} c_1 \\ s_1 \end{bmatrix}$$

e quindi

$$\begin{bmatrix} c_1 \\ s_1 \end{bmatrix} = \boldsymbol{\pi}^{-1}(q_2) \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}$$

ma $\boldsymbol{\pi}(q_2)$ è invertibile $\Leftrightarrow \det \boldsymbol{\pi}(q_2) = l_1^2 + l_2^2 + 2l_1l_2c_2 \neq 0$ ovvero $\Leftrightarrow \neg(l_1 = l_2 \wedge c_2 = -1)$.

Al fine di determinare c_1 e s_1 conviene tuttavia risolvere il sistema

$$\begin{cases} \hat{x} = l_1c_1 + l_2c_{12} \\ \hat{y} = l_1s_1 + l_2s_{12} \end{cases}$$

Ricordando che

$$\begin{aligned} \cos(\alpha + \beta) &= \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta) \\ \sin(\alpha + \beta) &= \sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta) \end{aligned}$$

la precedente relazione diventa

$$\begin{cases} \hat{x} = l_1c_1 + l_2(c_1c_2 - s_1s_2) \\ \hat{y} = l_1s_1 + l_2(s_1c_2 + c_1s_2) \end{cases}$$

Dalla prima equazione è possibile ricavare

$$c_1 = \frac{\hat{x} + l_2 s_1 s_2}{l_1 + l_2 c_2} \quad (4.2)$$

che sostituita nella seconda permette di determinare s_1 :

$$\hat{y} = l_1 s_1 + l_2 \left(s_1 c_2 + \left(\frac{\hat{x} + l_2 s_1 s_2}{l_1 + l_2 c_2} \right) s_2 \right) = s_1 \left(l_1 + l_2 c_2 + \frac{l_2^2 s_2^2}{l_1 + l_2 c_2} \right) + \frac{\hat{x} l_2 s_2}{l_1 + l_2 c_2}$$

e quindi

$$s_1 = \frac{\hat{y} - \frac{\hat{x} l_2 s_2}{l_1 + l_2 c_2}}{l_1 + l_2 c_2 + \frac{l_2^2 s_2^2}{l_1 + l_2 c_2}} = \frac{\hat{y}(l_1 + l_2 c_2) - \hat{x} l_2 s_2}{l_1^2 + 2l_1 l_2 c_2 + l_2^2 c_2^2 + l_2^2 s_2^2} = \frac{\hat{y}(l_1 + l_2 c_2) - \hat{x} l_2 s_2}{l_1^2 + 2l_1 l_2 c_2 + l_2^2}$$

Ma ricordando che $\hat{x}^2 + \hat{y}^2 = l_1^2 + l_2^2 + 2l_1 l_2 c_2$ si ottiene

$$s_1 = \frac{\hat{y}(l_1 + l_2 c_2) - \hat{x} l_2 s_2}{\hat{x}^2 + \hat{y}^2}$$

e sostituendo s_1 nella 4.2 si ottiene

$$c_1 = \frac{\hat{x}(l_1 + l_2 c_2) + \hat{y} l_2 s_2}{\hat{x}^2 + \hat{y}^2}$$

Infine l'angolo q_1 è dato da

$$\begin{cases} c_1 = \frac{\hat{x}(l_1 + l_2 c_2) + \hat{y} l_2 s_2}{\hat{x}^2 + \hat{y}^2} \\ s_1 = \frac{\hat{y}(l_1 + l_2 c_2) - \hat{x} l_2 s_2}{\hat{x}^2 + \hat{y}^2} \end{cases} \Rightarrow q_1 = \text{atan2}(s_1, c_1)$$

Variabile q_3

Avendo ricavato q_1 e q_2 , l'angolo q_3 è ricavabile dalla cinematica diretta

$$q_3 = \phi - q_1 - q_2$$

4.2.1 Soluzione: spazio operativo

Al fine di individuare lo spazio di lavoro del robot RRR planare si considera $l_1 = l_2 = l_3 = 1$.

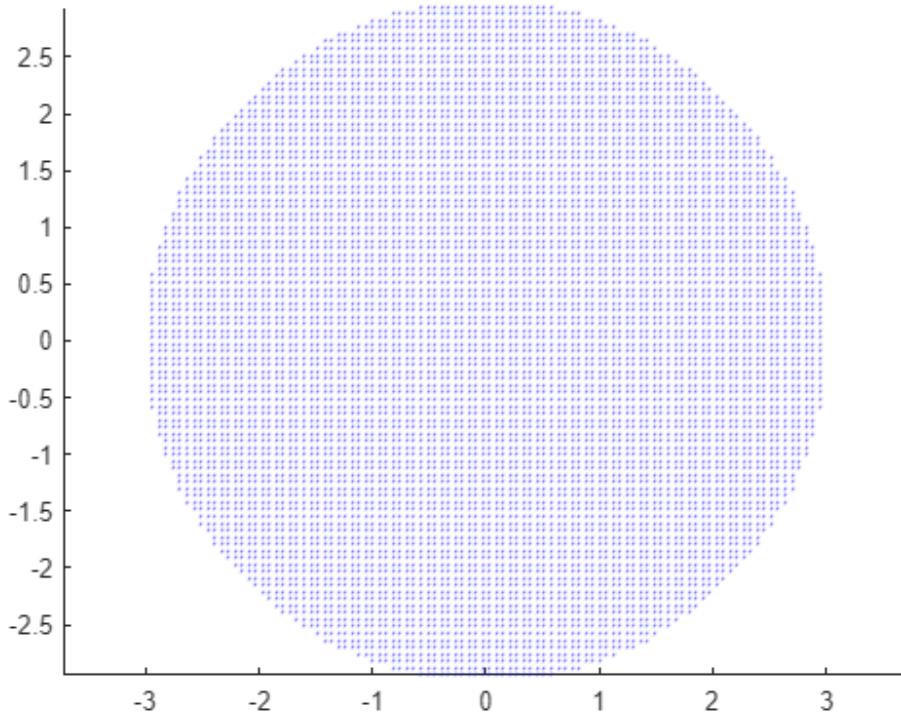
Spazio operativo primario e secondario del robot RRR planare

Spazio operativo primario

Lo spazio operativo primario è $WS_1 = \{p \in \mathbb{R}^2 : \|p\| \leq 3l\}$.

```
% parametri cerchio
center1 = [0, 0];
radius1 = 1+1+1;

figure
drawWS(center1, radius1, 'b');
axis equal
```

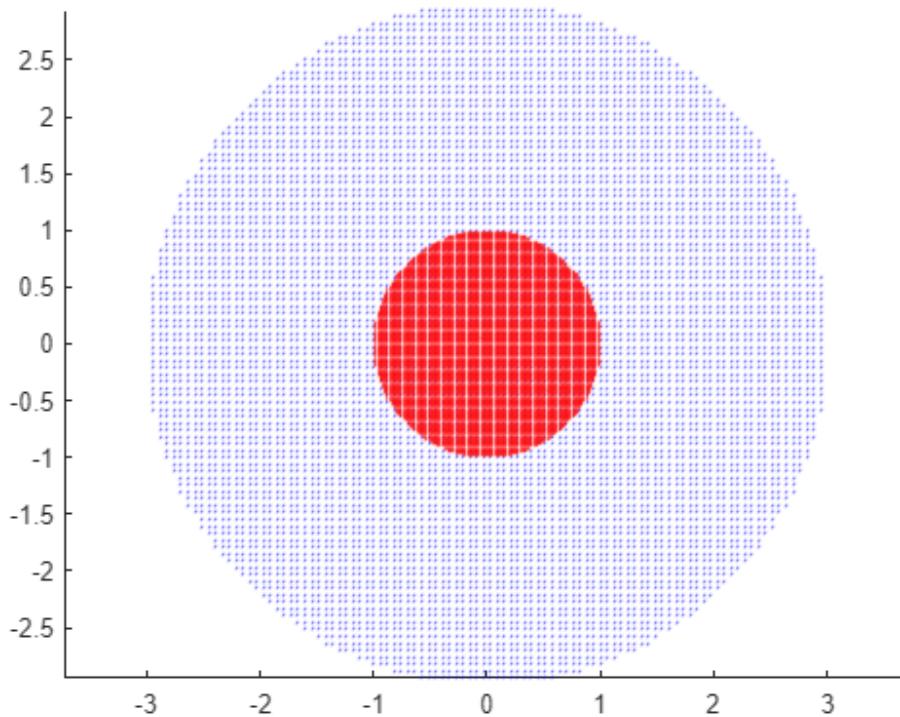


Spazio operativo secondario

Lo spazio operativo secondario è $WS_2 = \{p \in \mathbb{R}^2 : \|p\| \leq l\}$.

```
center2 = [0, 0];
radius2 = 1;

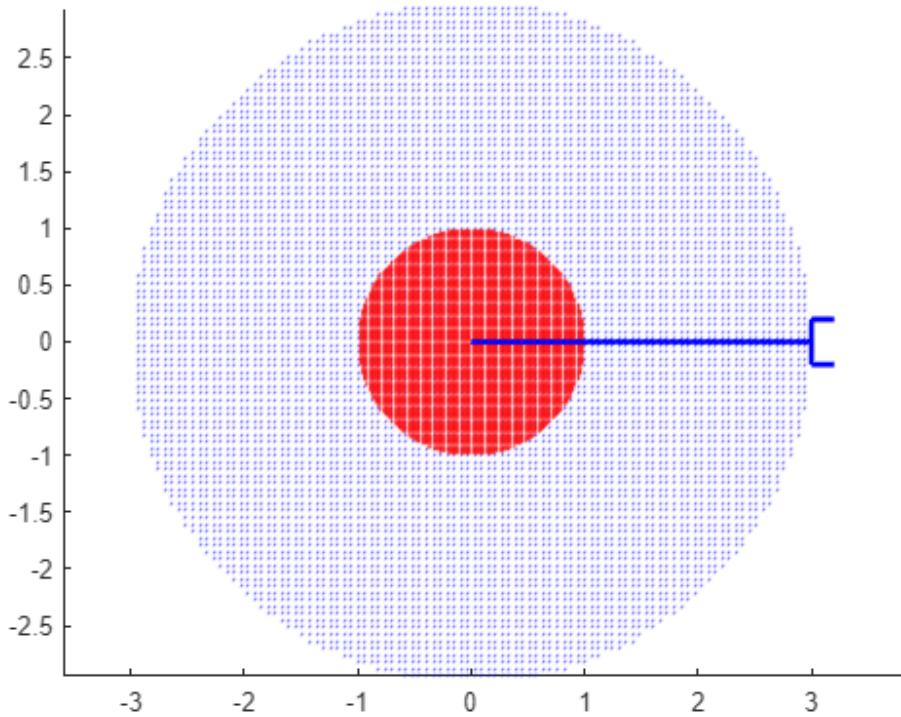
figure
drawWS(center1, radius1, 'b')
hold on
drawWS(center2, radius2, 'r')
```



Alcune configurazioni

Configurazione limite per WS_1

```
figure
drawWS(center1, radius1, 'b')
hold on
drawWS(center2, radius2, 'r')
drawPlanarRobot([1,1,1],[0,0,0])
```



Esempio in WS_2

Si vuole posizionare l'end-effector in $[p_x, p_y] = [0.1, -0.5] \in WS_2$ con orientamento $\phi = \frac{\pi}{2}$. Per fare ciò è necessario ricavare le variabili di giunto $[q_1, q_2, q_3]$.

Dalla cinematica inversa sappiamo che

$$q_2 = \text{atan}2(s_2, c_2) \text{ dove } \begin{cases} c_2 = \frac{\hat{x}^2 + \hat{y}^2 - l_1^2 - l_2^2}{2l_1l_2} = \frac{\hat{x}^2 + \hat{y}^2 - 2}{2} = \frac{(p_x^2 + p_y^2 + 1 - 2(p_x c_\phi + p_y s_\phi)) - 2}{2} \\ s_2 = \sqrt{1 - c_2^2} \end{cases}$$

dato che $\hat{x}^2 + \hat{y}^2 = p_x^2 + p_y^2 + 1 - 2(p_x c_\phi + p_y s_\phi)$ essendo $\hat{x} = p_x - l_3 c_\phi$ e $\hat{y} = p_y - l_3 s_\phi$.

```
% q2
px = 0.1; % x desiderata
py = -0.5; % y desiderata
phi = pi/2; % orientamento desiderato
c2 = (px^2+py^2+1-2*(px*cos(phi)+py*sin(phi))-2)/2;
s2 = sqrt(1-c2^2);
q2 = atan2(s2, c2)
```

q2 = 1.4404

Dalla cinematica inversa sappiamo che

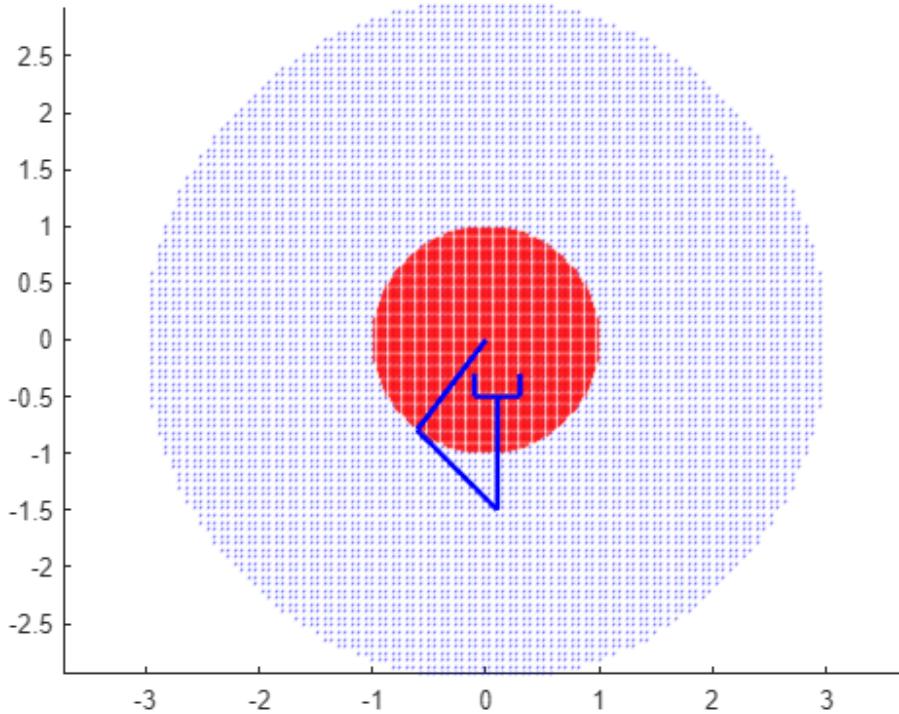
$$q_1 = \text{atan}2(s_1, c_1) \text{ dove } \begin{cases} c_1 = \frac{\hat{x}(l_1 + l_2c_2) - \hat{y}l_2s_2}{\hat{x}^2 + \hat{y}^2} = \frac{(p_x - l_3c_\phi)(l_1 + l_2c_2) + (p_y - l_3s_\phi)l_2s_2}{p_x^2 + p_y^2 + 1 - 2(p_xc_\phi + p_ys_\phi)} = \\ s_1 = \frac{\hat{y}(l_1 + l_2c_2) - \hat{x}l_2s_2}{\hat{x}^2 + \hat{y}^2} = \frac{(p_y - l_3s_\phi)(l_1 + l_2c_2) - (p_x - l_3c_\phi)l_2s_2}{p_x^2 + p_y^2 + 1 - 2(p_xc_\phi + p_ys_\phi)} \end{cases}$$

dato che $\hat{x}^2 + \hat{y}^2 = p_x^2 + p_y^2 + 1 - 2(p_xc_\phi + p_ys_\phi)$ essendo $\hat{x} = p_x - l_3c_\phi$ e $\hat{y} = p_y - l_3s_\phi$.

```
% q1
c1 = ((px-cos(phi))*(1+c2)+(py-sin(phi))*s2)/(px^2+py^2+1-2*(px*cos(phi)+py*sin(phi)));
s1 = ((py-sin(phi))*(1+c2)-(px-cos(phi))*s2)/(px^2+py^2+1-2*(px*cos(phi)+py*sin(phi)));
q1 = atan2(s1, c1);
% q3
q3 = phi-q1-q2
```

q3 = 2.3548

```
figure
drawWS(center1, radius1, 'b')
hold on
drawWS(center2, radius2, 'r')
drawPlanarRobot([1,1,1],[q1,q2,q3])
```



Si vuole ora posizionare l'end-effector nella stessa posizione $[p_x, p_y] = [0.1, -0.5] \in WS_2$ ma con orientamento

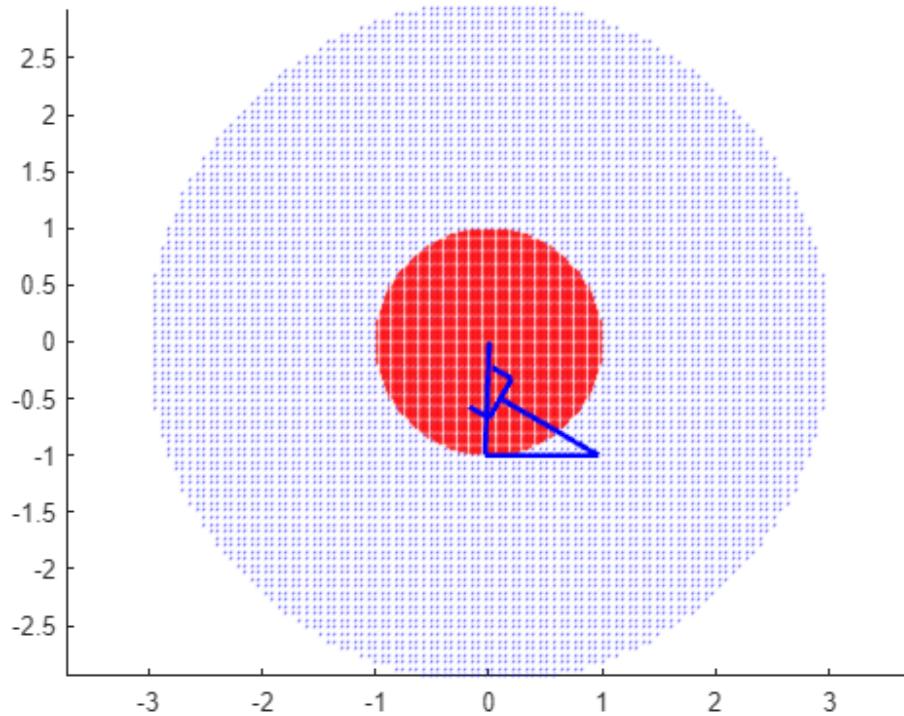
$$\phi = \frac{5\pi}{6}.$$

```

% q2
px = 0.1; % x desiderata
py = -0.5; % y desiderata
phi = 5*pi/6; % orientamento desiderato
c2 = (px^2+py^2+1-2*(px*cos(phi)+py*sin(phi))-2)/2;
s2 = sqrt(1-c2^2);
q2 = atan2(s2, c2);
% q1
c1 = ((px-cos(phi))*(1+c2)+(py-sin(phi))*s2)/(px^2+py^2+1-2*(px*cos(phi)+py*sin(phi)));
s1 = ((py-sin(phi))*(1+c2)-(px-cos(phi))*s2)/(px^2+py^2+1-2*(px*cos(phi)+py*sin(phi)));
q1 = atan2(s1, c1);
% q3
q3 = phi-q1-q2;

figure
drawWS(center1, radius1, 'b')
hold on
drawWS(center2, radius2, 'r')
drawPlanarRobot([1,1,1],[q1,q2,q3])

```



```

function [] = drawPlanarRobot(links, joints)
    % disegna un robot planare note la lunghezza dei link e l'angolo dei
    % giunti

    x = 0;
    y = 0;
    joint = 0;
    N = size(links, 2);
    for i= 1:N
        hold on
        joint = joint + joints(i);
        deltaX = links(i)*cos(joint);
        deltaY = links(i)*sin(joint);
        plot([x, x+deltaX], [y, y+deltaY], 'b-', 'LineWidth', 2);
        x = x+deltaX;
        y = y+deltaY;
    end

    % disegna end effector
    ang = pi/2-joint; % angolo alterno interno
    dim = 0.2; % dimensione EE
    ee1 = [x-dim*cos(ang), y+dim*sin(ang)];
    ee2 = [x+dim*cos(ang), y-dim*sin(ang)];
    plot([ee1(1), ee2(1)], [ee1(2), ee2(2)], 'b-', 'LineWidth', 2);
    ee11 = [ee1(1)+dim*cos(joint), ee1(2)+dim*sin(joint)];
    ee22 = [ee2(1)+dim*cos(joint), ee2(2)+dim*sin(joint)];
    plot([ee1(1), ee11(1)], [ee1(2), ee11(2)], 'b-', 'LineWidth', 2);
    plot([ee2(1), ee22(1)], [ee2(2), ee22(2)], 'b-', 'LineWidth', 2);
end

```

Code 4.1: Funzione MATLAB per disegnare robot planari

```

function [] = drawWS(center, radius, color)
centerX = center(1);
centerY = center(2);
dotSize = 2;

% punti quadrato
x = linspace(centerX - radius, centerX + radius, 100);
y = linspace(centerY - radius, centerY + radius, 100);
[X, Y] = meshgrid(x, y);

% distanza punti quadrato dal centro
dist = sqrt((X - centerX).^2 + (Y - centerY).^2);

```

```
% seleziona punti interni alla circonferenza  
pointsInsideCircle = dist <= radius;  
  
% Flatten the selected points into column vectors  
X_WS = X(pointsInsideCircle);  
Y_WS = Y(pointsInsideCircle);  
  
% Create the scatter plot with dotted pattern  
c = scatter(X_WS(:,1), Y_WS(:,1), dotSize, 'filled', 'MarkerFaceColor',  
color, 'MarkerEdgeColor', 'none');  
alpha(c, 0.5);  
axis equal;  
end
```

Code 4.2: Funzione MATLAB per disegnare lo spazio di lavoro

5. Esercitazione 5: cinematica inversa tramite disaccoppiamento

(5) *Cinematica inversa tramite disaccoppiamento polso-struttura portante e discussione delle configurazioni singolari per il manipolatore di Standford con polso sferico.*

5.1 Descrizione del problema

5.1.1 Proprietà di disaccoppiamento polso-struttura portante

Il polso sferico permette di disaccoppiare la cinematica della struttura portante da quella di orientamento, consentendo di pervenire ad una procedura sistematica per il calcolo di una soluzione analitica del problema inverso.

Affinché tale procedura sia percorribile deve valere la proprietà di disaccoppiamento polso-struttura portante.

Si ricorda che il problema inverso prevede, data la posa desiderata dell'organo terminale $\mathbf{T} = \begin{bmatrix} \mathbf{R} & \underline{d}_{06} \\ \mathbf{0}_{3 \times 1} & 1 \end{bmatrix}$, di determinare le variabili di giunto \underline{q} che permettono di raggiungere tale posa.

Tale problema può risultare particolarmente complesso. È però possibile semplificarlo se vale la seguente proprietà:

Property 5.1.1 (Disaccoppiamento polso struttura portante). $\exists \underline{d}_0, \underline{d}_1 \in \mathbb{R}^3 : \underline{d}_{36} = {}^3\mathbf{R}_6 \underline{d}_1 + \underline{d}_0$, dove ${}^3\mathbf{T}_6 = \begin{bmatrix} {}^3\mathbf{R}_6 & \underline{d}_{36} \\ \mathbf{0}_{3 \times 1} & 1 \end{bmatrix}$. La traslazione \underline{d}_{36} che porta la terna

6 sulla terna 3 è data da una traslazione \underline{d}_1 espressa nella terna 6 e un offset \underline{d}_0 espresso nella terna 3.

Dato che $\underline{d}_{06} = {}^0\mathbf{R}_3 \underline{d}_{36} + \underline{d}_{03}$ è possibile riscrivere il tutto come:

$$\underline{d}_{06} = {}^0\mathbf{R}_3 \underline{d}_{36} + \underline{d}_{03} = {}^0\mathbf{R}_3 \underbrace{({}^3\mathbf{R}_6 \underline{d}_1 + \underline{d}_0)}_{\text{disaccoppiamento polso}} + \underline{d}_{03} = {}^0\mathbf{R}_6 \underline{d}_1 + {}^0\mathbf{R}_3 \underline{d}_0 + \underline{d}_{03}$$

E quindi

$$\underline{d}_{06} = \mathbf{R} \underline{d}_1 + {}^0\mathbf{R}_3 \underline{d}_0 + \underline{d}_{03}$$

Ma \underline{d}_{06} e \mathbf{R} sono noti essendo la posa desiderata. Si può riscrivere

$$\underline{d}_{06} - \mathbf{R} \underline{d}_1 = {}^0\mathbf{R}_3 \underline{d}_0 + \underline{d}_{03}$$

e chiamando $\hat{\underline{d}} = \underline{d}_{06} - \mathbf{R} \underline{d}_1$ si ha la seguente equazione

$$\hat{\underline{d}} = {}^0\mathbf{R}_3 \underline{d}_0 + \underline{d}_{03}$$

con incognite ${}^0\mathbf{R}_3$ e \underline{d}_{03} .

Risolvendo tale equazione è quindi possibile trovare q_1 , q_2 e q_3 e quindi anche ${}^0\mathbf{R}_3$.

Notando che $\mathbf{R} = {}^0\mathbf{R}_3 {}^3\mathbf{R}_6$ è possibile dire che

$${}^3\mathbf{R}_6 = {}^0\mathbf{R}_3 {}^T \mathbf{R}$$

da cui si ricavano q_4 , q_5 e q_6 .

5.1.2 Manipolatore di stanford: cinematica diretta

Il manipolatore di Stanford risulta composto da un manipolatore sferico e da un polso sferico.

Manipolatore sferico RRP

Il manipolatore sferico RRP è composto da due giunti rotoidali e uno prismatico. Il termine “sferico” si riferisce alla forma sferica dello spazio di lavoro del manipolatore, cioè la regione dello spazio tridimensionale in cui il robot può muoversi e manipolare oggetti.

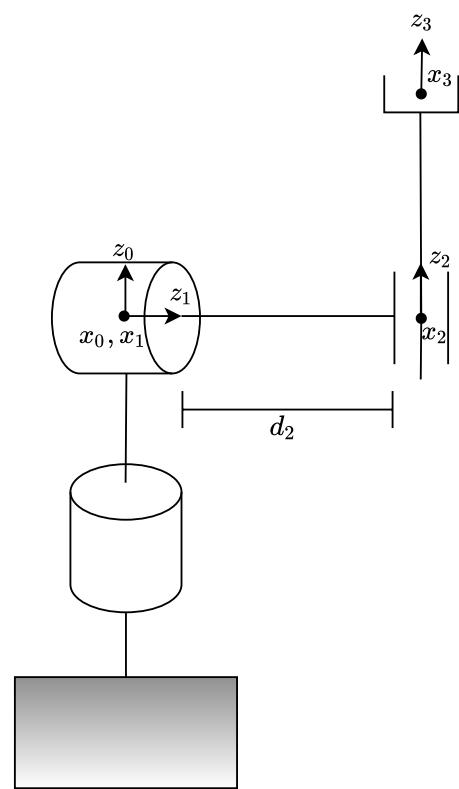


Figura 5.1: Manipolatore sferico RRP

Braccio	a_i	α_i	d_i	θ_i
1	0	$-\frac{\pi}{2}$	0	q_1
2	0	$\frac{\pi}{2}$	d_2	q_2
3	0	0	q_3	0

Tabella 5.1: Parametri DH per il manipolatore sferico RRP

Le matrici di trasformazione omogenea per i singoli giunti sono:

$${}^0\mathbf{A}_1(q_1) = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1\mathbf{A}_2(q_2) = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2\mathbf{A}_3(q_3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

E quindi

$${}^0\mathbf{T}_3(\underline{q}) = {}^0\mathbf{A}_1(q_1){}^1\mathbf{A}_2(q_2){}^2\mathbf{A}_3(q_3) = \begin{bmatrix} c_1c_2 & -s_1 & c_1s_2 & c_1s_2q_3 - s_1d_2 \\ s_1c_2 & c_1 & s_1s_2 & s_1s_2q_3 + c_1d_2 \\ -s_2 & 0 & c_2 & c_2q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Polso sferico RRR

Il polso sferico RRR è composto da tre giunti rotoidali. Un polso viene detto sferico se è costituito da tre giunti rotoidali i cui assi si intersecano in un punto comune, detto centro del polso.

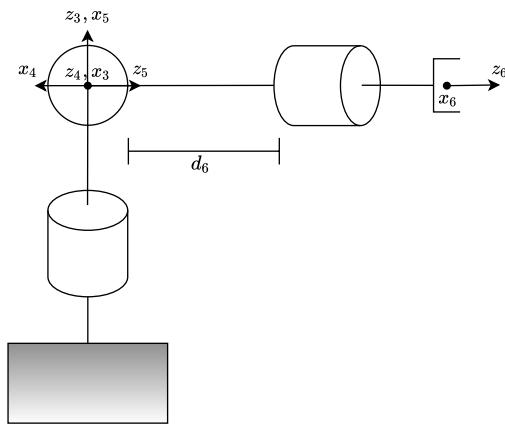


Figura 5.2: Polso sferico RRR

Braccio	a_i	α_i	d_i	θ_i
1	0	$-\frac{\pi}{2}$	0	q_4
2	0	$\frac{\pi}{2}$	0	q_5
3	0	0	d_6	q_6

Tabella 5.2: Parametri DH per il polso sferico RRR

Le matrici di trasformazione omogenea per i singoli giunti sono:

$$\begin{aligned} {}^3\mathbf{A}_4(q_4) &= \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^4\mathbf{A}_5(q_5) &= \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^5\mathbf{A}_6(q_6) &= \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

E quindi

$${}^3\mathbf{T}_6(\underline{q}) = {}^3\mathbf{A}_4(q_4){}^4\mathbf{A}_5(q_5){}^5\mathbf{A}_6(q_6) = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5c_6 - s_4c_6 & c_4s_5 & c_4s_5d_6 \\ s_4c_5c_6 - c_4s_6 & -s_4c_5s_6 - s_4c_6 & c_4s_5 & c_4s_5d_6 \\ -s_5c_6 & s_5s_6 & c_5 & c_5d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Manipolatore sferico + polso sferico

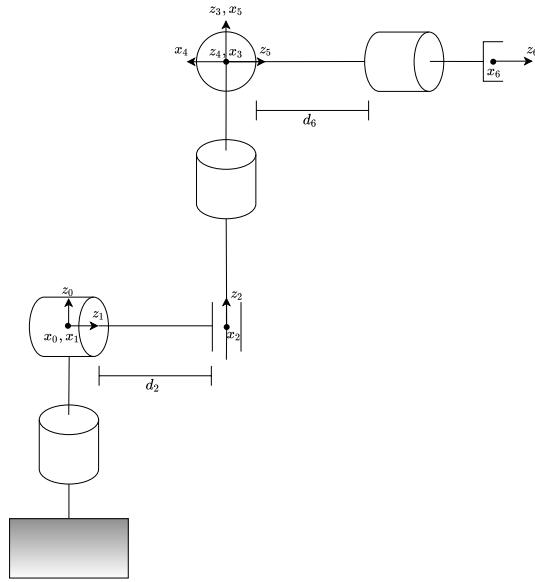


Figura 5.3: Manipolatore di Stanford

Poiché la terna 3 del manipolatore sferico coincide con la terna 3 del polso sferico, la funzione cinematica diretta può essere ottenuta semplicemente per composizione delle matrici di trasformazione.

$${}^0\mathbf{T}_6 = {}^0\mathbf{T}_3 {}^3\mathbf{T}_6 = \begin{bmatrix} n & s & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

dove

$$\underline{n} = \begin{bmatrix} -s_1(c_4s_6 + c_5c_6s_4) - c_1c_2(s_4s_6 - c_4c_5c_6) - c_1c_6s_2s_5 \\ c_1(c_4s_6 + c_5c_6s_4) - c_2s_1(s_4s_6 - c_4c_5c_6) - c_6s_1s_2s_5 \\ s_2(s_4s_6 - c_4c_5c_6) - c_2c_6s_5 \end{bmatrix}$$

$$\underline{s} = \begin{bmatrix} c_1s_2s_5s_6 - c_1c_2(c_6s_4 + c_4c_5s_6) - s_1(c_4c_6 - c_5s_4s_6) \\ c_1(c_4c_6 - c_5s_4s_6) - c_2s_1(c_6s_4 + c_4c_5s_6) + s_1s_2s_5s_6 \\ s_2(c_6s_4 + c_4c_5s_6) + c_2s_5s_6 \end{bmatrix}$$

$$\underline{a} = \begin{bmatrix} c_1c_5s_2 - s_1s_4s_5 + c_1c_2c_4s_5 \\ c_5s_1s_2 + c_1s_4s_5 + c_2c_4s_1s_5 \\ c_5s_2 - c_4c_2s_5 \end{bmatrix}$$

$$\underline{p} = \begin{bmatrix} q_3c_1s_2 - d_2s_1 + d_6c_1c_5s_2 - d_6s_1s_4s_5 + d_6c_1c_2c_4s_5 \\ d_2c_1 + q_3s_1s_2 + d_6c_5s_1s_2 + d_6c_1s_4s_5 + d_6c_2c_4s_1s_5 \\ q_3c_2 + d_6c_2c_5 - d_6c_4s_2s_5 \end{bmatrix}$$

5.2 Soluzione

In precedenza si è visto che ${}^3\mathbf{T}_6(\underline{q}) = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5c_6 - s_4c_6 & c_4s_5 & c_4s_5d_6 \\ s_4c_5c_6 - c_4s_6 & -s_4c_5s_6 - s_4c_6 & c_4s_5 & c_4s_5d_6 \\ -s_5c_6 & s_5s_6 & c_5 & c_5d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

e quindi $\underline{d}_{36} = \begin{bmatrix} c_4s_5d_6 \\ c_4s_5d_6 \\ c_5d_6 \end{bmatrix}$. Si deve quindi verificare che $\exists \underline{d}_0, \underline{d}_1 \in \mathbb{R}^3 : \underline{d}_{36} = {}^3\mathbf{R}_6\underline{d}_1 + \underline{d}_0$, ma:

- $\underline{d}_0 = \underline{0}$ dato che \underline{d}_{36} non presenta termini indipendenti dalle variabili di giunto;

- $\underline{d}_{36} = {}^3\mathbf{R}_6\underline{d}_1$ e quindi $\underline{d}_1 = {}^3\mathbf{R}_6^T\underline{d}_{36} = \begin{bmatrix} 0 \\ 0 \\ d_6 \end{bmatrix}$, tale termine è accettabile perché non dipende dalle variabili di giunto;

5.2.1 Cinematica inversa per il manipolatore sferico RRP

Chiamando

$$\hat{\underline{d}} \stackrel{\text{def}}{=} \underline{d}_{06} - \mathbf{R}\underline{d}_1$$

si ottiene

$$\hat{\underline{d}} = \underline{d}_{03} = \begin{bmatrix} c_1s_2q_3 - s_1d_2 \\ s_1s_2q_3 + c_1d_2 \\ c_1q_3 \end{bmatrix}$$

dato che ${}^0\mathbf{T}_3(\underline{q}) = \begin{bmatrix} c_1c_2 & -s_1 & c_1s_2 & c_1s_2q_3 - s_1d_2 \\ s_1c_2 & c_1 & s_1s_2 & s_1s_2q_3 + c_1d_2 \\ -s_2 & 0 & c_2 & c_2q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, inoltre si ottiene che

$$\hat{\underline{d}}^T \hat{\underline{d}} = \underline{d}_{03}^T \underline{d}_{03} = d_2^2 + q_3^2$$

e ricordando che la condizione per lo spazio operativo primario è $\hat{\underline{d}}^T \hat{\underline{d}} - d_2^2 \geq 0$ si ha che

$$q_3 = \pm \sqrt{\hat{\underline{d}}^T \hat{\underline{d}} - d_2^2}$$

Considerando:

- $q_3 \neq 0$:

$$\hat{d}_3 = q_3 c_2 \Rightarrow c_2 = \frac{\hat{d}_3}{q_3}$$

Se $|\frac{\hat{d}_3}{q_3}| \leq 1$ si ha

$$s_2 = \pm \sqrt{1 - c_2^2}$$

e quindi

$$q_2 = \text{atan2}(s_2, c_2)$$

Dato che $c_2 = \frac{\hat{d}_3}{q_3}$ e $q_3 = \pm \sqrt{\hat{d}^T \hat{d} - d_2^2}$ si ha che

$$-1 \leq \frac{\hat{d}_3}{\sqrt{\hat{d}^T \hat{d} - d_2^2}} \leq 1$$

e quindi

$$-\sqrt{\hat{d}^T \hat{d} - d_2^2} \leq \hat{d}_3 \leq \sqrt{\hat{d}^T \hat{d} - d_2^2}$$

che equivale alla condizione

$$\hat{d}_3^2 \leq |\hat{d}^T \hat{d} - d_2^2| = |\hat{d}_1^2 + \hat{d}_2^2 + \hat{d}_3^2 - d_2^2|$$

ma sapendo che $\hat{d}_1^2 + \hat{d}_2^2 + \hat{d}_3^2 - d_2^2 \geq 0$ si ha

$$\hat{d}_3^2 \leq \hat{d}_1^2 + \hat{d}_2^2 + \hat{d}_3^2 - d_2^2$$

ovvero

$$\hat{d}_1^2 + \hat{d}_2^2 \geq d_2^2 \quad (\text{condizione geometrica})$$

Si è quindi ricavato $q_3 = \pm \sqrt{\hat{d}^T \hat{d} - d_2^2}$ e $q_2 = \text{atan2}(s_2, c_2)$ per $q_3 \neq 0$ e $\hat{d}_1^2 + \hat{d}_2^2 \geq d_2^2$. Per ricavare q_1 è possibile invertire la relazione:

$$\begin{bmatrix} \hat{d}_1 \\ \hat{d}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} q_3 s_2 & -d_2 \\ d_2 & q_3 s_2 \end{bmatrix}}_{\lambda(q_2, q_3)} \begin{bmatrix} c_1 \\ s_1 \end{bmatrix}$$

Ma la matrice $\begin{bmatrix} q_3 s_2 & -d_2 \\ d_2 & q_3 s_2 \end{bmatrix}$ è invertibile $\Leftrightarrow \det \lambda(q_2, q_3) = q_3^2 s_2^2 + d_2^2 \neq 0$.

La risoluzione di tale sistema porta alla soluzione

$$c_1 = \frac{\hat{d}_1 q_3 s_2 + d_2 \hat{d}_2}{q_3^2 s_2^2 + d_2^2}$$

e

$$s_1 = \pm \sqrt{(1 - c_1^2)}$$

- $q_3 = 0$

Si ha $\hat{d}_3 = q_3 c_2 = 0$ ed essendo

$$\begin{cases} -\hat{d}_1 = d_2 s_1 \\ \hat{d}_2 = d_2 c_1 \end{cases}$$

si ottiene $q_1 = \text{atan2}\left(-\frac{\hat{d}_1}{d_2}, \frac{\hat{d}_2}{d_2}\right)$.

q_2 è arbitrario perché moltiplica $q_3 = 0$.

Configurazione singolari

Per quanto visto prima le configurazioni singolari, ovvero i casi in cui il numero di soluzioni è diverso dal numero di soluzioni nel caso generale, sono due:

- $\hat{d}_1^2 + \hat{d}_2^2 + \hat{d}_3^2 - d_2^2 = 0$

In questo caso si ha un solo valore di $q_3 = 0$, q_2 arbitrario e $q_1 = \text{atan2}\left(-\hat{d}_1, \hat{d}_2\right)$. Si hanno perciò infinite soluzioni.

Da notare che la condizione $\hat{d}_1^2 + \hat{d}_2^2 + \hat{d}_3^2 = d_2^2$ indica una sfera di raggio d_2 , dove d_2 è la lunghezza del braccio che congiunge il giunto 2 con il giunto 3.

- $c_2 = \pm 1$

In questo caso si hanno due soluzioni. Fissato $q_3 = \sqrt{\hat{d}^T \hat{d} - d_2^2}$ o $q_3 = -\sqrt{\hat{d}^T \hat{d} - d_2^2}$ si ha un solo valore di q_2 e di q_1 .

5.2.2 Cinematica inversa per il polso sferico

Avendo ricavato q_1 , q_2 e q_3 è ora nota la matrice ${}^0\mathbf{R}_3$, è quindi possibile calcolare

$${}^3\mathbf{R}_6 = {}^0\mathbf{R}_3 {}^T \mathbf{R}$$

da cui è possibile ricavare le restanti variabili di giunto q_4 , q_5 e q_6 .

Chiamando ${}^3\mathbf{R}_6 = \bar{\mathbf{R}}$ è possibile dire che

$$\begin{cases} c_5 = \bar{\mathbf{R}}_{33} \\ s_5 = \pm \sqrt{1 - \bar{\mathbf{R}}_{33}^2} \end{cases} \Rightarrow q_5 = \text{atan2}(s_5, c_5)$$

Da notare il fatto che essendo la matrice $\bar{\mathbf{R}}$ una matrice di rotazione, e quindi ortonormale, le colonne hanno modulo unitario, il valore di $\sqrt{1 - \bar{\mathbf{R}}_{33}^2}$ risulta quindi consistente con i possibili valori di una funzione sin.

Considerando:

- $c_5 = \bar{\mathbf{R}}_{33} \neq \pm 1$

$$\bar{\mathbf{R}}_{32} = s_5 s_6 \Rightarrow \begin{cases} s_6 = \frac{\bar{\mathbf{R}}_{32}}{\pm \sqrt{1 - \bar{\mathbf{R}}_{33}^2}} \\ c_6 = \pm \sqrt{1 - \frac{\bar{\mathbf{R}}_{32}^2}{1 - \bar{\mathbf{R}}_{33}^2}} \end{cases} \Rightarrow q_6 = \text{atan2}(s_6, c_6)$$

$$\begin{cases} \bar{\mathbf{R}}_{13} = s_5 c_4 \\ \bar{\mathbf{R}}_{23} = s_5 s_4 \end{cases} \Rightarrow \begin{cases} c_4 = \frac{\bar{\mathbf{R}}_{13}}{\pm \sqrt{1 - \bar{\mathbf{R}}_{33}^2}} \\ s_4 = \frac{\bar{\mathbf{R}}_{23}}{\pm \sqrt{1 - \bar{\mathbf{R}}_{33}^2}} \end{cases} \Rightarrow q_4 = \text{atan2}(s_4, c_4)$$

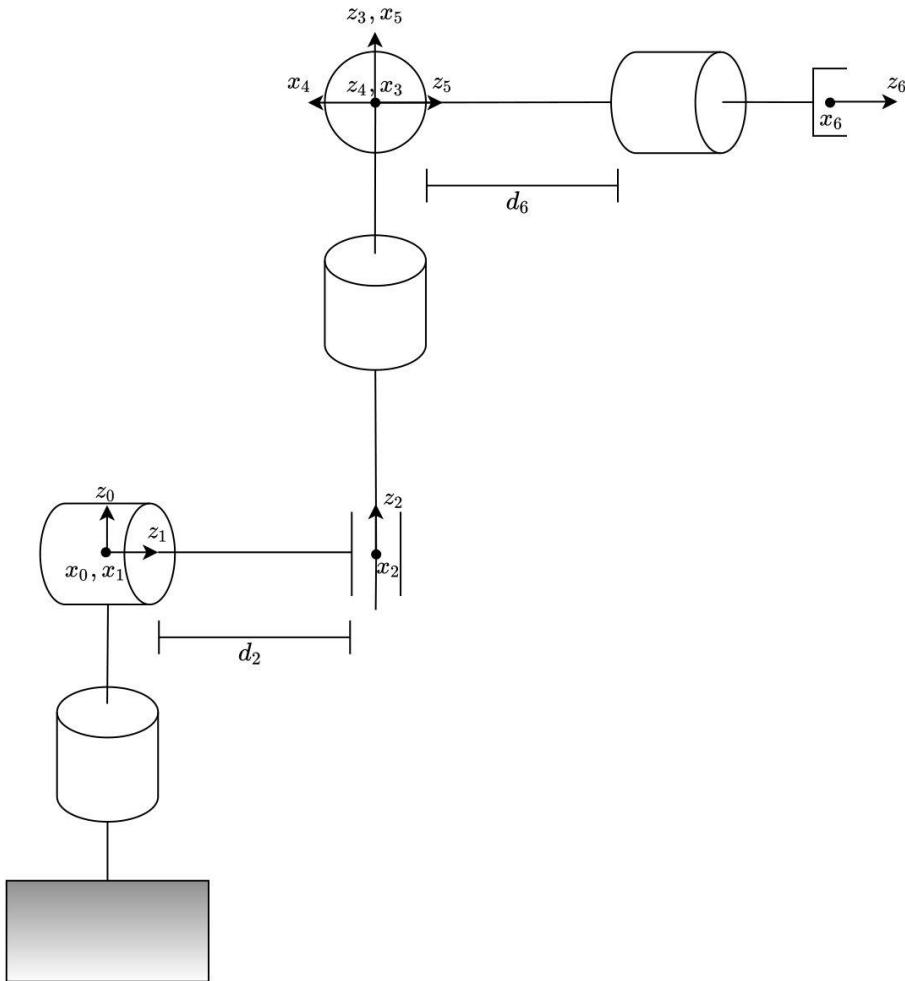
- $\bar{\mathbf{R}}_{33} = 1$

Si ha $q_5 = 0, \pi$ ottenendo così una singolarità. In questo caso è infatti possibile ricavare solo $\theta = q_4 + q_6$, ottenendo così infinite soluzioni per q_4 e q_6 . Da notare che

$$\theta = \text{atan2}(\bar{\mathbf{R}}_{21}, \bar{\mathbf{R}}_{11})$$

Cinematica inversa manipolatore di Stanford

Manipolatore di Stanford



```
syms q1 q2 q3 q4 q5 q6 d2 d6 real
DHstanford = [0, -pi/2, 0, q1;
              0, pi/2, d2, q2;
              0, 0, q3, 0;
              0, -pi/2, 0, q4;
              0, pi/2, 0, q5;
              0, 0, d6, q6]
```

```
DHstanford =
```

$$\begin{pmatrix} 0 & -\frac{\pi}{2} & 0 & q_1 \\ 0 & \frac{\pi}{2} & d_2 & q_2 \\ 0 & 0 & q_3 & 0 \\ 0 & -\frac{\pi}{2} & 0 & q_4 \\ 0 & \frac{\pi}{2} & 0 & q_5 \\ 0 & 0 & d_6 & q_6 \end{pmatrix}$$

```
TstanfordList = cinDirDH(DHstanford);
T01 = TstanfordList{1};
T12 = TstanfordList{2};
T23 = TstanfordList{3};
T34 = TstanfordList{4};
T45 = TstanfordList{5};
T56 = TstanfordList{6};
T03 = T01*T12*T23
```

T03 =

$$\begin{pmatrix} \cos(q_1) \cos(q_2) & -\sin(q_1) & \cos(q_1) \sin(q_2) & q_3 \cos(q_1) \sin(q_2) - d_2 \sin(q_1) \\ \cos(q_2) \sin(q_1) & \cos(q_1) & \sin(q_1) \sin(q_2) & d_2 \cos(q_1) + q_3 \sin(q_1) \sin(q_2) \\ -\sin(q_2) & 0 & \cos(q_2) & q_3 \cos(q_2) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T36 = T34*T45*T56
```

T36 =

$$\begin{pmatrix} \cos(q_4) \cos(q_5) \cos(q_6) - \sin(q_4) \sin(q_6) & -\cos(q_6) \sin(q_4) - \cos(q_4) \cos(q_5) \sin(q_6) & \cos(q_4) \sin(q_5) & d_6 \text{ co} \\ \cos(q_4) \sin(q_6) + \cos(q_5) \cos(q_6) \sin(q_4) & \cos(q_4) \cos(q_6) - \cos(q_5) \sin(q_4) \sin(q_6) & \sin(q_4) \sin(q_5) & d_6 \text{ si} \\ -\cos(q_6) \sin(q_5) & \sin(q_5) \sin(q_6) & \cos(q_5) & d \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

```
T06 = TstanfordList{7}
```

T06 =

$$\begin{pmatrix} -\cos(q_6) \sigma_8 - \sin(q_6) \sigma_4 & \sin(q_6) \sigma_8 - \cos(q_6) \sigma_4 & \sigma_5 - \sin(q_5) \sigma_{10} & q_3 \cos(q_1) \sin(q_2) \\ \cos(q_6) \sigma_6 + \sin(q_6) \sigma_3 & \cos(q_6) \sigma_3 - \sin(q_6) \sigma_6 & \sigma_7 & d_2 \cos(q_1) \\ \sin(q_2) \sin(q_4) \sin(q_6) - \cos(q_6) \sigma_2 & \sin(q_6) \sigma_2 + \cos(q_6) \sin(q_2) \sin(q_4) & \sigma_1 & \\ 0 & 0 & 0 & \end{pmatrix}$$

where

$$\sigma_1 = \cos(q_2) \cos(q_5) - \cos(q_4) \sin(q_2) \sin(q_5)$$

$$\sigma_2 = \cos(q_2) \sin(q_5) + \cos(q_4) \cos(q_5) \sin(q_2)$$

$$\sigma_3 = \cos(q_1) \cos(q_4) - \cos(q_2) \sin(q_1) \sin(q_4)$$

$$\sigma_4 = \cos(q_4) \sin(q_1) + \cos(q_1) \cos(q_2) \sin(q_4)$$

$$\sigma_5 = \cos(q_1) \cos(q_5) \sin(q_2)$$

$$\sigma_6 = \cos(q_5) \sigma_9 - \sin(q_1) \sin(q_2) \sin(q_5)$$

$$\sigma_7 = \sin(q_5) \sigma_9 + \cos(q_5) \sin(q_1) \sin(q_2)$$

$$\sigma_8 = \cos(q_5) \sigma_{10} + \cos(q_1) \sin(q_2) \sin(q_5)$$

$$\sigma_9 = \cos(q_1) \sin(q_4) + \cos(q_2) \cos(q_4) \sin(q_1)$$

$$\sigma_{10} = \sin(q_1) \sin(q_4) - \cos(q_1) \cos(q_2) \cos(q_4)$$

Test cinematica inversa

```
d_2 = 1;
d_6 = 1;
q_1 = 0.1;
q_2 = 0.2;
q_3 = 0.3;
q_4 = 0.4;
q_5 = 0.5;
q_6 = 0.6;
T_06 = subs(T06, [q1,q2,q3,q4,q5,q6,d2,d6], [q_1,q_2,q_3,q_4,q_5,q_6,d_2,d_6]);
round(cinInvStanford(T_06, d_2, d_6), 2)
```

ans = (0.1 0.2 0.3 0.4 0.5 0.6)

```

function [q] = cinInvStanford(T06, d2, d6)
d1 = [0; 0; d6];
d06 = T06(1:3, 4);
R = T06(1:3, 1:3);
dhat = d06 - R*d1;

% manipolatore sferico RRP
q3 = sqrt(dhat.*dhat-d2^2);

if(q3 == 0)
    c1 = -dhat(1)/d2;
    s1 = dhat(2)/d2;
    q1 = atan2(s1, c1);

    q2 = "arbitrario";

    q4 = "non definito";
    q5 = "non definito";
    q6 = "non definito";
else
    c2 = dhat(3)/q3;
    s2 = sqrt(1-c2^2);
    q2 = atan2(s2, c2);

    c1 = (dhat(1)*q3*s2+d2*dhat(2))/(q3^2*s2^2+d2^2);
    s1 = sqrt(1-c1^2);
    q1 = atan(s1, c1);

% polso sferico RRR
R03 = [cos(q1)*cos(q2), -sin(q1), cos(q1)*sin(q2);
        cos(q2)*sin(q1), cos(q1), sin(q1)*sin(q2);
        -sin(q2), 0, cos(q2)];

R = T06(1:3, 1:3);
R36 = R03.*R;

c5 = R36(3, 3);
s5 = sqrt(1-c5^2);
q5 = atan2(s5, c5);

if(c5 == 1 || c5 == -1)
    theta = atan2(R36(2,1), R36(1,1));
    q4 = "theta = q4 + q5 = " + theta;
    q5 = "theta = q4 + q5 = " + theta;

```

```

else
    s6 = R36(3,2)/sqrt(1-R36(3,3)^2);
    c6 = sqrt(1-R36(3,2)^2/(1-R36(3,3)^2));
    q6 = atan2(s6, c6);

    c4 = R36(1,3)/sqrt(1-R36(3,3)^2);
    s4 = R36(2,3)/sqrt(1-R36(3,3)^2);
    q4 = atan2(s4, c4);
end
end

q = round([q1, q2, q3, q4, q5, q6],2);

```

Code 5.1: Funzione MATLAB per la inematica inversa del manipolatore di Stanford

6. Esercitazione 6: vettore velocità angolare

(6) Relazione tra il vettore velocità angolare e le derivate degli angoli di Eulero per la rappresentazione roll-pitch-yaw.

6.1 Descrizione del problema

La cinematica differenziale caratterizza i legami tra le velocità dei giunti e le corrispondenti velocità lineare e angolare dell'organo terminale.

6.1.1 Relazione fondamentale della cinematica

Definition 6.1.1 (Corpo rigido). Un corpo rigido è un oggetto materiale in cui le distanze tra due suoi punti qualunque non cambia nel tempo, in formule

$$\|r_{ij}\| = \text{costante}$$

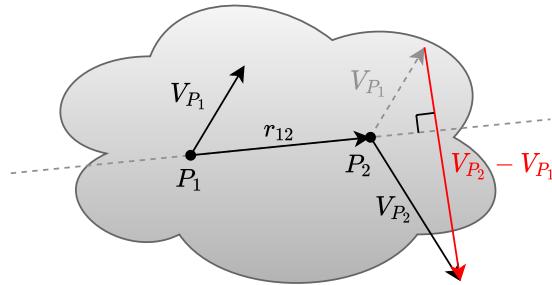


Figura 6.1: Velocità relativa tra due punti su un corpo rigido

Data la definizione di corpo rigido è possibile dire che la velocità relativa tra due punti qualsiasi è ortogonale al vettore che congiunge i due punti

$$\underline{v}_{P_2} - \underline{v}_{P_1} \perp \underline{r}_{12}$$

dato che, se ciò non fosse vero, la componente di $\underline{v}_{P_2} - \underline{v}_{P_1}$ lungo \underline{r}_{12} farebbe in modo che $\|\underline{r}_{ij}\| \neq \text{costante}$.

Dato che $\underline{v}_{P_2} - \underline{v}_{P_1} \perp \underline{r}_{12}$ è possibile dire che

$$\exists \underline{\omega}_1 \in \mathbb{R}^3 : \underline{v}_{P_2} - \underline{v}_{P_1} = \underline{\omega}_1 \times \underline{r}_{12} \quad (6.1)$$

In modo analogo si può dire:

$$\begin{aligned} \underline{v}_{P_3} - \underline{v}_{P_1} &= \underline{\omega}_1 \times \underline{r}_{13} \\ \underline{v}_{P_3} - \underline{v}_{P_2} &= \underline{\omega}_2 \times \underline{r}_{23} \end{aligned}$$

Property 6.1.1. Il vettore velocità angolare $\underline{\omega} = \underline{\omega}_1 = \underline{\omega}_2$ è lo stesso per tutte le coppie di punti, è quindi associato a tutto il corpo e non ai singoli punti.

Dimostrazione.

$$\begin{aligned} \underline{v}_{P_{31}} - \underline{v}_{P_{21}} &= (\underline{v}_{P_3} - \underline{v}_{P_1}) - (\underline{v}_{P_2} - \underline{v}_{P_1}) = \\ &= (\underline{\omega}_1 \times \underline{r}_{13}) - (\underline{\omega}_1 \times \underline{r}_{12}) = \\ &= \mathbf{S}(\underline{\omega}_1) \underline{r}_{13} - \mathbf{S}(\underline{\omega}_1) \underline{r}_{12} = \\ &= \mathbf{S}(\underline{\omega}_1) (\underline{r}_{13} - \underline{r}_{12}) \end{aligned}$$

Ed essendo $\underline{v}_{P_{31}} - \underline{v}_{P_{21}} = \underline{v}_{P_{32}} = \mathbf{S}(\underline{\omega}_2) \times \underline{r}_{23}$ e $\underline{r}_{13} - \underline{r}_{12} = \underline{r}_{23}$ si ha che

$$\mathbf{S}(\underline{\omega}_1) (\underline{r}_{13} - \underline{r}_{12}) = \mathbf{S}(\underline{\omega}_2) (\underline{r}_{23})$$

ovvero $\underline{\omega}_1 = \underline{\omega}_2 = \underline{\omega} \ \forall P_1, P_2, P_3$. □

Property 6.1.2 (Relazione fondamentale della cinematica). Dalla 6.1 e dalla proprietà 6.1.1 è possibile ottenere la relazione fondamentale della cinematica:

$$\underline{v}_{P_j} = \underline{v}_{P_i} + \underline{\omega} \times \underline{r}_{ij} \quad (6.2)$$

Ricordando che $\underline{\omega} \times \underline{r}_{ij} = \mathbf{S}(\underline{\omega}) \underline{r}_{ij}$ è possibile riscrivere la relazione come

$$\underline{v}_{P_j} = \underline{v}_{P_i} + \mathbf{S}(\underline{\omega}) \underline{r}_{ij}$$

6.1.2 Rotazioni infinitesimali

Si ricorda che l'obiettivo è caratterizzare i legami tra le velocità dei giunti e le corrispondenti velocità lineare \underline{v} e angolare $\underline{\omega}$ dell'organo terminale descritto dalla posa $\underline{r} = (p, \underline{\phi}) \in \mathbb{R}^6$.

Si è visto che tale posa può essere rappresentata anche in una forma non minima tramite la matrice di trasformazione $\mathbf{T} = \begin{bmatrix} \mathbf{R} & \underline{p} \\ \underline{0}_{1 \times 3} & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4}$.

È subito possibile notare che

$$\dot{\underline{p}} = \underline{v}$$

ma

$$\dot{\underline{\phi}} \neq \underline{\omega}$$

questo perché $\underline{\omega}$ è un vettore, ovvero un elemento di uno spazio vettoriale: si può ottenere come somma dei contributi $\underline{\omega}_1, \dots, \underline{\omega}_n$ (in qualsiasi ordine). Viceversa, $\dot{\underline{\phi}}$ (e $\underline{\phi}$) non è un elemento di uno spazio vettoriale: la rappresentazione di rotazioni successive, in generale, non si ottiene sommando gli angoli ϕ corrispondenti. In sintesi si può dire che le rotazioni non commutano e che $\underline{\omega}$ non è un differenziale esatto (l'integrale di $\underline{\omega}$ dipende dal cammino di integrazione).

Da notare che le rotazioni infinitesime invece commutano

$$\mathbf{R}(d\underline{\phi}) = \mathbf{R}(d\phi_x, d\phi_y, d\phi_z) = \begin{bmatrix} 1 & -d\phi_z & d\phi_y \\ d\phi_z & 1 & -d\phi_x \\ -d\phi_y & d\phi_x & 1 \end{bmatrix} = \mathbf{I} + \mathbf{S}(d\underline{\phi})$$

Anche le rotazioni finite effettuate intorno allo stesso asse fisso ovviamente commutano.

6.1.3 Derivata di una matrice di rotazione

L'equazione cinematica diretta di un manipolatore descrive la posa dell'organo terminale, in funzione delle variabili di giunto, in termini di un vettore di posizione e di una matrice di rotazione. Se l'obiettivo è quello di caratterizzare velocità lineare e angolare dell'organo terminale, è opportuno analizzare in primo luogo la derivata di una matrice di rotazione rispetto al tempo. Si supponga che la matrice di rotazione vari nel tempo, ovvero $\mathbf{R} = \mathbf{R}(t)$. Essendo $\mathbf{R}(t) \in SO(3)$ si ha la relazione

$$\mathbf{R}(t)\mathbf{R}(t)^T = \mathbf{I} \quad \forall t$$

che derivata rispetto al tempo fornisce:

$$\dot{\mathbf{R}}(t)\mathbf{R}(t)^T + \mathbf{R}(t)(\dot{\mathbf{R}}(t)^T) = 0$$

e ricordando che la derivata e la trasposizione sono due operatori lineari è vero che $(\mathbf{R}(t)^T) = (\dot{\mathbf{R}}(t))^T$, è quindi possibile riscrivere il tutto come

$$\dot{\mathbf{R}}(t)\mathbf{R}(t)^T + \mathbf{R}(t)(\dot{\mathbf{R}}(t))^T = 0$$

Chiamando $\mathbf{S} \stackrel{\text{def}}{=} \dot{\mathbf{R}}(t)\mathbf{R}(t)^T$ si ha che

$$\mathbf{S} + \mathbf{S}^T = 0$$

ovvero \mathbf{S} è una matrice antisimmetrica.

Moltiplicando per $\mathbf{R}(t)$ da destra ambo i membri della relazione

$$\dot{\mathbf{R}}(t)\mathbf{R}(t)^T = \mathbf{S}$$

si ottiene che

$$\dot{\mathbf{R}}(t) = \mathbf{S}(t)\mathbf{R}(t) \quad (6.3)$$

che consente di esprimere la derivata temporale di $\mathbf{R}(t)$ in funzione della matrice stessa.

Si consideri ora un generico vettore costante \underline{p}^* e il vettore $\underline{p}(t) = \mathbf{R}(t)\underline{p}^*$; la derivata temporale di $\underline{p}(t)$ risulta essere

$$\dot{\underline{p}}(t) = \dot{\mathbf{R}}(t)\underline{p}^*$$

che può essere riscritta come

$$\dot{\underline{p}}(t) = \mathbf{S}(t)\mathbf{R}(t)\underline{p}^*$$

Se con il vettore $\underline{\omega}(t)$ si indica la velocità angolare della terna $\mathbf{R}(t)$ all'istante di tempo t rispetto alla terna di riferimento, dalla meccanica è noto che

$$\dot{\underline{p}}(t) = \underline{\omega}(t) \times \mathbf{R}(t)\underline{p}^*$$

Risulta quindi evidente che l'operatore matriciale $\mathbf{S}(t)$ descrive il prodotto vettoriale tra il vettore $\underline{\omega}$ e il vettore $\mathbf{R}(t)\underline{p}^*$. La matrice $\mathbf{S}(t)$ è tale che i suoi elementi, simmetrici rispetto alla diagonale principale, rappresentano le componenti del vettore $\underline{\omega}(t) = [\omega_x \ \omega_y \ \omega_z]^T$ nella forma

$$\mathbf{S} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

È quindi possibile scrivere

$$\dot{\mathbf{R}} = \mathbf{S}(\underline{\omega})\mathbf{R}$$

Property 6.1.3. È possibile dimostrare che vale la seguente relazione

$$\mathbf{R}\mathbf{S}(\underline{\omega})\mathbf{R}^T = \mathbf{S}(\mathbf{R}\underline{\omega})$$

Derivata di matrici di rotazione elementari

Data la matrice di rotazione rispetto all'asse x

$$\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix}$$

è possibile calcolare la sua derivata:

$$\dot{\mathbf{R}}_x(\phi) = \dot{\phi} \begin{bmatrix} 0 & 0 & 0 \\ 0 & -s_\phi & -c_\phi \\ 0 & c_\phi & -s_\phi \end{bmatrix}$$

e quindi

$$\dot{\mathbf{R}}_x(\phi) \mathbf{R}_x^T(\phi) = \dot{\phi} \begin{bmatrix} 0 & 0 & 0 \\ 0 & -s_\phi & -c_\phi \\ 0 & c_\phi & -s_\phi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\dot{\phi} \\ 0 & \dot{\phi} & 0 \end{bmatrix} = \mathbf{S}(\underline{\omega})$$

per cui

$$\underline{\omega} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

dato che

$$\mathbf{S} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

In modo analogo si dimostra che i vettori velocità angolare $\underline{\omega}$ rispetto agli assi y e z sono rispettivamente

$$\underline{\omega} = \begin{bmatrix} 0 \\ \dot{\phi} \\ 0 \end{bmatrix} \quad \underline{\omega} = \begin{bmatrix} 0 \\ 0 \\ \dot{\phi} \end{bmatrix}$$

6.2 Soluzione

Si vuole ora individuare la relazione tra il vettore velocità angolare $\underline{\omega}$ e le derivate

degli angoli di Eulero $\begin{bmatrix} \dot{\gamma} \\ \dot{\beta} \\ \dot{\alpha} \end{bmatrix}$ per la rappresentazione roll-pitch-yaw.

Ricordando che la matrice di rotazione RPY è calcolabile come

$$\mathbf{R}_{RPY}(\alpha, \beta, \gamma) = \mathbf{R}_Z(\gamma)\mathbf{R}_Y(\beta)\mathbf{R}_X(\alpha)$$

la sua derivata è

$$\dot{\mathbf{R}} = \dot{\mathbf{R}}_Z\mathbf{R}_Y\mathbf{R}_X + \mathbf{R}_Z\dot{\mathbf{R}}_Y\mathbf{R}_X + \mathbf{R}_Z\mathbf{R}_Y\dot{\mathbf{R}}_X$$

e ricordando che

$$\begin{aligned}\dot{\mathbf{R}}_Z &= \dot{\gamma}\bar{\mathbf{R}}_X(\gamma) \\ \dot{\mathbf{R}}_Y &= \dot{\beta}\bar{\mathbf{R}}_Y(\beta) \\ \dot{\mathbf{R}}_X &= \dot{\alpha}\bar{\mathbf{R}}_Z(\alpha)\end{aligned}$$

è possibile dire che

$$\dot{\mathbf{R}} = \dot{\gamma}\bar{\mathbf{R}}_X(\gamma)\mathbf{R}_Y\mathbf{R}_X + \mathbf{R}_Z\dot{\beta}\bar{\mathbf{R}}_Y(\beta)\mathbf{R}_X + \mathbf{R}_Z\mathbf{R}_Y\dot{\alpha}\bar{\mathbf{R}}_Z(\alpha)$$

Osservando che

$$\mathbf{S}(\underline{w}) = \dot{\mathbf{R}}_{RPY}\mathbf{R}_{RPY}^T$$

e notando che

$$\mathbf{R}_{RPY}^T = \mathbf{R}_X^T(\alpha)\mathbf{R}_Y^T(\beta)\mathbf{R}_Z^T(\gamma)$$

è possibile dire che

$$\dot{\mathbf{R}}\mathbf{R}^T = \underbrace{\dot{\mathbf{R}}_Z\mathbf{R}_Z^T}_{\mathbf{T}_Z} + \mathbf{R}_Z\underbrace{\dot{\mathbf{R}}_Y\mathbf{R}_Y^T}_{\mathbf{T}_Y}\mathbf{R}_Z^T + \mathbf{R}_Z\mathbf{R}_Y\underbrace{\dot{\mathbf{R}}_X\mathbf{R}_X^T}_{\mathbf{T}_X}\mathbf{R}_Y^T\mathbf{R}_Z^T$$

ma

$$\mathbf{T}_Z = \dot{\gamma} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (6.4)$$

$$\mathbf{T}_Y = \dot{\beta}\mathbf{R}_Z \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \mathbf{R}_Z^T = \dot{\beta} \begin{bmatrix} 0 & 0 & c_\gamma \\ 0 & 0 & s_\gamma \\ -c_\gamma & -s_\gamma & 0 \end{bmatrix} \quad (6.5)$$

$$\mathbf{T}_X = \dot{\alpha}\mathbf{R}_Z\mathbf{R}_Y \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{R}_Y^T\mathbf{R}_Z^T = \dot{\alpha} \begin{bmatrix} 0 & s_\beta & c_\beta s_\gamma \\ -s_\beta & 0 & -c_\beta c_\gamma \\ -c_\beta s_\gamma & c_\beta c_\gamma & 0 \end{bmatrix} \quad (6.6)$$

Si ha quindi che la matrice

$$\mathbf{S}(\underline{w}) = \dot{\gamma}\bar{\mathbf{T}}_Z + \dot{\beta}\bar{\mathbf{T}}_Y(\gamma) + \dot{\alpha}\bar{\mathbf{T}}_X(\beta, \gamma) \quad (6.7)$$

è lineare nelle derivate degli angoli di eulero. Il che significa che \underline{w} è linearmente dipendente alle derivate degli angoli di eulero, in formule

$$\underline{w} = \mathbf{M}(\beta, \gamma) \begin{bmatrix} \dot{\gamma} \\ \dot{\beta} \\ \dot{\alpha} \end{bmatrix}$$

Per ricavare i termini della matrice $\mathbf{M}(\beta, \gamma)$ è possibile osservare che

$$\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} \dot{\gamma} \\ \dot{\beta} \\ \dot{\alpha} \end{bmatrix}$$

e quindi per:

- $\dot{\beta} = 0$ e $\dot{\alpha} = 0$ si ha

$$\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} M_{11}\dot{\gamma} \\ M_{21}\dot{\gamma} \\ M_{31}\dot{\gamma} \end{bmatrix}$$

Dalla 6.7 e dalla 6.4 si ha che

$$\mathbf{S}(\underline{w}) = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} = \dot{\gamma} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

e quindi

$$\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} M_{11}\dot{\gamma} \\ M_{21}\dot{\gamma} \\ M_{31}\dot{\gamma} \end{bmatrix}$$

per cui necessariamente

$$\begin{bmatrix} M_{11} \\ M_{21} \\ M_{31} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- $\dot{\gamma} = 0$ e $\dot{\alpha} = 0$ si ha

$$\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} M_{12}\dot{\beta} \\ M_{22}\dot{\beta} \\ M_{32}\dot{\beta} \end{bmatrix}$$

Dalla 6.7 e dalla 6.5 si ha che

$$\mathbf{S}(\underline{w}) = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} = \dot{\beta} \begin{bmatrix} 0 & 0 & c_\gamma \\ 0 & 0 & s_\gamma \\ -c_\gamma & -s_\gamma & 0 \end{bmatrix}$$

e quindi

$$\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} -s_\beta \dot{\beta} \\ c_\beta \dot{\beta} \\ 0 \end{bmatrix} = \begin{bmatrix} M_{12} \dot{\beta} \\ M_{22} \dot{\beta} \\ M_{32} \dot{\beta} \end{bmatrix}$$

per cui necessariamente

$$\begin{bmatrix} M_{12} \\ M_{22} \\ M_{32} \end{bmatrix} = \begin{bmatrix} -s_\beta \\ c_\beta \\ 0 \end{bmatrix}$$

- $\dot{\gamma} = 0$ e $\dot{\beta} = 0$ si ha

$$\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} M_{13} \dot{\alpha} \\ M_{23} \dot{\alpha} \\ M_{33} \dot{\alpha} \end{bmatrix}$$

Dalla 6.7 e dalla 6.6 si ha che

$$\mathbf{S}(\underline{w}) = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} = \dot{\alpha} \begin{bmatrix} 0 & s_\beta & c_\beta s_\gamma \\ -s_\beta & 0 & -c_\beta c_\gamma \\ -c_\beta s_\gamma & c_\beta c_\gamma & 0 \end{bmatrix}$$

e quindi

$$\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} c_\beta c_\gamma \dot{\alpha} \\ c_\beta s_\gamma \dot{\alpha} \\ -s_\beta \dot{\alpha} \end{bmatrix} = \begin{bmatrix} M_{13} \dot{\alpha} \\ M_{23} \dot{\alpha} \\ M_{33} \dot{\alpha} \end{bmatrix}$$

per cui necessariamente

$$\begin{bmatrix} M_{13} \\ M_{23} \\ M_{33} \end{bmatrix} = \begin{bmatrix} c_\beta c_\gamma \\ c_\beta s_\gamma \\ -s_\beta \end{bmatrix}$$

Mettendo insieme i risultati ottenuti è possibile dire che:

$$\mathbf{M} = \begin{bmatrix} 0 & -s_\gamma & c_\beta c_\gamma \\ 0 & c_\gamma & c_\beta s_\gamma \\ 1 & 0 & -s_\beta \end{bmatrix}$$

La matrice \mathbf{M} fornisce la relazione tra \underline{w} e gli angoli di eulero. Se $\det \mathbf{M}(\beta, \gamma) = -\cos(\beta) \neq 0$ allora a partire da \underline{w} è possibile conoscere la derivata degli angoli di Eulero $\dot{\alpha}, \dot{\beta}, \dot{\gamma}$ ed essendo essi integrabili si possono ricavare α, β, γ .

Risulta ora doveroso notare che dal punto di vista fisico \underline{w} rappresenta la velocità angolare rispetto alla terna base, mentre $\dot{\gamma}, \dot{\beta}$ e $\dot{\alpha}$ rappresentano la velocità angolare rispetto ad un sistema di riferimento che varia al variare della posa dell'organo

terminale. Sembra che la scelta \underline{w} sia migliore, tuttavia è possibile notare che l'integrale dei termini $\dot{\gamma}, \dot{\beta}$ e $\dot{\alpha}$ corrisponde a γ, β e α ed esprime quindi le variazioni impresse sugli angoli di Eulero per passare da un orientamento a un altro, l'integrale di \underline{w} non ammette invece una interpretazione fisica.

7. Esercitazione 7: mobilità e singolarità robot RRR

(7) Analisi di mobilità del manipolatore planare RRR e analisi della singolarità.

7.1 Descrizione del problema

Lo spazio $R(\mathbf{J}(\underline{q}))$ è chiamato spazio delle velocità generalizzate (con componenti lineari e angolari). Esso rappresenta le velocità che possono essere assunte istantaneamente a partire dalla configurazione \underline{q} dall'end-effector al variare delle velocità di giunto.

Property 7.1.1. Se $\mathbf{J}(\underline{q})$ (immagine di \mathbf{J}) ha rango m massimo allora, nella configurazione \underline{q} , il robot può muovere il suo end-effector in tutte le direzioni dello spazio operativo \mathbb{R}^m .

Property 7.1.2. Se $\rho(\mathbf{J}(\underline{q})) < m$ allora esistono direzioni di \mathbb{R}^m in cui l'end-effector non può muoversi. Sono le direzioni identificate da $N(\mathbf{J}^T(\underline{q}))$ (nucleo di \mathbf{J}).

Property 7.1.3. Se $N(\mathbf{J}(\underline{q})) \neq \{0\}$ allora esistono velocità di giunto che non forniscono velocità all'end-effector, esse vengono chiamate automovimenti. La condizioni $N(\mathbf{J}(\underline{q})) \neq \{0\}$ si verifica sempre se $m < n$, ossia nei robot ridondanti.

Le configurazioni cinematiche singolari sono le configurazioni in cui si ha caduta di rango dello Jacobiano con conseguente perdita di mobilità dell'end-effector. In generale, in tali configurazioni la cinematica diretta ammette un numero di soluzioni diverso dal caso generale. Non è inoltre possibile invertire la cinematica

differenziale, ovvero trovare velocità di giunto che realizzino una arbitraria velocità dell'organo terminale del robot.

Vicino alle singolarità potrebbero essere necessarie velocità di giunto elevate per ottenere una velocità anche bassa dell'end-effector.

Conoscere le singolarità aiuta ad evitarle nella pianificazione e nel controllo del moto. Se $m = n$, bisogna trovare le configurazioni \underline{q} in cui $\det \mathbf{J}(\underline{q}) = 0$. Se invece $m < n$, le singolarità sono le configurazioni \underline{q} in cui tutte le sottomatrici $m \times m$ di \mathbf{J} sono singolari (caduta di rango della matrice) oppure quelle per cui $\det(\mathbf{J}(\underline{q})\mathbf{J}^T(\underline{q})) = 0$.

7.2 Soluzione

Si considera ora il robot RRR planare con bracci identicamente lunghi $l_1 = l_2 = l_3 = l = 1$.

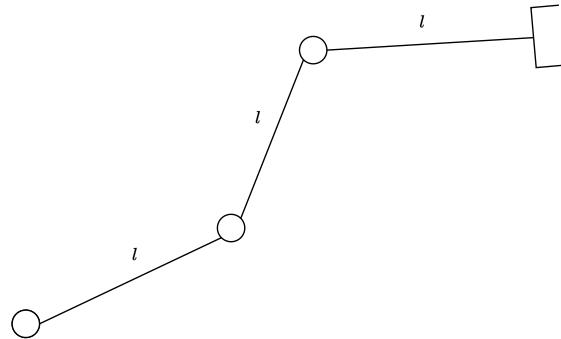


Figura 7.1: Robot RRR planare con bracci identicamente lunghi

In questo caso lo spazio dei compiti ha dimensione $m = 2$ e si utilizza quindi il vettore $\underline{r} = \begin{bmatrix} x \\ y \end{bmatrix}$ per esprimere la posizione dell'end-effector.

Come visto nell'esercitazione 4 lo spazio operativo primario è dato da $WS_1 = \{\underline{p} \in \mathbb{R}^2 : \|\underline{p}\| \leq 3\}$, mentre lo spazio operativo secondario è $WS_2 = \{\underline{p} \in \mathbb{R}^2 : \|\underline{p}\| \leq 1\}$.

I parametri DH per tale manipolatore sono riassunti nella seguente tabella:

Braccio	a_i	α_i	d_i	θ_i
1	1	0	0	q_1
2	1	0	0	q_2
3	1	0	0	q_3

Tabella 7.1: Parametri DH per il manipolatore RRR

e ricordando che la matrice di trasformazione dalla terna i alla terna $i - 1$ è data da una rototraslazione del tipo:

$${}^{i-1}\mathbf{A}_i(q_i) = \begin{bmatrix} c(\theta_i) & -s(\theta_i)c(\alpha_i) & s(\theta_i)s(\alpha_i) & a_i c(\theta_i) \\ s(\theta_i) & c(\theta_i)c(\alpha_i) & -c(\theta_i)s(\alpha_i) & a_i s(\theta_i) \\ 0 & s(\alpha_i) & c(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

si ottengono le seguenti matrici di trasformazione:

$${}^0\mathbf{A}_1(q_1) = \begin{bmatrix} c_1 & -s_1 & 0 & c_1 \\ s_1 & c_1 & 0 & s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^1\mathbf{A}_2(q_2) = \begin{bmatrix} c_2 & -s_2 & 0 & c_2 \\ s_2 & c_2 & 0 & s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^2\mathbf{A}_3(q_3) = \begin{bmatrix} c_3 & -s_3 & 0 & c_3 \\ s_3 & c_3 & 0 & s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

e quindi la matrice di trasformazione completa è

$${}^0\mathbf{A}_3(\underline{q}) = \begin{bmatrix} c_{123} & -s_{123} & 0 & c_1 + c_{12} + c_{123} \\ s_{123} & c_{123} & 0 & s_1 + s_{12} + s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La posizione dell'end-effector è data da

$$\underline{r} = \begin{bmatrix} c_{123} & -s_{123} & 0 & c_1 + c_{12} + c_{123} \\ s_{123} & c_{123} & 0 & s_1 + s_{12} + s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} c_1 + c_{12} + c_{123} \\ s_1 + s_{12} + s_{123} \\ 0 \\ 1 \end{bmatrix}$$

La posa dell'end-effector in termini di rappresentazione minima è:

$$\underline{p} = \begin{bmatrix} p_x \\ p_y \\ \phi \end{bmatrix} = \begin{bmatrix} c_1 + c_{12} + c_{123} \\ s_1 + s_{12} + s_{123} \\ q_1 + q_2 + q_3 \end{bmatrix}$$

Si è visto che lo Jacobiano geometrico può essere calcolato nel seguente modo:

$$\underbrace{\begin{bmatrix} \mathbf{J}_L(\underline{q}) \\ - \\ \mathbf{J}_A(\underline{q}) \end{bmatrix}}_{\in \mathbb{R}^{6 \times n}} = \begin{bmatrix} \underline{J}_{L_1}(\underline{q}) & \dots & \underline{J}_{L_n}(\underline{q}) \\ \underline{J}_{A_1}(\underline{q}) & \dots & \underline{J}_{A_n}(\underline{q}) \end{bmatrix}$$

dove i termini \underline{J}_{L_i} e \underline{J}_{A_i} possono essere calcolati seguendo la seguente tabella:

	prismatico	rotoidale
\underline{J}_{L_i}	\underline{z}_{i-1}	$\underline{z}_{i-1} \times \underline{P}_{i-1,E}$
\underline{J}_{A_i}	0	\underline{z}_{i-1}

Tabella 7.2: Calcolo delle colonne della matrice dello Jacobiano geometrico

Ma per il robot RRR planare si ha che

$$\underline{z}_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \underline{z}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \underline{z}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

ed essendo le matrici di trasformazione parziali

$$\begin{aligned} {}^0\mathbf{A}_3(q_1, q_2, q_3) &= \begin{bmatrix} c_{123} & -s_{123} & 0 & c_1 + c_{12} + c_{123} \\ s_{123} & c_{123} & 0 & s_1 + s_{12} + s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^0\mathbf{A}_1(q_1) &= \begin{bmatrix} c_1 & -s_1 & 0 & c_1 \\ s_1 & c_1 & 0 & s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^0\mathbf{A}_2(q_1) &= \begin{bmatrix} c_{12} & -s_{12} & 0 & c_1 + c_{12} \\ s_{12} & c_{12} & 0 & s_1 + s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

si ha che

$$\begin{aligned} \underline{P}_{0,E} &= \underline{P}_{0,E} - \underline{P}_{0,0} = \begin{bmatrix} c_1 + c_{12} + c_{123} \\ s_1 + s_{12} + s_{123} \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} c_1 + c_{12} + c_{123} \\ s_1 + s_{12} + s_{123} \\ 0 \end{bmatrix} \\ \underline{P}_{1,E} &= \underline{P}_{0,E} - \underline{P}_{0,1} = \begin{bmatrix} c_1 + c_{12} + c_{123} \\ s_1 + s_{12} + s_{123} \\ 0 \end{bmatrix} - \begin{bmatrix} c_1 \\ s_1 \\ 0 \end{bmatrix} = \begin{bmatrix} c_{12} + c_{123} \\ s_{12} + s_{123} \\ 0 \end{bmatrix} \\ \underline{P}_{2,E} &= \underline{P}_{0,E} - \underline{P}_{0,2} = \begin{bmatrix} c_1 + c_{12} + c_{123} \\ s_1 + s_{12} + s_{123} \\ 0 \end{bmatrix} - \begin{bmatrix} c_1 + c_{12} \\ s_1 + s_{12} \\ 0 \end{bmatrix} = \begin{bmatrix} c_{123} \\ s_{123} \\ 0 \end{bmatrix} \end{aligned}$$

Le componenti lineari dello Jacobiano sono quindi

$$\begin{aligned}\underline{J}_{L_1} &= \underline{z}_0 \times \underline{P}_{0,E} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} c_1 + c_{12} + c_{123} \\ s_1 + s_{12} + s_{123} \\ 0 \end{bmatrix} = \\ &= \det \begin{bmatrix} i & j & k \\ 0 & 0 & 1 \\ c_1 + c_{12} + c_{123} & s_1 + s_{12} + s_{123} & 0 \end{bmatrix} = \begin{bmatrix} -s_1 - s_{12} - s_{123} \\ c_1 + c_{12} + c_{123} \\ 0 \end{bmatrix} \\ \underline{J}_{L_2} &= \underline{z}_1 \times \underline{P}_{1,E} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} c_{12} + c_{123} \\ s_{12} + s_{123} \\ 0 \end{bmatrix} = \det \begin{bmatrix} i & j & k \\ 0 & 0 & 1 \\ c_{12} + c_{123} & s_{12} + s_{123} & 0 \end{bmatrix} = \begin{bmatrix} -s_{12} - s_{123} \\ c_{12} + c_{123} \\ 0 \end{bmatrix} \\ J_{L_3} &= z_2 \times P_{2,E} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} c_{123} \\ s_{123} \\ 0 \end{bmatrix} = \det \begin{bmatrix} i & j & k \\ 0 & 0 & 1 \\ c_{123} & s_{123} & 0 \end{bmatrix} = \begin{bmatrix} -s_{123} \\ c_{123} \\ 0 \end{bmatrix}\end{aligned}$$

mentre le componenti rotazionali sono

$$\underline{J}_{A_1} = \underline{z}_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \underline{J}_{A_2} = \underline{z}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \underline{J}_{A_3} = \underline{z}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Infine è quindi possibile dire che lo Jacobiano geometrico è

$$\mathbf{J}(\underline{q}) = \begin{bmatrix} -s_1 - s_{12} - s_{123} & -s_{12} - s_{123} & -s_{123} \\ c_1 + c_{12} + c_{123} & c_{12} + c_{123} & c_{123} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Lo Jacobiano analitico è invece ricavabile per differenziazione

$$\mathbf{J}_r(\underline{q}) = \begin{bmatrix} \frac{\partial p_x}{\partial q_1} & \frac{\partial p_x}{\partial q_2} & \frac{\partial p_x}{\partial q_3} \\ \frac{\partial p_y}{\partial q_1} & \frac{\partial p_y}{\partial q_2} & \frac{\partial p_y}{\partial q_3} \\ \frac{\partial \phi}{\partial q_1} & \frac{\partial \phi}{\partial q_2} & \frac{\partial \phi}{\partial q_3} \end{bmatrix} = \begin{bmatrix} -s_1 - s_{12} - s_{123} & -s_{12} - s_{123} & -s_{123} \\ c_1 + c_{12} + c_{123} & c_{12} + c_{123} & c_{123} \\ 1 & 1 & 1 \end{bmatrix}$$

Al fine di individuare le singolarità è opportuno considerare solo le componenti \dot{p}_x e \dot{p}_y della velocità lineare nel piano, si analizza dunque

$$\mathbf{J}(\underline{q}) = \begin{bmatrix} -s_1 - s_{12} - s_{123} & -s_{12} - s_{123} & -s_{123} \\ c_1 + c_{12} + c_{123} & c_{12} + c_{123} & c_{123} \end{bmatrix}$$

che risulta essere una matrice non quadrata. Al fine di determinare in quali configurazioni tale matrice perde di rango si valuta il determinante:

$$\det(\mathbf{J}_r(\underline{q})\mathbf{J}_r^T(\underline{q})) = 0$$

Analisi delle singolarità per il robot RRR planare

Lo jacobiano analitico del robot planare RRR è

```
syms q1 q2 q3
J = [-sin(q1)-sin(q1+q2)-sin(q1+q2+q3), -sin(q1+q2)-sin(q1+q2+q3), -sin(q1+q2+q3);
      cos(q1)+cos(q1+q2)+cos(q1+q2+q3), cos(q1+q2)+cos(q1+q2+q3), cos(q1+q2+q3)]
```

```
J =
\begin{pmatrix} -\sin(q_1 + q_2 + q_3) - \sin(q_1 + q_2) - \sin(q_1) & -\sin(q_1 + q_2 + q_3) - \sin(q_1 + q_2) & -\sin(q_1 + q_2 + q_3) \\ \cos(q_1 + q_2 + q_3) + \cos(q_1 + q_2) + \cos(q_1) & \cos(q_1 + q_2 + q_3) + \cos(q_1 + q_2) & \cos(q_1 + q_2 + q_3) \end{pmatrix}
```

Al fine di individuare le singolarità si analizzano le soluzioni dell'equazione $\det(\mathbf{JJ}^T) = 0$.

```
JJT = J*(J.' )
```

```
JJT =
\begin{pmatrix} \sigma_2^2 + \sigma_4^2 + (\sigma_4 + \sin(q_1 + q_2))^2 & \sigma_1 \\ \sigma_1 & \sigma_3^2 + \sigma_5^2 + (\sigma_5 + \cos(q_1 + q_2))^2 \end{pmatrix}
```

where

$$\sigma_1 = -\sigma_2 \sigma_3 - \sigma_5 \sigma_4 - (\sigma_5 + \cos(q_1 + q_2)) (\sigma_4 + \sin(q_1 + q_2))$$

$$\sigma_2 = \sigma_4 + \sin(q_1 + q_2) + \sin(q_1)$$

$$\sigma_3 = \sigma_5 + \cos(q_1 + q_2) + \cos(q_1)$$

$$\sigma_4 = \sin(q_1 + q_2 + q_3)$$

$$\sigma_5 = \cos(q_1 + q_2 + q_3)$$

```
simplify(det(JJT))
```

```
ans =
\cos(q_2) - \cos(2 q_2 + q_3) - \frac{\cos(2 q_2)}{2} - \cos(2 q_3) - \cos(2 q_2 + 2 q_3) - \cos(q_2 + 2 q_3) + \cos(q_3) + \frac{5}{2}
```

Secondo WolframAlpha tale equazione si annulla in:

$$q_2 = \pi n_1 \quad q_3 = 2\pi n_2 \quad n_1 \in \mathbb{Z} \quad n_2 \in \mathbb{Z}$$

$$q_2 = 2\pi n_1 + \pi \quad q_3 = 2\pi n_2 + \pi \quad n_1 \in \mathbb{Z} \quad n_2 \in \mathbb{Z}$$

$$q_2 = 2\pi n_2 \quad q_3 = 2\pi n_1 + \pi \quad n_1 \in \mathbb{Z} \quad n_2 \in \mathbb{Z}$$

Si ha quindi singolarità in

$$q_2 = 0 \wedge q_3 = 0$$

$$q_2 = \pi \wedge q_3 = 0$$

$$q_2 = \pi \wedge q_3 = \pi$$

$$q_2 = 0 \wedge q_3 = \pi$$

Rango di J

Al fine di individuare le singolarità è anche possibile determinare le configurazioni per cui la matrice J perde di rango. Si considerano quindi le sue sottomatrici 2×2 .

J12

J12 = J(1:2,1:2)

J12 =

$$\begin{pmatrix} -\sin(q_1 + q_2 + q_3) - \sin(q_1 + q_2) - \sin(q_1) & -\sin(q_1 + q_2 + q_3) - \sin(q_1 + q_2) \\ \cos(q_1 + q_2 + q_3) + \cos(q_1 + q_2) + \cos(q_1) & \cos(q_1 + q_2 + q_3) + \cos(q_1 + q_2) \end{pmatrix}$$

```
detJ12 = simplify(det(J12))
```

$$\detJ{1}{2} = \sin(q_2 + q_3) + \sin(q_2)$$

Quindi il blocco J_{12} (prime due colonne di J) perde di rango nel caso in cui $q_3 = -2q_2$.

```
J121 = simplify(subs(J,q3,-2*q2))
```

J121 =

$$\begin{pmatrix} -\sin(q_1) & (2 \cos(q_2) + 1) & -2 \cos(q_2) \sin(q_1) & -\sin(q_1 - q_2) \\ \cos(q_1) & (2 \cos(q_2) + 1) & 2 \cos(q_1) \cos(q_2) & \cos(q_1 - q_2) \end{pmatrix}$$

```
det(J121(:,[1 2]))
```

ans = 0

```
simplify(det(J121(:,[1 3])))
```

ans = -sin(2 q2) - sin(q2)

Tale determinante si annulla solo per

$$q_2 = n\pi \quad n \in \mathbb{Z}$$

$$q_2 = \frac{\pi}{3} + 2\pi n \quad n \in \mathbb{Z}$$

```
simplify(det(J121(:,[2 3])))
```

ans = -sin(2 q2)

Tale determinante si annulla per $q_2 = 0, \pi$.

Quindi la matrice **J** perde di rango per

$$q_2 = 0 \wedge q_3 = 0$$

$$q_2 = \pi \wedge q_3 = 0$$

J23

```
J23 = J(1:2,2:3)
```

J23 =

$$\begin{pmatrix} -\sin(q_1 + q_2 + q_3) - \sin(q_1 + q_2) & -\sin(q_1 + q_2 + q_3) \\ \cos(q_1 + q_2 + q_3) + \cos(q_1 + q_2) & \cos(q_1 + q_2 + q_3) \end{pmatrix}$$

```
detJ23 = simplify(det(J23))
```

detJ23 = sin(q3)

Il blocco 2x2 **J**₂₃ costituito dalle ultime due colonne di **J** perde di rango nel caso in cui $q_3 = 0 \vee q_3 = \pi$.

```
J231 = subs(J,[q3],[0])
```

J231 =

$$\begin{pmatrix} -2 \sin(q_1 + q_2) - \sin(q_1) & -2 \sin(q_1 + q_2) & -\sin(q_1 + q_2) \\ 2 \cos(q_1 + q_2) + \cos(q_1) & 2 \cos(q_1 + q_2) & \cos(q_1 + q_2) \end{pmatrix}$$

```
simplify(det(J231(:,[1 3])))
```

$$\text{ans} = \sin(q_2)$$

Nel caso $q_3 = 0$ la matrice **J** perde di rango se $q_2 = 0 \vee q_2 = \pi$.

```
J231 = subs(J,[q3],[pi])
```

J231 =

$$\begin{pmatrix} -\sin(q_1) & 0 & \sin(q_1 + q_2) \\ \cos(q_1) & 0 & -\cos(q_1 + q_2) \end{pmatrix}$$

```
simplify(det(J231(:,[1 3])))
```

$$\text{ans} = -\sin(q_2)$$

Nel caso $q_3 = \pi$ la matrice **J** perde di rango se $q_2 = 0 \vee q_2 = \pi$.

Quindi la matrice **J** perde di rango per

$$q_2 = 0 \wedge q_3 = 0$$

$$q_2 = \pi \wedge q_3 = 0$$

$$q_2 = 0 \wedge q_3 = \pi$$

$$q_2 = \pi \wedge q_3 = \pi$$

J13

```
J13 = J(1:2,[1 3])
```

J13 =

$$\begin{pmatrix} -\sin(q_1 + q_2 + q_3) - \sin(q_1 + q_2) - \sin(q_1) & -\sin(q_1 + q_2 + q_3) \\ \cos(q_1 + q_2 + q_3) + \cos(q_1 + q_2) + \cos(q_1) & \cos(q_1 + q_2 + q_3) \end{pmatrix}$$

```
detJ13 = simplify(det(J13))
```

$$\text{detJ13} = \sin(q_2 + q_3) + \sin(q_3)$$

Quindi il blocco **J**₁₃ (prima e ultima colonna di **J**) perde di rango nel caso in cui $q_2 = -2q_3$.

```
J131 = simplify(subs(J,q2,-2*q3))
```

J131 =

$$\begin{pmatrix} -\sin(q_1 - q_3) - \sin(q_1 - 2q_3) - \sin(q_1) & -\sin(q_1 - q_3) - \sin(q_1 - 2q_3) & -\sin(q_1 - q_3) \\ \cos(q_1 - q_3) + \cos(q_1 - 2q_3) + \cos(q_1) & \cos(q_1 - q_3) + \cos(q_1 - 2q_3) & \cos(q_1 - q_3) \end{pmatrix}$$

```
simplify(det(J131(:,1:2)))
```

$$\text{ans} = -\sin(2q_3) - \sin(q_3)$$

Tale determinante si annulla solo per

$$q_3 = n\pi \quad n \in \mathbb{Z}$$

$$q_3 = \frac{\pi}{3} + 2\pi n \quad n \in \mathbb{Z}$$

```
simplify(det(J131(:,2:3)))
```

$$\text{ans} = \sin(q_3)$$

Tale determinante si annulla per $q_3 = 0, \pi$.

Quindi la matrice **J** perde di rango per

$$q_2 = 0 \wedge q_3 = 0$$

$$q_2 = \pi \wedge q_3 = 0$$

Considerazioni finali

Sia le soluzioni di $\det(\mathbf{J}\mathbf{J}^T) = 0$ calcolate con WolframAlpha che l'analisi delle sottomatrici di **J** forniscono le seguenti singolarità:

$$1) q_2 = 0 \wedge q_3 = 0$$

$$2) q_2 = \pi \wedge q_3 = 0$$

$$3) q_2 = \pi \wedge q_3 = \pi$$

$$4) q_2 = 0 \wedge q_3 = \pi$$

Test grafico

Si vogliono verificare graficamente i 4 casi di singolarità precedentemente calcolati. Dato che nessuno dei 4 casi di singolarità dipende da q_1 si pone $q_1 = 0$.

```
q_1 = 0;
```

Caso 1

```
q_2 = 0;
q_3 = 0;
```

In questo caso le direzioni consentite sono date da

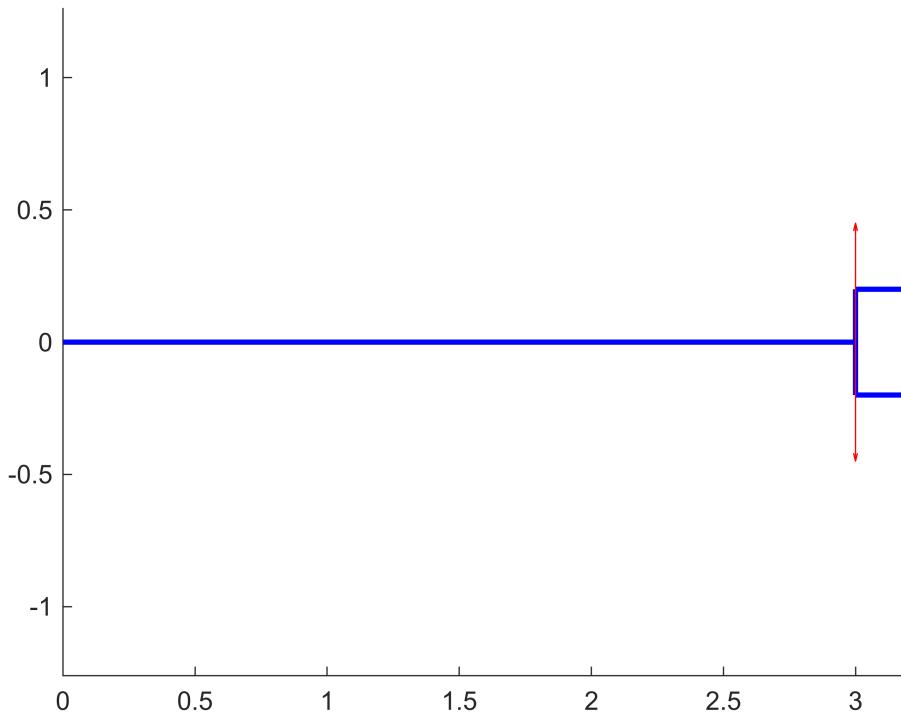
```
subs(J, [q1, q2, q3], [q_1, q_2, q_3])
```

```
ans =
```

$$\begin{pmatrix} 0 & 0 & 0 \\ 3 & 2 & 1 \end{pmatrix}$$

Risulta quindi evidente che le uniche direzioni consentite sono quelle del tipo $\alpha \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Tutti i giunti sono in grado di realizzare tali direzioni in maniera indipendente.

```
figure
drawPlanarRobot([1,1,1], [q_1, q_2, q_3])
axis equal
v1 = 0.5*[0 1];
v2 = 0.5*[0 -1];
ox = cos(q_1)+cos(q_1+q_2)+cos(q_1+q_2+q_3);
oy = sin(q_1)+sin(q_1+q_2)+sin(q_1+q_2+q_3);
quiver(ox, oy, v1(1), v1(2), 'r');
quiver(ox, oy, v2(1), v2(2), 'r');
```



Caso 2

```
q_2 = pi;
q_3 = 0;
```

In questo caso le direzioni consentite sono date da

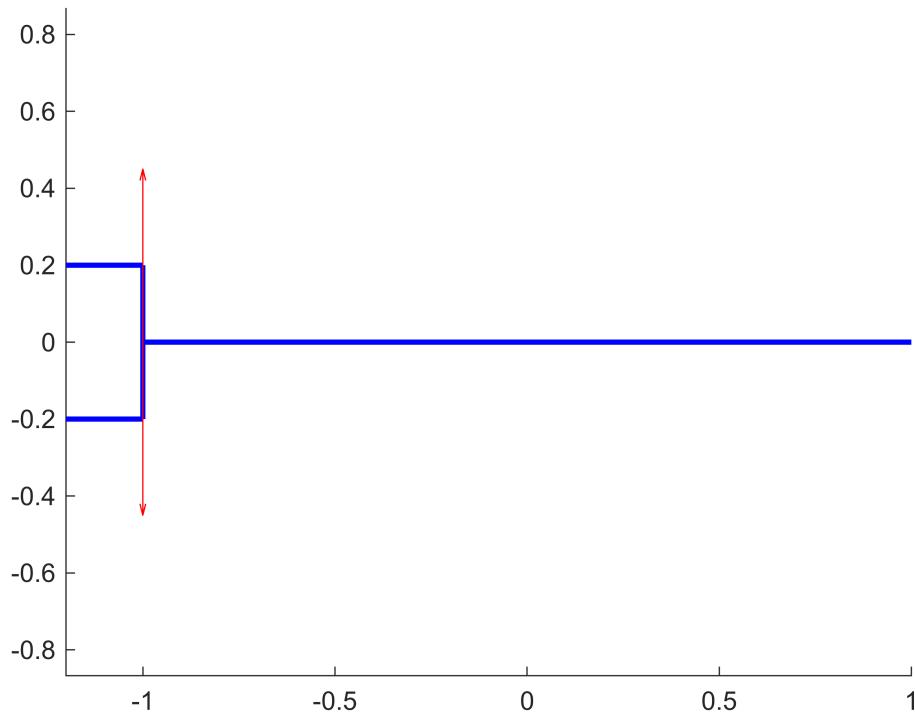
```
subs(J, [q1, q2, q3], [q_1, q_2, q_3])
```

```
ans =
```

$$\begin{pmatrix} 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

Risulta quindi evidente che le uniche direzioni consentite sono quelle del tipo $\alpha \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Tutti i giunti sono in grado di realizzare tali direzioni in maniera indipendente.

```
figure
drawPlanarRobot([1,1,1], [q_1, q_2, q_3])
axis equal
v1 = 0.5*[0 1];
v2 = 0.5*[0 -1];
ox = cos(q_1)+cos(q_1+q_2)+cos(q_1+q_2+q_3);
oy = sin(q_1)+sin(q_1+q_2)+sin(q_1+q_2+q_3);
quiver(ox, oy, v1(1), v1(2), 'r');
quiver(ox, oy, v2(1), v2(2), 'r');
```



Caso 3

```
q_2 = pi;
q_3 = pi;
```

In questo caso le direzioni consentite sono date da

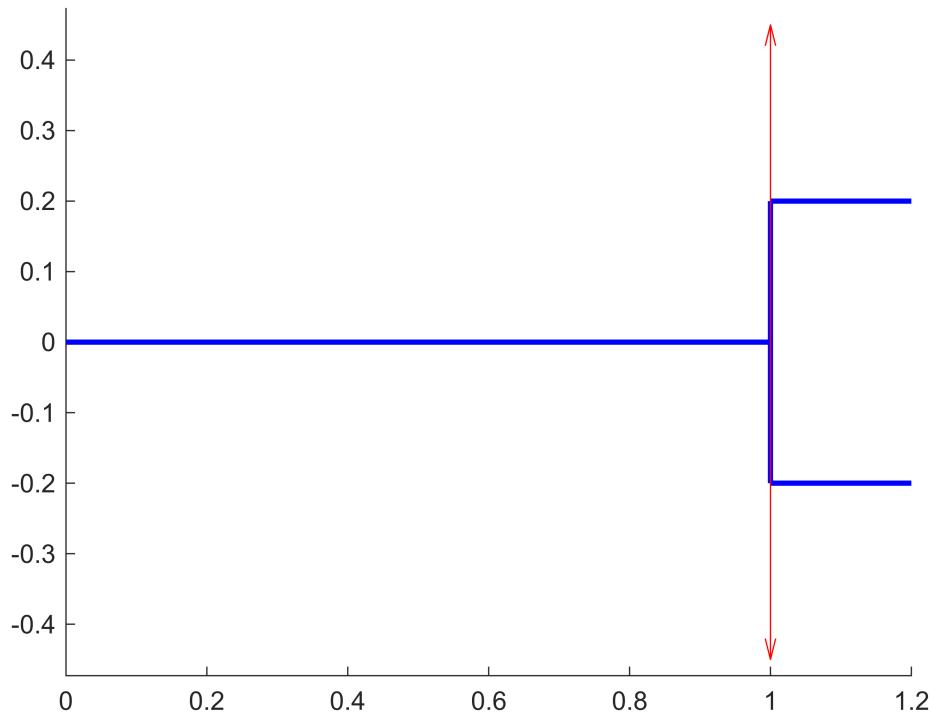
```
subs(J, [q1, q2, q3], [q_1, q_2, q_3])
```

```
ans =
```

$$\begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

Risulta quindi evidente che le uniche direzioni consentite sono quelle del tipo $\alpha \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Il secondo giunto non contribuisce.

```
figure
drawPlanarRobot([1,1,1], [q_1, q_2, q_3])
axis equal
v1 = 0.5*[0 1];
v2 = 0.5*[0 -1];
ox = cos(q_1)+cos(q_1+q_2)+cos(q_1+q_2+q_3);
oy = sin(q_1)+sin(q_1+q_2)+sin(q_1+q_2+q_3);
quiver(ox, oy, v1(1), v1(2), 'r');
quiver(ox, oy, v2(1), v2(2), 'r');
```



Caso 4

```
q_2 = 0;
q_3 = pi;
```

In questo caso le direzioni consentite sono date da

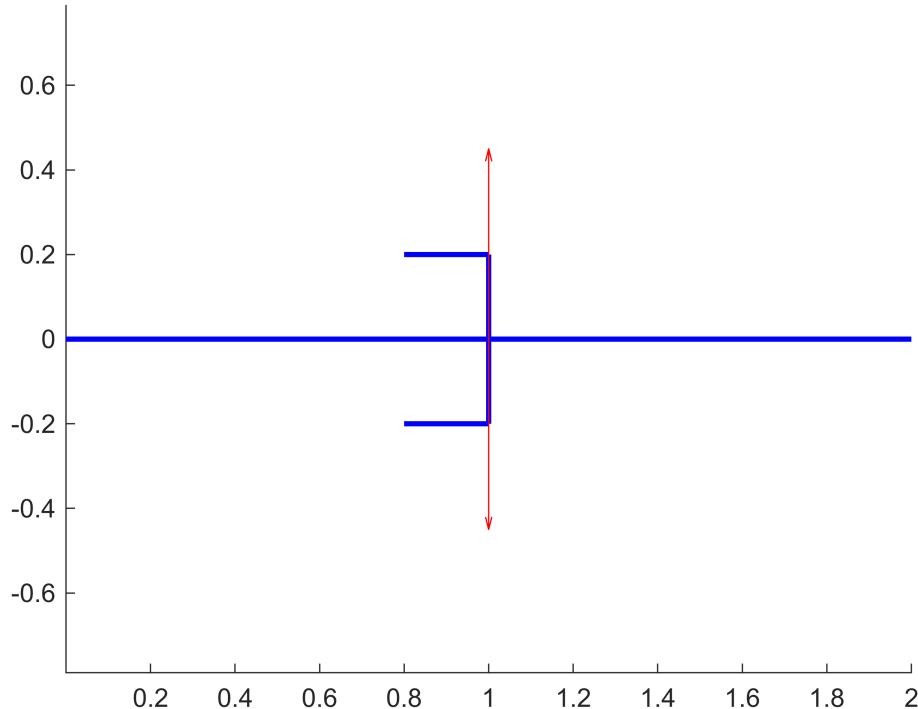
```
subs(J, [q1, q2, q3], [q_1, q_2, q_3])
```

```
ans =
```

$$\begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \end{pmatrix}$$

Risulta quindi evidente che le uniche direzioni consentite sono quelle del tipo $\alpha \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Il secondo giunto non contribuisce.

```
figure
drawPlanarRobot([1,1,1], [q_1, q_2, q_3])
axis equal
v1 = 0.5*[0 1];
v2 = 0.5*[0 -1];
ox = cos(q_1)+cos(q_1+q_2)+cos(q_1+q_2+q_3);
oy = sin(q_1)+sin(q_1+q_2)+sin(q_1+q_2+q_3);
quiver(ox, oy, v1(1), v1(2), 'r');
quiver(ox, oy, v2(1), v2(2), 'r');
```



Caso q_1 diverso da 0

Si vuole ora analizzare un caso in cui $q_1 \neq 0$.

```
q_1 = pi/6;
q_2 = 0;
```

```
q_3 = pi;
```

In questo caso le direzioni consentite sono date da

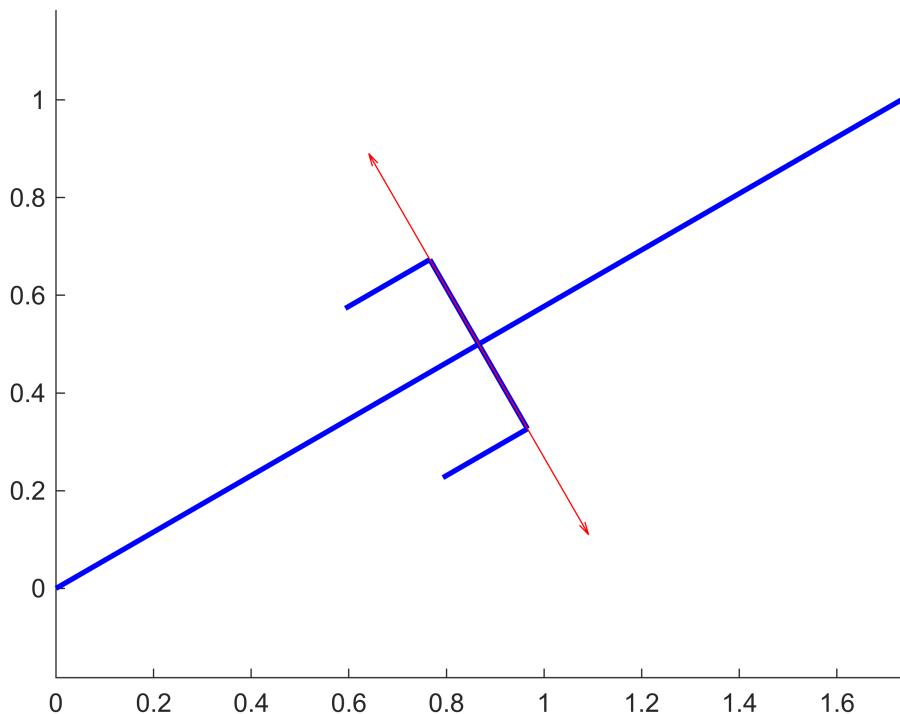
```
subs(J, [q1, q2, q3], [q_1, q_2, q_3])
```

```
ans =
```

$$\begin{pmatrix} -\frac{1}{2} & 0 & \frac{1}{2} \\ \frac{\sqrt{3}}{2} & 0 & -\frac{\sqrt{3}}{2} \end{pmatrix}$$

Risulta quindi evidente che le uniche direzioni consentite sono quelle del tipo $\alpha \begin{pmatrix} -1 \\ \sqrt{3} \end{pmatrix}$. Il secondo giunto non contribuisce.

```
figure
drawPlanarRobot([1,1,1], [q_1, q_2, q_3])
axis equal
v1 = 0.5*[-1 sqrt(3)]/norm([-1 sqrt(3)]);
v2 = 0.5*[1 -sqrt(3)]/norm([1 -sqrt(3)]);
ox = cos(q_1)+cos(q_1+q_2)+cos(q_1+q_2+q_3);
oy = sin(q_1)+sin(q_1+q_2)+sin(q_1+q_2+q_3);
quiver(ox, oy, v1(1), v1(2), 'r');
quiver(ox, oy, v2(1), v2(2), 'r');
```



8. Esercitazione 8: analisi polso sferico

(8) Calcolo dello Jacobiano geometrico del polso sferico con origine dell'ultima terna coincidente col centro del polso. Analisi delle singolarità e confronto con quanto ottenuto nell'esercitazione numero 5.

8.1 Descrizione del problema

8.1.1 Jacobiano geometrico

Lo jacobiano geometrico è nella forma

$$\underbrace{\begin{bmatrix} \underline{v}_E \\ \underline{\omega}_E \end{bmatrix}}_{\in \mathbb{R}^6} = \underbrace{\begin{bmatrix} \mathbf{J}_L(\underline{q}) \\ - \\ \mathbf{J}_A(\underline{q}) \end{bmatrix}}_{\in \mathbb{R}^{6 \times n}} \underbrace{\dot{\underline{q}}}_{\in \mathbb{R}^n} = \begin{bmatrix} \underline{J}_{L_1}(\underline{q}) & \dots & \underline{J}_{L_n}(\underline{q}) \\ \underline{J}_{A_1}(\underline{q}) & \dots & \underline{J}_{A_n}(\underline{q}) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} = \mathbf{J}(\underline{q}) \dot{\underline{q}}$$

dove \underline{J}_{L_i} e \underline{J}_{A_i} sono vettori colonna 3×1 e rappresentano il legame tra velocità lineare e angolare e derivata della variabile di giunto i -esima.

Si ha quindi che le velocità dell'end effector sono date dalla relazione

$$\begin{cases} \underline{v}_E = \underline{J}_{L_1}(\underline{q})\dot{q}_1 + \dots + \underline{J}_{L_n}(\underline{q})\dot{q}_n \\ \underline{\omega}_E = \underline{J}_{A_1}(\underline{q})\dot{q}_1 + \dots + \underline{J}_{A_n}(\underline{q})\dot{q}_n \end{cases}$$

Il metodo sistematico per trovare tale Jacobiano sfrutta la sovrapposizione degli effetti. Per trovare la colonna i infatti si annullano i termini $\dot{q}_j \forall j \neq i$.

8.1.2 Velocità di un braccio

Considerando l' i -esimo braccio

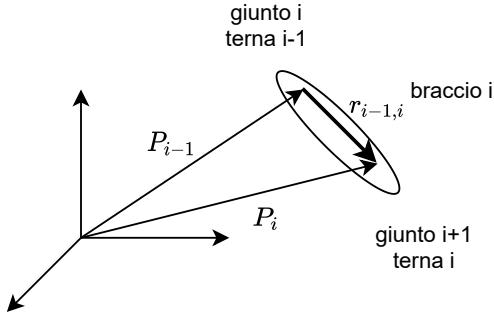


Figura 8.1: Braccio i -esimo

è possibile dire che

$$\underline{P}_i = \underline{P}_{i-1} + \mathbf{R}_{i-1}^{i-1} \underline{r}_{i-1,i}$$

e quindi la velocità lineare della terna i rispetto alla terna 0 è

$$\dot{\underline{P}}_i = \dot{\underline{P}}_{i-1} + \mathbf{R}_{i-1}^{i-1} \dot{\underline{r}}_{i-1,i} + \underline{\omega}_{i-1} \times (\mathbf{R}_{i-1}^{i-1} \underline{r}_{i-1,i}) = \dot{\underline{P}}_{i-1} + \dot{\underline{r}}_{i-1,i} + \underline{\omega}_{i-1} \times \underline{r}_{i-1,i}$$

Si ricorda che si utilizza la seguente notazione $\mathbf{R}_n = {}^0\mathbf{R}_n$.

Inoltre considerando che

$$\mathbf{R}_i = \mathbf{R}_{i-1}^{i-1} \mathbf{R}_i$$

e che

$$\dot{\mathbf{R}}_i = \mathbf{S}(\underline{\omega}_i) \mathbf{R}_i = \mathbf{S}(\underline{\omega}_{i-1}) \mathbf{R}_i + \mathbf{R}_{i-1} \mathbf{S}({}^{i-1}\underline{\omega}_{i-1,i})^{i-1} \mathbf{R}_i$$

dove:

- $\underline{\omega}_i$ rappresenta la velocità angolare del braccio i rispetto a RF_0 ;
- $\underline{\omega}_{i-1}$ rappresenta la velocità angolare del braccio $i - 1$ rispetto a RF_0 ;
- ${}^{i-1}\underline{\omega}_{i-1,i}$ rappresenta la velocità angolare del braccio i rispetto al braccio $i - 1$;

ma notando che

$$\mathbf{R}_{i-1} \mathbf{S}({}^{i-1}\underline{\omega}_{i-1,i})^{i-1} \mathbf{R}_i = \mathbf{R}_{i-1} \mathbf{S}({}^{i-1}\underline{\omega}_{i-1,i}) \underbrace{\mathbf{R}_{i-1}^T \mathbf{R}_{i-1}}_I {}^{i-1} \mathbf{R}_i$$

e utilizzando le seguenti proprietà:

Property 8.1.1.

$$\mathbf{R}\mathbf{S}(\underline{w})\mathbf{R}^T = \mathbf{S}(\mathbf{R}\underline{w})$$

Property 8.1.2.

$$\mathbf{S}(x) + \mathbf{S}(y) = \mathbf{S}(x + y)$$

è possibile dire che

$$\begin{aligned}\dot{\mathbf{R}}_i &= \mathbf{S}(\underline{\omega}_{i-1})\mathbf{R}_i + \mathbf{R}_{i-1}\mathbf{S}^{(i-1)}\underline{\omega}_{i-1,i})^{i-1}\mathbf{R}_i = \\ &= \mathbf{S}(\underline{\omega}_{i-1})\mathbf{R}_i + \mathbf{R}_{i-1}\mathbf{S}^{(i-1)}\underline{w}_{i-1,i})\mathbf{R}_{i-1}^{T(i-1)}\mathbf{R}_i = \\ &= \mathbf{S}(\underline{w}_{i-1})\mathbf{R}_i + \mathbf{S}(\mathbf{R}_{i-1}^{i-1}\underline{w}_{i-1,i})\mathbf{R}_i = \\ &= \mathbf{S}(\underline{w}_{i-1})\mathbf{R}_i + \mathbf{S}(\underline{w}_{i-1,i})\mathbf{R}_i = \\ &= \mathbf{S}(\underline{w}_{i-1} + \underline{w}_{i-1,i})\mathbf{R}_i = \mathbf{S}(\underline{w}_i)\mathbf{R}_i\end{aligned}$$

Si ha quindi che la velocità angolare del braccio i rispetto alla terna RF_0 è

$$\underline{w}_i = \underline{w}_{i-1} + \mathbf{R}_{i-1}^{i-1}\underline{w}_{i-1,i}$$

e per quanto visto prima quella lineare è

$$\dot{\underline{P}}_i = \dot{\underline{P}}_{i-1} + \underline{v}_{i-1,i} + \underline{w}_{i-1} \times \underline{r}_{i-1,i}$$

8.1.3 Metodo sistematico per un giunto prismatico

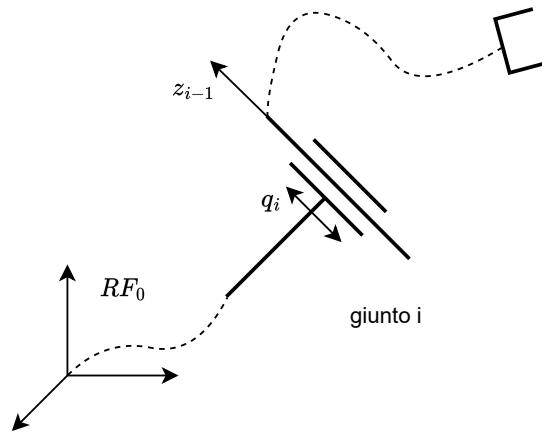


Figura 8.2: Giunto prismatico

Per un giunto prismatico la velocità angolare del braccio i rispetto al braccio $i - 1$ è nulla, si ha infatti $\underline{w}_i = \underline{w}_{i-1}$ e quindi

$$\underline{w}_{i-1,i} = 0$$

Inoltre la velocità lineare è pari in modulo alla derivata della variabile di giunto q_i e ha come direzione quella dell'asse di giunto \underline{z}_{i-1} relativo al giunto i espresso nel riferimento RF_0 , in formule:

$$\underline{v}_{i-1,i} = \dot{q}_i \underline{z}_{i-1}$$

e quindi la velocità lineare assoluta è $\dot{P}_i = \dot{P}_{i-1} + \underline{v}_{i-1,i} + \underline{w}_{i-1} \times \underline{r}_{i-1,i}$.

In definitiva si può dire che per un giunto prismatico si ha:

$$\begin{cases} \underline{J}_{L_i}(q)\dot{q}_i = \underline{z}_{i-1}\dot{q}_i \\ \underline{J}_{A_i}(q)\dot{q}_i = 0 \end{cases}$$

8.1.4 Metodo sistematico per un giunto rotoidale

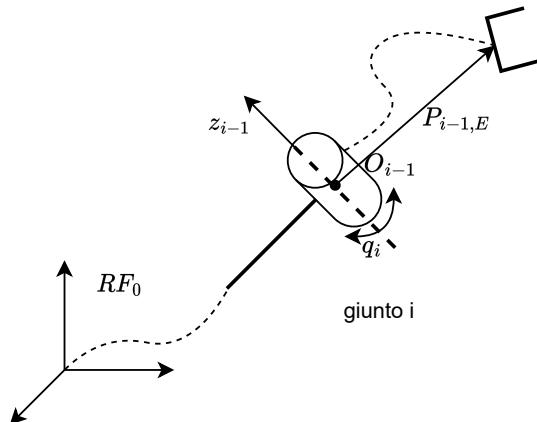


Figura 8.3: Giunto rotoidale

Per un giunto rotoidale la velocità angolare assoluta può essere calcolata sommando la velocità angolare del giunto precedente a quella angolare rispetto al giunto precedente $\underline{w}_i = \underline{w}_{i-1} + \dot{q}_i \underline{z}_{i-1}$, la velocità relativa tra i due giunti è quindi

$$\underline{w}_{i-1,i} = \dot{q}_i \underline{z}_{i-1}$$

La velocità lineare relativa tra due giunti consecutivi $i - 1$ e i è quindi

$$\underline{v}_{i-1,i} = \underline{w}_{i-1,i} \times \underline{r}_{i-1,i}$$

Si ha quindi che la velocità lineare assoluta è $\dot{\underline{P}}_i = \dot{\underline{P}}_{i-1} + \underline{w}_i \times \underline{r}_{i-1,i}$.

In definitiva si può dire che per un giunto rotoidale si ha:

$$\begin{cases} \underline{J}_{L_i}(\underline{q})\dot{q}_i = (\underline{z}_{i-1} \times \underline{P}_{i-1,E})\dot{q}_i \\ \underline{J}_{A_i}(\underline{q})\dot{q}_i = \underline{z}_{i-1}\dot{q}_i \end{cases}$$

8.1.5 Metodo sistematico per il calcolo dello Jacobiano geometrico

Si è visto che lo Jacobiano geometrico può essere scomposto nel seguente modo:

$$\underbrace{\begin{bmatrix} \underline{v}_E \\ \underline{\omega}_E \end{bmatrix}}_{\in \mathbb{R}^6} = \underbrace{\begin{bmatrix} \underline{J}_L(\underline{q}) \\ - \\ \underline{J}_A(\underline{q}) \end{bmatrix}}_{\in \mathbb{R}^{6 \times n}} \underbrace{\begin{bmatrix} \dot{q} \end{bmatrix}}_{\in \mathbb{R}^n} = \begin{bmatrix} \underline{J}_{L_1}(\underline{q}) & \dots & \underline{J}_{L_n}(\underline{q}) \\ \underline{J}_{A_1}(\underline{q}) & \dots & \underline{J}_{A_n}(\underline{q}) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} = \underline{J}(\underline{q})\dot{\underline{q}}$$

dove i termini \underline{J}_{L_i} e \underline{J}_{A_i} possono essere calcolati seguendo la seguente tabella:

	prismatico	rotoidale
\underline{J}_{L_i}	\underline{z}_{i-1}	$\underline{z}_{i-1} \times \underline{P}_{i-1,E}$
\underline{J}_{A_i}	0	\underline{z}_{i-1}

Tabella 8.1: Calcolo delle colonne della matrice dello Jacobiano geometrico

Risulta quindi necessario che i termini \underline{z}_{i-1} e $\underline{P}_{i-1,E}$ possano essere estratti dalla cinematica diretta.

È evidente che

$$\underline{z}_{i-1} = {}^0\mathbf{R}_{i-1}{}^{i-1}\underline{z}_{i-1} = {}^0\mathbf{R}_{i-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

il versore parallelo all'asse del giunto i è quindi dato dalla terza colonna della matrice di rotazione che porta la terna $i - 1$ sulla terna 0. In effetti le colonne di tale matrice sono i versori della terna $i - 1$ espressi nella terna 0.

Il termine $\underline{P}_{i-1,E}$ rappresenta la posizione relativa dell'organo terminale rispetto alla terna $i - 1$. È possibile ricavare tale vettore tramite la seguente operazione vettoriale

$$\underline{P}_{i-1,E} = \underline{P}_{0,E} - \underline{P}_{0,i-1}$$

dove i termini $\underline{P}_{0,E}$ e $\underline{P}_{0,i-1}$ sono ricavabili rispettivamente dalla cinematica totale e dalla cinematica parziale, in particolare essi sono le prime tre righe dell'ultima colonna delle matrici di trasformazione omogenea ${}^0\mathbf{T}_E$ e ${}^0\mathbf{T}_{i-1}$.

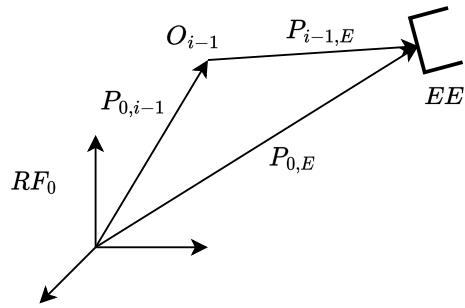
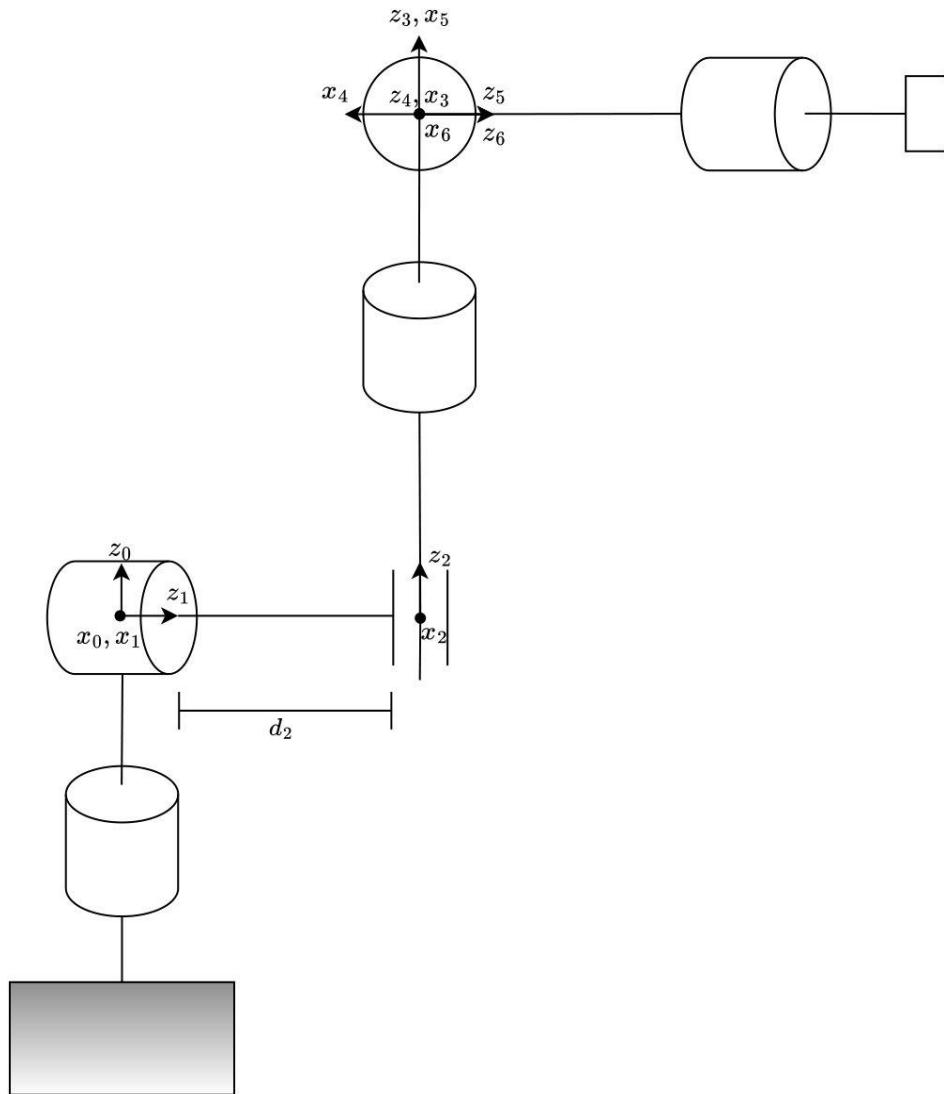


Figura 8.4: Posizione relativa dell'EE rispetto alla terna $i - 1$

8.2 Soluzione

Jacobiano geometrico del polso sferico

Manipolatore di Stanford con O_6 = centro polso sferico



```
syms q1 q2 q3 q4 q5 q6 d2 real
DHstanford = [0, -pi/2, 0, q1;
              0, pi/2, d2, q2;
              0, 0, q3, 0;
              0, -pi/2, 0, q4;
              0, pi/2, 0, q5;
              0, 0, 0, q6]
```

DHstanford =

$$\begin{pmatrix} 0 & -\frac{\pi}{2} & 0 & q_1 \\ 0 & \frac{\pi}{2} & d_2 & q_2 \\ 0 & 0 & q_3 & 0 \\ 0 & -\frac{\pi}{2} & 0 & q_4 \\ 0 & \frac{\pi}{2} & 0 & q_5 \\ 0 & 0 & 0 & q_6 \end{pmatrix}$$

```
J = simplify(jacGeom(DHstanford, [1,1,0,1,1,1]))
```

$$J = \begin{pmatrix} -d_2 \cos(q_1) - q_3 \sin(q_1) \sin(q_2) & q_3 \cos(q_1) \cos(q_2) & \sigma_2 & 0 & 0 \\ q_3 \cos(q_1) \sin(q_2) - d_2 \sin(q_1) & q_3 \cos(q_2) \sin(q_1) & \sigma_1 & 0 & 0 \\ 0 & -q_3 \sin(q_2) & \cos(q_2) & 0 & 0 \\ 0 & -\sin(q_1) & 0 & \sigma_2 & -\cos(q_4) \sin(q_1) - \cos(q_1) \cos(q_2) \sin(q_3) \\ 0 & \cos(q_1) & 0 & \sigma_1 & \cos(q_1) \cos(q_4) - \cos(q_2) \sin(q_1) \sin(q_3) \\ 1 & 0 & 0 & \cos(q_2) & \sin(q_2) \sin(q_4) \end{pmatrix}$$

where

$$\sigma_1 = \sin(q_1) \sin(q_2)$$

$$\sigma_2 = \cos(q_1) \sin(q_2)$$

Si ha quindi $\mathbf{J}(q_1, \dots, q_5) = \begin{pmatrix} \mathbf{J}_{11} & 0 \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{pmatrix}$ per cui $\det \mathbf{J}(q_1, \dots, q_5) = \det \mathbf{J}_{11} \det \mathbf{J}_{22}$.

```
detJ = simplify(det(J))
```

$$\det J = q_3^2 \sin(q_2) \sin(q_5)$$

$\det \mathbf{J}_{11}(q_1, \dots, q_3) = 0$ fornisce le singolarità della struttura portante.

```
J11 = J(1:3, 1:3)
```

$J_{11} =$

$$\begin{pmatrix} -d_2 \cos(q_1) - q_3 \sin(q_1) \sin(q_2) & q_3 \cos(q_1) \cos(q_2) & \cos(q_1) \sin(q_2) \\ q_3 \cos(q_1) \sin(q_2) - d_2 \sin(q_1) & q_3 \cos(q_2) \sin(q_1) & \sin(q_1) \sin(q_2) \\ 0 & -q_3 \sin(q_2) & \cos(q_2) \end{pmatrix}$$

```
detJ11 = simplify(det(J11))
```

$$\det J_{11} = -q_3^2 \sin(q_2)$$

$\det J_{22}(q_1, \dots, q_5) = 0$ fornisce le singolarità del polso sferico.

$$J_{22} = J(4:6, 4:6)$$

$$J_{22} =$$

$$\begin{pmatrix} \cos(q_1) \sin(q_2) & -\cos(q_4) \sin(q_1) - \cos(q_1) \cos(q_2) \sin(q_4) & \cos(q_1) \cos(q_5) \sin(q_2) - \sin(q_5) (\sin(q_1) \sin(q_4) \\ \sin(q_1) \sin(q_2) & \cos(q_1) \cos(q_4) - \cos(q_2) \sin(q_1) \sin(q_4) & \sin(q_5) (\cos(q_1) \sin(q_4) + \cos(q_2) \cos(q_4) \sin(q_4) \\ \cos(q_2) & \sin(q_2) \sin(q_4) & \cos(q_2) \cos(q_5) - \cos(q_4) \sin(q_4) \end{pmatrix}$$

$$\det J_{22} = \text{simplify}(\det(J_{22}))$$

$$\det J_{22} = -\sin(q_5)$$

Si ha quindi singolarità per $q_5 = 0, \pi$.

Jacobiano geometrico del polso sferico

$$DHpolso = [0, -\pi/2, 0, q4; \\ 0, \pi/2, 0, q5; \\ 0, 0, 0, q6]$$

$$DHpolso =$$

$$\begin{pmatrix} 0 & -\frac{\pi}{2} & 0 & q4 \\ 0 & \frac{\pi}{2} & 0 & q5 \\ 0 & 0 & 0 & q6 \end{pmatrix}$$

$$tList = \text{cinDirDH}(DHpolso); \\ T = \text{simplify}(tList[4])$$

$$T =$$

$$\begin{pmatrix} \cos(q_4) \cos(q_5) \cos(q_6) - \sin(q_4) \sin(q_6) & -\cos(q_6) \sin(q_4) - \cos(q_4) \cos(q_5) \sin(q_6) & \cos(q_4) \sin(q_5) & 0 \\ \cos(q_4) \sin(q_6) + \cos(q_5) \cos(q_6) \sin(q_4) & \cos(q_4) \cos(q_6) - \cos(q_5) \sin(q_4) \sin(q_6) & \sin(q_4) \sin(q_5) & 0 \\ -\cos(q_6) \sin(q_5) & \sin(q_5) \sin(q_6) & \cos(q_5) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Jpolso = \text{simplify}(\text{jacGeom}(DHpolso, [1,1,1]))$$

$$Jpolso =$$

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -\sin(q_4) & \cos(q_4) \sin(q_5) \\ 0 & \cos(q_4) & \sin(q_4) \sin(q_5) \\ 1 & 0 & \cos(q_5) \end{pmatrix}$$

Per trovare le singolarità si considerano i casi in cui lo Jacobiano perde di rango, ovvero:

```
Jp = Jpolso(4:6, :)
```

```
Jp =
(0 -sin(q4) cos(q4) sin(q5))
(0 cos(q4) sin(q4) sin(q5))
(1 0 cos(q5))
```

```
simplify(det(Jp))
```

```
ans = -sin(q5)
```

```

function [jacGeom] = jacGeom(parDH, types)
    % ordine parametri: a alpha d theta type
    % type => 0 prismatico, 1 rotoidale
    N = size(parDH,1); % numero di giunti
    tList = cinDirDH(parDH); % lista di trasformazioni i-(i-1) e 0-N

    % calcolo la lista di trasformazioni 0-i
    tList0i{1} = eye(4); % trasformazione 0-0
    for i=1:N
        tList0i{i+1} = simplify(tList0i{i}*tList{i}); % trasformazione 0-(i+1)
    end

    TN = simplify(tList0i{N+1}); % cinematica diretta 0-N
    POE = TN(1:3, 4); % posa end effector
    for i=1:N
        T = tList0i{i};
        z = T(1:3,1:3)*[0; 0; 1]; % z_{i-1}
        % prismatico
        if(types(i) == 0)
            JL = z;
            JA = [0; 0; 0];
        % rotoidale
        else
            P0i = T(1:3, 4);
            P1E = simplify(POE-P0i); % P_{i-1,E}
            JL = simplify(cross(z, P1E));
            JA = z;
        end
        jacGeom(1:3, i) = JL;
        jacGeom(4:6, i) = JA;
    end
end

```

Code 8.1: Funzione MATLAB per il calcolo dello jacobiano geometrico

9. Esercitazione 9: algoritmo di Newton e del gradiente

(9) *Implementazione degli algoritmi di Newton e del gradiente per il manipolatore RR planare e analisi dei massimi e punti di sella della funzione “norma dell’errore cartesiano al quadrato”.*

9.1 Descrizione del problema

I metodi analitici per l’inversione della cinematica diretta $\underline{r} = f_r(\underline{q})$ non sono sempre utilizzabili, si vuole quindi formulare un metodo numerico (iterativo) da poter applicare sempre.

9.1.1 Algoritmo di Newton

Si definisce un errore

$$\underline{e} = \underline{r} - f_r(\underline{q})$$

L’algoritmo iterativo da progettare ha quindi come obiettivo quello di fornire una evoluzione per \underline{q} tale che $\underline{q} \rightarrow \bar{\underline{q}}$ in modo che $\bar{\underline{e}}(\bar{\underline{q}}) = \underline{r} - f_r(\bar{\underline{q}}) = 0$. Per fare ciò si considera la sua derivata

$$\dot{\underline{e}} = \dot{\underline{r}} - \mathbf{J}_r(\underline{q})\dot{\underline{q}}$$

Considerando \mathbf{J} quadrata e non singolare e imponendo

$$\dot{\underline{e}} = \mathbf{K}\underline{e}$$

si fa tendere l’errore a 0 quando gli autovalori di \mathbf{K} sono tutti a parte reale negativa, ovvero $\sigma(\mathbf{K}) \in \mathbb{C}^-$.

Si ha quindi che

$$\dot{\underline{r}} - \mathbf{J}_r(\underline{q})\dot{\underline{q}} = \mathbf{K}\underline{e}$$

da cui si ottiene

$$\dot{\underline{q}} = \mathbf{J}_r^{-1}(\underline{q})[\dot{\underline{r}} - \mathbf{K}\underline{e}]$$

da cui deriva l'algoritmo di Newton. La condizione iniziale va scelta in modo che la matrice $\mathbf{J}_r(\underline{q})$ sia non singolare. Tale algoritmo funziona finché $\mathbf{J}_r(\underline{q})$ è non singolare.

In generale, se \mathbf{J} è non quadrata e non singolare si può sfruttare la pseudoinversa:

$$\dot{\underline{q}} = \mathbf{J}_r^\# (\dot{\underline{r}} - \mathbf{K}\underline{e}) + (\mathbf{I} - \mathbf{J}_r^\# \mathbf{J}_r)\dot{\underline{q}}_0$$

dove $\dot{\underline{q}}_0$ è scelto in maniera arbitraria.

Nel caso quadrato non singolare l'algoritmo risulta quindi essere una procedura iterativa che ad ogni passo calcola:

$$\underline{q}(k+1) = \underline{q}(k) + T\mathbf{J}^{-1}(\underline{q}(k))[\dot{\underline{r}}(k) - \mathbf{K}\underline{e}(k)]$$

dove T è il periodo di discretizzazione.

9.1.2 Algoritmo del gradiente

Come nel caso dell'algoritmo di Newton si considera l'errore

$$\underline{e} = \underline{r} - f_r(\underline{q})$$

e la sua derivata

$$\dot{\underline{e}} = \dot{\underline{r}} - \mathbf{J}_r(\underline{q})\dot{\underline{q}}$$

Tuttavia in questo caso, invece di imporre $\dot{\underline{e}} = \mathbf{K}\underline{e}$ affinché l'errore converga, si sfrutta la teoria di Lyapunov.

In particolare, si utilizza una funzione di Lyapunov del tipo

$$V(\underline{e}) = \frac{1}{2}\underline{e}^T \underline{e}$$

la sua derivata è

$$\dot{V}(\underline{e}) = [\nabla V(\underline{e})]^T \dot{\underline{e}} = \underline{e}^T \dot{\underline{e}}$$

dato che ovviamente $\nabla V(\underline{e}) = \underline{e}$.

Assumendo un riferimento costante $\dot{\underline{r}} = 0$ si ottiene

$$\dot{V}(\underline{e}) = -\underline{e}^T \mathbf{J}_r(\underline{q})\dot{\underline{q}}$$

e al fine di ottenere $\dot{V}(\underline{e})$ semidefinita negativa si prende $\dot{\underline{q}} = \mathbf{J}_r^T(\underline{q})\underline{e}$. Quindi si ha

$$\dot{V}(\underline{e}) = -\underline{e}^T \mathbf{J}_r(\underline{q}) \mathbf{J}_r^T(\underline{q}) \underline{e}$$

ed essendo $\mathbf{J}_r(\underline{q})\mathbf{J}_r(\underline{q})$ definita positiva, $\dot{V}(\underline{e})$ è definita negativa.

Per il principio di Lasalle si può dimostrare che l'errore non converge a 0 ma rimane limitato. Si sa inoltre che $\mathbf{J}_r^T(\underline{q})\underline{e}$ tenderà a 0, il che non garantisce che \underline{e} tenda a 0 dato che $\mathbf{J}_r(\underline{q})$ potrebbe essere singolare. Nel caso di singolarità l'algoritmo si ferma in un punto con un errore che potrebbe essere non nullo.

L'algoritmo del gradiente prevede il calcolo

$$\dot{\underline{q}} = \alpha \mathbf{J}_r^T(\underline{q})(\underline{r} - f(\underline{q}))$$

dove $\alpha > 0$. Si tratta di un algoritmo del gradiente perché se consideriamo la funzione $H(\underline{q}) = \frac{1}{2}\|\underline{r} - f(\underline{q})\|^2$, il suo gradiente è proprio $\nabla H(\underline{q}) = -\mathbf{J}_r^T(\underline{q})(\underline{r} - f(\underline{q}))$. Ci si muove nella direzione dell'antigradiente di $H(\underline{q})$, ovvero nella direzione dove si minimizza l'errore, dato che H rappresenta l'errore quadratico. In forma discreta si calcola quindi

$$\underline{q}(k+1) = \underline{q}(k) + \alpha \mathbf{J}_r^T(\underline{q}(k))[\underline{r} - f(\underline{q}(k))]$$

dove α è lo step size.

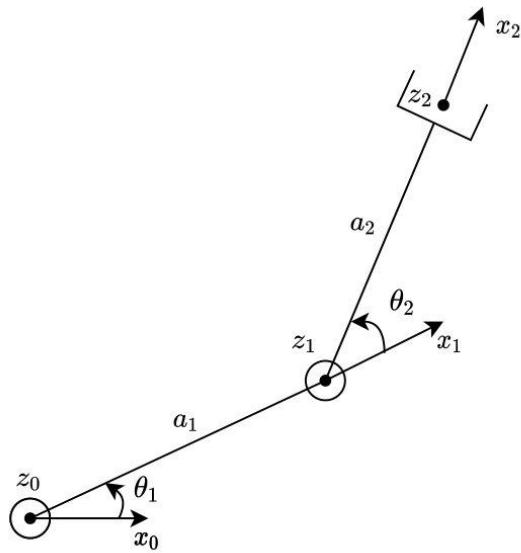
Il vantaggio del metodo del gradiente rispetto al metodo di Newton è che esso non richiede il calcolo della matrice inversa \mathbf{J}_r^{-1} . Il metodo del gradiente tuttavia non garantisce la convergenza in punti di minimo, potrebbe infatti convergere in punti di massimo o di sella (che corrispondono alle configurazioni singolari).

In generale i metodi iterativi per l'inversione della cinematica presentano uno svantaggio, per poterli applicare sono richieste le condizioni iniziali.

9.2 Soluzione

Algoritmo di Newton e del gradiente per il robot RR planare

Robot RR planare



```
clear
clc
syms l1 l2 q1 q2
DHRRplanare = [1, 0, 0, q1;
                1, 0, 0, q2]
```

```
DHRRplanare =

$$\begin{pmatrix} 1 & 0 & 0 & q_1 \\ 1 & 0 & 0 & q_2 \end{pmatrix}$$

tList = cinDirDH(DHRRplanare);
T02 = tList{3}
```

$$T02 = \begin{pmatrix} \cos(q_1 + q_2) & -\sin(q_1 + q_2) & 0 & \cos(q_1 + q_2) + \cos(q_1) \\ \sin(q_1 + q_2) & \cos(q_1 + q_2) & 0 & \sin(q_1 + q_2) + \sin(q_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
r = T02(1:2, 4)
```

```
r =

$$\begin{pmatrix} \cos(q_1 + q_2) + \cos(q_1) \\ \sin(q_1 + q_2) + \sin(q_1) \end{pmatrix}$$

```

```
Jgeom = jacGeom(DHRRplanare, [1 1])
```

```
Jgeom =
```

$$\begin{pmatrix} -\sin(q_1 + q_2) - \sin(q_1) & -\sin(q_1 + q_2) \\ \cos(q_1 + q_2) + \cos(q_1) & \cos(q_1 + q_2) \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{pmatrix}$$

```
Jr = Jgeom(1:2, 1:2)
```

Jr =

$$\begin{pmatrix} -\sin(q_1 + q_2) - \sin(q_1) & -\sin(q_1 + q_2) \\ \cos(q_1 + q_2) + \cos(q_1) & \cos(q_1 + q_2) \end{pmatrix}$$

```
Jinv = simplify(inv(Jr))
```

Jinv =

$$\begin{pmatrix} \frac{\cos(q_1 + q_2)}{\sin(q_2)} & \frac{\sin(q_1 + q_2)}{\sin(q_2)} \\ -\frac{\cos(q_1 + q_2) + \cos(q_1)}{\sin(q_2)} & -\frac{\sin(q_1 + q_2) + \sin(q_1)}{\sin(q_2)} \end{pmatrix}$$

Algoritmo del gradiente

```
q_i = [-1; -1]; % condizioni iniziali
rRif = [1; 1]; % riferimento
[q_f_grad, iter_grad, deltaQ_grad, currentError_grad] = algGradiente(q_i, rRif);
qEnd = q_f_grad(:, end)
```

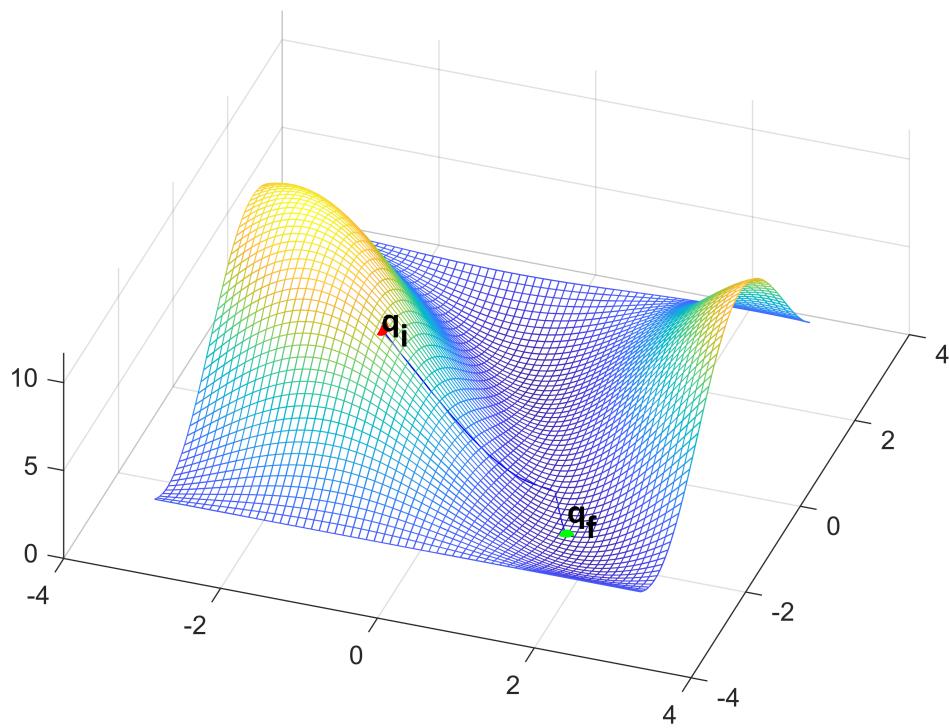
qEnd = 2x1
1.5707
-1.5706

```
fr = [cos(qEnd(1))+cos(qEnd(1)+qEnd(2)); +sin(qEnd(1))+sin(qEnd(1)+qEnd(2))]
```

fr = 2x1
1.0001
1.0001

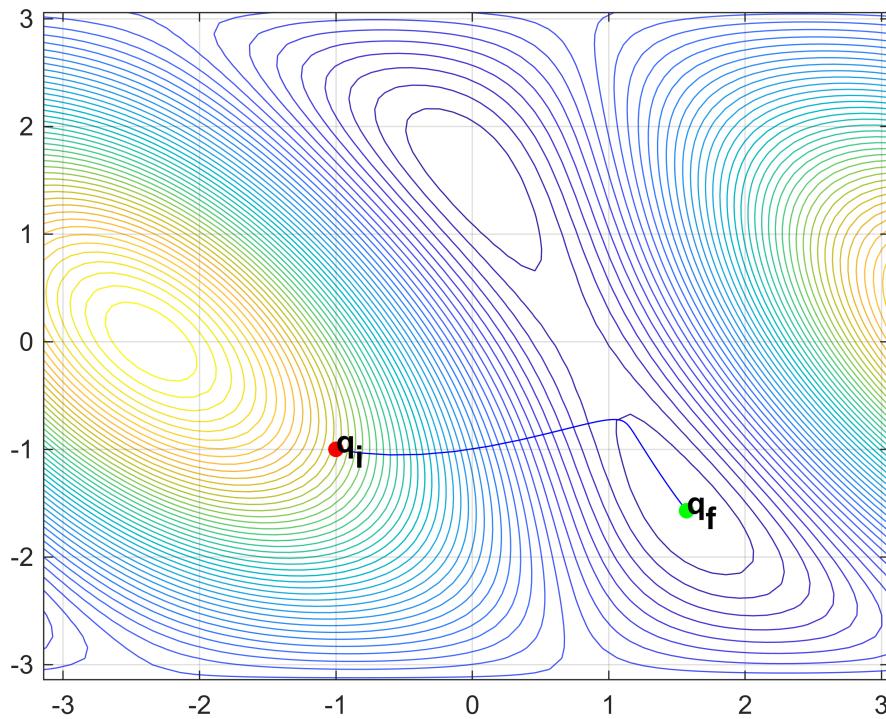
3D plot

```
plot3Dtraj(q_f_grad, rRif, 'b')
```



2D plot

```
plot2Dtraj(q_f_grad, rRif, 'b')
```



Algoritmo di Newton

```
q_i = [-1; -1]; % condizioni iniziali
rRif = [1; 1]; % riferimento
[q_f_newton, iter_newton, deltaQ_newton, currentError_newton] = algNewton(q_i, rRif);
```

```
iter = 103
deltaQ = 9.9043e-06
currentError = 6.8587e-05
```

```
qEnd = q_f_newton(:, end)
```

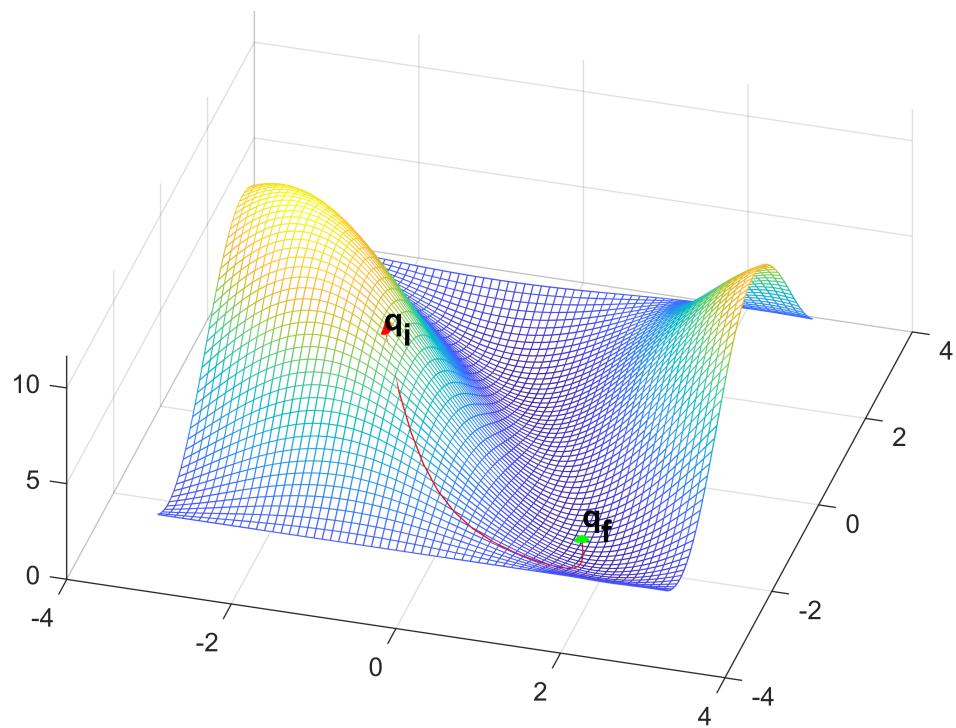
```
qEnd = 2×1
1.5708
-1.5709
```

```
fr = [cos(qEnd(1))+cos(qEnd(1)+qEnd(2)); +sin(qEnd(1))+sin(qEnd(1)+qEnd(2))]
```

```
fr = 2×1
1.0000
0.9999
```

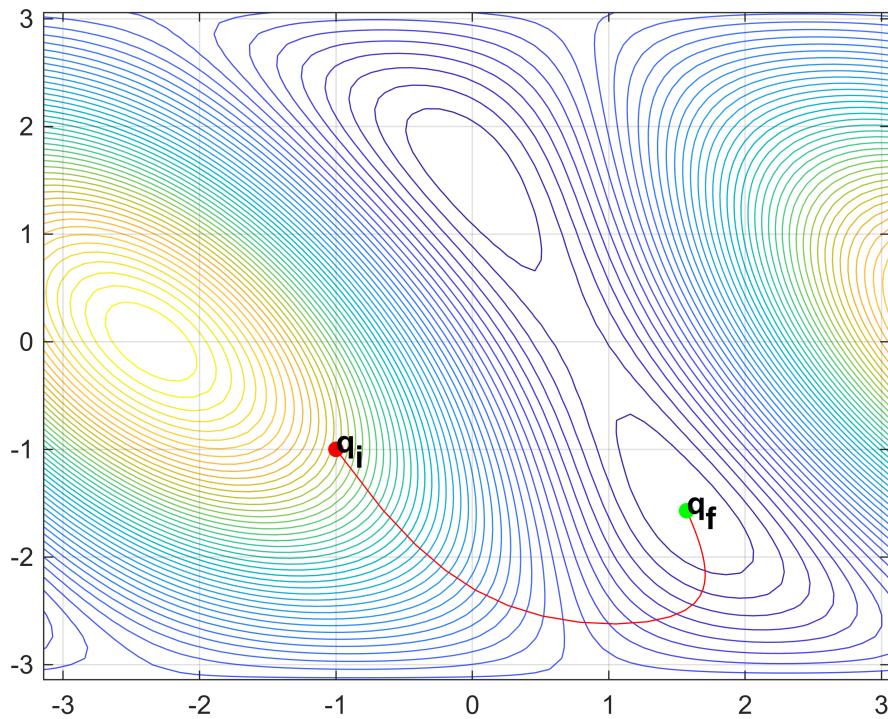
3D plot

```
plot3Dtraj(q_f_newton, rRif, 'r')
```



2D plot

```
plot2Dtraj(q_f_newton, rRif, 'r')
```



Confronto

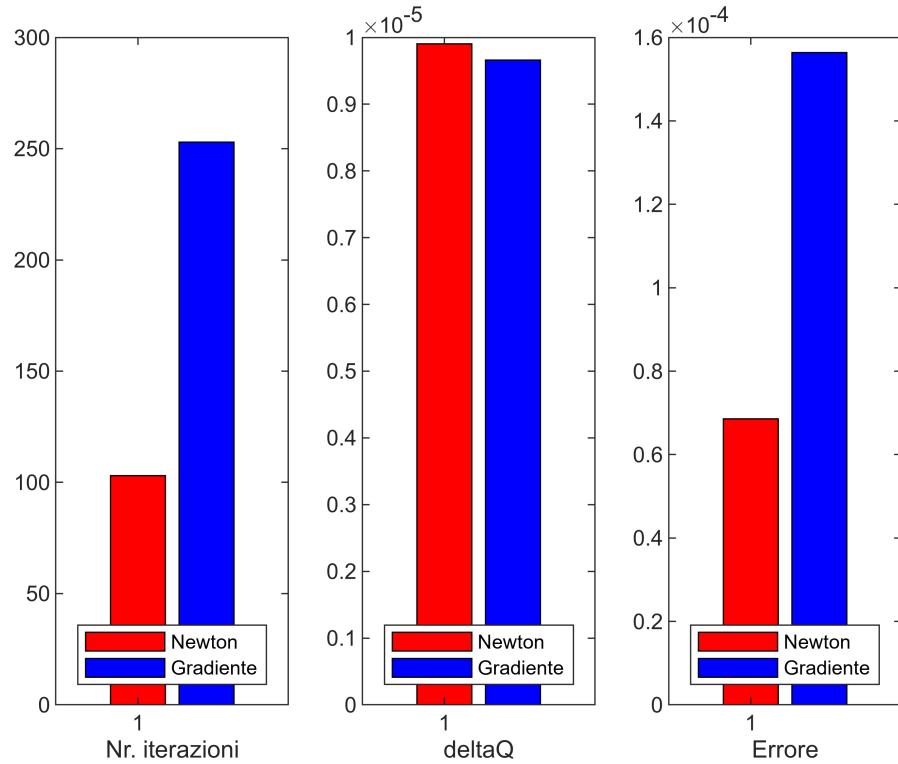
```

figure;
subplot(1,3,1);
b = bar(1, iter_newton, 'FaceColor', 'r');
hold on
b = bar(2, iter_grad, 'FaceColor', 'b');
legend('Newton','Gradiente')
xlabel('Nr. iterazioni');
legend('Location', 'south');

subplot(1,3,2);
b = bar(1, deltaQ_newton, 'FaceColor', 'r');
hold on
b = bar(2, deltaQ_grad, 'FaceColor', 'b');
legend('Newton','Gradiente')
xlabel('deltaQ');
legend('Location', 'south');

subplot(1,3,3);
b = bar(1, currentError_newton, 'FaceColor', 'r');
hold on
b = bar(2, currentError_grad, 'FaceColor', 'b');
legend('Newton','Gradiente')
xlabel('Errore');
legend('Location', 'south');

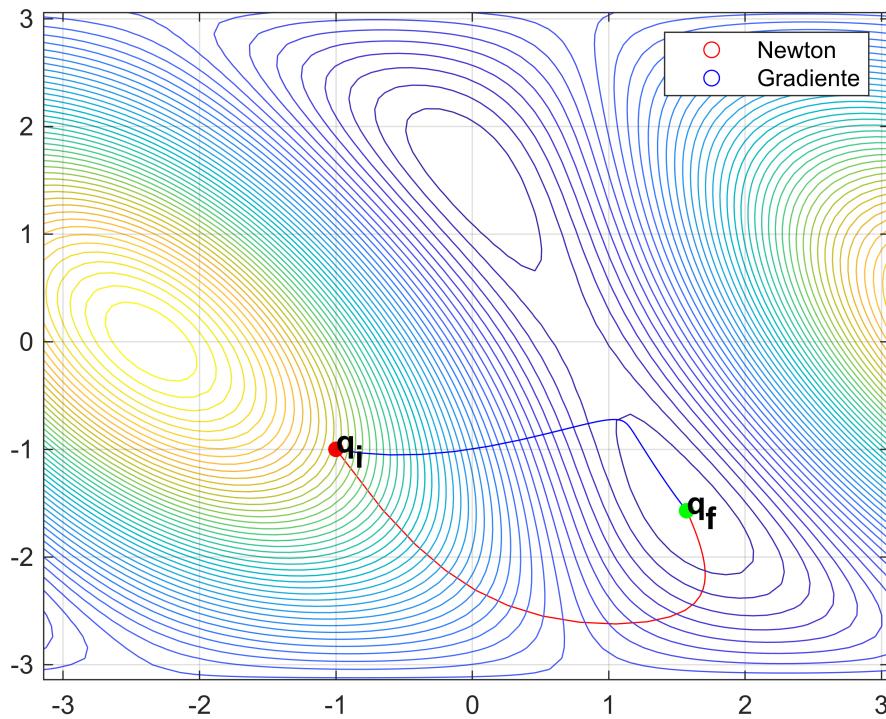
```



```

figure
plot2Dtraj(q_f_newton, rRif, 'r')
hold on
plot2Dtraj(q_f_grad, rRif, 'b')
hold on
h = zeros(2, 1);
h(1) = plot(NaN,NaN,'or');
h(2) = plot(NaN,NaN,'ob');
legend(h,'Newton','Gradiente')
hold off

```



Studio punti estremanti

```
e2v = (rRif-r).^2;
e2 = e2v(1) + e2v(2)
```

$$e2 = (\cos(q_1 + q_2) + \cos(q_1) - 1)^2 + (\sin(q_1 + q_2) + \sin(q_1) - 1)^2$$

```
g = gradient(e2, [q1, q2])
```

$$\begin{aligned} g = \\ \left(\begin{array}{l} 2(\cos(q_1 + q_2) + \cos(q_1))\sigma_1 - 2(\sin(q_1 + q_2) + \sin(q_1))\sigma_2 \\ 2\cos(q_1 + q_2)\sigma_1 - 2\sin(q_1 + q_2)\sigma_2 \end{array} \right) \end{aligned}$$

where

$$\sigma_1 = \sin(q_1 + q_2) + \sin(q_1) - 1$$

$$\sigma_2 = \cos(q_1 + q_2) + \cos(q_1) - 1$$

```
sol = solve(g==0, [q1, q2])
```

```
sol = struct with fields:
    q1: [5x1 sym]
    q2: [5x1 sym]
```

```
estr = [sol.q1, sol.q2]
```

```
estr =
```

$$\begin{pmatrix} 0 & \frac{\pi}{2} \\ \frac{\pi}{4} & 0 \\ -\frac{3\pi}{4} & 0 \\ \frac{\pi}{2} & -\frac{\pi}{2} \\ \frac{\pi}{4} & \pi \end{pmatrix}$$

```
H = hessian(e2, [q1, q2])
```

```
H =
```

$$\begin{pmatrix} 2\sigma_3^2 - 2\sigma_3(\cos(q_1 + q_2) + \cos(q_1) - 1) - 2\sigma_2(\sin(q_1 + q_2) + \sin(q_1) - 1) + 2\sigma_2^2 & \sigma_1 \\ \sigma_1 & 2\cos(q_1 + q_2)^2 - \sigma_4 \end{pmatrix}$$

where

$$\sigma_1 = 2\sin(q_1 + q_2)\sigma_2 - \sigma_4 + 2\cos(q_1 + q_2)\sigma_3 - \sigma_5$$

$$\sigma_2 = \sin(q_1 + q_2) + \sin(q_1)$$

$$\sigma_3 = \cos(q_1 + q_2) + \cos(q_1)$$

$$\sigma_4 = 2\sin(q_1 + q_2)(\sin(q_1 + q_2) + \sin(q_1) - 1)$$

$$\sigma_5 = 2\cos(q_1 + q_2)(\cos(q_1 + q_2) + \cos(q_1) - 1)$$

```
nrSol = size(estr, 1)
```

```
nrSol = 5
```

```
for i=1:nrSol
    Hp = subs(H, [q1, q2], estr(i, :));
    %disp("Autovalori soluzione " + i)
    autovalori = simplify(eig(Hp));

    if(autovalori > 0)
        % hessiana definita positiva => punto di minimo
        estr(i, :)
        disp("è punto di minimo")
    elseif(autovalori < 0)
        % hessiana definita negativa => punto di massimo
```

```

estr(i, :)
disp(" è punto di massimo")
elseif(any(autovalori > 0) && any(autovalori < 0))
    % hessiana indefinita => punto di sella
    estr(i, :)
    disp(" è punto di sella")
else
    % hessiana semidefinita positiva/negativa => punto estremante
    % indefinito
    estr(i, :)
    disp(" è punto estremante indefinito")
end
end

```

ans =

$$\left(0 \quad \frac{\pi}{2}\right)$$

è punto di minimo
ans =

$$\left(\frac{\pi}{4} \quad 0\right)$$

è punto di sella
ans =

$$\left(-\frac{3\pi}{4} \quad 0\right)$$

è punto di massimo
ans =

$$\left(\frac{\pi}{2} \quad -\frac{\pi}{2}\right)$$

è punto di minimo
ans =

$$\left(\frac{\pi}{4} \quad \pi\right)$$

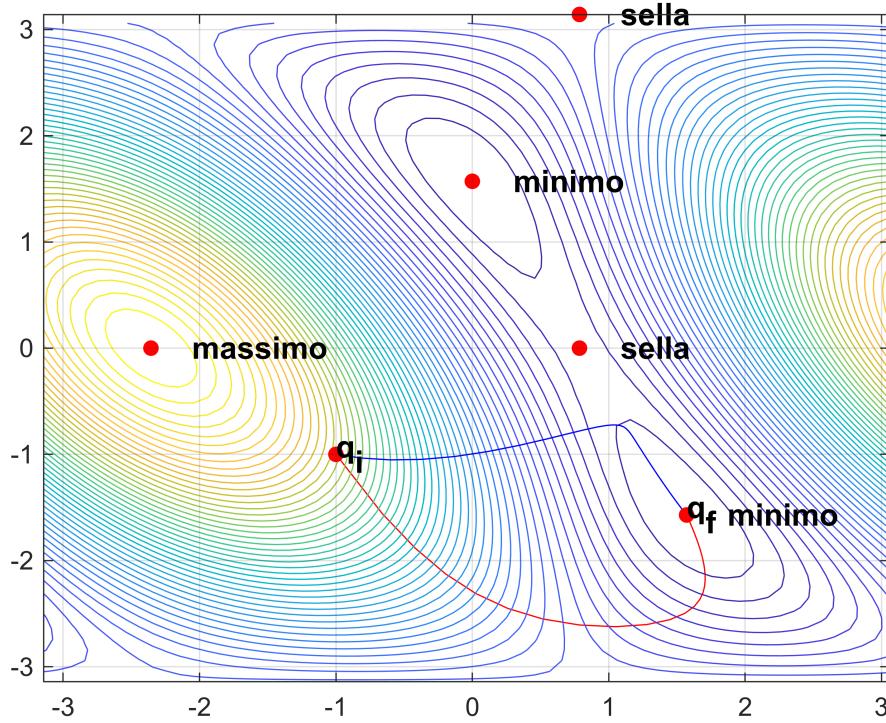
è punto di sella

```

figure
plot2Dtraj(q_f_newton, rRif, 'r')
hold on
plot2Dtraj(q_f_grad, rRif, 'b')
hold on
mappa = ["minimo", "sella", "massimo", "minimo", "sella"];

for i=1:nrSol
    q_estr = estr(i, :);
    scatter(q_estr(1), q_estr(2), 'filled', 'MarkerFaceColor', 'r');
    text(q_estr(1)+0.3, q_estr(2), mappa(i), 'FontSize', 12, 'FontWeight', 'bold');
end
legend off
hold off

```



Le singolarità si ottengono per $\det(\mathbf{J}) = 0$, e dato che

```
detJr = simplify(det(Jr))
```

```
detJr = sin(q2)
```

si hanno singolarità per $q_2 = 0 \vee q_2 = \pi$. I punti di sella e di massimo trovati sono tutti in singolarità.

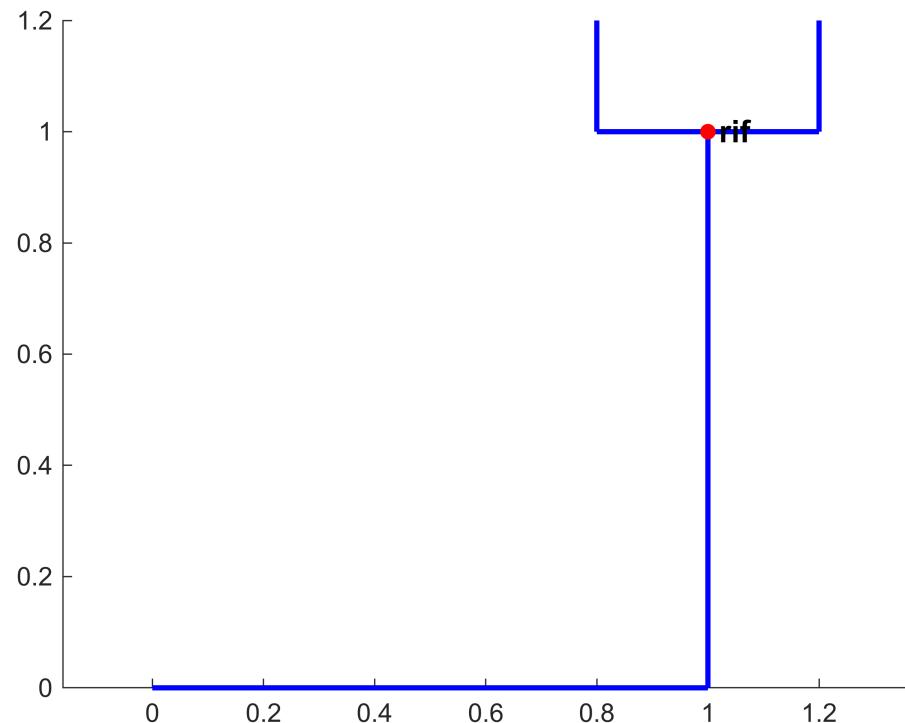
```
for i=1:nrSol
    Jrq = subs(Jr, [q1, q2], estr(i, :));
    disp("Nucleo di Jr: ")
    nucleo = null(Jrq)
    display("Rango di Jr per il punto estremante " + mat2str(string(estr(i, :))) ...
        + " di " + mappa(i) + ": rank(Jr)=" + rank(Jrq))
    figure
    drawPlanarRobot([1,1],estr(i, :))
    scatter(rRif(1), rRif(2), 'filled', 'MarkerFaceColor', 'r');
    text(rRif(1)+0.02, rRif(2), "rif", 'FontSize', 12, 'FontWeight', 'bold');
    axis equal
end
```

Nucleo di Jr:

nucleo =

Empty sym: 2-by-0

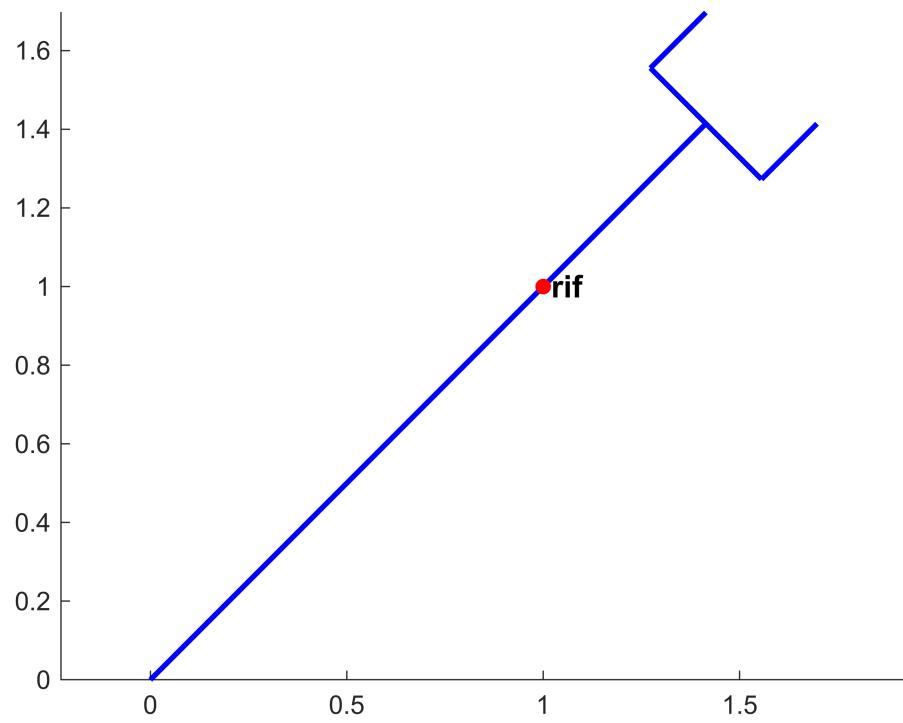
"Rango di Jr per il punto estremante $[0 \pi/2]$ di minimo: $\text{rank}(Jr)=2$ "



Nucleo di J_r :
nucleo =

$$\begin{pmatrix} -\frac{1}{2} \\ 1 \end{pmatrix}$$

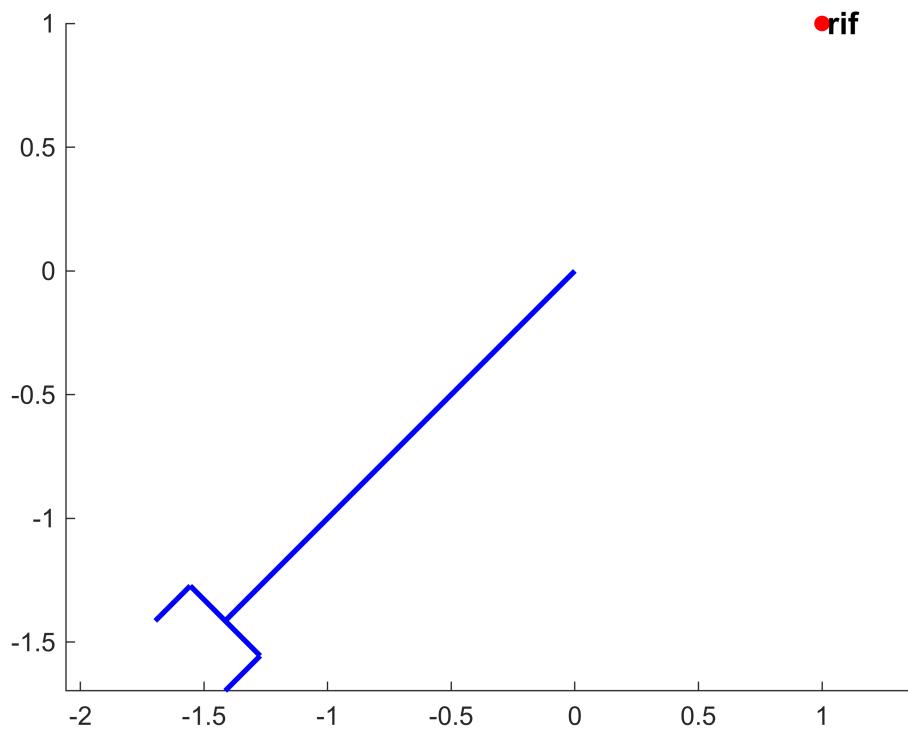
"Rango di J_r per il punto estremante $[\pi/4 \ 0]$ di sella: $\text{rank}(Jr)=1$ "



Nucleo di \mathbb{J}_r :
nucleo =

$$\begin{pmatrix} -\frac{1}{2} \\ 1 \end{pmatrix}$$

"Rango di \mathbb{J}_r per il punto estremante $[-(3\pi)/4, 0]$ di massimo: rank(\mathbb{J}_r)=1"

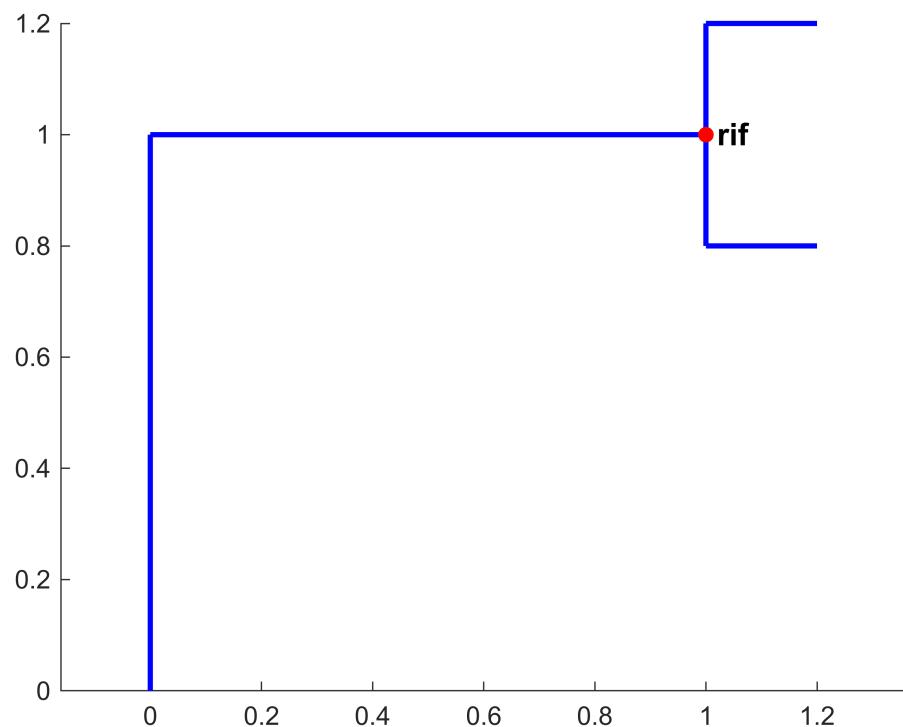


Nucleo di Jr :

nucleo =

Empty sym: 2-by-0

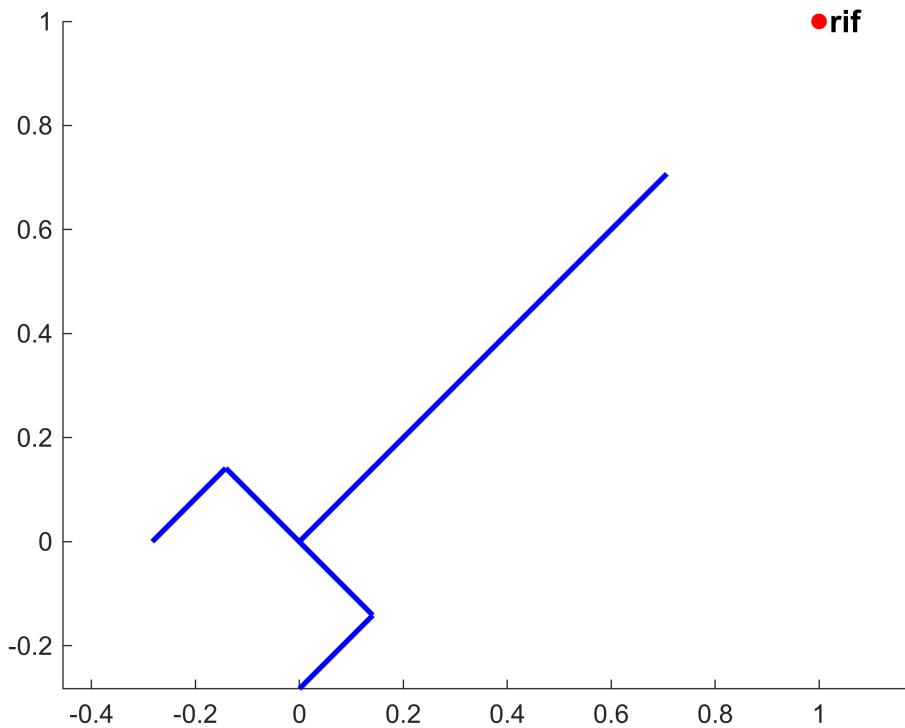
"Rango di Jr per il punto estremante $[\pi/2 \ -\pi/2]$ di minimo: rank(Jr)=2"



Nucleo di Jr :
nucleo =

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

"Rango di Jr per il punto estremante $[\pi/4 \ \pi]$ di sella: $\text{rank}(Jr)=1$ "



Test con punti estremanti come condizione iniziale

Algoritmo del gradiente

```
rRif = [1; 1]; % riferimento
for i=1:nrSol
    q_i = estr(i, :); % condizioni iniziali
    display("Condizione iniziale: q_i=" + mat2str(string(estr(i, :))))
    [q_f_grad, iter_grad, deltaQ_grad, currentError_grad] = algGradiente(q_i, rRif)
    qEnd = q_f_grad(:, end)
    fr = [cos(qEnd(1))+cos(qEnd(1)+qEnd(2)); +sin(qEnd(1))+sin(qEnd(1)+qEnd(2))]
    %plot3Dtraj(q_f_grad, rRif)
    %figure(i)
    %plot2Dtraj(q_f_grad, rRif)
    %hold off
end
```

```
"Condizione iniziale: q_i=[ "0" "pi/2"]"
```

```
q_f_grad =
```

$$\begin{pmatrix} 0 & 0 \\ \frac{\pi}{2} & \frac{\pi}{2} \end{pmatrix}$$

```
iter_grad = 2
```

```
deltaQ_grad = 0
```

```
currentError_grad = 0
```

```
qEnd =
```

```


$$\begin{pmatrix} 0 \\ \frac{\pi}{2} \end{pmatrix}$$

fr =

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

"Condizione iniziale: q_i=[ "pi/4" "0" ]"
q_f_grad =

$$\begin{pmatrix} \frac{\pi}{4} & \frac{\pi}{4} \\ 0 & 0 \end{pmatrix}$$

iter_grad = 2
deltaQ_grad = 0
currentError_grad =

$$\sqrt{2 (\sqrt{2} - 1)^2}$$

qEnd =

$$\begin{pmatrix} \frac{\pi}{4} \\ 0 \end{pmatrix}$$

fr =

$$\begin{pmatrix} \sqrt{2} \\ \sqrt{2} \end{pmatrix}$$

"Condizione iniziale: q_i=[ "-(3*pi)/4" "0" ]"
q_f_grad =

$$\begin{pmatrix} -\frac{3\pi}{4} & -\frac{3\pi}{4} \\ 0 & 0 \end{pmatrix}$$

iter_grad = 2
deltaQ_grad = 0
currentError_grad =

$$\sqrt{2 (\sqrt{2} + 1)^2}$$

qEnd =

$$\begin{pmatrix} -\frac{3\pi}{4} \\ 0 \end{pmatrix}$$

fr =

$$\begin{pmatrix} -\sqrt{2} \\ -\sqrt{2} \end{pmatrix}$$

"Condizione iniziale: q_i=[ "pi/2" "-pi/2" ]"
q_f_grad =

```

$$\begin{pmatrix} \frac{\pi}{2} & \frac{\pi}{2} \\ -\frac{\pi}{2} & -\frac{\pi}{2} \end{pmatrix}$$

iter_grad = 2
 deltaQ_grad = 0
 currentError_grad = 0
 qEnd =

$$\begin{pmatrix} \frac{\pi}{2} \\ -\frac{\pi}{2} \end{pmatrix}$$

 fr =

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

 "Condizione iniziale: q_i=["pi/4" "pi"]"
 q_f_grad =

$$\begin{pmatrix} \frac{\pi}{4} & \frac{\pi}{4} \\ \pi & \pi \end{pmatrix}$$

 iter_grad = 2
 deltaQ_grad = 0
 currentError_grad = $\sqrt{2}$
 qEnd =

$$\begin{pmatrix} \frac{\pi}{4} \\ \pi \end{pmatrix}$$

 fr =

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Agoritmo di Newton

```
rRif = [1; 1]; % riferimento
for i=1:nrSol
    q_i = estr(i, :); % condizioni iniziali
    display("Condizione iniziale: q_i=" + mat2str(string(estri(i, :))))
    [q_f_newton, iter_newton, deltaQ_newton, currentError_newton] = algNewton(q_i, rRif);
    qEnd = q_f_newton(:, end)
    fr = [cos(qEnd(1))+cos(qEnd(1)+qEnd(2)); +sin(qEnd(1))+sin(qEnd(1)+qEnd(2))]
    %plot3Dtraj(q_f_newton, rRif)
    %figure
    %plot2Dtraj(q_f_newton, rRif)
    %hold off
end

"Condizione iniziale: q_i=[ "0" "pi/2"]"
iter = 2
```

```

deltaQ = 0
currentError = 0
qEnd =

$$\begin{pmatrix} 0 \\ \frac{\pi}{2} \end{pmatrix}$$

fr =

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

"Condizione iniziale: q_i=[\"pi/4\" \"0\"]"
iter = 3
deltaQ = NaN
currentError = NaN
qEnd =

$$\begin{pmatrix} \text{NaN} \\ \text{NaN} \end{pmatrix}$$

fr =

$$\begin{pmatrix} \text{NaN} \\ \text{NaN} \end{pmatrix}$$

"Condizione iniziale: q_i=[\"-(3*pi)/4\" \"0\"]"
iter = 3
deltaQ = NaN
currentError = NaN
qEnd =

$$\begin{pmatrix} \text{NaN} \\ \text{NaN} \end{pmatrix}$$

fr =

$$\begin{pmatrix} \text{NaN} \\ \text{NaN} \end{pmatrix}$$

"Condizione iniziale: q_i=[\"pi/2\" \"-pi/2\"]"
iter = 2
deltaQ = 0
currentError = 0
qEnd =

$$\begin{pmatrix} \frac{\pi}{2} \\ -\frac{\pi}{2} \end{pmatrix}$$

fr =

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

"Condizione iniziale: q_i=[\"pi/4\" \"pi\"]"
iter = 2
deltaQ = NaN
currentError =  $\sqrt{2}$ 
qEnd =

```

$\begin{pmatrix} -\infty \\ \text{NaN} \end{pmatrix}$

fr =

$\begin{pmatrix} \text{NaN} \\ \text{NaN} \end{pmatrix}$

```

function [q_f, iter, deltaQ, currentError] = algNewton(condIniz,
riferimento)
% iperparametri
maxIter = 1000; % massimo numero di iterazioni
minQ = 0.00001; % minima variazione delle variabili di giunto
minE = 0.00001; % distanza minima dal punto di arrivo
alpha = 0.1; % passo

% inizializzazione variabili
q_i = condIniz; % condizioni iniziali
rRif = riferimento; % riferimento
iter = 1; % contatore per le iterazioni
deltaQ = realmax; % variazione delle variabili di giunto in modulo
currentError = realmax; % errore corrente
q_f = q_i; % andamento di q, ultimo valore = valore finale, primo
valore = cond. iniz.

% passo iterativo
while(iter < maxIter && deltaQ > minQ && currentError > minE)
    q = q_f(:, end);
    q_1 = q(1);
    q_2 = q(2);
    Jinv = 1/sin(q_2)*[cos(q_1+q_2), sin(q_1+q_2); -cos(q_1)-cos(q_1+
q_2), -sin(q_1)-sin(q_1+q_2)];
    fr = [cos(q_1)+cos(q_1+q_2); +sin(q_1)+sin(q_1+q_2)];
    newQ = q + alpha*Jinv*(rRif-fr);
    q_f = [q_f, newQ];
    deltaQ = norm(newQ-q);
    currentError = norm(rRif-fr);
    iter = iter + 1;
end
iter
deltaQ
currentError
end

```

Code 9.1: Algoritmo di Newton

```

function [q_f, iter, deltaQ, currentError] = algGradiente(condIniz,
riferimento)
% iperparametri
maxIter = 1000; % massimo numero di iterazioni
minQ = 0.00001; % minima variazione delle variabili di giunto
minE = 0.00001; % distanza minima dal punto di arrivo

```

```

alpha = 0.1; % passo

% inizializzazione variabili
q_i = condIniz; % condizioni iniziali
rRif = riferimento; % riferimento
iter = 1; % contatore per le iterazioni
deltaQ = realmax; % variazione delle variabili di giunto in modulo
currentError = realmax; % errore corrente
q_f = q_i; % andamento di q, ultimo valore = valore finale, primo
valore = cond. iniz.

% passo iterativo
while(iter < maxIter && deltaQ > minQ && currentError > minE)
    q = q_f(:, end);
    q_1 = q(1);
    q_2 = q(2);
    JT = [-sin(q_1)-sin(q_1+q_2), cos(q_1)+cos(q_1+q_2); -sin(q_1+q_2)
), cos(q_1+q_2)];
    fr = [cos(q_1)+cos(q_1+q_2); +sin(q_1)+sin(q_1+q_2)];
    newQ = q + alpha*JT*(rRif-fr);
    q_f = [q_f, newQ];
    deltaQ = norm(newQ-q);
    currentError = norm(rRif-fr);
    iter = iter + 1;
end
end

```

Code 9.2: Algoritmo del gradiente

```

function [] = plot2Dtraj(q_f, rRif, color)
if ~exist('color','var')
    % third parameter does not exist, so default it to something
    color = 'b';
end

% plot delle curve di livello della norma dell'errore cartesiano al
quadrato
[q11, q22] = meshgrid(-pi:.1:pi,-pi:.1:pi);
f1 = cos(q11)+cos(q11+q22);
f2 = sin(q11)+sin(q11+q22);
e1 = rRif(1)-f1;
e2 = rRif(2)-f2;
n_e = e1.^2+e2.^2; % norma dell'errore cartesiano al quadrato
figure(2)

```

```

contour(q11, q22, n_e, 50) % curve di livello
grid on;
hold on;
% plot traiettoria di q
hold on
f1 = cos(q_f(1,:))+cos(q_f(1,:)+q_f(2,:));
f2 = sin(q_f(1,:))+sin(q_f(1,:)+q_f(2,:));
e1 = rRif(1)-f1;
e2 = rRif(2)-f2;
zTra = e1.^2+e2.^2;
plot(q_f(1,:),q_f(2,:), 'Color', color)

% disegna condizione iniziale
scatter(q_f(1,1), q_f(2,1), 'filled', 'MarkerFaceColor', 'r');
text(q_f(1,1), q_f(2,1), zTra(1), 'q_i', 'FontSize', 12, 'FontWeight',
, 'bold');

% disegna la soluzione finale
scatter(q_f(1,end), q_f(2,end), 'filled', 'MarkerFaceColor', 'g');
text(q_f(1,end), q_f(2,end), 'q_f', 'FontSize', 12, 'FontWeight',
bold);
hold off
end

```

Code 9.3: Disegna la traiettoria q_f nelle curve di livello della norma dell'errore al quadrato

```

function [] = plot3Dtraj(q_f, rRif, color)
% plot 3D della traeittoria di q per il robot RR planare

if ~exist('color','var')
    % third parameter does not exist, so default it to something
    color = 'b';
end

% plot norma dell'errore cartesiano al quadrato
[q11, q22] = meshgrid(-pi:.1:pi,-pi:.1:pi);
f1 = cos(q11)+cos(q11+q22);
f2 = sin(q11)+sin(q11+q22);
e1 = rRif(1)-f1;
e2 = rRif(2)-f2;
n_e = e1.^2+e2.^2; % norma dell'errore cartesiano al quadrato
figure()
mesh(q11,q22,n_e)

```

```

% plot traiettoria di q
hold on
f1 = cos(q_f(1,:))+cos(q_f(1,:)+q_f(2,:));
f2 = sin(q_f(1,:))+sin(q_f(1,:)+q_f(2,:));
e1 = rRif(1)-f1;
e2 = rRif(2)-f2;
zTra = e1.^2+e2.^2;
plot3(q_f(1,:),q_f(2,:),zTra, 'Color', color)

% disegna condizione iniziale
scatter3(q_f(1,1), q_f(2,1), zTra(1), 'filled', 'MarkerFaceColor', 'r');
text(q_f(1,1), q_f(2,1), zTra(1), 'q_i', 'FontSize', 12, 'FontWeight',
, 'bold');

% disegna la soluzione finale
scatter3(q_f(1,end), q_f(2,end), zTra(end), 'filled', 'MarkerFaceColor', 'g');
text(q_f(1,end), q_f(2,end), zTra(end)+1, 'q_f', 'FontSize', 12, 'FontWeight',
, 'bold');
hold off
end

```

Code 9.4: Disegna la traiettoria q_f nel grafico 3D della norma dell'errore al quadrato

10. Esercitazione 10: cinematica inversa analitica e numerica robot RRP

(10) *Cinematica inversa analitica e numerica del robot polare RRP.*

10.1 Descrizione del problema

La cinematica inversa analitica prevede di trovare in modo analitico le variabili di giunto q che realizzano la posa assegnata \mathbf{T} . Quella numerica invece prevede l'uso dell'algoritmo del gradiente o di Newton.

10.2 Soluzione

10.2.1 Soluzione analitica

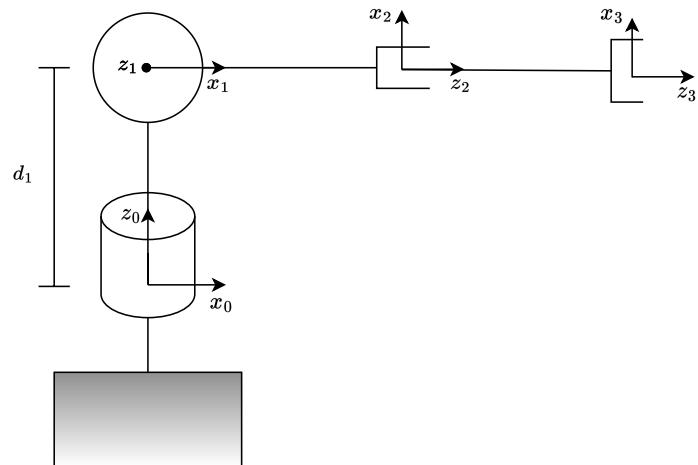


Figura 10.1: Manipolatore polare RRP

Braccio	a_i	α_i	d_i	θ_i
1	0	$\frac{\pi}{2}$	d_1	q_1
2	0	$\frac{\pi}{2}$	0	$q_2 + \frac{\pi}{2}$
3	0	0	q_3	0

Tabella 10.1: Parametri DH per il manipolatore RRP polare

$$\begin{aligned}
{}^0\mathbf{T}_1 &= \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^1\mathbf{T}_2 &= \begin{bmatrix} -s_2 & 0 & c_2 & 0 \\ c_2 & 0 & s_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^2\mathbf{T}_3 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^0\mathbf{T}_3 &= \begin{bmatrix} -c_1 s_2 & s_1 & c_1 c_2 & q_3 c_1 c_2 \\ -s_1 s_2 & -c_1 & s_1 c_2 & q_3 s_1 c_2 \\ c_2 & 0 & s_2 & d_1 + q_3 s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} q_3 c_1 s_2 \\ q_3 s_1 s_2 \\ d_1 + q_3 c_2 \end{bmatrix}$$

Innanzitutto conviene eliminare l'offset d_1 nel seguente modo:

$$P_z - d_1 = q_3 s_2$$

Si prova dunque a quadrare e sommare le componenti

$$\begin{aligned}
P_x^2 + P_y^2 + (P_z - d_1)^2 &= q_3^2(c_2^2 c_1^2 + c_2^2 s_1^2 + s_2^2) = \\
&= q_3^2(c_2^2(c_1^2 + s_1^2) + s_2^2) = q_3^2(c_2^2 + s_2^2) = q_3^2
\end{aligned}$$

Quindi essendo

$$P_x^2 + P_y^2 + (P_z - d_1)^2 = q_3^2$$

si ha

$$q_3 = \pm \sqrt{P_x^2 + P_y^2 + (P_z - d_1)^2}$$

Se:

- $q_3 = 0$:

q_1 e q_2 sono indeterminati dato che si ha lo stesso \underline{P} $\forall q_1, q_2$ (si è in una configurazione singolare).

- $q_3 \neq 0$:

Si ha

$$s_2 = \frac{P_z - d_1}{q_3}$$

e

$$\begin{aligned} c_2 &= \pm \sqrt{1 - s_2^2} = \pm \sqrt{1 - \frac{(P_z - d_1)^2}{q_3^2}} = \pm \sqrt{\frac{q_3^2 - (P_z - d_1)^2}{q_3^2}} = \\ &= \pm \frac{1}{q_3} \sqrt{q_3^2 - (P_z - d_1)^2} = \pm \frac{1}{q_3} \sqrt{P_x^2 + P_y^2} \end{aligned}$$

e quindi

$$q_2 = \text{atan2}(s_2, c_2)$$

Inoltre essendo

$$\begin{cases} P_x = q_3 c_2 c_1 \\ P_y = q_3 c_2 s_1 \end{cases}$$

e quando $c_2 \neq 0 \Leftrightarrow P_x^2 + P_y^2 \neq 0$ si può dire che

$$\begin{cases} c_1 = \frac{P_x}{q_3 c_2} \\ s_1 = \frac{P_y}{q_3 c_2} \end{cases}$$

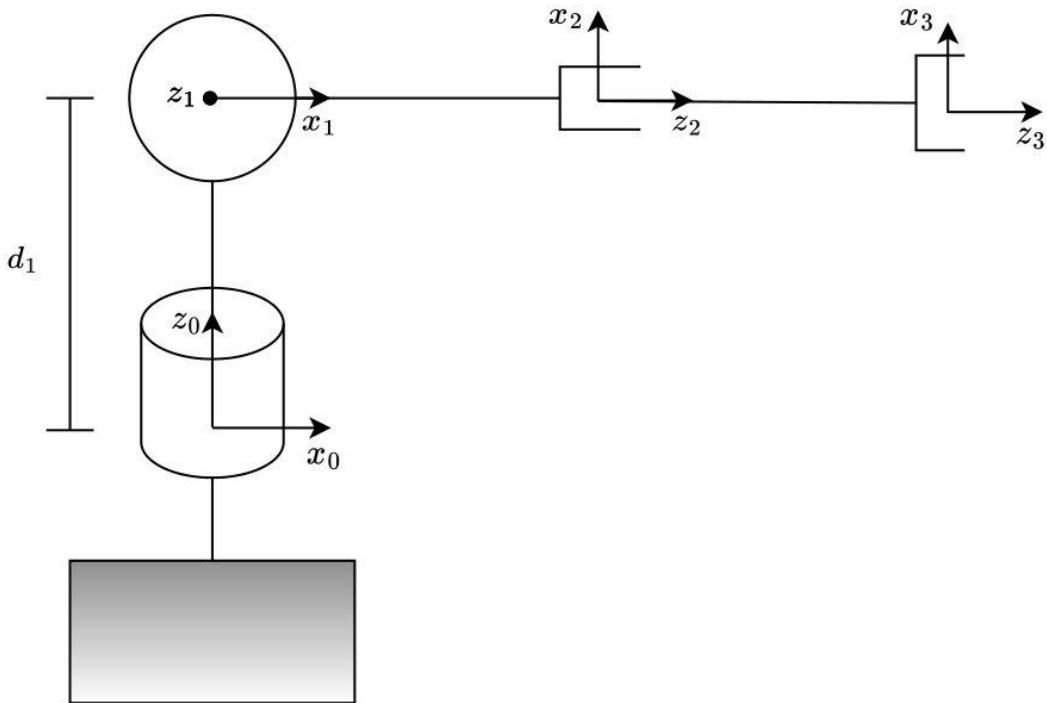
e quindi

$$q_1 = \text{atan2}(s_1, c_1)$$

10.2.2 Soluzione numerica

Cinematica inversa del robot RRP polare

Robot polare RRP



```
syms q1 q2 q3 d1
DHRRPpolare = [0, pi/2, d1, q1;
                0, pi/2, 0, q2+pi/2;
                0, 0, q3, 0]
```

```
DHRRPpolare =

$$\begin{pmatrix} 0 & \frac{\pi}{2} & d_1 & q_1 \\ 0 & \frac{\pi}{2} & 0 & q_2 + \frac{\pi}{2} \\ 0 & 0 & q_3 & 0 \end{pmatrix}$$

```

```
tList = cinDirDH(DHRRPpolare);
T03 = tList{4}
```

```
T03 =

$$\begin{pmatrix} -\cos(q_1)\sin(q_2) & \sin(q_1) & \cos(q_1)\cos(q_2) & q_3\cos(q_1)\cos(q_2) \\ -\sin(q_1)\sin(q_2) & -\cos(q_1) & \cos(q_2)\sin(q_1) & q_3\cos(q_2)\sin(q_1) \\ \cos(q_2) & 0 & \sin(q_2) & d_1 + q_3\sin(q_2) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```

```
r = T03(1:3, 4)
```

```
r =
```

$$\begin{pmatrix} q_3 \cos(q_1) \cos(q_2) \\ q_3 \cos(q_2) \sin(q_1) \\ d_1 + q_3 \sin(q_2) \end{pmatrix}$$

```
Jr = jacobian(r, [q1, q2, q3])
```

$$Jr = \begin{pmatrix} -q_3 \cos(q_2) \sin(q_1) & -q_3 \cos(q_1) \sin(q_2) & \cos(q_1) \cos(q_2) \\ q_3 \cos(q_1) \cos(q_2) & -q_3 \sin(q_1) \sin(q_2) & \cos(q_2) \sin(q_1) \\ 0 & q_3 \cos(q_2) & \sin(q_2) \end{pmatrix}$$

```
Jr.'
```

$$ans = \begin{pmatrix} -q_3 \cos(q_2) \sin(q_1) & q_3 \cos(q_1) \cos(q_2) & 0 \\ -q_3 \cos(q_1) \sin(q_2) & -q_3 \sin(q_1) \sin(q_2) & q_3 \cos(q_2) \\ \cos(q_1) \cos(q_2) & \cos(q_2) \sin(q_1) & \sin(q_2) \end{pmatrix}$$

Cinematica inversa analitica

```
P = [1; 1; 1];
d_1 = 0.5;
q_an = cinInvAnaliticaRRPpolare(P, d_1)
```

```
q_an = 3x1
0.7854
0.3398
1.5000
```

```
round(subs(r, [q1, q2, q3, d1], [q_an(1), q_an(2), q_an(3), d_1]), 10)
```

```
ans =
1.0
1.0
1.0
```

Cinematica inversa numerica

```
qi = [1; 0.4; 0];
q_num = cinInvNumericaRRPpolare(P, d_1, qi)
```

```
q_num = 3x1
0.7854
0.3398
1.5000
```

```
round(subs(r, [q1, q2, q3, d1], [q_num(1), q_num(2), q_num(3), d_1]), 10)
```

```
ans =
```

$$\begin{pmatrix} 0.9999999946 \\ 0.9999999946 \\ 0.9999999973 \end{pmatrix}$$

```
errore = round(norm(q_an - q_num), 10)
```

```
errore = 8.1000e-09
```

```

function [q] = cinInvAnaliticaRRPpolare(P, d1)
Px = P(1);
Py = P(2);
Pz = P(3);

q3 = sqrt(Px^2 + Py^2 + (Pz-d1)^2);
if(q3 == 0)
    q1 = NaN;
    q2 = NaN;
else
    s2 = (Pz-d1)/q3;
    c2 = 1/q3*sqrt(Px^2+Py^2);
    q2 = atan2(s2, c2);

    if(c2 == 0)
        q1 = NaN;
    else
        c1 = Px/(q3*c2);
        s1 = Py/(q3*c2);
        q1 = atan2(s1, c1);
    end
end
q = [q1; q2; q3];
end

```

Code 10.1: Cinematica inversa analitica del robot RRP polare

```

function [q] = cinInvNumericaRRPpolare(P, d1, condIniz)
% iperparametri
maxIter = 1000; % massimo numero di iterazioni
minQ = 0.000000001; % minima variazione delle variabili di giunto
minE = 0.000000001; % distanza minima dal punto di arrivo
alpha = 0.1; % passo

% inizializzazione variabili
q_i = condIniz; % condizioni iniziali
rRif = P; % riferimento
iter = 1; % contatore per le iterazioni
deltaQ = realmax; % variazione delle variabili di giunto in modulo
currentError = realmax; % errore corrente
q_f = q_i; % andamento di q, ultimo valore = valore finale, primo
valore = cond. iniz.

% passo iterativo

```

```

while(iter < maxIter && deltaQ > minQ && currentError > minE)
    q = q_f(:, end);
    q_1 = q(1);
    q_2 = q(2);
    q_3 = q(3);
    JT = [-q_3*cos(q_2)*sin(q_1), q_3*cos(q_1)*cos(q_2), sym(0);
           -q_3*cos(q_1)*sin(q_2), -q_3*sin(q_1)*sin(q_2), q_3*cos(
    q_2);
           cos(q_1)*cos(q_2), cos(q_2)*sin(q_1), sin(q_2)];
    fr = [q_3*cos(q_1)*cos(q_2);
           q_3*cos(q_2)*sin(q_1);
           d1 + q_3*sin(q_2)];
    newQ = double(q + alpha*JT*(rRif-fr));
    q_f = [q_f, newQ];
    deltaQ = norm(newQ-q);
    currentError = norm(rRif-fr);
    iter = iter + 1;
end

q = q_f(:, end);
end

```

Code 10.2: Cinematica inversa numerica del robot RRP polare

11. Esercitazione 11: manipolabilità RR

(11) *Analisi di manipolabilità del manipolatore planare RR. Studio della configurazione a massima manipolabilità.*

11.1 Descrizione del problema

Si vuole vedere quanto facilmente il robot possa muoversi nelle varie direzioni dello spazio operativo.

Ricordando che $\dot{\underline{q}} = \mathbf{J}^\#(\underline{q})\underline{v}$ e $\dot{\underline{q}}^T = \underline{v}^T \mathbf{J}^{\#T}(\underline{q})$ e considerando velocità di giunto a norma unitaria $\dot{\underline{q}}^T \dot{\underline{q}} = 1$ si ottiene

$$\dot{\underline{q}}^T \dot{\underline{q}} = \underline{v}^T \mathbf{J}^{\#T}(\underline{q}) \mathbf{J}^\#(\underline{q}) \underline{v} = 1$$

e fissando \underline{q} si ha

$$\underline{v}^T \mathbf{J}^{\#T} \mathbf{J}^\# \underline{v} = 1$$

tale equazione descrive un elissoide in \mathbb{R}^n chiamato ellissoide di manipolabilità. Esso rappresenta una regione nello spazio di lavoro del robot in cui il robot è in grado di raggiungere velocità e orientamenti desiderati.

A punti sulla superficie della sfera nello spazio delle velocità ai giunti corrispondono punti sulla superficie dell'ellissoide nello spazio delle velocità all'organo terminale. Nella direzione dell'asse maggiore dell'ellissoide, l'organo terminale può muoversi a velocità maggiore, mentre nella direzione dell'asse minore si hanno velocità minori. Quanto più l'ellissoide ha forma sferica tanto più il manipolatore può muoversi in maniera identica in tutte le direzioni.

Si vuole caratterizzare la dimensione e la forma dell'elissoide di manipolabilità.

Indicando con $\lambda_i(\mathbf{J}\mathbf{J}^T)$ gli autovalori di $\mathbf{J}\mathbf{J}^T$ si utilizzano come misura i valori

$$\sigma_i(\mathbf{J}) = \sqrt{\lambda_i(\mathbf{J}\mathbf{J}^T)} \in \mathbb{R} \quad \forall i = 1, \dots, m$$

che corrispondono ai valori singolari di \mathbf{J} , essi rappresentano l’“importanza” di ciascuna dimensione presente nella matrice originale. Da notare che l’argomento della radice è ≥ 0 , dato che una matrice $\mathbf{A}\mathbf{A}^T$ è semidefinita positiva e quindi i suoi autovalori sono ≥ 0 .

I valori singolari $\sigma_i(\mathbf{J})$ rappresentano la lunghezza dei semiassi principali dell’elissoide $\underline{v}^T \mathbf{J}^{\#T} \mathbf{J}^\# \underline{v} = 1$. Gli autovettori \underline{u}_i relativi agli autovalori λ_i identificano la retta che supporta il semiasse.

I valori singolari possono essere quindi usati per misurare la grandezza dell’elissoide di manipolabilità. Nel caso in cui $\mathbf{J}\mathbf{J}^T$ è singolare i semiassi si annullano.

Una possibile misura rappresentativa globale dell’abilità di manipolazione è data dal volume dell’elissoide. La quantità

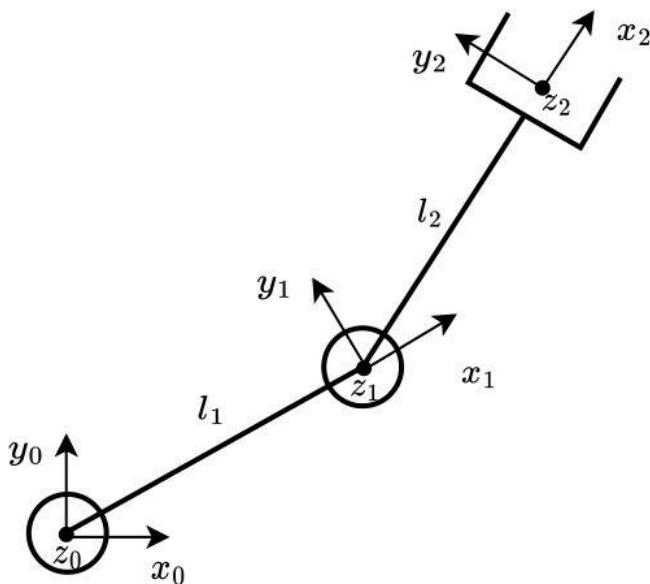
$$w = \sqrt{\det(\mathbf{J}\mathbf{J}^T)} = \prod_{i=1}^m \sigma_i$$

è proporzionale al volume dell’elissoide e prende il nome di misura di manipolabilità.

11.2 Soluzione

Manipolabilità del robot RR planare

Robot RR planare



```
clear
clc
syms q1 q2 l1 l2 real
l1 = 1;
l2 = 1;
DHRRplanare = [l1, 0, 0, q1;
                l2, 0, 0, q2]
```

```
DHRRplanare =

$$\begin{pmatrix} 1 & 0 & 0 & q_1 \\ 1 & 0 & 0 & q_2 \end{pmatrix}$$

```

```
tList = cinDirDH(DHRRplanare);
T03 = tList{3}
```

```
T03 =

$$\begin{pmatrix} \cos(q_1 + q_2) & -\sin(q_1 + q_2) & 0 & \cos(q_1 + q_2) + \cos(q_1) \\ \sin(q_1 + q_2) & \cos(q_1 + q_2) & 0 & \sin(q_1 + q_2) + \sin(q_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```

```
r = T03(1:2, 4);
J = [diff(r(1), q1), diff(r(1), q2);
      diff(r(2), q1), diff(r(2), q2)]
```

```
J =
```

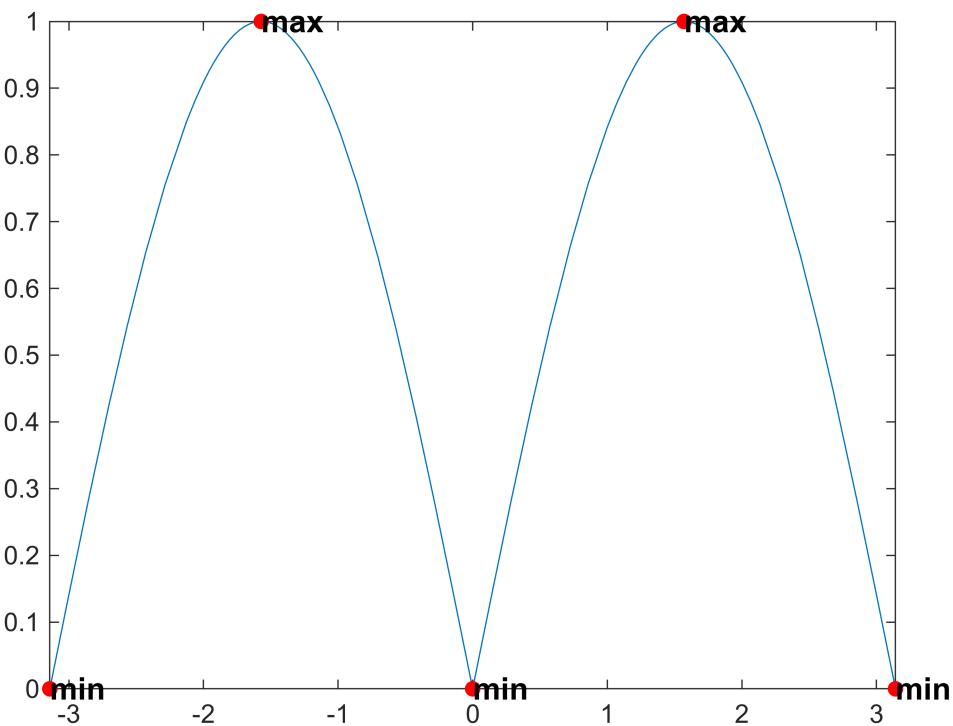
$$\begin{pmatrix} -\sin(q_1 + q_2) & -\sin(q_1 + q_2) \\ \cos(q_1 + q_2) + \cos(q_1) & \cos(q_1 + q_2) \end{pmatrix}$$

Elissoide di manipolabilità

```
w = simplify(sqrt(det(J*J.')))
```

```
w = |sin(q2)|
```

```
figure
fplot(w, [-pi, pi])
hold on
maxManConf = [-pi/2, pi/2];
minManConf = [-pi, 0, pi];
for i=1:size(maxManConf, 2)
    q_2 = maxManConf(i);
    scatter(q_2, subs(w, q2, q_2), 'filled', 'MarkerFaceColor', 'r');
    text(q_2, subs(w, q2, q_2), "max", 'FontSize', 12, 'FontWeight', 'bold');
end
for i=1:size(minManConf, 2)
    q_2 = minManConf(i);
    scatter(q_2, subs(w, q2, q_2), 'filled', 'MarkerFaceColor', 'r');
    text(q_2, subs(w, q2, q_2), "min", 'FontSize', 12, 'FontWeight', 'bold');
end
hold off
```



```

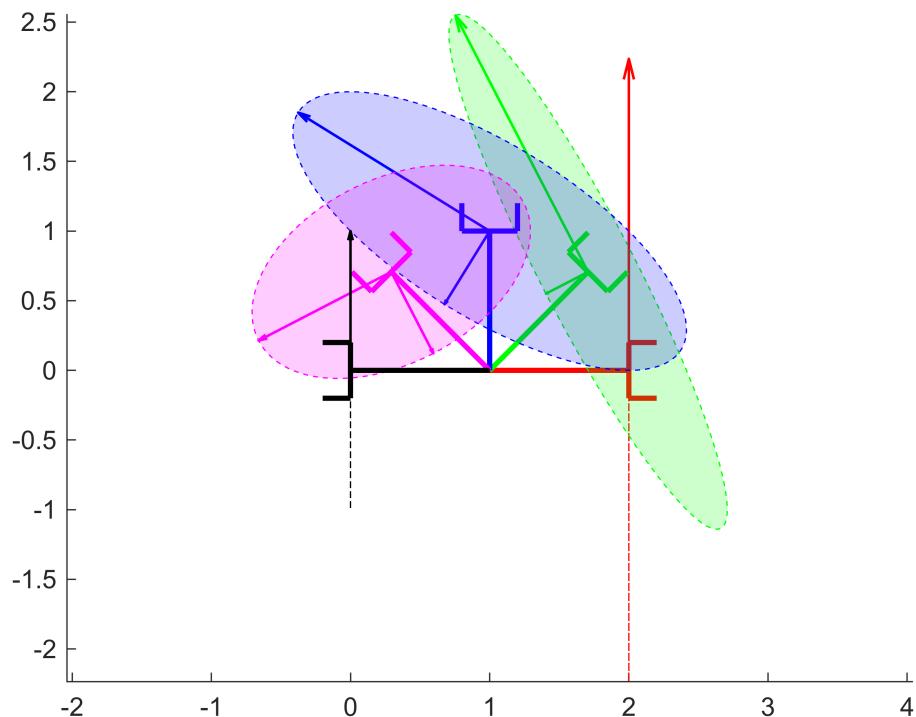
q = linspace(0, pi, 5);
q_1 = 0;
colors = [ 'r', 'g', 'b', 'm', 'k' ];

figure
for i=1:size(q,2)

    q_2 = q(i);
    center = subs(r, [q1, q2], [q_1, q_2]);
    Jq = double(subs(J, [q1, q2], [q_1, q_2]));
    drawManEllipse(Jq, center, colors(i));

    drawPlanarRobot([1, 1], [q_1, q_2], colors(i))
    axis equal
end

```



```

function [] = drawManEllipse(J, center, color)
    [V, D] = eig(J.*J.');
```

% autovettori (colonne di V) e autovalori (diagonale di D)

```

lambda = diag(D); % autovalori
sigma = sqrt(lambda); % valori singolari
%w = sqrt(det(J.*J.'))

```

```

u1 = double(V(1:end, 1)); % semiasse 1
u2 = double(V(1:end, 2)); % semiasse 2

u1 = u1/norm(u1);
u2 = u2/norm(u2);

u1length = double(sigma(1)); % lunghezza semiasse 1
u2length = double(sigma(2)); % lunghezza semiasse 2

% punti dell'ellisse
theta = linspace(0, 2*pi, 100);
x = u1length * cos(theta) * u1(1) + u2length * sin(theta) * u2(1) +
center(1);
y = u1length * cos(theta) * u1(2) + u2length * sin(theta) * u2(2) +
center(2);

% disegna ellisse
hold on
f = fill(x, y, color);
set(f, 'edgecolor', color, 'LineStyle', '--');
alpha(f,.2)
%plot(x, y, '--r', 'LineWidth', 2, 'Color', color);

% disegna assi
quiver(center(1), center(2), u1(1)*u1length, u1(2)*u1length, 0, 'b',
...
'LineWidth', 1, 'Color', color, 'MarkerSize', 1);
quiver(center(1), center(2), u2(1)*u2length, u2(2)*u2length, 0, 'b',
...
'LineWidth', 1, 'Color', color, 'MarkerSize', 1);

axis equal;
end

```

Code 11.1: Funzione MATLAB per disegnare l'ellissi di manipolabilità

12. Esercitazione 12: analisi leggi di controllo

(12) *Analisi delle leggi di controllo PD, PD+compensazione di gravità e PID per un punto materiale in caduta libera verticale*

12.1 Descrizione del problema

Un manipolatore ha un modello dinamico del tipo

$$\mathbf{M}(\underline{q})\ddot{\underline{q}} + \underline{c}(\underline{q}, \dot{\underline{q}}) + \underline{g}(\underline{q}) = \underline{u}$$

si assume ora che \underline{u} sia l'ingresso di controllo.

Teorema di Lyapunov

Una funzione candidata di Lyapunov è una funzione del tipo $V : \mathbb{R}^n \rightarrow \mathbb{R}$ tale che $V(\underline{x}_e) = 0$ e $V(\underline{x}_e) > 0 \quad \forall \underline{x} \neq \underline{x}_e$.

Il teorema di Lyapunov afferma che è condizione sufficiente alla stabilità

$$\exists V : \dot{V}(\underline{x}) = \langle \nabla V(\underline{x}), f(\underline{x}) \rangle \leq 0 \quad \forall \underline{x}$$

Inoltre si ha che è condizione sufficiente alla stabilità asintotica

$$\exists V : \dot{V}(\underline{x}) < 0 \quad \forall \underline{x} \neq \underline{x}_e$$

Teorema di Lasalle

Il teorema di Lasalle (o principio di invarianza di Lasalle) afferma che se

$$\exists V : \dot{V}(\underline{x}) \leq 0$$

allora le soluzioni di $\dot{\underline{x}} = f(\underline{x})$ convergono nel più grande insieme invariante $M \subset S = \{\underline{x} : \dot{V}(\underline{x}) = 0\}$. Un insieme invariante rispetto a una funzione è un insieme che viene mappato in sé stesso dalla funzione.

12.1.1 Controllo P.D.

Dato il modello

$$\mathbf{M}(\underline{q})\ddot{\underline{q}} + \underline{c}(\underline{q}, \dot{\underline{q}}) + g(\underline{q}) = \underline{u}$$

si vuole trovare \underline{u} tale che $\underline{q} \rightarrow \underline{q}_d$ e $\dot{\underline{q}} \rightarrow 0$, ovvero posizionamento a \underline{q}_d desiderata e velocità nulla.

Il controllo P.D. prevede l'utilizzo di una legge di controllo del tipo

$$\underline{u} = \mathbf{K}_P(\underline{q}_d - \underline{q}) - \mathbf{K}_D\dot{\underline{q}}$$

Theorem 12.1.1. Se $g(\underline{q}) = 0$ e \mathbf{K}_P e \mathbf{K}_D sono matrici simmetriche e definite positive allora $(\underline{q}, \dot{\underline{q}}) \rightarrow (\underline{q}_d, 0)$. È necessario che $g(\underline{q})$ sia nullo perché altrimenti, dato che all'equilibrio il controllo si annulla (derivata $\dot{\underline{q}}$ ed errore nullo), ci sarebbe il termine forzante $g(\underline{q})$ che sposterebbe il manipolatore dall'equilibrio.

Dimostrazione. Data una funzione $V = \frac{1}{2}\dot{\underline{q}}\mathbf{M}(\underline{q})\dot{\underline{q}} + \frac{1}{2}\underline{e}^T\mathbf{K}_P\underline{e}$ dove $\underline{e} = \underline{q}_d - \underline{q}$ che si annulla quando l'errore è nullo $\underline{e} = 0$ e la velocità è nulla $\dot{\underline{q}} = 0$, ovvero nel punto di equilibrio che si vuole raggiungere, si ha che

$$\begin{aligned} \dot{V} &= \dot{\underline{q}}\mathbf{M}(\underline{q})\dot{\underline{q}} - \underline{e}^T\mathbf{K}_P\dot{\underline{q}} = \\ &= \dot{\underline{q}}^T(\underline{u} - \mathbf{S}\dot{\underline{q}} + \underbrace{\frac{1}{2}\dot{\mathbf{M}}\dot{\underline{q}}}_{=0}) - \underline{e}^T\mathbf{K}_D\dot{\underline{q}} = \\ &= \dot{\underline{q}}^T\underline{u} - \underline{e}^T\mathbf{K}_P\dot{\underline{q}} = \\ &= \dot{\underline{q}}^T(\mathbf{K}_P\underline{e} - \mathbf{K}_D\dot{\underline{q}}) - \underline{e}^T\mathbf{K}_P\dot{\underline{q}} = \\ &= \dot{\underline{q}}^T\mathbf{K}_P\underline{e} - \dot{\underline{q}}^T\mathbf{K}_D\dot{\underline{q}} - \underline{e}^T\mathbf{K}_P\dot{\underline{q}} = \\ &= -\dot{\underline{q}}^T\mathbf{K}_D\dot{\underline{q}} \end{aligned}$$

e quindi $\dot{V} = -\dot{\underline{q}}^T\mathbf{K}_D\dot{\underline{q}} \leq 0$. Per il teorema di Lyapunov è possibile quindi affermare che il punto di equilibrio $(\underline{e}, \dot{\underline{q}}) = (0, 0)$ è stabile (ovvero partendo vicino a

quel punto si rimane vicino), ma non è possibile dire nulla sulla stabilità asintotica dato che \dot{V} potrebbe annullarsi anche in punti non di equilibrio.

È quindi necessario applicare il teorema di Lasalle che afferma che le soluzioni del sistema convergono nel più grande invariante $M \subset S = \{(\underline{e}, \dot{\underline{q}}) : \dot{V}(\underline{e}, \dot{\underline{q}}) = 0\} = \{(\underline{e}, 0) : \underline{e} \in \mathbb{R}^n\}$ dato che $\dot{V} = -\dot{\underline{q}}^T \mathbf{K}_D \dot{\underline{q}}$.

Per calcolare l'insieme invariante M è necessario considerare il sistema

$$\mathbf{M}(\underline{q}) \ddot{\underline{q}} + \underline{c}(\underline{q}, \dot{\underline{q}}) = \mathbf{K}_P \underline{e} - \mathbf{K}_D \dot{\underline{q}}$$

e ristringendosi all'insieme S (ovvero $\dot{\underline{q}} = 0$) diventa

$$\mathbf{M}(\underline{q}) \ddot{\underline{q}} = \mathbf{K}_P \underline{e}$$

ovvero

$$\ddot{\underline{q}} = \mathbf{M}^{-1}(\underline{q}) \mathbf{K}_P \underline{e} = 0$$

dato che se $\dot{\underline{q}} = 0$ anche $\ddot{\underline{q}} = 0$ e quindi necessariamente

$$\underline{e} = 0$$

dato che $\boldsymbol{\pi}_e = \mathbf{M}^{-1}(\underline{q}) \mathbf{K}_P = 0$ è invertibile.

Si è quindi dimostrato che il più grande insieme invariante contenuto in S è proprio $(\underline{e}, \dot{\underline{q}}) = (0, 0)$. Per cui il teorema di Lasalle garantisce che le traiettorie convergono a $(0, 0)$, ovvero lo schema di controllo P.D. in assenza di gravità garantisce la regolazione dell'equilibrio quando le matrici \mathbf{K}_D e \mathbf{K}_P sono definite positive. \square

Da notare come la legge di controllo appena definita risulti completamente indipendente dalla struttura del manipolatore, essa infatti utilizza solo la conoscenza di \underline{q} e $\dot{\underline{q}}$.

12.1.2 Controllo P.D. con gravità

Nel caso in cui $g(\underline{q}) \neq 0$ è possibile estendere la legge di controllo precedentemente vista nel seguente modo

$$\underline{u} = \mathbf{K}_P(\underline{q}_d - \underline{q}) - \mathbf{K}_D \dot{\underline{q}} + g(\underline{q})$$

Tale schema di controllo ha le stesse proprietà di quello visto precedentemente dato che la dimostrazione è analoga basta semplificare i termini $g(\underline{q})$.

Tale legge di controllo risulta però essere non lineare, dato che, a differenza di prima, \underline{u} presenta la funzione $g(\underline{q})$ che in generale è non lineare.

La $g(\underline{q})$ potrebbe essere non nota, è però possibile utilizzare una sua stima $\hat{g}(\underline{q})$. In questo caso è possibile dimostrare che $\dot{\underline{q}} \rightarrow 0$ ma in generale $\underline{q} \rightarrow \bar{\underline{q}} \neq \underline{q}_d$ dove $\bar{\underline{q}}$ dipende dalla condizione iniziale. Se il guadagno \mathbf{K}_P è abbastanza alto di solito $\bar{\underline{q}} \sim \underline{q}_d$. I guadagni \mathbf{K}_P e \mathbf{K}_D possono ovviamente essere presi grandi quanto si vuole purché siano definiti positivi.

Alternativamente, invece di compensare la gravità perfettamente in ogni istante, è possibile compensarla solo nella configurazione finale

$$\underline{u} = \mathbf{K}_P(\underline{q}_d - \underline{q}) - \mathbf{K}_D\dot{\underline{q}} + g(\underline{q}_d)$$

ci sono ovviamente problemi nel transitorio, $g(\underline{q}_d)$ agisce infatti come un disturbo costante. Anche questa legge funziona bene solo in caso di \mathbf{K}_P abbastanza grande.

12.1.3 Controllo P.I.D. per la compensazione della gravità

Un altro modo possibile per la compensazione della gravità è utilizzare una legge di controllo P.I.D.

$$\underline{u} = \mathbf{K}_P(\underline{q}_d - \underline{q}) + \mathbf{K}_I \int_0^t (\underline{q}_d - \underline{q}(\tau)) d\tau - \mathbf{K}_D\dot{\underline{q}}$$

È noto che il termine integrale è capace di compensare disturbi costanti. Tuttavia l'azione della gravità in generale non è costante, tende però ad esserlo all'avvicinarsi del punto di equilibrio, vicino ad esso infatti $g(\underline{q}) \rightarrow g(\underline{q}_d)$.

Per tale legge di controllo è però possibile solo fornire condizioni sufficienti per la stabilità locale esponenziale, per fare ciò si linearizza il sistema.

12.2 Soluzione

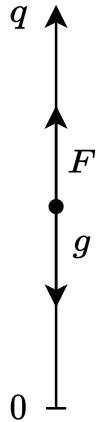


Figura 12.1: Punto materiale in caduta libera

Per la seconda legge della dinamica si ha:

$$F(t) - mg(t) = m\ddot{q}(t)$$

e definendo l'errore come $e(t) = q_d - q(t)$ si ottiene $F(t) - mg(t) = -m\dot{e}(t)$ ovvero

$$F(t) = mg(t) - m\ddot{e}(t)$$

dove $F(t)$ è la legge di controllo.

12.2.1 Controllo PID

Si utilizza una legge di controllo PID del tipo:

$$F(t) = K_P e(t) + K_D \dot{e}(t) + K_I \int_0^t e(\tau) d\tau$$

si ottiene quindi

$$mg(t) - m\ddot{e}(t) = K_P e(t) + K_D \dot{e}(t) + K_I \int_0^t e(\tau) d\tau$$

da cui si ricava:

$$g(t) = \ddot{e}(t) + \frac{1}{m} \left(K_P e(t) + K_D \dot{e}(t) + K_I \int_0^t e(\tau) d\tau \right)$$

che corrisponde all'ingresso del sistema.

Si vuole ora calcolare la funzione di trasferimento

$$H(s) = \frac{E(s)}{G(s)}$$

ma

$$G(s) = s^2 E(s) + \frac{K_P}{m} E(s) + \frac{K_D}{m} s E(s) + \frac{K_I}{ms} E(s)$$

e quindi

$$H(s) = \frac{E(s)}{G(s)} = \frac{ms}{ms^3 + K_D s^2 + K_P s + K_I}$$

Se $\dot{g}(t) = 0$ ovvero $g(t) = g_0$ costante, è possibile dire che $G(s) = \frac{g_0}{s}$ e quindi

$$E(s) = \frac{mg_0}{ms^3 + K_D s^2 + K_P s + K_I}$$

Dato che si vuole $e(t) \rightarrow 0$ è necessario che i poli di $E(s)$ siano a parte reale negativa. In questo modo infatti si avrà $e(t) = \sum_i m_i(t)$ dove $m_i(t)$ sono funzioni esponenziali con esponente negativo, ovvero convergenti a 0. Con tale condizione è possibile individuare i valori di K_P , K_D e K_I tali che il sistema risulti stabile.

Al fine di valutare gli zeri del polinomio $d(s) = ms^3 + K_D s^2 + K_P s + K_I$ è possibile utilizzare il criterio di Routh. Si costruisce quindi la tabella di Routh

$$\begin{bmatrix} a_n & a_{n-2} \\ a_{n-1} & a_{n-3} \\ b_{n-1} & b_{n-2} \\ c_{n-2} & c_{n-3} \end{bmatrix}$$

dove:

- a_n, a_{n-1}, a_{n-2} e a_{n-3} sono i coefficienti del polinomio;

$$\bullet \quad b_{n-1} = \frac{\det \begin{bmatrix} a_n & a_{n-2} \\ a_{n-1} & a_{n-3} \end{bmatrix}}{-a_{n-1}}, \quad b_{n-2} = 0;$$

$$\bullet \quad c_{n-2} = \frac{\det \begin{bmatrix} a_{n-1} & a_{n-3} \\ b_{n-1} & b_{n-3} \end{bmatrix}}{-b_{n-1}}, \quad c_{n-3} = 0;$$

Per $d(s)$ si ha quindi la seguente tabella

$$\begin{bmatrix} m & K_P \\ K_D & K_I \\ -\frac{mK_I - K_D K_P}{K_D} & 0 \\ K_I & 0 \end{bmatrix}$$

Secondo il criterio di Routh ogni variazione (permanenza) del segno dei coefficienti della prima colonna corrisponde a una radice con parte reale positiva (negativa). Dato che $m > 0$ affinché tutte le radici di $d(s)$ siano a parte reale negativa è necessario che

$$\begin{cases} K_D > 0 \\ -\frac{mK_I - K_D K_P}{K_D} > 0 \\ K_I > 0 \end{cases}$$

ovvero

$$\begin{cases} K_D > 0 \\ K_P > m \frac{K_I}{K_D} \\ K_I > 0 \end{cases}$$

12.2.2 Controllo PD

In questo caso si ha una legge di controllo del tipo:

$$F(t) = K_P e(t) + K_D \dot{e}(t)$$

si ottiene quindi

$$mg(t) - m\ddot{e}(t) = K_P e(t) + K_D \dot{e}(t)$$

Per cui

$$E(s) = \frac{mg_0}{ms^3 + K_D s^2 + K_P s}$$

e quindi la tabella di Routh è

$$\begin{bmatrix} m & K_P \\ K_D & 0 \\ K_P & 0 \end{bmatrix}$$

Affinché si abbia $e(t) \rightarrow 0$ è necessario e sufficiente che

$$\begin{cases} K_D > 0 \\ K_P > 0 \end{cases}$$

dato che $m > 0$.

12.2.3 Controllo PD + compensazione della gravità

Si utilizza una legge di controllo del tipo:

$$F(t) = K_P e(t) + K_P \dot{e}(t) + mg(t)$$

si ha quindi che

$$m\ddot{q}(t) = K_P e(t) + K_D \dot{e}(t)$$

che in Lapalce diventa

$$ms^2 Q(s) = K_P E(s) + K_D s E(s)$$

La funzione di trasferimento risulta essere

$$H(s) = \frac{Q(s)}{E(s)} = \frac{K_P + K_D s}{ms^2}$$

da cui si ricava

$$Q(s) = \frac{K_P + K_D s}{ms^2} E(s)$$

Ricordando che $e(t) = q_d - q(t)$ è possibile dire che

$$E(s) = \frac{q_d}{s} - \frac{K_P + K_D s}{ms^2} E(s)$$

per cui si ha

$$E(s) = \frac{q_d ms}{ms^2 + K_D s + K_P}$$

perciò affinché l'errore converga a 0 è necessario che il polinomio $ms^2 + K_D s + K_P$ abbia tutti gli zeri a parte reale negativa, tale condizione è vera per la regola di Routh $\Leftrightarrow K_D > 0 \wedge K_P > 0$, dato che $m > 0$.

13. Esercitazione 13: modello dinamico robot PR

(13) *Modello dinamico di un manipolatore PR in assenza di gravità.*

13.1 Descrizione del problema

Nella dinamica si vuole mettere in relazione le forze generalizzate \underline{u} con il moto del robot \underline{q} .

La dinamica diretta permette di stabilire l'evoluzione di \underline{q} e $\dot{\underline{q}}$ dati \underline{q}_0 e $\dot{\underline{q}}_0$. È possibile ricavare la dinamica diretta sfruttando il metodo di Lagrange.

La dinamica inversa permette di determinare $\underline{u}_d(t)$ dati $(\underline{q}_d, \dot{\underline{q}}_d, \ddot{\underline{q}}_d)$. È possibile ricavare la dinamica inversa sfruttando il metodo di Eulero-Lagrange, tale metodo è in realtà utilizzabile anche per ricavare la dinamica diretta.

13.1.1 Metodo di Lagrange

Il metodo di Lagrange è un metodo analitico energy-based per lo studio di un sistema meccanico. Esso prende in considerazione un sistema meccanico descritto da un vettore di coordinate generalizzate $\underline{q} \in \mathbb{R}^n$ a cui si associa la funzione lagrangiana

$$L(\underline{q}(t), \dot{\underline{q}}(t), t) = T(\underline{q}, \dot{\underline{q}}) - U(\underline{q})$$

Il principio di minima azione, o principio di Hamilton, afferma che il percorso seguito da un sistema fisico tra due punti nello spazio e nel tempo è quello che minimizza l'integrale dell'azione lungo quel percorso. L'integrale azione è definito

come l'integrale della funzione lagrangiana in un intervallo di tempo

$$S = \int_{t_1}^{t_2} L(\underline{q}, \dot{\underline{q}}, t) dt$$

Si vogliono quindi trovare $\underline{q}(t)$ e $\dot{\underline{q}}(t)$ che minimizzano S . È possibile dimostrare che la soluzione a tale problema soddisfa il seguente insieme di equazioni differenziali:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\underline{q}}_i} - \frac{\partial L}{\partial \underline{q}_i} = u_i \quad \forall i = 1, \dots, n$$

dove u_i è la componente i -esima del vettore delle forze generalizzate, ovvero quelle forze che compiono lavoro lungo la componente i -esima. Le forze generalizzate rappresentano le forze esterne o le influenze che agiscono sul sistema e che non derivano direttamente dall'energia potenziale. Possono essere forze di attrito, forze di trascinamento o altre forze non conservative.

13.1.2 Energia di corpi rigidi in moto nello spazio

Corpo singolo

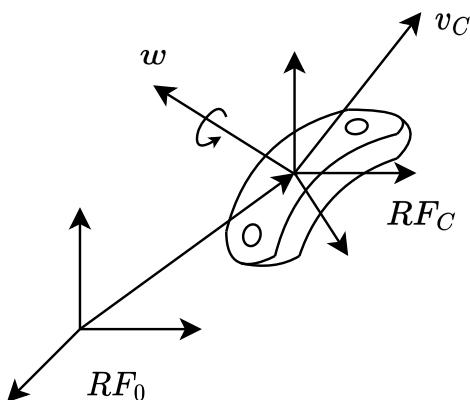


Figura 13.1: Corpo rigido

Si considera un corpo rigido in moto in \mathbb{R}^3 , un sistema di riferimento RF_0 solidale al terreno e un sistema di riferimento RF_C solidale al corpo e centrato nel centro di massa, con v_c si indica la velocità del corpo, mentre con w si indica la sua velocità angolare.

La massa del corpo è calcolabile come

$$m = \int_B \rho(x, y, z) dx dy dz = \int_b dm$$

La posizione del centro di massa nel riferimento assoluto RF_0 è

$$\underline{r}_C = \frac{1}{m} \int_B \underline{r} dm$$

La sua energia cinetica è

$$T = \frac{1}{2} \int_B \underline{v}^T(x, y, z) \underline{v}(x, y, z) dm$$

dove $\underline{v} = \underline{v}_C + \mathbf{S}(\underline{w})\underline{r}$ (relazione fondamentale della cinematica). Si ha quindi che

$$\begin{aligned} T &= \frac{1}{2} \int_B (\underline{v}_C + \mathbf{S}(\underline{w})\underline{r})^T (\underline{v}_C + \mathbf{S}(\underline{w})\underline{r}) dm = \\ &= \frac{1}{2} \int_B \underline{v}_C^T \underline{v}_C dm + \int_B \underline{v}_C^T \mathbf{S}(\underline{w}) \underline{r} dm + \frac{1}{2} \int_B \underline{r}^T \mathbf{S}^T(\underline{w}) \mathbf{S}(\underline{w}) \underline{r} dm = \\ &= \underbrace{\frac{1}{2} m \underline{v}_C^T \underline{v}_C}_{\text{en. cin. traslazionale}} + \underbrace{\underline{v}_C^T \mathbf{S}(\underline{w}) \int_B \underline{r} dm}_{=0} + \underbrace{\frac{1}{2} \underline{w}^T [\int_B \mathbf{S}^T(\underline{r}) \mathbf{S}(\underline{w}) dm] \underline{w}}_{\text{en. cin. rotazionale}} \end{aligned}$$

La precedente espressione è nota come teorema di Koning. Da notare che $\int_B \underline{r} dm = 0$ perché esso rappresenta la posizione del centro di massa in RF_C che è centrato nel centro di massa.

Il termine integrale della componente rotazionale rappresenta la matrice di inerzia rispetto al centro di massa:

$$\mathbf{I}_C = \int_B \mathbf{S}^T(\underline{r}) \mathbf{S}(\underline{r}) dm$$

dove $dm = \rho(x, y, z) dx dy dz$ e $\underline{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$. Il vettore \underline{w} e \mathbf{I}_C devono essere espressi nello stesso sistema di riferimento. La matrice \mathbf{I}_C è costante perché calcolata rispetto ad un riferimento solidale al corpo.

Energia cinetica di un manipolatore seriale

Per un manipolatore seriale l'energia cinetica è

$$T = \sum_{i=1}^N T_i$$

dato che per un manipolatore seriale si ha

$$T_i = T_i(q_j, \dot{q}_j) \quad j \leq i$$

ovvero l'energia cinetica dell' i -esimo braccio dipende solo dal braccio stesso e da quelli precedenti nella catena.

L'energia cinetica rotazionale dell' i -esimo corpo è

$$\begin{aligned} {}^0 \underline{w}_i^T {}^0 \mathbf{I}_{C_i}(\underline{q}) {}^0 \underline{w}_i &= ({}^0 \mathbf{R}_i(\underline{q})^i \underline{w}_i)^T ({}^0 \mathbf{I}_{C_i}(\underline{q})) ({}^0 \mathbf{R}_i(\underline{q})^i \underline{w}_i) = \\ &= {}^i \underline{w}_i^T \underbrace{({}^0 \mathbf{R}_i^T(\underline{q}) ({}^0 \mathbf{I}_{C_i}(\underline{q})) ({}^0 \mathbf{R}_i(\underline{q})^i \underline{w}_i)}_{{}^0 \mathbf{I}_{C_i}} \end{aligned}$$

Da notare che

$${}^0 \mathbf{I}_{C_i} = ({}^0 \mathbf{R}_i^T(\underline{q}) ({}^0 \mathbf{I}_{C_i}(\underline{q})) ({}^0 \mathbf{R}_i(\underline{q})))$$

non è costante, mentre ${}^0 \mathbf{I}_{C_i}(\underline{q})$ sì. La precedente relazione ci dice come la matrice di inerzia si trasforma da un riferimento all'altro.

La velocità del centro di massa dell' i -esimo corpo è

$$\underline{v}_{C_i} = \mathbf{J}_{L_i}(\underline{q}) \dot{\underline{q}} = \left[\underbrace{\begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix}}_{i \text{ colonne}} \cdots \underbrace{\begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix}}_{n-i \text{ colonne}} \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix} \right] \dot{\underline{q}}$$

mentre la velocità angolare è

$$\underline{w}_i = \mathbf{J}_{A_i} \dot{\underline{q}}$$

Utilizzando le espressioni precedentemente ricavate è possibile riscrivere l'energia cinetica dell' i -esimo corpo come

$$\begin{aligned} T_i &= m_i \underline{v}_{C_i}^T \underline{v}_{C_i} + \underline{w}_i^T \mathbf{I}_{C_i} \underline{w}_i = \\ &= m_i (\mathbf{J}_{L_i}(\underline{q}) \dot{\underline{q}})^T (\mathbf{J}_{L_i}(\underline{q}) \dot{\underline{q}}) + (\mathbf{J}_{A_i} \dot{\underline{q}})^T \mathbf{I}_{C_i} (\mathbf{J}_{A_i} \dot{\underline{q}}) = \\ &= \dot{\underline{q}}^T (\mathbf{J}_{L_i}^T(\underline{q}) \mathbf{J}_{L_i}(\underline{q}) + \mathbf{J}_{A_i}^T \mathbf{I}_{C_i} \mathbf{J}_{A_i}) \dot{\underline{q}} \end{aligned}$$

L'energia cinetica totale è

$$T = \frac{1}{2} \dot{\underline{q}}^T \sum_{i=1}^N [\mathbf{J}_{L_i}^T(\underline{q}) \mathbf{J}_{L_i}(\underline{q}) + \mathbf{J}_{A_i}^T \mathbf{I}_{C_i} \mathbf{J}_{A_i}] \dot{\underline{q}}$$

chiamando $\mathbf{M}(\underline{q}) = \mathbf{J}_{L_i}^T(\underline{q}) \mathbf{J}_{L_i}(\underline{q}) + \mathbf{J}_{A_i}^T \mathbf{I}_{C_i} \mathbf{J}_{A_i}$ matrice generalizzata d'inerzia si ottiene l'espressione

$$T = \frac{1}{2} \dot{\underline{q}}^T \mathbf{M} \dot{\underline{q}}$$

La matrice generalizzata d'inerzia è simmetrica $\mathbf{M}(\underline{q}) = \mathbf{M}^T(\underline{q}) \forall \underline{q}$, inoltre essa è definita positiva $\forall \underline{q}$. È simmetrica per costruzione, inoltre non può essere semidefinita positiva perché altrimenti si potrebbe avere energia cinetica nulla per velocità di giunto diversa da 0.

Energia potenziale di un manipolatore seriale

Con considerazioni analoghe a quelle fatte per l'energia cinetica è possibile osservare che l'energia potenziale totale del corpo è la somma delle singole energie potenziali

$$U = \sum_{i=1}^N U_i$$

inoltre

$$U_i = U_i(q_j) \quad j \leq i$$

ovvero l'energia potenziale dell' i -esimo braccio dipende solo dal braccio stesso e da quelli precedenti nella catena.

L'energia dell' i -esimo braccio è esprimibile come

$$U_i = -m_i g^T \mathbf{r}_{C_i}$$

da notare che la posizione del centro di massa \mathbf{r}_{C_i} è nota dalla cinematica diretta.

Lagrangiano di un manipolatore seriale

La funzione lagrangiana della struttura è

$$L(\underline{q}(t), \dot{\underline{q}}(t), t) = T(\underline{q}, \dot{\underline{q}}) - U(\underline{q})$$

e si ottiene un sistema di n equazioni differenziali

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k} = u_k \quad k = 1, \dots, N$$

tutte del secondo ordine. Di solito $\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_k}$ dipende solo dall'energia cinetica e $\frac{\partial L}{\partial q_k}$ dipende solo dall'energia potenziale.

Per la generica coordinata k si ottiene

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_k} = \sum_j m_{kj} \ddot{q}_j + \sum_{i,j} \frac{\partial m_{ij}}{\partial q_i} \dot{q}_i \dot{q}_j$$

e

$$\frac{\partial L}{\partial q_k} = \frac{1}{2} \sum_{i,j} \frac{\partial m_{ij}}{\partial q_k} \dot{q}_i \dot{q}_j - \frac{\partial U}{\partial q_k}$$

dove $m_{ij} = M[i][j]$.

Utilizzando particolari convenzioni è possibile esprimere l'espressione $\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k} = u_k$ come

$$\sum_j m_{kj}(\underline{q}) \ddot{q}_j + \sum_{i,j} i, j C_{kij}(\underline{q}) \dot{q}_i \dot{q}_j + \frac{\partial U}{\partial q_k} = u_k$$

dove C_{kij} sono chiamati simboli di Christoffel che dipendono dalle derivate degli elementi m_{kj} e m_{ij} rispetto a q_i , q_j e q_k . I tre termini possono essere interpretati nel seguente modo:

- $\sum_j m_{kj}(\underline{q}) \ddot{q}_j$ termini inerziali del tipo ma ;
- $\sum_{i,j} i, j C_{kij}(\underline{q}) \dot{q}_i \dot{q}_j$ termini centrifughi e di Coriolis;
- $\frac{\partial U}{\partial q_k}$ termine gravitazionale;

Il modello vettoriale completo è quindi

$$\mathbf{M}(\underline{q}) \ddot{\underline{q}} + \underline{c}(\underline{q}, \dot{\underline{q}}) + g(\underline{q}) = \underline{u}$$

dove le componenti del vettore \underline{c} sono $c_k(\underline{q}, \dot{\underline{q}}) = \dot{\underline{q}}^T \mathbf{C}_k(\underline{q}) \dot{\underline{q}}$ e $\mathbf{C}_k(\underline{q}) = \frac{1}{2} \left(\frac{\partial \underline{M}_k}{\partial \underline{q}} + \left(\frac{\partial \underline{M}_k}{\partial \underline{q}} \right)^T - \frac{\partial \mathbf{M}}{\partial q_k} \right)$

dove \underline{M}_k è la k -esima colonna della matrice \mathbf{M} .

Una rappresentazione equivalente è

$$\mathbf{M}(\underline{q}) \ddot{\underline{q}} + \mathbf{S}(\underline{q}, \dot{\underline{q}}) \dot{\underline{q}} + g(\underline{q}) = \underline{u}$$

dove $S_{kj}(\underline{q}, \dot{\underline{q}}) = \sum_i C_{kij}(\underline{q}) \dot{q}_i$ e $\mathbf{C}(\underline{q}, \dot{\underline{q}}) = \mathbf{S}(\underline{q}, \dot{\underline{q}}) \dot{\underline{q}}$.

Property 13.1.1. La matrice $\dot{\mathbf{M}} - 2\mathbf{S}$ è antisimmetrica. Quindi $\underline{x}^T (\dot{\mathbf{M}} - 2\mathbf{S}) \underline{x} = 0 \forall \underline{x}$. Proprietà utile in alcuni schemi di controllo.

Conservazione dell'energia meccanica per una struttura robotica

L'energia meccanica totale di una struttura robotica è

$$E = T + U$$

affinché E sia conservativa è necessario che $\dot{E} = 0$.

Si ha che

$$\dot{E} = \dot{\underline{q}}^T \mathbf{M}(\underline{q}) \ddot{\underline{q}} + \frac{1}{2} \dot{\underline{q}}^T \dot{\mathbf{M}}(\underline{q}) \dot{\underline{q}} + \frac{\partial U}{\partial \underline{q}} \dot{\underline{q}}$$

ma $\mathbf{M}(\underline{q}) \ddot{\underline{q}} = \underline{u} - \mathbf{S}(\underline{q}, \dot{\underline{q}}) \dot{\underline{q}} - g(\underline{q})$ e quindi

$$\begin{aligned} \dot{E} &= \dot{\underline{q}}^T (\underline{u} - \mathbf{S}(\underline{q}, \dot{\underline{q}}) \dot{\underline{q}} - g(\underline{q})) + \frac{1}{2} \dot{\underline{q}}^T \dot{\mathbf{M}}(\underline{q}) \dot{\underline{q}} + \frac{\partial U}{\partial \underline{q}} \dot{\underline{q}} = \\ &= \dot{\underline{q}}^T \underline{u} + \underbrace{\frac{1}{2} \dot{\underline{q}}^T (\dot{\mathbf{M}}(\underline{q}) - 2\mathbf{S}(\underline{q}, \dot{\underline{q}})) \dot{\underline{q}}}_{=0} = \dot{\underline{q}}^T \underline{u} \end{aligned}$$

e ricordando che \underline{u} è il vettore delle forze generalizzate (ovvero le forze non conservative) l'espressione $\dot{E} = \dot{\underline{q}}^T \underline{u}$ ci dice che la variazione dell'energia meccanica dipende solo dalle forze non conservative.

Se ad esempio $\underline{u} = \mathbf{D} \dot{\underline{q}}$ (attrito viscoso) si ha $\dot{E} = \dot{\underline{q}}^T \mathbf{D} \dot{\underline{q}}$ dove \mathbf{D} è definita negativa (in questo modo l'energia meccanica diminuisce).

13.2 Soluzione

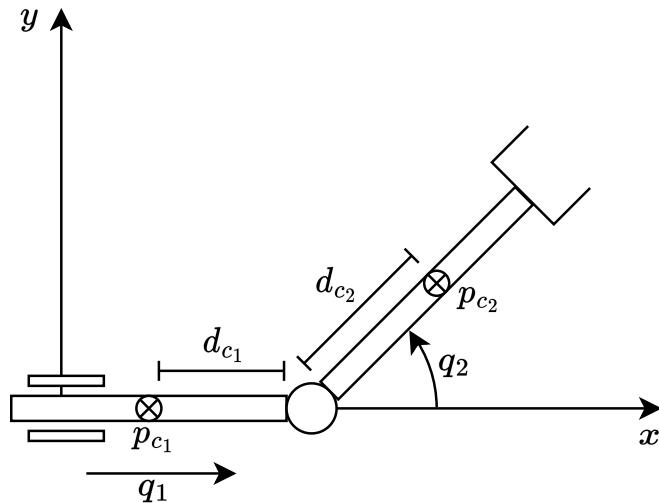


Figura 13.2: Robot PR

Al fine di poter scrivere la lagrangiana L conviene prima calcolare l'energia cinetica e potenziale dei due bracci.

Energia cinetica e potenziale del primo braccio

L'energia potenziale risulta essere pari ad una costante dato che il robot si trova su un piano:

$$U = \text{costante}$$

La generica energia cinetica di un braccio robotico è calcolabile come

$$T_i = \frac{1}{2} m_i \underline{v}_{c_i}^T \underline{v}_{c_i} + \frac{1}{2} \underline{w}_i^T \mathbf{I}_{c_i} \underline{w}_i$$

In questo caso si ha ovviamente $\underline{w}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ dato che si tratta del primo link e si ha un giunto prismatico.

La posizione del centro di massa del primo braccio è

$$\underline{p}_{c_1} = \begin{bmatrix} q_1 - d_{c_1} \\ 0 \\ 0 \end{bmatrix}$$

e quindi si ha

$$\underline{v}_{c_1}^T \underline{v}_{c_1} = \dot{\underline{p}}_{c_1}^T \dot{\underline{p}}_{c_1} = [\dot{q}_1 \ 0 \ 0] \begin{bmatrix} \dot{q}_1 \\ 0 \\ 0 \end{bmatrix} = \dot{q}_1^2$$

Per cui l'energia cinetica del braccio 1 risulta essere

$$T_1 = \frac{1}{2} m_1 \dot{q}_1^2$$

Energia cinetica e potenziale del secondo braccio

Anche in questo caso l'energia potenziale risulta essere pari ad una costante dato che il robot si trova su un piano:

$$U = \text{costante}$$

In questo caso si ha che la posizione del centro di massa del secondo braccio è

$$\underline{p}_{c_2} = \begin{bmatrix} q_1 + d_{c_2} \cos(q_2) \\ d_{c_2} \sin(q_2) \\ 0 \end{bmatrix}$$

per cui si ha che

$$\begin{aligned} \underline{v}_{c_2}^T \underline{v}_{c_2} &= \dot{\underline{p}}_{c_2}^T \dot{\underline{p}}_{c_2} = [q_1 + d_{c_2} \cos(q_2) \ d_{c_2} \sin(q_2) \ 0] \begin{bmatrix} q_1 + d_{c_2} \cos(q_2) \\ d_{c_2} \sin(q_2) \\ 0 \end{bmatrix} = \\ &= \dot{q}_1^2 - 2\dot{q}_2\dot{q}_1 d_{c_2} q_2 + \dot{q}_2^2 d_{c_2}^2 \sin^2(q_2) + \dot{q}_2^2 d_{c_2}^2 \cos^2(q_2) = \\ &= \dot{q}_1^2 - 2\dot{q}_2\dot{q}_1 d_{c_2} q_2 + \dot{q}_2^2 d_{c_2}^2 \end{aligned}$$

il termine traslazionale dell'energia cinetica è

$$T_{2t} = \frac{1}{2} m_2 (\dot{q}_1^2 - 2\dot{q}_2\dot{q}_1 d_{c_2} q_2 + \dot{q}_2^2 d_{c_2}^2)$$

Inoltre essendo $\underline{w}_2 = \begin{bmatrix} 0 \\ 0 \\ \dot{q}_2 \end{bmatrix}$ è possibile dire che il termine rotazionale dell'energia cinetica è

$$T_{2r} = \frac{1}{2} \underline{w}_2^T \mathbf{I}_{c_2} \underline{w}_2 = \frac{1}{2} I_{c_2,z} \dot{q}_2^2$$

L'energia cinetica complessiva del secondo braccio è quindi

$$T_2 = T_{2t} + T_{2r} = \frac{1}{2} m_2 (\dot{q}_1^2 - 2\dot{q}_2\dot{q}_1 d_{c_2} q_2 + \dot{q}_2^2 d_{c_2}^2) + \frac{1}{2} I_{c_2,z} \dot{q}_2^2$$

Lagrangiana del sistema

La lagrangiana del sistema è

$$L = T - U = \frac{1}{2} m_1 \dot{q}_1^2 + \frac{1}{2} m_2 (\dot{q}_1^2 - 2\dot{q}_2\dot{q}_1 d_{c_2} q_2 + \dot{q}_2^2 d_{c_2}^2) + \frac{1}{2} I_{c_2,z} \dot{q}_2^2 - costante$$

Il modello dinamico è dato da:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = u_i \quad \forall i = 1, 2$$

i=1

Per quanto riguarda la prima equazione

$$\frac{\partial L}{\partial \dot{q}_1} = m_1 \dot{q}_1 + m_2 \dot{q}_1 - m_2 \dot{q}_2 d_{c_2} \sin(q_2) = \dot{q}_1(m_1 + m_2) - \dot{q}_2 m_2 d_{c_2} \sin(q_2)$$

e quindi

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_1} = \ddot{q}_1(m_1 + m_2) - \ddot{q}_2 m_2 d_{c_2} \sin(q_2) - \dot{q}_2^2 m_2 d_{c_2} \cos(q_2)$$

inoltre

$$\frac{\partial L}{\partial q_1} = 0$$

Infine si può quindi dire che

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_1} - \frac{\partial L}{\partial q_1} = \ddot{q}_1(m_1 + m_2) - \ddot{q}_2 m_2 d_{c_2} \sin(q_2) - \dot{q}_2^2 m_2 d_{c_2} \cos(q_2) = u_1$$

i=2

Per quanto riguarda la seconda equazione si può dire che

$$\frac{\partial L}{\partial \dot{q}_2} = -m_2 \dot{q}_1 d_{c2} \sin(q_2) + m_2 d_{c2}^2 \dot{q}_2 + I_{c2,z} \dot{q}_2$$

e quindi

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_2} = -\ddot{q}_1 m_2 d_{c2} \sin(q_2) - m_2 \dot{q}_1 d_{c2} \cos(q_2) \dot{q}_2 + m_2 d_{c2}^2 \ddot{q}_2 + I_{c2} \ddot{q}_2$$

inoltre

$$\frac{\partial L}{\partial q_2} = -m_2 \dot{q}_1 \dot{q}_2 d_{c2} \cos(q_2)$$

Infine si può quindi dire che

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_2} - \frac{\partial L}{\partial q_2} = -\ddot{q}_1 m_2 d_{c2} \sin(q_2) + m_2 d_{c2}^2 \ddot{q}_2 + I_{c2,z} \ddot{q}_2 = u_2$$

Modello dinamico

Il sistema di equazioni

$$\begin{cases} \ddot{q}_1(m_1 + m_2) - \ddot{q}_2 m_2 d_{c2} \sin(q_2) - \dot{q}_2^2 m_2 d_{c2} \cos(q_2) = u_1 \\ -\ddot{q}_1 m_2 d_{c2} \sin(q_2) + m_2 d_{c2}^2 \ddot{q}_2 + I_{c2,z} \ddot{q}_2 = u_2 \end{cases}$$

costituisce il modello dinamico del manipolatore. Esso inoltre può essere riscritto in forma vettoriale

$$\underbrace{\begin{bmatrix} m_1 + m_2 & -m_2 d_{c2} \sin(q_2) \\ -m_2 d_{c2} \sin(q_2) & I_{c2,z} + m_2 d_{c2}^2 \end{bmatrix}}_{\mathbf{M}(\underline{q})} \underbrace{\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix}}_{\ddot{\underline{q}}} + \underbrace{\begin{bmatrix} -\dot{q}_2^2 m_2 d_{c2} \cos(q_2) \\ 0 \end{bmatrix}}_{\underline{C}(\underline{q}, \dot{\underline{q}})} = \underbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}_{\underline{u}}$$

ovvero

$$\mathbf{M}(\underline{q}) \ddot{\underline{q}} + \underline{C}(\underline{q}, \dot{\underline{q}}) = \underline{u}$$