

# Progetto di Robotica Industriale

Giorgio Ubbriaco  
209899

## Sommario

Abstract .....	3
Introduction .....	3
Model .....	3
$\lambda$ and $\sigma$ definition .....	4
Time Definition .....	4
Triangle .....	5
Circumference .....	8
Triangle and Circumference with Time .....	12

## Abstract

Il progetto in questione viene presentato facendo riferimento a piccoli cenni di teoria ed evidenziando gli aspetti implementativi delle tecniche e dei modelli considerati e mettendo in luce i risultati ottenuti. È stato utilizzato, come supporto ai plot di simulazione, il Robotics Toolbox sviluppato da Peter Corke.

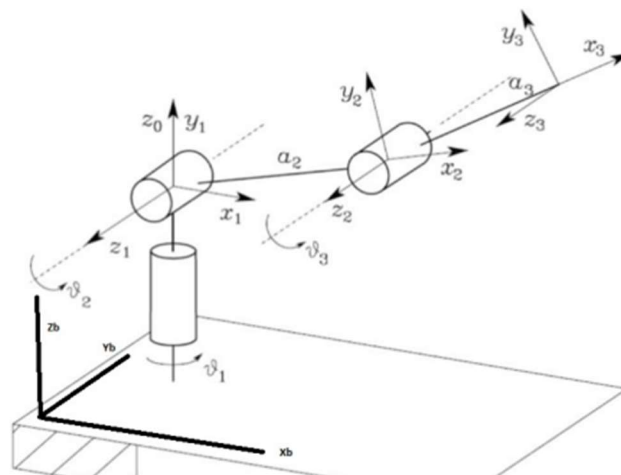
## Introduction

L'obiettivo di questo progetto di robotica industriale è quello di progettare due algoritmi che permettono al robot industriale di tipologia *puma560* di descrivere, in un tempo totale pari a 40 secondi, la traiettoria di un triangolo e quella di una circonferenza passante per tre punti sotto citati, tenendo conto di una tabella di D-H rappresentante il robot antropomorfo in questione.

$$P_1 = \begin{pmatrix} 0.8 \\ 0.8 \\ 0.5 \end{pmatrix} \quad P_2 = \begin{pmatrix} 1.2 \\ 0.8 \\ 0.5 \end{pmatrix} \quad P_3 = \begin{pmatrix} 1.0 \\ 1.2 \\ 0.5 \end{pmatrix}$$

## Model

Un robot industriale di tipologia antropomorfa è un robot a base fissa, cioè manipolatore, capace di riprodurre alcune caratteristiche dell'uomo, di imitarne tratti distintivi come l'aspetto e i movimenti.



tale che i sistemi di riferimento (0) e (b) sono legati dalla seguente matrice di rototraslazione:

$$\hat{R}_0^b = \begin{pmatrix} 1 & 0 & 0 & 0.5 \\ 0 & 1 & 0 & 0.5 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

link	a (m)	$\alpha$	d (m)	$\theta$
1	0	$\frac{\pi}{2}$	0	$\theta_1$
2	0.9	0	0	$\theta_2$
3	0.9	0	0	$\theta_3$

Da tale tabella si può evincere come il manipolatore in questione è composto da tre bracci di cui il primo di lunghezza nulla e i restanti due pari a  $0.9\text{ m}$ . Esso presenta un angolo di rotazione intorno alla normale comune, in riferimento al primo link, pari a  $\frac{\pi}{2}$  *radianti* e distanze dell'asse  $Z_{n-1}$  dalla normale comune nulle.

## $\lambda$ and $\sigma$ definition

Si definisce un  $\lambda$  utile alla discretizzazione dei percorsi. Infatti, come verrà citato nei prossimi paragrafi, sia il triangolo che la circonferenza verranno divise in tre percorsi differenti ognuna avente come estremi due punti dei tre prima citati. Il  $\lambda$  è stato ottenuto considerando un polinomio standardizzato di grado tre poiché i vincoli di posizione e velocità da determinare sono quattro. Volendo, si poteva scegliere un polinomio di grado maggiore ma per tale applicazione è stato sufficiente considerare questo grado appena citato. Inoltre, viene definito un  $\sigma$  affinché si possa definire nel migliore dei modi il polinomio in questione.

In questo caso, è stato scelto un passo di  $\sigma$  pari a 0.05 e, pertanto, un  $\sigma$  avente come estremi 0 e 1 discretizzato appunto a passo di 0.05.

## Time Definition

Il tempo totale da considerare, come prima già citato, per la copertura sia del percorso del triangolo sia per il percorso della circonferenza è pari a *40 secondi*. Pertanto, considerando tre punti nello spazio, ogni percorso che congiunge un punto all'altro, dovrà essere "coperto" in un tempo pari a  $\frac{40}{3}$ . Quindi, si definisce il tempo necessario per coprire un percorso generico dal punto  $P_i$  al punto  $P_j$  nel seguente modo:

$$t_{i,j} = T_i + \lambda \cdot (T_j - T_i)$$

dove  $T_i = T_j + \left(\frac{T_{tot}}{3}\right) \forall i, j = 1, 2, 3 \mid i \neq j$ . Bisogna precisare che  $i > j$  solo nel caso in cui rispettivamente  $i = 1, 2 \wedge j = 2, 3$  e  $j > i$  solo nel caso in cui rispettivamente  $i = 1 \wedge j = 3$ . Inoltre, il caso particolare si ha inizialmente quando il manipolatore si trova nel punto  $P_i = P_1$  e deve dirigersi nel punto  $P_j = P_2$  e, pertanto, considerando  $T_1 = 0$ .

## Triangle

Il percorso del triangolo è stato ottenuto per discretizzazione considerando ogni percorso semplice che ha come estremi  $P_i$  e  $P_j$ . Per la precisione, l'insieme dei punti che caratterizza il percorso  $\Omega_k$  è stato ottenuto nella seguente maniera:

$$\Omega_k = P_i + \lambda(i) \cdot (P_j - P_i)$$

Quindi, considerando un percorso avente andamento lineare, si può notare come esso passa perfettamente per i punti  $P_1$ ,  $P_2$  e  $P_3$ .

Dopo aver ottenuto, l'andamento si può calcolare, tramite cinematica inversa, l'andamento delle variabili di giunto.

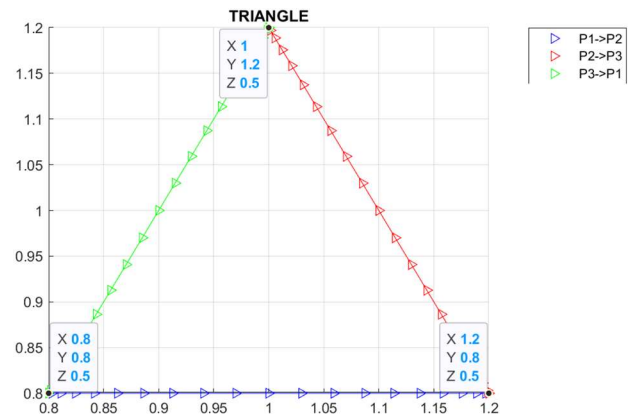
Dopo aver ottenuto questi ultimi valori, si è in grado di calcolare le velocità relative alla copertura del percorso e le velocità delle variabili di giunto. Le

velocità lineari dell'organo terminale si calcolano considerando la seguente relazione:

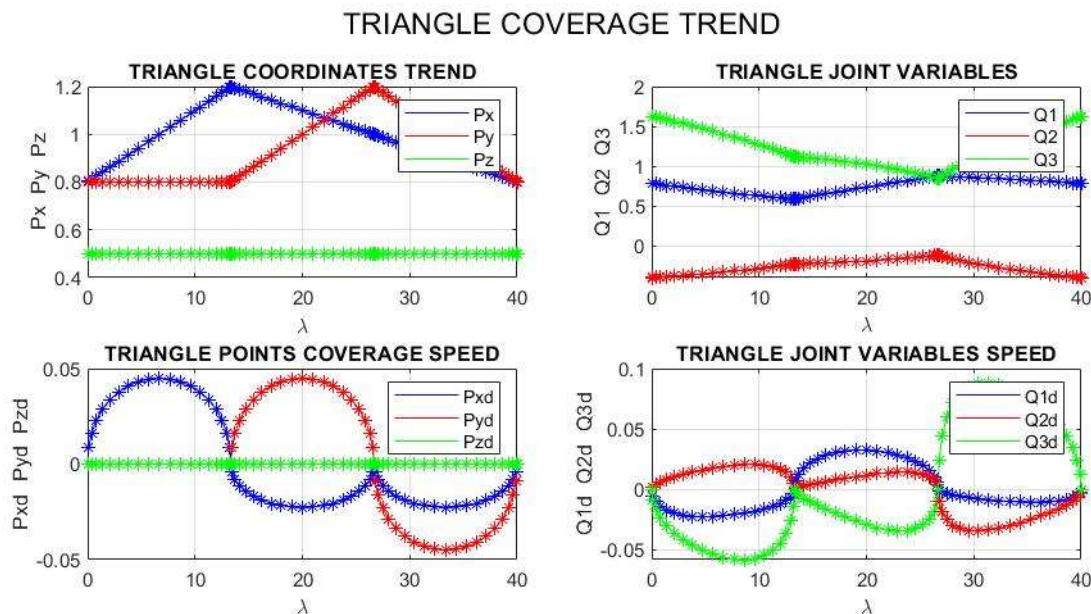
$$P_{d,n} = \frac{P_j - P_i}{T_j - T_i} \cdot \dot{\lambda}_i$$

Invece, per calcolare le velocità relative alle variabili di giunto si calcola inizialmente il Jacobiano  $J$  facendo riferimento sia ai link sia alle variabili di giunto. Essendo che il Jacobiano  $J$  è caratterizzato da solo tre righe linearmente indipendenti e avendo a disposizione tre gradi di libertà, è opportuno considerare solo il Jacobiano geometrico  $JP$ , cioè il blocco superiore ( $3 \times 3$ ) dello Jacobiano  $J$ . Si effettua un controllo relativo al determinante del Jacobiano geometrico ottenuto. Se quest'ultimo è nullo allora si è di fronte ad una singolarità cinematica e l'esecuzione termina con un errore, altrimenti si calcolano le velocità relative alle variabili di giunto facendo riferimento alla seguente relazione:

$$Q_{d,n} = JP^{-1} \cdot P_{d,i}$$



Dopo che si sono ottenuti tutti i dati relativi al percorso in questione, è possibile effettuare qualche analisi tramite alcuni plot. Pertanto, effettuando un plot relativo all'andamento delle coordinate dei punti  $P_x$ ,  $P_y$  e  $P_z$  rispetto al  $\lambda$ , si può notare come variano i percorsi rispetto alla variazione del parametro di discretizzazione  $\lambda$ . Si può notare come la coordinata  $P_x$  parta da un valore pari

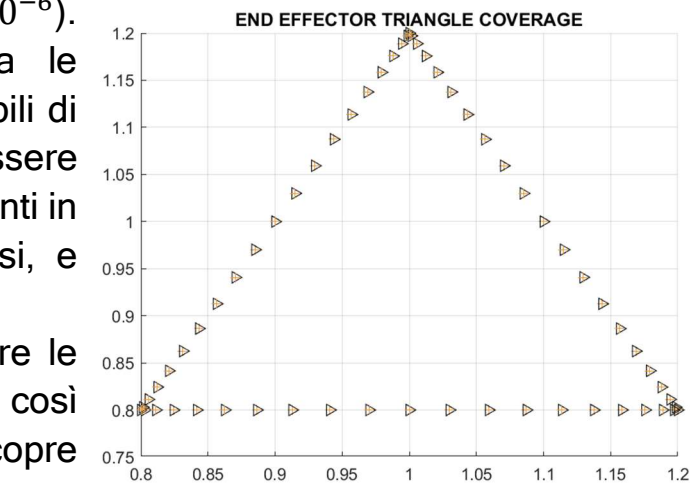
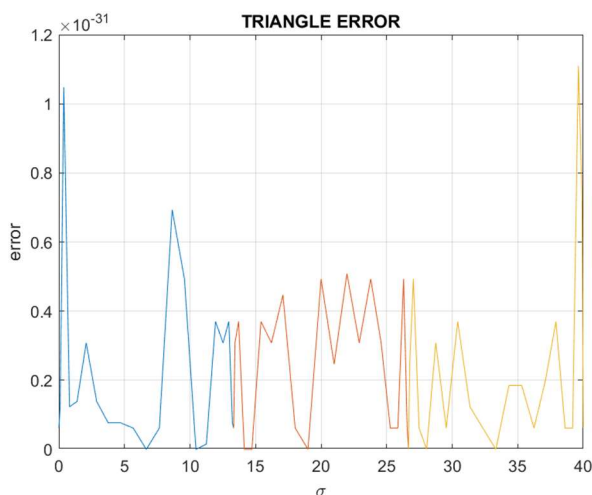


a 0.8 per arrivare ad un valore pari a 1.2 e, infine, terminare in un valore pari a 0.8 che sono per l'appunto le coordinate  $x$  dei punti  $P_1$ ,  $P_2$  e  $P_3$ . Analogamente succede anche per  $P_y$  e  $P_z$ . Si deve notare come, nelle coordinate raggiunte, ci sia un addensamento di punti poiché il manipolatore, passando tra la copertura di un percorso ad un altro, deve percorrere poco spazio e, pertanto, è come se il robot in questi punti diminuisca la velocità fino a fermarsi. Infatti, effettuando un plot relativo alla velocità di copertura del percorso, si può notare come le velocità nei punti prima citati siano nulle mentre nei restanti punti siano diverse da zero. Questo perché (per definizione di derivata essendo che essa rappresenta la variazione in un determinato punto) la derivata si annulla in presenza di staticità, cioè nei momenti in cui il manipolatore copre gli stessi punti (mantiene una posizione := "sta fermo"), mentre presenta valori diversi da zero quando è presente una variazione che in questo caso corrisponde ad una variazione di posizione del manipolatore. Inoltre, effettuando un altro plot relativo all'andamento delle variabili di giunto rispetto a  $\lambda$  (il quale risulta addensarsi nei punti in cui il robot rallenta fino a fermarsi come sopra citato), possiamo notare come il suo andamento è lineare dato che i percorsi descritti dal manipolatore sono anch'essi lineari. Tanto è vero che tale andamento potrebbe essere calcolato anche tramite la relazione  $\psi_n = Q_i + \lambda(i) \cdot (Q_j - Q_i)$  presupponendo di aver calcolato a priori tramite cinematica inversa le variabili

di giunto nei punti  $P_i$  e  $P_j$ . Tale approccio risulta essere valido poiché il manipolatore riesce perfettamente a coprire il percorso in questione e, pertanto, a passare per i punti  $P_1$ ,  $P_2$  e  $P_3$  mantenendo comunque un errore praticamente nullo (dell'ordine di  $10^{-6}$ ).

Infine, un ulteriore plot riguarda le variazioni delle velocità delle variabili di giunto, le quali possiamo notare essere pari a zero in corrispondenza dei punti in cui il robot rallenta fino a fermarsi, e diverse da zero negli altri punti.

Inoltre, molto importante è calcolare le posizioni che assume l'end effector così da vedere effettivamente come copre tale percorso e, soprattutto, con quale errore lo fa rispetto alla traiettoria di riferimento sopra citata. Quindi, sfruttando

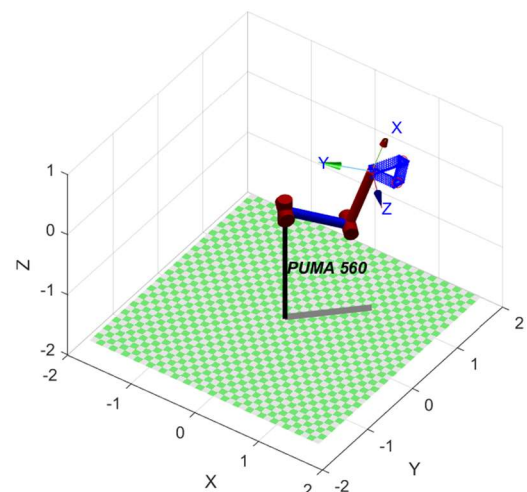


la cinematica inversa, viene calcolata posizione  $i$  –esima che assume l'end-effector nel coprire il percorso. Si può notare infatti che, i punti della traiettoria coperta dall'end effector (“+” di colore arancione) sono sovrapposti a quelli della traiettoria di riferimento (“>” di colore nero).

Inoltre, si può analizzare l'errore, cioè lo scarto tra l'andamento dell'end effector e l'andamento di riferimento. Si può notare

come esso sia praticamente nullo. Infatti, presenta valori davvero molto ridotti (dell'ordine di  $10^{-31}$ ).

Pertanto, si può effettuare un plot dinamico tramite l'ausilio del Robotics Toolbox di Peter Corke. Dopo aver definito, quindi, ogni link specificando i parametri  $\theta, d, \alpha$  e la tipologia di giunto (se rotoidale o prismatico), si può definire la struttura del robot tramite il costruttore *SerialLink* a cui si può attribuire il nome che identificherà il robot. Pertanto, abilitando a riga di comando il plot dinamico, sarà possibile osservare come il manipolatore riesce a passare perfettamente per i punti richiesti descrivendo il percorso del triangolo prima citato.





## Circumference

Il percorso della circonferenza è stato ottenuto considerando i tre archi di circonferenza aventi come estremi  $P_1$ ,  $P_2$  e  $P_3$ . Vengono calcolati gli angoli corrispondenti per tracciare gli archi in questione tramite la seguente relazione:

$$\theta_n = \theta_i + \lambda_i \cdot (\theta_j - \theta_i)$$

così da calcolare il punto  $n$  –esimo dell'arco (percorso)  $\Omega_k$  tramite la seguente relazione:

$$P_n = C + R \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix}$$

dove  $C$  ed  $R$  sono rispettivamente il centro ed il raggio della circonferenza in questione. Tali valori appena citati è possibile calcolarli tramite l'equazione della circonferenza ottenuta considerando il sistema lineare, caratterizzato da tre equazioni della sfera in cui sono stati sostituiti i tre punti in questione e, successivamente, facendo l'intersezione tra l'equazione della sfera ottenuta tramite il sistema lineare e l'equazione del piano  $z = 0.5$ , che è la coordinata che hanno in comune i tre punti in questione. Considerando, pertanto, l'equazione della sfera

$$x^2 + y^2 + z^2 + \alpha x + \beta y + \gamma z + \delta = 0$$

e sostituendo le coordinate dei punti, si ottiene

$$\begin{cases} 0.8^2 + 0.8^2 + 0.5^2 + 0.8\alpha + 0.8\beta + 0.5\gamma + \delta = 0 \\ 1.2^2 + 0.8^2 + 0.5^2 + 1.2\alpha + 0.8\beta + 0.5\gamma + \delta = 0 \\ 1.0^2 + 1.2^2 + 0.5^2 + 1.0\alpha + 1.2\beta + 0.5\gamma + \delta = 0 \end{cases}$$

tale che

$$\begin{cases} 0.8\alpha + 0.8\beta + 0.5\gamma + \delta = -1.53 \\ 1.2\alpha + 0.8\beta + 0.5\gamma + \delta = -2.33 \\ 1.0\alpha + 1.2\beta + 0.5\gamma + \delta = -2.69 \end{cases}$$

risolvendo il sistema tramite il metodo di eliminazione di Gauss-Jordan, si ottiene

$$\begin{cases} \alpha = -2 \\ \beta = -1.9 \\ \gamma = 3.18 - 2t \\ \delta = t \end{cases}$$

Si può notare come il sistema sia sotto determinato. Quindi, significa che esiste un "fascio" di sfere passante per i punti  $P_1$ ,  $P_2$  e  $P_3$ , cioè il sistema ammette infinite soluzioni. Considerando  $t = 0$  e, pertanto,  $\delta = 0$ , si ottiene la seguente equazione della sfera:

$$x^2 + y^2 + z^2 - 2x - 1.9y + 3.18z = 0$$

Sostituendo la coordinata  $z = 0.5$  che i punti hanno in comune, si ottiene

$$x^2 + y^2 - 2x - 1.9y + 1.84 = 0$$



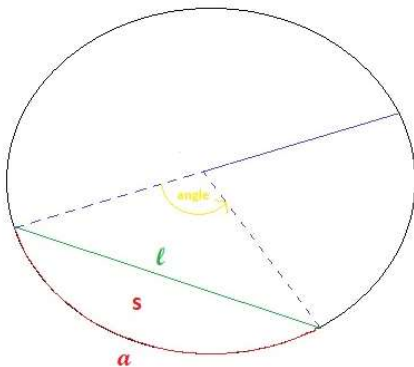
che corrisponde all'equazione della circonferenza passante per i punti  $P_1$ ,  $P_2$  e  $P_3$ . Pertanto, si può calcolare il centro ed il raggio della circonferenza rispettivamente come  $C = \left(-\frac{\alpha}{2}, -\frac{\beta}{2}, z\right)$  ed  $R = \sqrt{\frac{\alpha^2}{4} + \frac{\beta^2}{4} - \gamma}$ . Tenendo conto che  $\alpha = -2$ ,  $\beta = -1.9$ ,  $\gamma = 1.84$  e  $z = 0.5$ , si ottiene

$$C = (1, 0.95, 0.5) \quad R = 0.25$$

Successivamente, si effettua il calcolo dell'altezza corrispondente al triangolo ottenuto nella seguente maniera: si deve ricordare che  $d$  indica il diametro della circonferenza,  $p$  il segmento che congiunge il punto  $P_i$  generico e il centro della circonferenza,  $h$  l'altezza ottenuta come perpendicolare a  $d$  congiungente a  $P_i$  e, infine, *angle* la variazione di angolo ottenuta tra l'angolo di riferimento  $\theta = 360^\circ$  e l'angolo formato dal segmento  $p$  rispetto a  $\theta = 0$  cambiato di segno. Effettuando i calcoli si ottiene, quindi, un'altezza pari a 0.15. Pertanto, si calcola l'angolo *angle* tramite la seguente relazione:

$$\text{angle} = \arcsin \frac{h}{R}$$

A questo punto, si definiscono gli angoli corrispondenti ai punti  $P_1$ ,  $P_2$  e  $P_3$  così da discretizzarli secondo la relazione  $\theta_n = \theta_i + \lambda_i \cdot (\theta_j - \theta_i)$  e, successivamente, si calcolano i punti  $P_n$  appartenenti ai vari archi di circonferenza e, infine, si applica la cinematica inversa per ottenere le variabili di giunto corrispondenti.



Volendo il calcolo di *angle* poteva essere fatto in un ulteriore modo: considerando il settore circolare  $s$  e la corda  $l$  corrispondente all'arco di circonferenza  $a$  che coincide con il percorso tra due punti generici  $P_i$  e  $P_j$ , si ha la seguente

relazione:  $\text{angle} = 2 \cdot \arcsin \frac{a}{2 \cdot R}$ .

Pertanto, il punto generico  $P_n$  può essere ottenuto considerando una discretizzazione del tipo

$$P_n = R \cdot \theta \cdot \lambda_i$$

con  $0 \leq \lambda \leq 1$ .

In entrambi i modi si riesce ad ottenere una discretizzazione di tutti e tre i percorsi così da ottenere l'intera circonferenza passante per i tre punti. Considerando il primo approccio descritto (che è quello che è stato

implementato) e plottando i punti ottenuti, si può notare come i tre percorsi hanno come estremi i punti  $P_1$ ,  $P_2$  e  $P_3$ . Pertanto, dopo aver ottenuto i punti, si calcola l'andamento delle variabili di giunto tramite cinematica inversa. Dopo aver ottenuto quest'ultimi

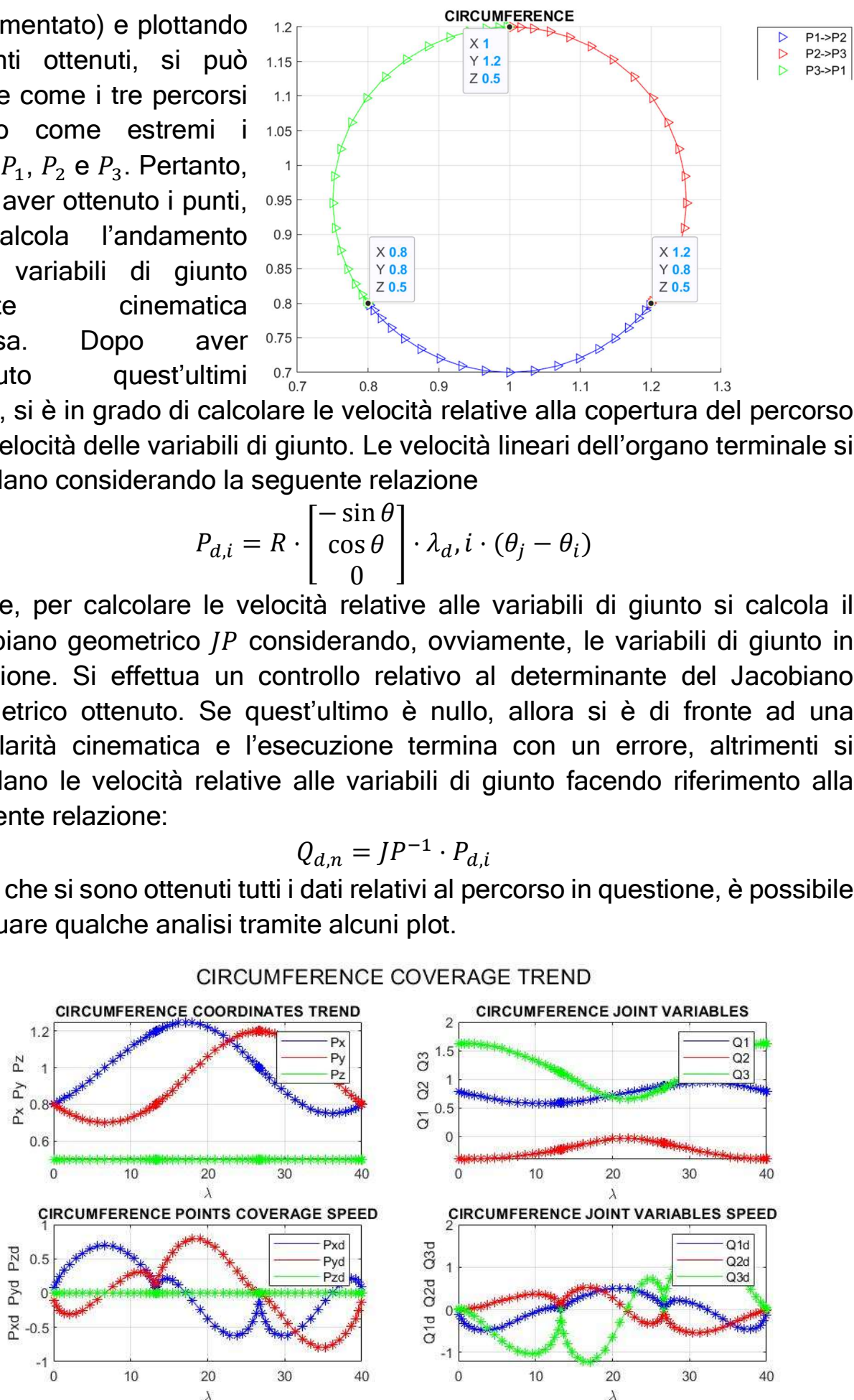
valori, si è in grado di calcolare le velocità relative alla copertura del percorso e le velocità delle variabili di giunto. Le velocità lineari dell'organo terminale si calcolano considerando la seguente relazione

$$P_{d,i} = R \cdot \begin{bmatrix} -\sin \theta \\ \cos \theta \\ 0 \end{bmatrix} \cdot \lambda_{d,i} \cdot (\theta_j - \theta_i)$$

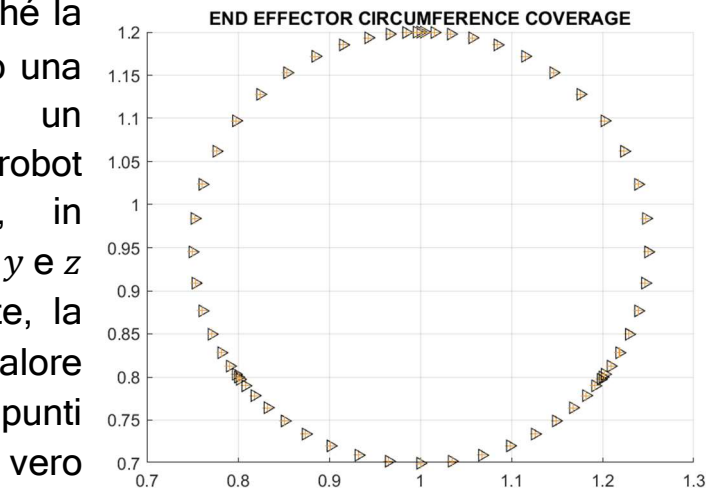
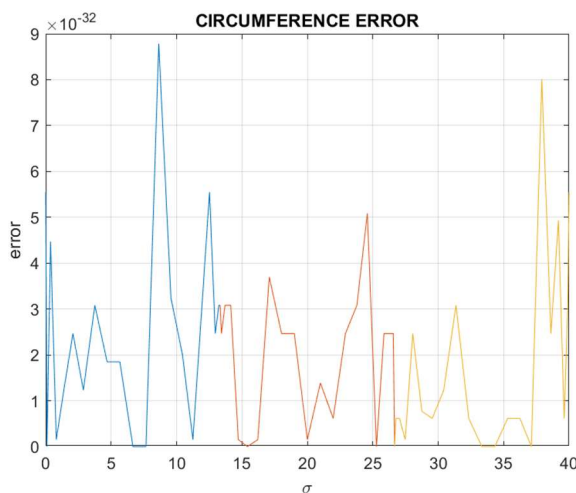
Invece, per calcolare le velocità relative alle variabili di giunto si calcola il Jacobiano geometrico  $JP$  considerando, ovviamente, le variabili di giunto in questione. Si effettua un controllo relativo al determinante del Jacobiano geometrico ottenuto. Se quest'ultimo è nullo, allora si è di fronte ad una singolarità cinematica e l'esecuzione termina con un errore, altrimenti si calcolano le velocità relative alle variabili di giunto facendo riferimento alla seguente relazione:

$$Q_{d,n} = JP^{-1} \cdot P_{d,i}$$

Dopo che si sono ottenuti tutti i dati relativi al percorso in questione, è possibile effettuare qualche analisi tramite alcuni plot.

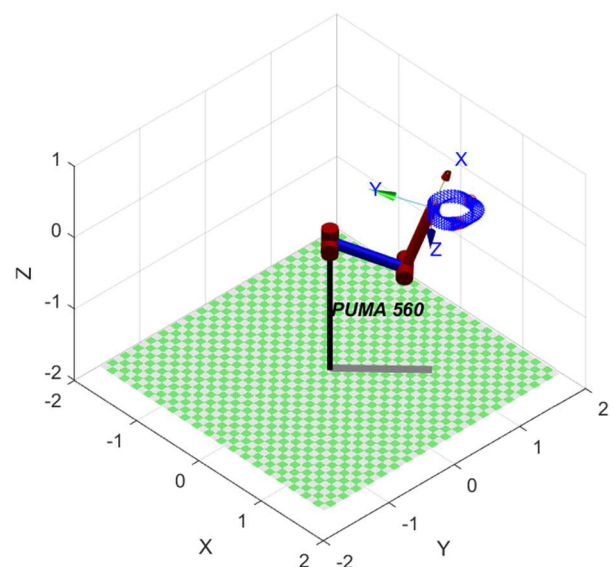


Si può notare come, in questo caso, l'andamento delle coordinate dei punti  $P_x$ ,  $P_y$  e  $P_z$  sia di tipo sinusoidale poiché la traiettoria descritta è per l'appunto una circonferenza. Si può vedere un accumulo dei punti, poiché il robot rallenta, si ferma e riparte, in corrispondenza delle coordinate  $x$ ,  $y$  e  $z$  dei punti  $P_1$ ,  $P_2$  e  $P_3$ . Ovviamente, la coordinata  $z$  rimane costante sul valore di 0.5 essendo comune a tutti e tre i punti (piano  $z$  su cui giacciono). Tanto è vero che, la derivata corrispondente per  $z = 0.5$ , essendo sempre costante senza alcuna variazione, è pari a 0. Anche in corrispondenza dei punti di accumulo, essendo che il robot in corrispondenza di essi rallenta, si può notare come la derivata e, quindi, la sua velocità diminuisca fino a diventare nulla e poi aumentare nuovamente per intraprendere il percorso successivo. Analogamente per le variabili di giunto ottenute e per la velocità delle variabili di giunto ottenute si può notare come un rallentamento dell'organo terminale comporta una diminuzione delle velocità delle variabili di giunto in questione.



Si calcolano anche in questo caso le posizioni assunte dall'end effector e la differenza, cioè l'errore, che c'è tra di esso e la traiettoria di riferimento. Si può notare come la traiettoria descritta dall'end effector ("+" di colore arancione) sia praticamente sovrapposta a quella di riferimento (">" di colore nero). Inoltre, analizzando l'errore è possibile notare come esso sia praticamente nullo (dell'ordine di  $10^{-32}$ ).

Pertanto, sfruttando nuovamente il Robotics Toolbox di Peter Corke, si effettua un plot dinamico del

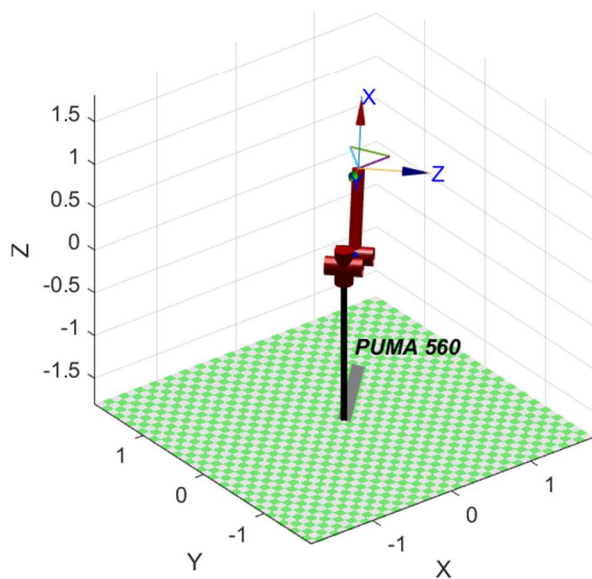


manipolatore. Si può notare come esso riesce a passare perfettamente nei punti richiesti descrivendo la traiettoria della circonferenza prima citata.

## Triangle and Circumference with Time

Affinché il manipolatore passi per i tre punti richiesti, descrivendo la traiettoria del triangolo precedentemente citato in 40 secondi, è stato necessario utilizzare la function *jtraj* che calcola una traiettoria nello spazio congiunta tra due configurazioni. Per la precisione nella function sono stati specificati tre parametri:  $Q_i$ , cioè la configurazione iniziale,  $Q_f$ , cioè la configurazione finale ed  $m$ , cioè un numero di step in cui compiere tale percorso. Bisogna precisare che tale calcolo è stato effettuato per ogni percorso (segmento/arco) tra un punto ed il prossimo considerando come configurazione iniziale quella corrente nel punto di inizio e quella finale come quella di arrivo nell'altro punto. In questo caso le configurazioni coincidono con le variabili di giunto in questione.

TRIANGLE COVERAGE PATH IN 40 SECONDS



CIRCUMFERENCE COVERAGE PATH IN 40 SECONDS

