

# Progetto Automazione e Robotica

Federica Ferraguti  
federica.ferraguti@unimore.it

Il progetto deve essere svolto utilizzando i software MATLAB e CoppeliaSim (per la visualizzazione) e costituisce parte integrante dell'esame di Automazione e Robotica.

La **relazione** contenente le risposte alle richieste, la descrizione del metodo risolutivo e del codice deve essere inviata al docente tramite mail **almeno 48 ore prima dell'appello di esame** che si vuole sostenere. Nella stessa email devono essere allegati i **codici sorgente**, raccolti in un file zip.

Il progetto pesa +/- 3 punti sul voto finale dell'esame. Si può accedere all'appello d'esame solo se ogni punto del progetto è stato sviluppato.



Figura 1: Il manipolatore "Puma 560"

I parametri relativi al robot "Puma 560" presentato in Fig. 1 sono espressi nelle tabelle 1 e 2 per i 6 giunti che compongono il manipolatore.

Parametri	1	2	3	4	5	6
$\alpha$	$\pi/2$	0	$-\pi/2$	$\pi/2$	$-\pi/2$	0
$a$	0	0.4323	0	0	0	0
$\theta$	0	0	0	0	0	0
$d$	0	0	0.1501	0.4331	0	0

Tabella 1: I parametri di Denavit-Hartenberg del manipolatore "Puma 560"

Parametri	1	2	3	4	5	6
$m$	0	17.4000	4.8000	0.8200	0.3400	0.0900
$r_x$	0	-0.3638	-0.0203	0	0	0
$r_y$	0	0.0060	-0.0141	0.0190	0	0
$r_z$	0	0.2275	0.0700	0	0	0.0320
$I_{xx}$	0	0.1300	0.0660	0.0018	0.0003	$.15e-3$
$I_{yy}$	0.3500	0.5240	0.0860	0.0013	0.0004	$.15e-3$
$I_{zz}$	0	0.5390	0.0125	0.0018	0.0003	$.04e-3$
$I_{xy}$	0	0	0	0	0	0
$I_{yz}$	0	0	0	0	0	0
$I_{xz}$	0	0	0	0	0	0

Tabella 2: I parametri dinamici del manipolatore "Puma 560"

La Tab. 1 contiene i parametri di Denavit-Hartenberg del manipolatore mentre la Tab. 2 contiene i parametri utili per determinare la dinamica del manipolatore:  $m$  rappresenta la massa del link,  $r_x, r_y, r_z$  rappresentano le coordinate della posizione del baricentro del link rispetto all'origine del sistema di riferimento,  $I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{yz}$  rappresentano le componenti della matrice di inerzia del link.

Si richiede di:

1. Determinare le 6 matrici  $A_i^{i-1}$  che rappresentano le rototraslazioni necessarie per passare dal sistema di riferimento  $i-1$  associato ad un link (definito secondo la convenzione di Denavit-Hartenberg) al successivo sistema di riferimento  $i$ . Descrivere il procedimento utilizzato.
2. Determinare la matrice complessiva di rototraslazione  $T_6^0$  del manipolatore per le seguenti posizioni (esprese nelle variabili di giunto):
  - a.  $q(1) = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$
  - b.  $q(2) = [0 \ 0 \ -\pi/X \ 0 \ 0 \ 0]^T$
  - c.  $q(3) = [0 \ \pi/X \ -\pi \ 0 \ 0 \ 0]^T$
  - d.  $q(4) = [0 \ \pi/4 \ -\pi \ 0 \ \pi/X \ 0]^T$
  - e.  $q(5) = [0 \ 0 \ -\pi/2 \ \pi/6 \ 0 \ -\pi/X]^T$
  - f.  $q(6) = [0 \ -\pi/X \ 0 \ 0 \ -\pi/4 \ 0]^T$

dove  $X$  è da sostituire con l'ultima cifra a destra del numero di matricola dello studente. Qualora  $X = 0, 1$  considerare  $X = 2$ .

Nota: Implementare in MATLAB le matrici del punto 1. e calcolare la matrice  $T_6^0$  complessiva. Verificare che il risultato ottenuto coincida con quello restituito dalla funzione del Toolbox di Corke per il calcolo della cinematica diretta.

3. Pianificare una traiettoria di posizione rettilinea nello spazio operativo, con profilo trapezoidale di velocità per congiungere i punti  $p_i = [0.0X \ -0.45 \ -0.05 \ 0 \ 0 \ 0]^T$  e  $p_f = [0.60 \ 0.15 \ 0.0X \ 0 \ 0 \ 0]^T$ , dove  $X$  è da sostituire con la penultima cifra a destra del numero di matricola dello studente. Gli orientamenti sono definiti con la convenzione RPY. Tale traiettoria deve essere percorsa in 3 secondi partendo e terminando con velocità nulla e con accelerazione iniziale e decelerazione finale (uguali tra di loro) della durata di 0.5 secondi ciascuna.

Esprimere poi la stessa traiettoria nello spazio dei giunti, partendo dalla conversione dei punti iniziale e finale nello spazio dei giunti. Considerare gli stessi vincoli su tempi e velocità.

Nota: Implementare in MATLAB la traiettoria sia nello spazio operativo che nello spazio dei giunti e visualizzare i grafici di posizione, velocità ed accelerazione in entrambi gli spazi. Verificare che il risultato sia analogo a quello calcolato dalla funzione preimpostata di MATLAB costruita a lezione.

Muovere il robot in CoppeliaSim con le posizioni di giunto calcolate (movimentazione solo cinematica).

4. Determinare le coppie necessarie da applicare al manipolatore per descrivere la traiettoria calcolata precedentemente.

Nota: Visualizzare in MATLAB le coppie calcolate.

5. Si supponga il manipolatore fermo in posizione iniziale  $\mathbf{p}_i = [0.10 \ -0.4\mathbf{X} \ -0.15 \ 0 \ 0 \ 0]^T$  dove  $\mathbf{X}$  è da sostituire con la penultima cifra a destra del numero di matricola dello studente. L'orientamento è definito con la convenzione RPY. Si progetti un controllo di tipo PD con compensazione di gravità nello spazio di giunto con le matrici per l'inseguimento della traiettoria desiderata calcolata precedentemente.

Nota: Implementare in MATLAB e visualizzare l'andamento dell'ingresso  $u(t)$ . Comandare le coppie al modello del robot in CoppeliaSim e generare un video della simulazione. Riportare i parametri  $\mathbf{P}$  e  $\mathbf{D}$  utilizzati ed esporre come si modificano i risultati ottenuti al variare di questi ultimi.

6. Per risolvere il problema di inseguimento di cui al punto sopra si progetti un controllo di tipo a dinamica inversa. Si confrontino i risultati con quelli ottenuti al punto precedente.

Per lo svolgimento del progetto **NON** si deve utilizzare il modello del Puma 560 già presente in MATLAB, poiché i parametri sono stati adattati e semplificati, quindi non risulterebbero coerenti.