**Relazione del Progetto di Robotica**

**Pianificazione delle Traiettorie**

*Giuseppe Montanaro, matricola 62719*

*Simone Smaldore, matricola 62728*

**Introduzione**

Il progetto sviluppato comprende due applicativi in ambiente MATLAB per la pianificazione di traiettorie dato un percorso nello spazio dei giunti. Gli algoritmi tra cui l’utente può scegliere sono i seguenti:

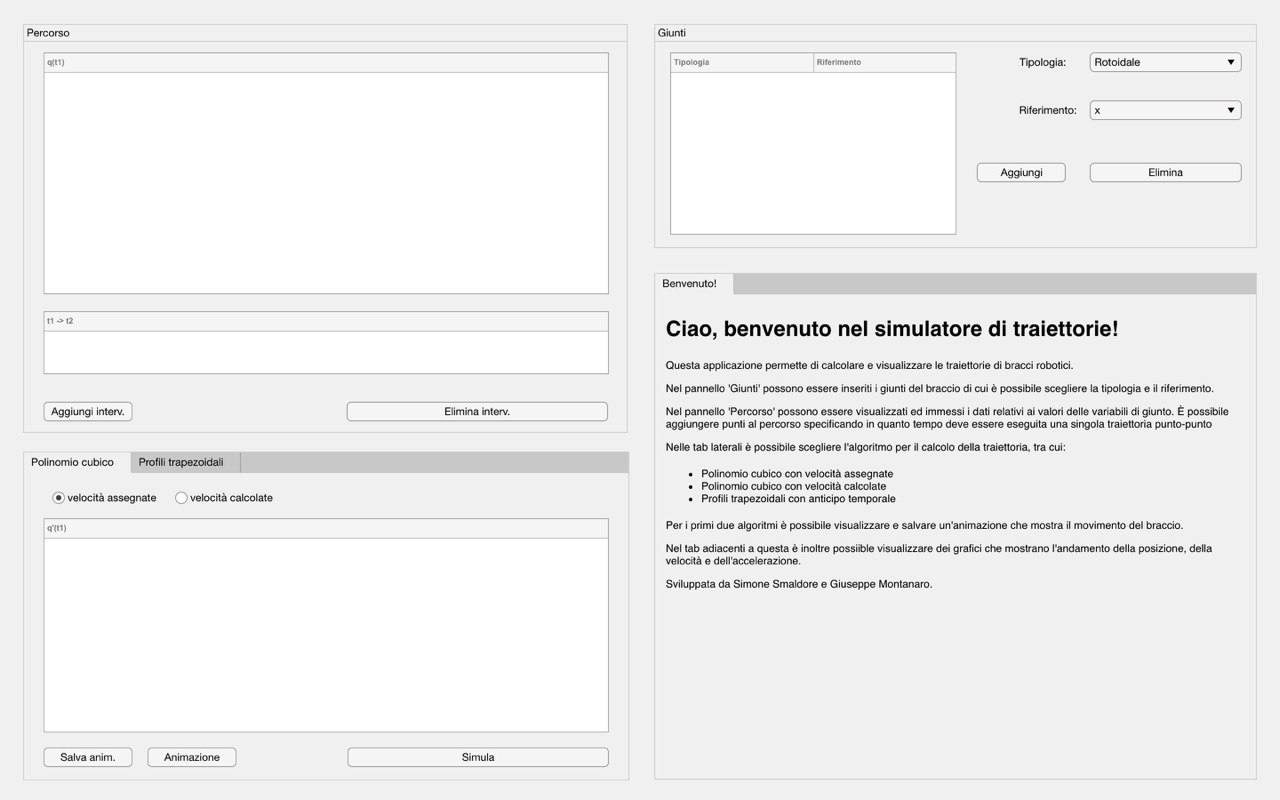
* sequenza di ***polinomi cubici con velocità assegnate*** nei punti di percorso;
* sequenza di ***polinomi cubici con velocità calcolate*** nei punti di percorso;
* sequenza di ***profili trapezoidali con anticipo temporale*** (passaggio in prossimità dei punti di via) con anticipo specificato dall’utente.

È stata sviluppata una prima versione più basilare da *linea di comando* (CLI) per velocizzare la scrittura ed il testing degli algoritmi. Successivamente è stata progettata un’*interfaccia grafica* (GUI) per permettere all’utente di inserire le informazioni necessarie in maniera più intuitiva ed ottenere una visualizzazione dei dati più ordinata. Il codice del progetto è scaricabile al seguente [link](https://drive.google.com/drive/folders/1pXfT7Zjib8STOCQ5gP_FJ37OC2pBakDT?usp=sharing).

Per utilizzare la versione da linea di comando occorre eseguire lo script Principale.m, invece, per la versione con GUI, va avviata l’applicazione main.mlapp. Le due implementazioni hanno di diverso unicamente le modalità con cui l’utente interagisce con l’applicazione, infatti la logica che governa entrambe è la stessa. Per l’inserimento dei dati, la CLI usufruisce di apposite funzioni (immettiPercorso, leggiTempi, leggiVelocita, leggiParametriC), mentre la GUI, com’è facile intuire, adopera bottoni e tabelle. È giusto sottolineare che le funzioni per stampare i grafici degli algoritmi sono specifiche per le due implementazioni al solo scopo di adattarsi al meglio ai due ambienti differenti; nonostante ciò il risultato in output è ovviamente sempre lo stesso. Per tale motivazione si analizzerà solo l’applicativo con interfaccia grafica.

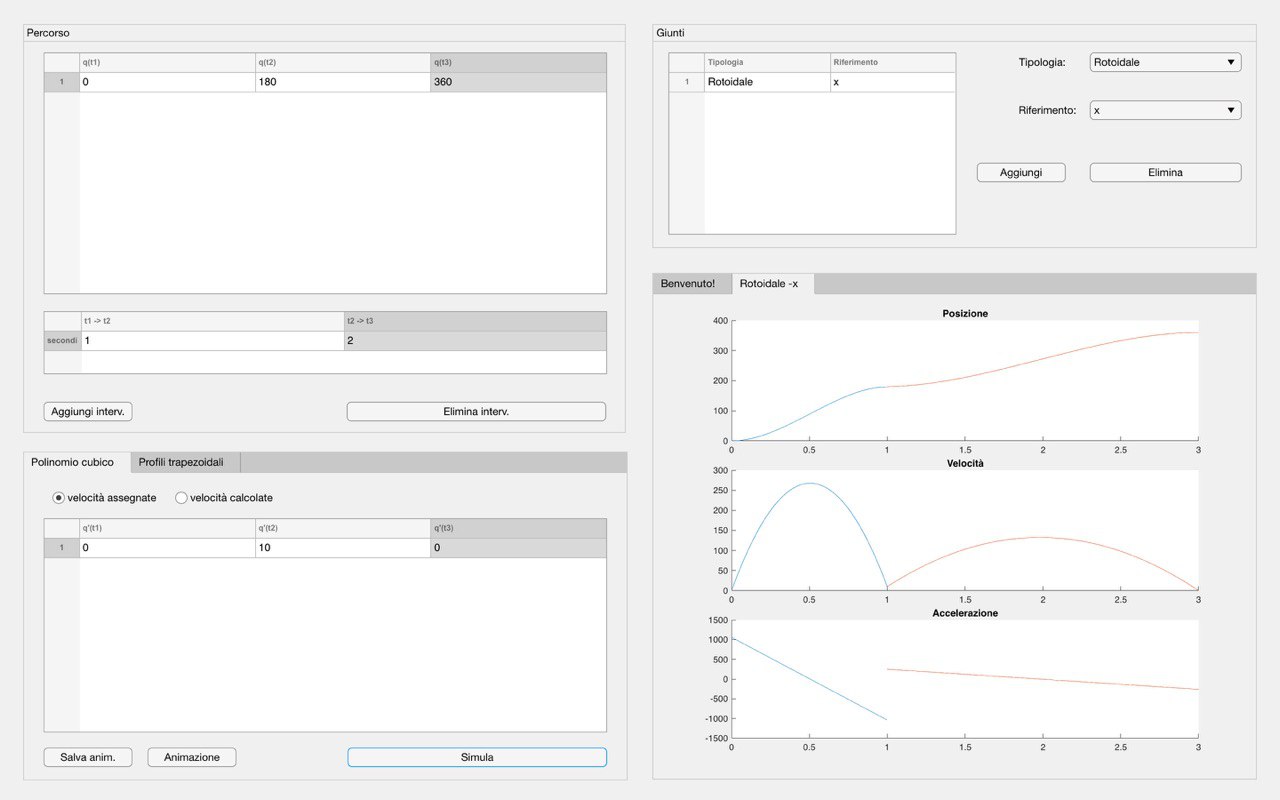
**Interfaccia e Funzionamento Generale**

Nella schermataè presente *in basso a destra* un tab di benvenuto che spiega in linea di generale il funzionamento dell’applicazione. Successivamente i grafici saranno stampati in nuovi tab adiacenti a questo. In *alto a destra* c’è un pannello che si occupa della ***gestione dei******giunti***. In questo è presente una tabella indicante i giunti attualmente inseriti dall’utente, quest’ultimo ha la possibilità di aggiungere giunti rotoidali o prismatici che ruotano o si distendono rispetto ad uno degli assi della terna base. In *alto a sinistra* si può ***descrivere il percorso*** nello spazio dei giunti specificando il valore delle variabili di giunto nei vari istanti di tempo, per semplicità di immissione si è scelto di esprimere le variabili di giunto rotoidali in gradi e non con i radianti. È possibile utilizzare la tabella sottostante per inserire la durata degli intervalli di tempo, i quali possono essere aggiunti ed eliminati tramite gli appositi pulsanti, come istante iniziale si considera sempre . La schermata in *basso a sinistra*, va analizzata in maniera più dettagliata poichè cambierà comportamento in base all’algoritmo.



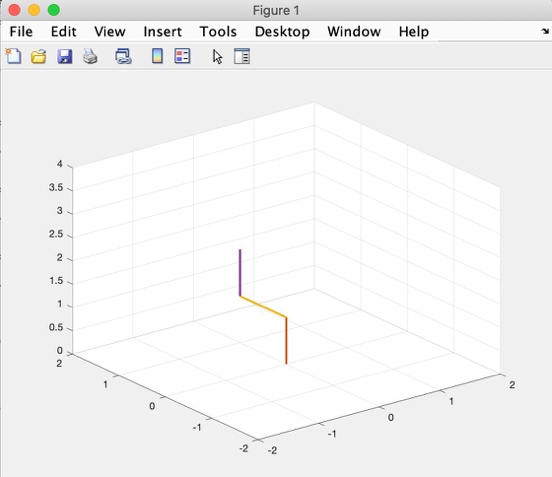
**Sequenza di polinomi cubici con velocità assegnate nei punti di percorso**

Per eseguire il primo algoritmo occorre selezionare *“velocità assegnate”* e indicare le velocità nella tabella successiva. Una volta cliccato il pulsante “*Simula”* verrà invocata la corrispondente funzione di ***callback***, la quale si occupa unicamente di chiamare a sua volta le funzioni generaPolinomiCubici e stampaGraficoGUI le quali si occupano rispettivamente della logica dell’algoritmo e della stampa dei grafici di posizione, velocità e accelerazione negli intervalli inseriti per ogni giunto considerato.



Per interpolare tra due singoli punti è stato considerato un polinomio di terzo grado nella forma:

I coefficienti del polinomio in questione sono stati calcolati attraverso l’imposizione dei seguenti vincoli, i quali sono stati forniti dall’utente per ogni singolo percorso punto-punto:

L’applicazione consente anche di visualizzare una figura MALTAB in 3D che **simula la traiettoria** rispetto ai valori indicati. È possibile accedervi cliccando sul pulsante *“Animazione”*. Purtroppo per motivi di tempo e a causa di una complessità che supera eccessivamente i propositi di questo progetto, non è stato possibile sviluppare una simulazione soddisfacente sotto tutti i punti di vista. La traiettoria, infatti, verrà simulata correttamente solo per i giunti rotoidali, mentre i giunti prismatici si muoveranno solamente rispetto direzioni della terna base ). L’applicazione permette anche di salvare l’animazione in formato mp4.

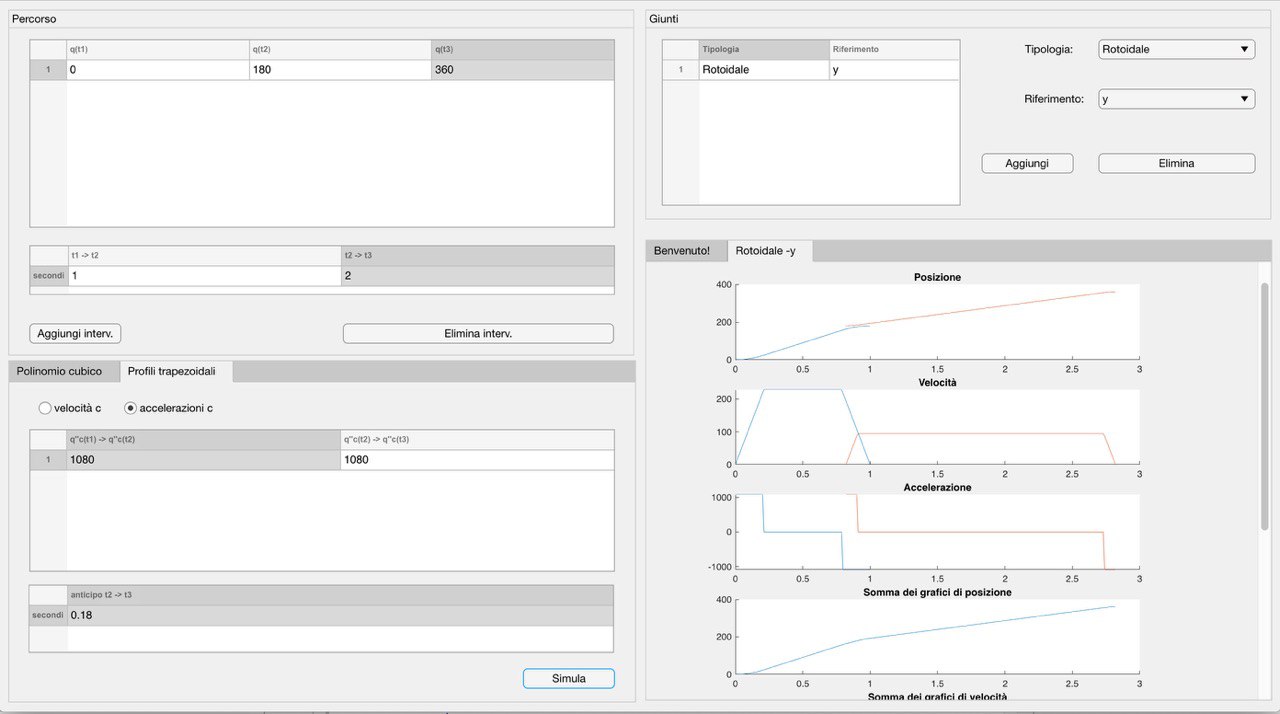
**Sequenza di polinomi cubici con velocità calcolate nei punti di percorso**

Questo algoritmo è molto simile al precedente: l’unica sostanziale differenza è che una volta selezionata la corrispondente opzione, non occorrerà più inserire le velocità nella tabella poiché queste sono calcolate autonomamente dall’algoritmo a partire dal percorso e dai tempi assegnati, grazie alla funzione la funzione calcolaVelocita. Per implementare questo metodo sono state utilizzate le seguenti formule:

Con .

**Sequenza di profili trapezoidali con anticipo temporale**

L’interfaccia permette di specificare valori o in base alla selezione effettuata nella checkbox. È possibile specificare gli anticipi per i trapezi, dei quali occorre indicare la singola durata (gli anticipi vanno considerati tra il trapezio k-esimo ed il trapezio k-1-esimo poiché verranno cumulati in automatico dall’algoritmo).



Cliccando il tasto *“Simula”* verrà invocata la funzione generaTrapezi tramite il callback. La funzione calcola i ,e per ogni intervallo di tempo basandosi sulle le seguenti formule adoperando quelle corrette in base alla scelta dell’utente di immettere le velocità o le accelerazioni:

Prima di effettuare la simulazione vengono effettuate delle convalide per verificare di aver calcolato un consistente (cioè reale e minore o uguale di ). A tal fine sono applicati i seguenti vincoli:

Nel caso in cui anche un solo valore non rispetti tali limitazioni, l’algoritmo non prosegue e viene mostrato un messaggio di errore. Una volta superate le convalide viene eseguito il metodo stampaGraficotrapeziGUI per generare i grafici di posizione, velocità, accelerazione, somma di posizione, somma di velocità e somma di accelerazione per ognuno dei giunti considerati, implementando nel metodo le seguenti formule:

**Codice e Algoritmi**

In questa ultima sezione è discusso in codice più nel dettaglio, in particolare sono analizzate le funzioni che permettono il calcolo della traiettoria che costituiscono il cuore del progetto.

**Sequenza di polinomi cubici con velocità assegnate nei punti di percorso**

****

La funzione GeneraPolinomiCubici si occupa di generare i parametri per il giunto j-esimo nell’intervallo di tempo i-esimo. Questi vengono immagazzinati in posizione (j, i, :) in una matrice tridimensionale di dimensioni mxnx4 (n.giunti x num. intervalli x num. parametri). Nel ciclo interno viene risolto un sistema lineare tra la matrice Q parametrizzata dagli estremi dell’intervallo e i valori immessi dall’utente per posizione e velocità.

**Sequenza di polinomi cubici con velocità calcolate nei punti di percorso**



In questa funzione sono state implementate le formule esposte precedentemente.

**Sequenza di profili trapezoidali con anticipo temporale**

****

La funzione generaTrapezi calcola i valori , e i quali sono memorizzati in matrici mxn (n.giunti x num. intervalli). isVelocita determina quando è true il fatto che l’utente ha immesso i valori di altrimenti l’algoritmo assume che siano di . Ad ogni iterazione sono applicate le formule descritte precedentemente.



In questa funzione sono applicate le convalide per verificare che abbia valori consistenti.



La funzione calcolaValoriGiunti restituisce tre matrici tridimensionali in cui ogni valore rappresenta il valore di posizione, velocita o accelerazione del giunto j-esimo, all’intervallo k-esimo dell’istante dt-esimo dell’intervallo k-esimo. La variabile passo è utilizzata per gestire il campionamento dell’intervallo t in modo tale che ad ogni iterazione la lunghezza di ogni campione è la stessa. Le variabili intervalloRel e intervalloLen sono utilizzate nelle formule già discusse al fine di rendere relativo all’inizio dell’intervallo ogni istante di tempo t(dt).





Questa funzione è adoperata principalmente per il disegno dei grafici ed in questo momento viene attuato l’anticipo temporale. Infatti nel ciclo vengono considerati valori di posizione, velocità e accelerazione del giunto j-esimo nell’intervallo k-esimo (questa funzione viene chiamata da un callback dell’interfaccia che itera su tutti i giunti, in modo da mostrare i grafici di ogni giunto in una singola tab). i valori vengono anticipati adoperando la variabile anticipo e vengono salvati nella riga k-esima di una nuova matrice. Quest’ultima viene utilizzata per stampare i grafici sommando tutte le righe per ottenere i valori delle ordinate del grafico. Nel caso della posizione per trovare i valori corretti viene usata la funzione calcolaConSovrapposizione.