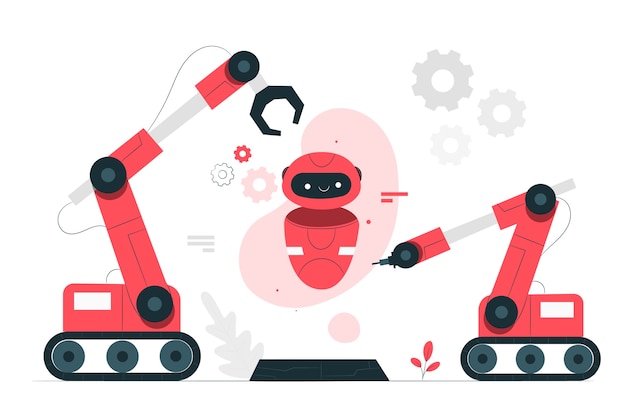
PROGETTO ROBOTICA INDUSTRIALE E ROBOTICA MOBILE



A.A.: 2022/2023

Riccardo Florio

209502

Sommario

[Introduzione 1](#_Toc135925810)

[ROBOTICA INDUSTRIALE 2](#_Toc135925811)

[Traccia 2](#_Toc135925812)

[Triangolo 3](#_Toc135925813)

[Circonferenza 6](#_Toc135925814)

[ROBOTICA MOBILE 9](#_Toc135925815)

[Traccia 9](#_Toc135925816)

# Introduzione

La robotica è un campo in rapida evoluzione che svolge un ruolo sempre più centrale nell'automazione di processi e nella realizzazione di soluzioni intelligenti. Nel contesto di questo progetto di robotica, il lavoro è stato suddiviso in due parti: robotica industriale e robotica mobile.

Nella prima parte, ci si concentra sulla robotica industriale, analizzando la struttura di un robot antropomorfo e studiando le sue capacità di movimento. Essa è ampiamente utilizzata in diversi settori, come l'automazione delle linee di produzione, l'assemblaggio di componenti e la manipolazione di oggetti pesanti. Durante questa fase del progetto, viene esaminata la struttura meccanica di un robot antropomorfo, comprendendo le sue articolazioni e le modalità di controllo. Sono stati sviluppati anche algoritmi per far percorrere ad esso delle traiettorie specifiche, al fine di simulare attività come il prelievo e il posizionamento di oggetti in un ambiente industriale.

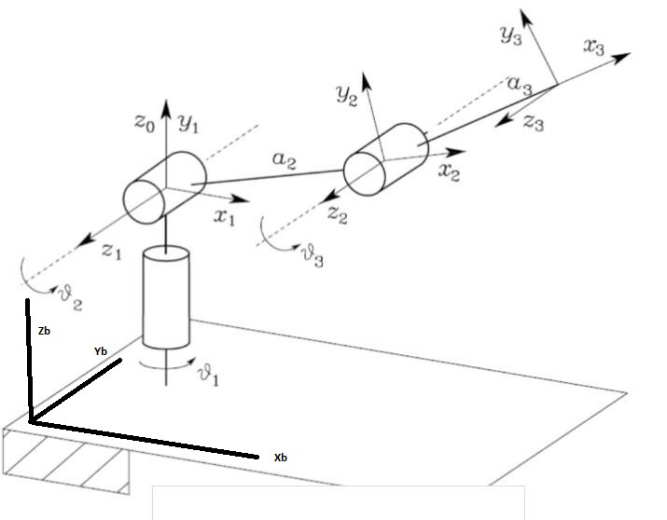
Nella seconda parte del progetto, ci si concentra sulla robotica mobile e sullo sviluppo di un robot capace di muoversi autonomamente in un ambiente prestabilito. Essa sta diventando sempre più importante in vari contesti, come la logistica, l'esplorazione di ambienti ostili e l'assistenza agli anziani. Durante questa fase, viene progettato e realizzato un robot dotato di sensori e algoritmi di navigazione, che gli consentono di percepire l'ambiente circostante e di pianificare i suoi movimenti. È stato anche sviluppato un sistema di controllo per far sì che il robot si sposti in modo sicuro ed efficiente nel percorso prestabilito.

Attraverso questo progetto di robotica, abbiamo affrontato sfide tecniche e concettuali nel campo dell'automazione e dell'intelligenza artificiale. Il nostro obiettivo principale era quello di esplorare le potenzialità dei robot antropomorfi e dei robot mobili, applicandoli a contesti specifici come la robotica industriale e quella mobile. Speriamo che questa relazione possa fornire una panoramica dettagliata delle nostre attività, dei risultati ottenuti e delle prospettive future nella ricerca e nello sviluppo di robot avanzati.

# ROBOTICA INDUSTRIALE

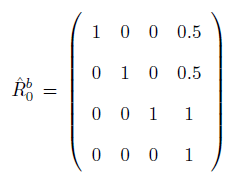
## Richiesta

Si consideri il robot antropomorfo in figura



la cui struttura è modellata dalla tabella di Denavit – Hartenberg (*Figura 1*) ed in cui i sistemi di riferimento (0) e (b) sono legati dalla matrice di rototraslazione (*Figura 2*).

w



Figura

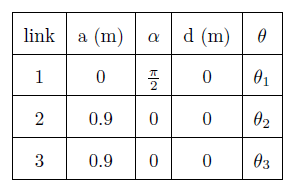
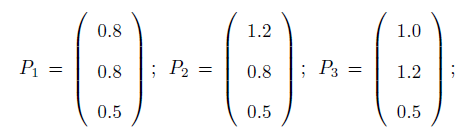


Figura 1

Siano i punti:



Si determinino gli andamenti temporali delle variabili di giunto (posizione e velocità) affinché l'origine del SR3:

1. descriva un triangolo secondo la sequenza P1 -> P2 -> P3 -> P1
2. descriva una circonferenza passante per i punti P1 -> P2 -> P3 -> P1

in entrambi i casi il tempo totale di percorrenza della curva deve essere di 40sec.

P.S. - le distanze sono espresse in metri; gli angoli in radianti.

## Percorso triangolo

Si noti innanzitutto che il percorso triangolare può essere visto come l’unione di tre percorsi rettilinei, per la precisione tre segmenti. Un segmento può essere parametrizzato come segue:

Per ottenere lambda viene usata la funzione *poly3.m*, alla quale si passa il parametro *sigma*; è una variabile indipendente adimensionale che varia tra 0 ed 1. Ciò viene eseguito nella parte iniziale dello script *Industriale.mlx*, scritto utilizzando il software ‘Matlab’. Nella medesima parte di codice vengono anche definite alcune variabili che saranno utili nel seguito, come i punti P1, P2 e P3, ovvero i vertici del triangolo, il vettore L, contenente le lunghezze dei bracci (estrapolate dalla tabella di Denavit – Hartenberg) ed i tempi di esecuzione del movimento. I tempi di esecuzione sono ottenuti partizionando in tre parti il tempo totale che il robot deve impiegare a percorrere la traiettoria, circa *13.33 sec*.

|  |
| --- |
| % Bracci del robot  global L  L = [0; 0.9; 0.9];  % Vertici del triangolo  P1 = [0.8; 0.8; 0.5];  P2 = [1.2; 0.8; 0.5];  P3 = [1.0; 1.2; 0.5];  % Tempi di esecuzione  T1 = 0; T2 = 13.3333;  T3 = 28.6667; T4 = 40;  % Sigma  DSIGMA = 0.03;  sigma = 0:DSIGMA:1;  N = length(sigma);  % Lambda  lambda = poly3(sigma);  lambda\_d = poly3d(sigma); |

Nel codice precedente viene anche definita la variabile *lambda\_d* mediante l’apposita funzione *poly3d*, che altro non è se non la derivata di *lambda*, la quale sarà utile più avanti per il calcolo delle velocità dei giunti.

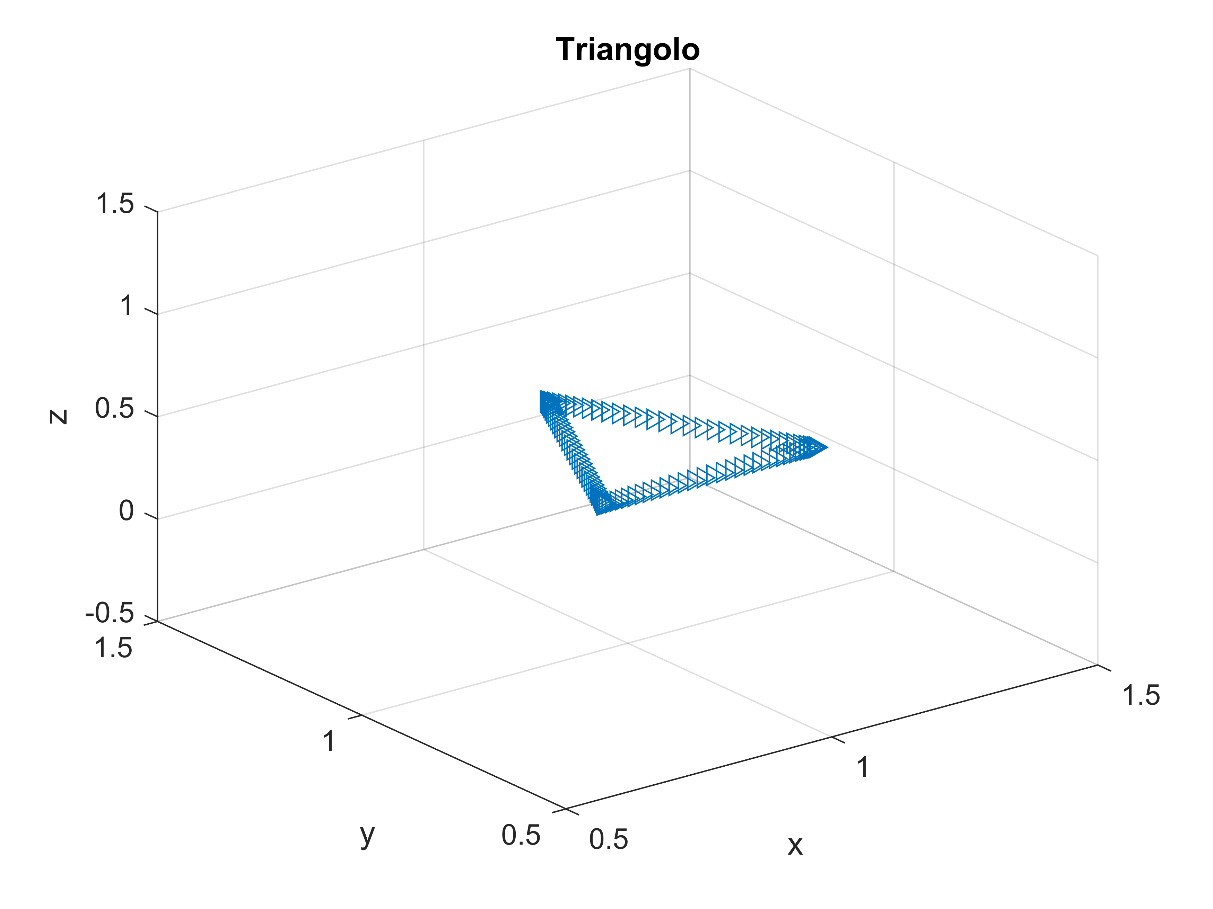
Si procede, quindi, col percorrere il tratto da P1 verso P2: ad ogni iterazione vengono calcolati la posizione dell’end – effector usando la funzione vista prima, il valore delle variabili di giunto mediante la cinematica inversa (funzione *Antropomorfo\_Cin\_Inv.m*) ed il valore della velocità dei giunti. Per il calcolo delle velocità si applica la cinematica differenziale inversa, in particolare si sfrutta la relazione:

🡪

|  |
| --- |
| % Percorso P1->P2  for i = 1:N  % Parametrizzazione del percorso  P = P1+lambda(i)\*(P2-P1);  % Calcolo della cinematica inversa  Q = Antropomorfo\_Cin\_Inv(L, P);  % Calcolo di J(Q)  J = Jacobiano\_Antropomorfo(L,Q);  % Risoluzione del sistema di eq. lineari J(Q)\*Qd=Pd  Pd = (P2-P1)\*lambda\_d(i)/(T2-T1);  Qd = inv(J)\*Pd;    QQ(i,:)=Q; % Variabili di giunto all'istante i  PP(i,:)=P; % Posizione dell'end effector all'istante i  QQd(i,:)=Qd; % Velocita' dei giunti all'istante i  End |

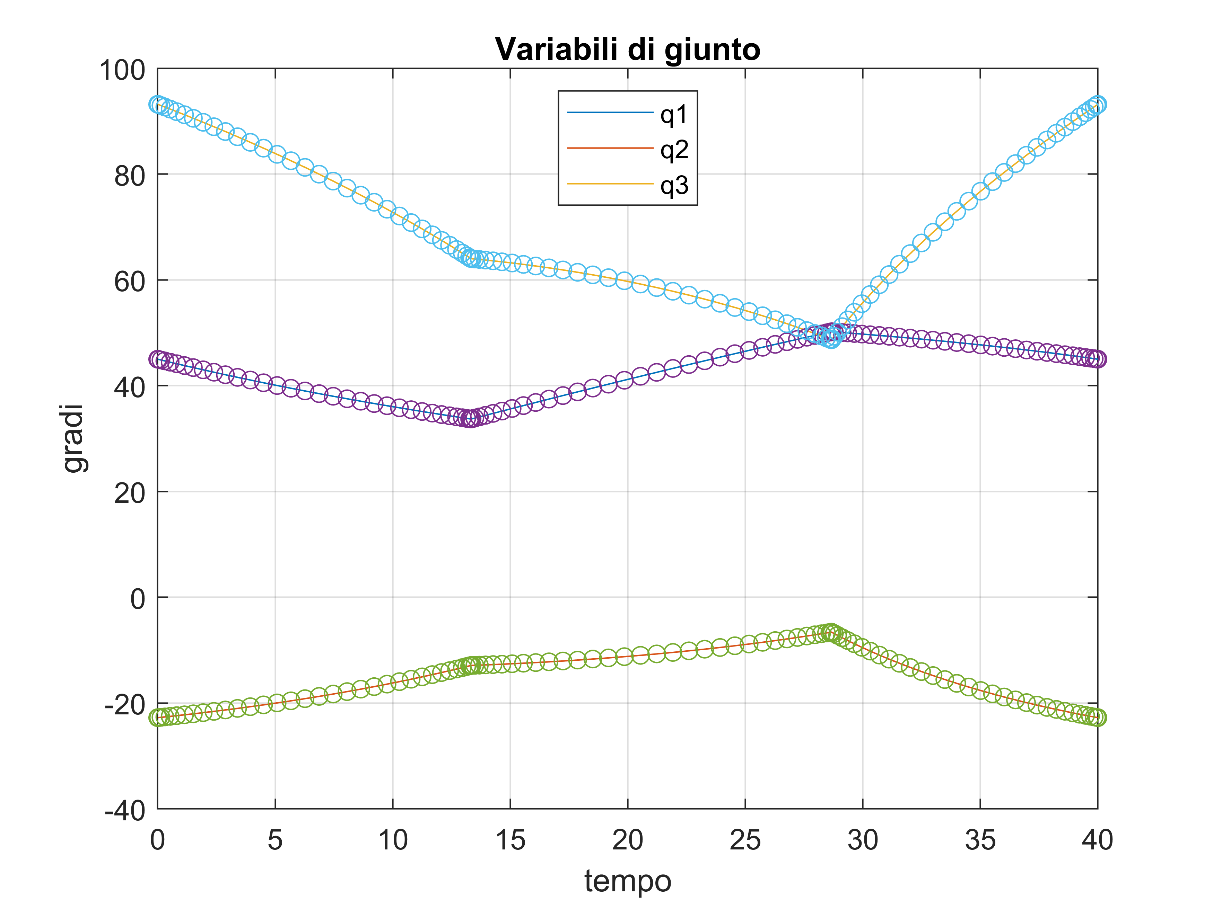
Analogamente si calcolano i percorsi P2 -> P3 e P3 -> P1.

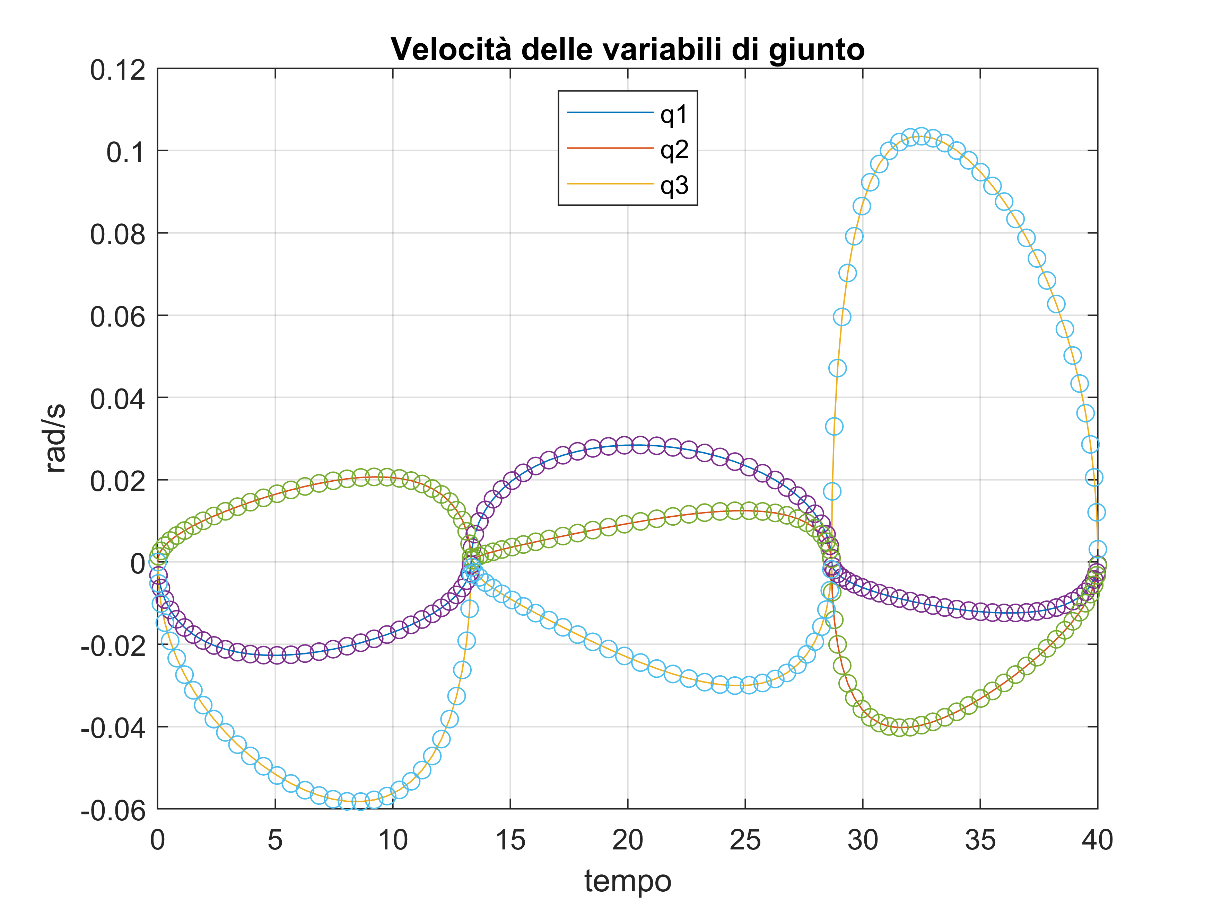
A questo punto viene eseguito il plot del percorso



Infine, si possono visualizzare gli andamenti temporali delle variabili di giunto dopo aver parametrizzato il tempo e convertito gli angoli espressi in radianti in angoli espressi in gradi:

|  |
| --- |
| % Plot variabili di giunto  t1 = T1+lambda\*(T2-T1);  t2 = T2+lambda\*(T3-T2);  t3 = T3+lambda\*(T4-T3);  t = [t1 t2(2:N) t3(2:N)];  plot(t,QQ\*180/pi);  .... |





## Percorso circonferenza

Anche in questo caso occorre parametrizzare il percorso che si vuole seguire. La generica equazione della circonferenza nel piano è data da:

In particolare, nel caso preso in esame, estendendo in :

ovvero:

dall'equazione della circonferenza che passa per P1, P2 e P3, ricaviamo il centro ed il raggio:

Prima di procedere con l'implementazione è importante precisare che il percorso che si vuole seguire deve partire dal punto P1; pertanto, l'intervallo in cui varia theta deve essere: .

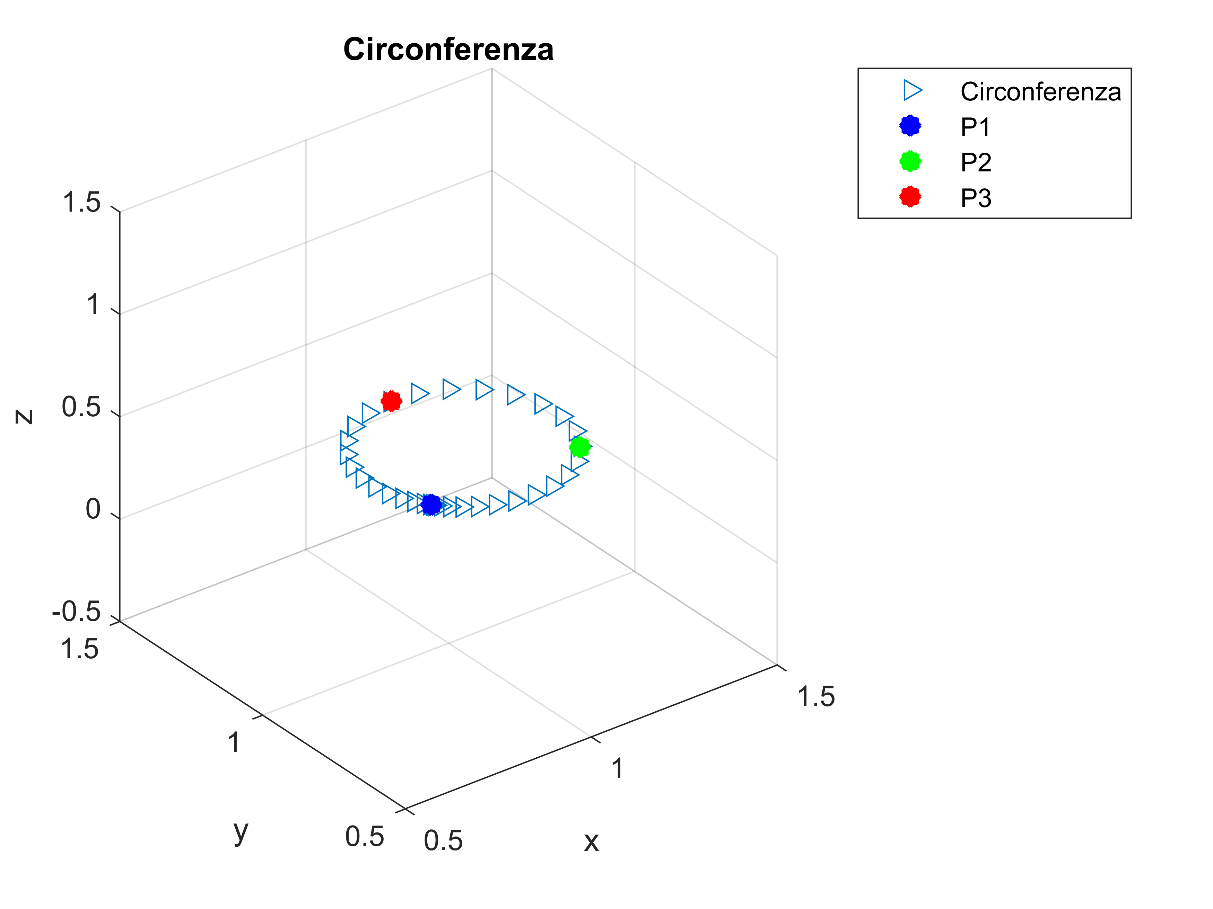
Utilizzando i principi della trigonometria, si ottiene che .

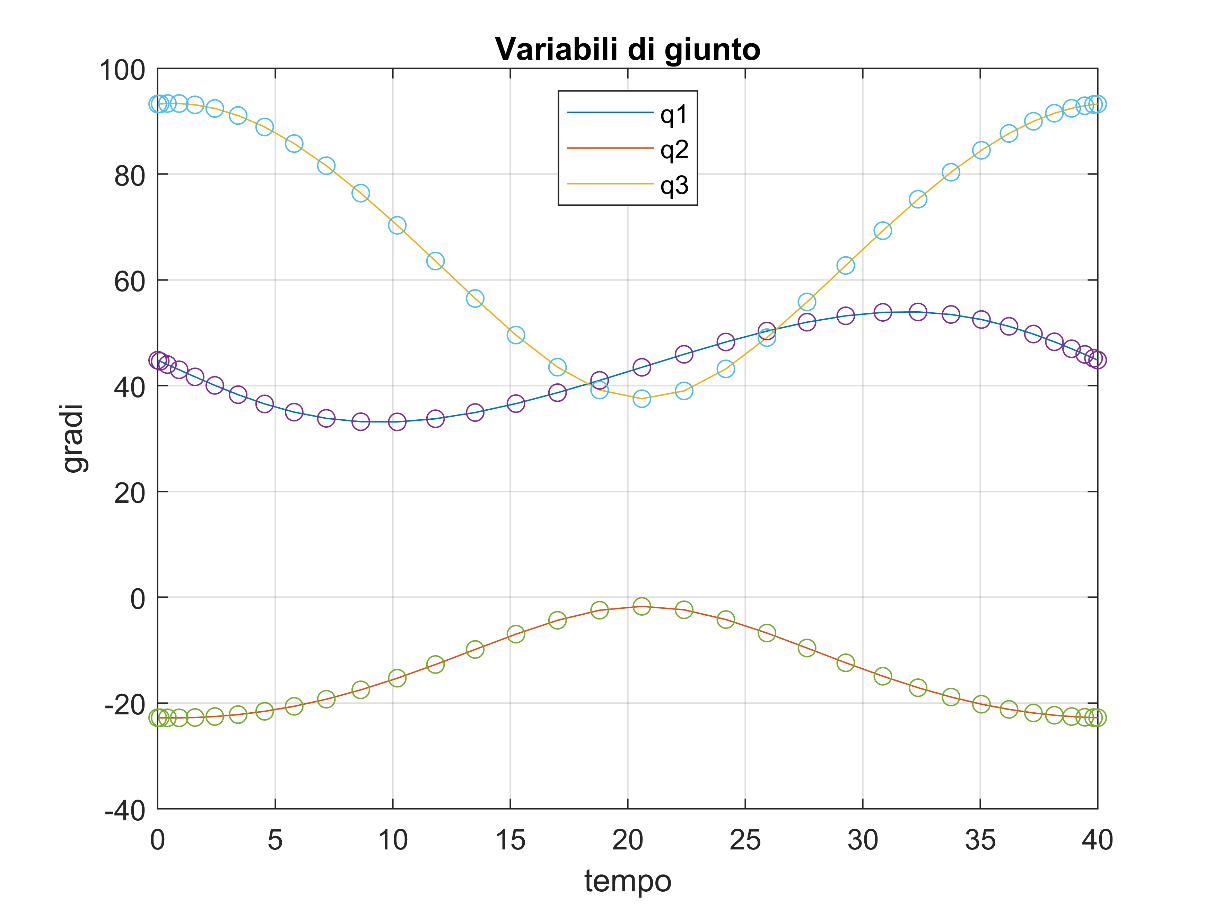
Inoltre, è necessario esprimere theta in funzione di λ: .

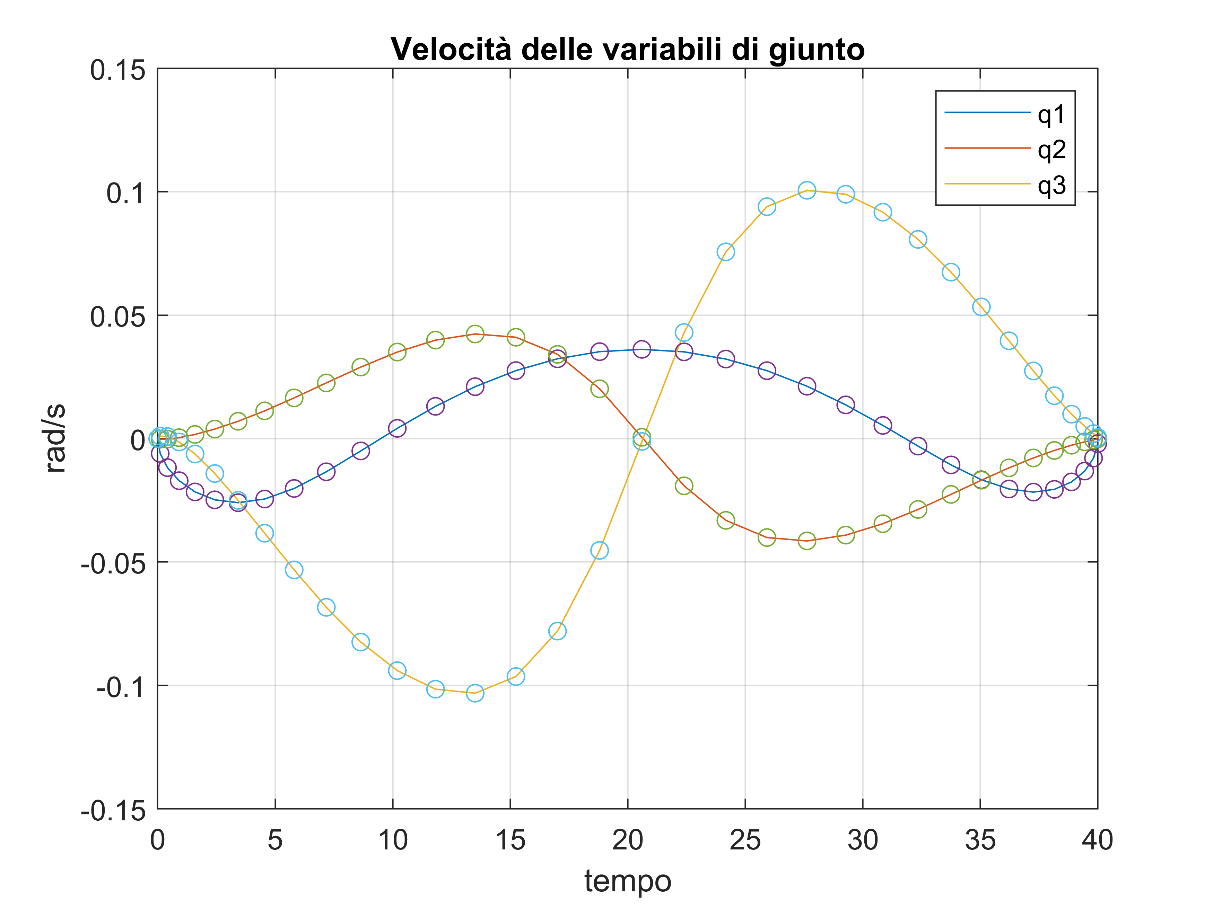
Nel caso in esame: .

Analogamente al caso precedente, si percorre il tratto P1 -> P2 -> P3 -> P1 e si ricavano le variabili di giunto applicando la cinematica inversa e la cinematica differenziale inversa:

|  |
| --- |
| % Percorso Circonferenza  for i=1:N  P = PC+r\*[cos(theta(i)); sin(theta(i)); 0];  Q = Antropomorfo\_Cin\_Inv(L,P);  J = Jacobiano\_Antropomorfo(L,Q');  Pd=(r\*[-sin(theta(i)), cos(theta(i)),  0]\*lambda\_d(i)\*2\*pi)/(T4-T1);  Qd = inv(J)\*Pd';  QQc(i,:)=Q; PPc(i,:)=P; QQdc(i,:)=Qd;  end |







# ROBOTICA MOBILE

## Richiesta

Dopo aver definito un ambiente con ostacoli, si progetti un codice che sia in grado di far muovere un robot da un punto di start ad un punto di goal con un determinato orientamento finale. Per farlo bisogna implementare il seguente schema di controllo:

Immagine che contiene testo, schermata, diagramma, Carattere

Descrizione generata automaticamente

In particolare,

* l’utente deve poter scegliere il Task, ovvero posizione iniziale, finale ed orientamento del robot
* l’utente deve poter scegliere quale tra i seguenti Planner usare: Artificial Potential Fields, Discrete Potential Fields, Voronoi Diagrams, Visibility Graphs (di quest’ultimo la versione completa è preferibile)
* l’utente deve poter scegliere anche il tipo di Controller che guida il robot in modo da fargli seguire la traiettoria definita al punto precedente: Lineare, Non Lineare, Input/Output Linearization
* il Mobile Robot deve essere di tipo uniciclo o differential drive
* l’Environment è prestabilito
* non è necessario implementare il blocco Sensors

## Guida

Una volta lanciato lo script *Mobile.mlx*, verrà plottato l’ambiente all’interno del quale si andrà a muovere il robot. Scorrendo in basso dovrebbe apparire.

A questo punto il programma attenderà che nella command window vengano inseriti i dati richiesti. Dovrebbe verificarsi una situazione del genere:

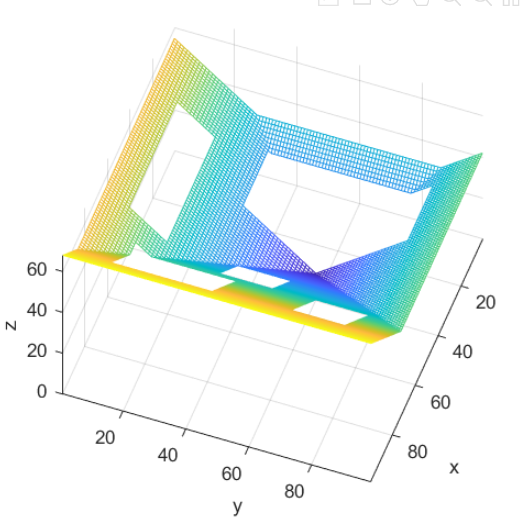
Immagine che contiene testo, schermo, schermata, Rettangolo

Descrizione generata automaticamente

Dopodiché il programma continuerà la sua esecuzione e scorrendo in basso si potranno visualizzare graficamente i risultati ottenuti in base ai parametri inseriti. All’occorrenza, sarà chiesto di inserire eventuali parametri come appena visto.

## Discrete Potential Fields

Ricordare di descrivere il caso anomalo in cui non si può generare una traiettoria



Dalla figura si nota che ci sono due punti in cui il robot si può fermare senza generare una traiettoria per arrivare al goal. Questo a causa della forma degli ostacoli e della natura stessa del metodo dei potenziali artificiali discreti.