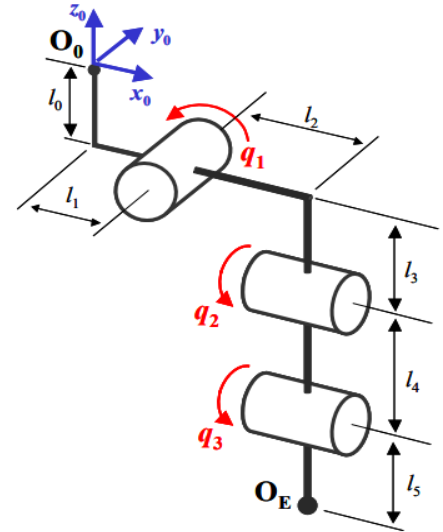


ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ

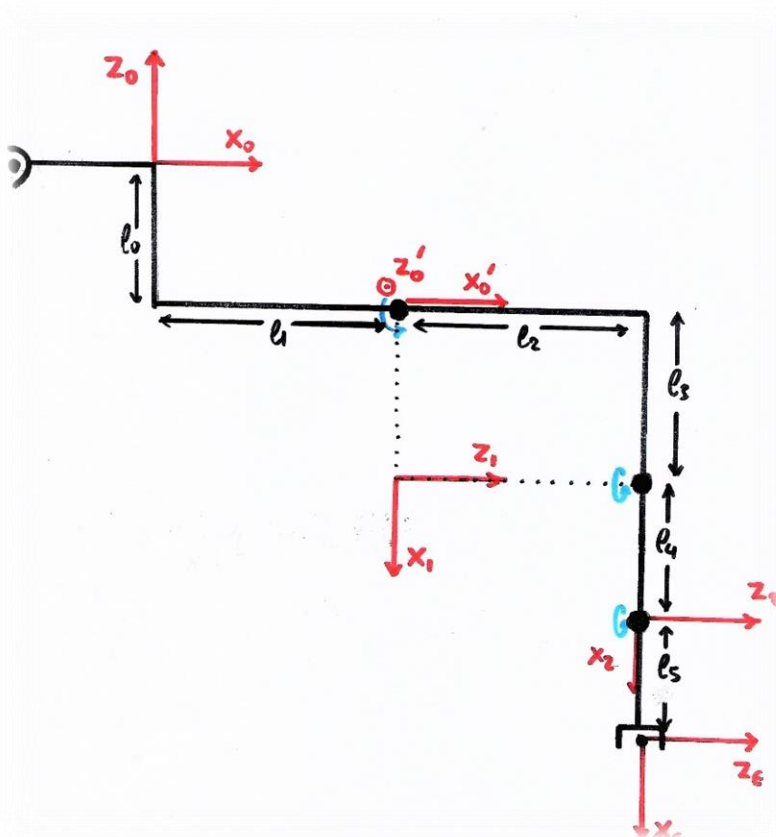
3-DOF Robotic Manipulator

Στην συγκεκριμένη εργασία καλούμαστε να αναλύσουμε το κινηματικό μοντέλο του διπλανού ρομποτικού χειριστή 3 στροφικών βαθμών ελευθερίας καθώς και να εφαρμόσουμε αυτή τη θεωρητική ανάλυση στον σχεδιασμό κατάλληλης τροχιάς για την αποδοτική εκτέλεση μίας επιθυμητής κίνησης του τελικού εργαλείου του βραχίονα. Επιλέγεται συγκεκριμένα μία περιοδική ευθύγραμμη μετατόπιση μεταξύ 2 δεδομένων σημείων του χώρου. Τόσο κατά τη θεωρητική ανάλυση, όσο κυρίως και κατά την προσομοίωση του ρομπότ, αξιοποιήθηκε το εργαλείο MATLAB, ούτως ώστε να απλοποιηθούν ιδιαίτερα πολύπλοκοι υπολογισμοί και να σχεδιαστούν τα ζητούμενα γραφήματα. Ο κώδικας που χρησιμοποιήθηκε επεξηγείται σε όλη την έκταση της εργασίας και παρατίθεται σε ξεχωριστά αρχεία.



Μέρος Α: Θεωρητική Ανάλυση

1. Ανάλυση Denavit-Hartenberg



Ακολουθώντας τη μέθοδο Denavit-Hartenberg (DH) προσδιορίζουμε τα πλαίσια αναφοράς των συνδέσμων ως εξής: Ξεκινάμε από το αρχικό πλαίσιο παγκόσμιων συντεταγμένων O_0 που δίνεται από την εκκώνηση. Οι z άξονες των αρθρώσεων θα ταυτίζονται με τους άξονες περιστροφής τους. Ο z_0 δεν είναι παράλληλος στον άξονα περιστροφής της q_1 οπότε θα χρειαστεί να μεταβούμε σε αυτή με επιπλέον πλαίσιο. Συνεπώς τοποθετούμε πλαίσιο O'_0 στην τομή μεταξύ κοινής καθέτου του άξονα της q_1 και z_0 με τον άξονά της, δηλαδή πάνω στην πρώτη άρθρωση. Στη συνέχεια, για να μεταβούμε στην q_2 θα πρέπει να τοποθετήσουμε πλαίσιο O_1 στην τομή μεταξύ κοινής καθέτου του άξονά της και z'_0 με τον άξονά της, δηλαδή σε σημείο έξω από το ρομπότ. Για να μεταβούμε στην q_3 θα πρέπει να τοποθετήσουμε πλαίσιο O_2 στην τομή μεταξύ κοινής καθέτου του άξονά της και z_1 με τον άξονά της, δηλαδή πάνω στην άρθρωση. Τέλος, για να μεταβούμε στο τελικό εργαλείο θα πρέπει να τοποθετήσουμε πλαίσιο O_E πάνω στο τελικό εργαλείο. Επιλέγουμε να μην αλλάξουμε τη διεύθυνση των αξόνων.

Οι x άξονες των πλαισίων τοποθετούνται παράλληλα στην εκάστοτε κάθετο, ενώ οι y άξονες τοποθετούνται στο τέλος, ούτως ώστε το εκάστοτε τριανδονικό σύστημα να είναι δεξιόστροφο. Παραθέτουμε δίπλα γραφικά τα πλαίσια αναφοράς που προσδιορίστηκαν από την παραπάνω διαδικασία. Τα βήματα μετάβασης από το αρχικό στο τελικό πλαίσιο περιλαμβάνουν κινήσεις $Tra_{x,z}$ & $Rot_{x,z}$ που παρατίθενται κατά σειρά στον παρακάτω πίνακα παραμέτρων:

DH – Table	$\theta - Rot_z$	$d - Tra_z$	$a - Tra_x$	$\alpha - Rot_x$
$0 \rightarrow 0'$	0	$-l_0$	l_1	$\pi/2$
$0' \rightarrow 1$	$q_1 - \pi/2$	0	l_3	$-\pi/2$
$1 \rightarrow 2$	q_2	l_2	l_4	0
$2 \rightarrow 3$	q_3	0	l_5	0

Σημείωση: Για τη συνέχεια υποθέτουμε $l_3 = 0$, δηλαδή το πλαίσιο O_1 θα έρθει στο ύψος του ρομποτικού βραχίονα.

2. Ευθεία Κινηματική Εξίσωση

Μετασχηματίζουμε κάθε γραμμή του DH-Table στην αντίστοιχη ομογενή μήτρα μετασχηματισμού:

$$A_{0'}^0 = \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & -l_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_1^{0'} = \begin{bmatrix} s_1 & 0 & c_1 & 0 \\ -c_1 & 0 & s_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_2^1 = \begin{bmatrix} c_2 & -s_2 & 0 & l_4 c_2 \\ s_2 & c_2 & 0 & l_4 s_2 \\ 0 & 0 & 1 & l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_E^2 = \begin{bmatrix} c_3 & -s_3 & 0 & l_5 c_3 \\ s_3 & c_3 & 0 & l_5 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Υπολογίζουμε τώρα σταδιακά την κινηματική εξίσωση του βραχίονα:

$$\begin{aligned} A_1^0 &= A_{0'}^0 * A_1^{0'} = \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & -l_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} s_1 & 0 & c_1 & 0 \\ -c_1 & 0 & s_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_1 & 0 & c_1 & l_1 \\ 0 & 1 & 0 & 0 \\ -c_1 & 0 & s_1 & -l_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ A_2^0 &= A_1^0 * A_2^1 = \begin{bmatrix} s_1 & 0 & c_1 & l_1 \\ 0 & 1 & 0 & 0 \\ -c_1 & 0 & s_1 & -l_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} c_2 & -s_2 & 0 & l_4 c_2 \\ s_2 & c_2 & 0 & l_4 s_2 \\ 0 & 0 & 1 & l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_2 s_1 & -s_1 s_2 & c_1 & l_1 + l_2 c_1 + l_4 c_2 s_1 \\ s_2 & c_2 & 0 & l_4 s_2 \\ -c_1 c_2 & c_1 s_2 & s_1 & -l_0 + l_2 s_1 - l_4 c_1 c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ T &= A_2^1 * A_E^2 = \begin{bmatrix} c_2 s_1 & -s_1 s_2 & c_1 & l_1 + l_2 c_1 + l_4 c_2 s_1 \\ s_2 & c_2 & 0 & l_4 s_2 \\ -c_1 c_2 & c_1 s_2 & s_1 & -l_0 + l_2 s_1 - l_4 c_1 c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} c_3 & -s_3 & 0 & l_5 c_3 \\ s_3 & c_3 & 0 & l_5 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow \\ T &= \begin{bmatrix} c_{23} s_1 & -s_1 s_{23} & c_1 & l_1 + l_2 c_1 + l_4 c_2 s_1 + l_5 s_1 c_{23} \\ s_{23} & c_{23} & 0 & l_4 s_2 + l_5 s_{23} \\ -c_1 c_{23} & c_1 s_{23} & s_1 & -l_0 + l_2 s_1 - l_4 c_1 c_2 - l_5 c_1 c_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

3. Ευθύ Διαφορικό Κινηματικό Μοντέλο

Η Ιακωβιανή μήτρα του μηχανισμού θα έχει διαστάσεις 6×3 , μιας και είναι 3DOF, και θα είναι της μορφής:

$$J = \begin{bmatrix} z_{0'} \times (p_e - p_{0'}) & z_1 \times (p_e - p_1) & z_2 \times (p_e - p_2) \\ z_{0'} & z_1 & z_2 \end{bmatrix}$$

όπου:

$z_{0'} = [0 \ 0 \ 1]^T$ ο άξονας στροφής της πρώτης άρθρωσης, τον λαμβάνουμε από την τρίτη στήλη του $A_{0'}^0$

$z_1 = [c_1 \ 0 \ s_1]^T$ ο άξονας στροφής της δεύτερης άρθρωσης, τον λαμβάνουμε από την τρίτη στήλη του A_1^0

$z_2 = [c_1 \ 0 \ s_1]^T$ ο άξονας στροφής της τρίτης άρθρωσης, τον λαμβάνουμε από την τρίτη στήλη του A_2^0

$$p_{0'} = [l_1 \ 0 \ -l_0]^T \quad p_1 = p_{0'}$$

$$p_2 = \begin{bmatrix} l_1 + l_2 c_1 + l_4 c_2 s_1 \\ l_4 s_2 \\ -l_0 + l_2 s_1 - l_4 c_1 c_2 \end{bmatrix} \quad p_e = \begin{bmatrix} l_1 + l_2 c_1 + l_4 c_2 s_1 + l_5 s_1 c_{23} \\ l_4 s_2 + l_5 s_{23} \\ -l_0 + l_2 s_1 - l_4 c_1 c_2 - l_5 c_1 c_{23} \end{bmatrix}$$

Συνεπώς η Ιακωβιανή μήτρα είναι η εξής:

$$J = \begin{bmatrix} -l_2 s_1 + l_4 c_1 c_2 + l_5 c_1 c_{23} & -s_1(l_4 s_2 + l_5 s_{23}) & -l_5 s_1 s_{23} \\ 0 & l_4 c_2 + l_5 c_{23} & l_5 c_{23} \\ l_2 c_1 + l_4 s_1 c_2 + l_5 s_1 c_{23} & c_1(l_4 s_2 + l_5 s_{23}) & l_5 c_1 s_{23} \\ 0 & c_1 & c_1 \\ -1 & 0 & 0 \\ 0 & s_1 & s_1 \end{bmatrix}$$

4. Αντίστροφο Διαφορικό Κινηματικό Μοντέλο

Οι ιδιομορφίες του εξεταζόμενου συστήματος προκύπτουν στα σημεία μηδενισμού της άνω 3X3 ορίζουσας της Ιακωβιανής μήτρας, αφού μας ενδιαφέρει η γραμμική ταχύτητα του τελικού εργαλείου. Προκύπτει το εξής:

$$\begin{vmatrix} -l_2 s_1 + l_4 c_1 c_2 + l_5 c_1 c_{23} & -s_1(l_4 s_2 + l_5 s_{23}) & -l_5 s_1 s_{23} \\ 0 & l_4 c_2 + l_5 c_{23} & l_5 c_{23} \\ l_2 c_1 + l_4 s_1 c_2 + l_5 s_1 c_{23} & c_1(l_4 s_2 + l_5 s_{23}) & l_5 c_1 s_{23} \end{vmatrix} = 0 \rightarrow l_4 l_5 s_3 (l_4 c_2 + l_5 c_{23}) = 0 \rightarrow \begin{cases} s_3 = 0 \\ l_4 c_2 = -l_5 c_{23} \end{cases}$$

Η πρώτη περίπτωση συμβαίνει όταν $q_3 = 0$ ή $q_3 = \pi$, δηλαδή όταν ο τελευταίος σύνδεσμος εκτείνεται ή διπλώνει στον προηγούμενο αντίστοιχα. Πρόκειται για ιδιομορφία αγκώνα. Μέσω της κινηματικής εξίσωσης παρατηρούμε πως κατά τη δεύτερη περίπτωση, όπου $l_4 c_2 = -l_5 c_{23}$, ο μηχανισμός βρίσκεται σε θέση, την οποία οι q_1, q_2 δεν την επηρεάζουν. Έχουμε λοιπόν απώλεια βαθμών ελευθερίας στην κίνηση.

Για τιμές της ορίζουσας διάφορες από τις παραπάνω μπορούμε να θεωρήσουμε τον αντίστροφο 3X3 πίνακα $\text{inv}(J(1:3,:))$ και με τη βοήθεια αυτού και του διανύσματος $[u_x \ u_y \ u_z]$ των γραμμικών ταχυτήτων του τελικού εργαλείου, να υπολογίσουμε τις σχέσεις από τις οποίες προκύπτουν οι γωνιακές ταχύτητες των αρθρώσεων:

$$[w_1 \ w_2 \ w_3]^T = J' * [u_x \ u_y \ u_z]^T$$

Υπολογίζουμε τον αντίστροφο πίνακα μέσω MATLAB και προκύπτει:

$$\begin{aligned} J'_{11} &= \frac{c_1}{l_4 c_2 + l_5 c_{23}} & J'_{12} &= 0 & J'_{13} &= \frac{s_1}{l_4 c_2 + l_5 c_{23}} & J'_{22} &= \frac{s_{23}}{l_4 s_3} \\ J'_{21} &= -\frac{l_5 c_2^2 s_1 - l_5 s_1 + l_5 c_3^2 s_1 - 2l_5 c_2^2 c_3^2 s_1 - l_2 c_1 c_2 c_3 + l_2 c_1 s_2 s_3 - l_4 c_2^2 c_3 s_1 + l_4 c_2 s_1 s_2 s_3 + 2l_5 c_2 c_3 s_1 s_2 s_3}{l_4^2 c_2 s_3 - l_4 l_5 s_2 + l_4 l_5 c_3^2 s_2 + l_4 l_5 c_2 c_3 s_3} \\ J'_{23} &= \frac{l_5 c_2^2 c_1 - l_5 c_1 + l_5 c_3^2 c_1 - 2l_5 c_2^2 c_3^2 c_1 + l_2 s_1 c_2 c_3 - l_2 s_1 s_2 s_3 - l_4 c_2^2 c_3 c_1 + l_4 c_1 c_2 s_2 s_3 + 2l_5 c_1 c_2 c_3 s_2 s_3}{l_4^2 c_2 s_3 - l_4 l_5 s_2 + l_4 l_5 c_3^2 s_2 + l_4 l_5 c_2 c_3 s_3} \\ J'_{31} &= -\frac{l_2 c_1 + l_4 c_2 s_1 + l_5 c_2 c_3 s_1 - l_5 s_1 s_2 s_3}{l_4 l_5 s_3} & J'_{32} &= -\frac{l_4 s_2 + l_5 s_{23}}{l_4 l_5 s_3} & J'_{33} &= -\frac{l_2 s_1 - l_4 c_1 c_2 - l_5 c_1 c_2 c_3 + l_5 c_1 s_2 s_3}{l_4 l_5 s_3} \end{aligned}$$

Ο αντίστοιχος κώδικας για τον υπολογισμό του αντίστροφου διαφορικού μοντέλου:

```
Jdet = simplify(det(J(1:3,:)));
% Singularities
vars = [q1 q2 q3];
[s1,s2,s3,parameters,conditions] = solve(Jdet==0,vars,"ReturnConditions",true);
Jdet = 14*15*sin(q3)*(14*cos(q2)+15*cos(q2+q3)); % final simplified Jdet
% Inverse
Jinv = simplify(inv(J(1:3,:))); % could also be adjoint(J(1:3,:))/Jdet
w = Jinv*[u1 u2 u3]'; % angular velocities from known end-point velocities
```

5. Αντίστροφο Γεωμετρικό Μοντέλο

Από την ευθεία κινηματική εξίσωση T λαμβάνουμε την 4^η στήλη $p_e = [p_x \ p_y \ p_z]$ που αναπαριστά τη θέση του τελικού εργαλείου δράσης και το επιλύουμε ως προς τις μεταβλητές των αρθρώσεων. Με αυτό τον τρόπο, η επιθυμητή θέση του τελικού εργαλείου θα μεταφράζεται στις στροφές των αρθρώσεων που απαιτούνται:

$$\begin{cases} p_x = l_1 + l_2 c_1 + l_4 c_2 s_1 + l_5 s_1 c_{23} & (1) \\ p_y = l_4 s_2 + l_5 s_{23} & (2) \\ p_z = -l_0 + l_2 s_1 - l_4 c_1 c_2 - l_5 c_1 c_{23} & (3) \end{cases} \Rightarrow$$

Από την (1) λαμβάνουμε: $p_x = l_1 + l_2 c_1 + s_1 p_{2x}$ ενώ από την (3): $p_z = -l_0 + l_2 s_1 - c_1 p_{2x}$ όπου $p_{2x} = l_4 c_2 + l_5 c_{23}$. Μετασχηματίζουμε κατάλληλα τις 2 αυτές σχέσεις και τις προσθέτουμε κατά μέλη:

$$\begin{cases} (p_x - l_1)^2 = (l_2 c_1 + s_1 p_{2x})^2 \\ (p_z + l_0)^2 = (l_2 s_1 - c_1 p_{2x})^2 \end{cases} \Rightarrow (p_x - l_1)^2 + (p_z + l_0)^2 = l_2^2 + p_{2x}^2 \Rightarrow p_{2x} = \pm \sqrt{(p_x - l_1)^2 + (p_z + l_0)^2}$$

Αναλύοντας και αντικαθιστώντας την τιμή για το p_{2x} λαμβάνουμε:

$$\begin{cases} l_4 c_2 + l_5 c_{23} = p_{2x} \\ l_4 s_2 + l_5 s_{23} = p_y \end{cases} \Rightarrow \begin{cases} (l_4 c_2 + l_5 c_{23})^2 = (p_x - l_1)^2 + (p_z + l_0)^2 - l_2^2 \\ p_y^2 = l_4^2 s_2^2 + l_5^2 s_{23}^2 + 2l_4 l_5 s_2 s_{23} \end{cases} \quad \text{Προσθέτοντας κατά μέλη:}$$

$$(p_x - l_1)^2 + (p_z + l_0)^2 - l_2^2 + p_y^2 = l_4^2 s_2^2 + l_5^2 s_{23}^2 + 2l_4 l_5 s_2 s_{23} + l_4^2 c_2^2 + l_5^2 c_{23}^2 + 2l_4 l_5 c_2 c_{23} = l_4^2 + l_5^2 + 2l_4 l_5 c_3$$

$$\Rightarrow c_3 = \frac{(p_x - l_1)^2 + (p_z + l_0)^2 - l_2^2 - l_4^2 - l_5^2 + p_y^2}{2l_4 l_5} \Rightarrow q_3 = \pm \arccos\left(\frac{(p_x - l_1)^2 + (p_z + l_0)^2 - l_2^2 - l_4^2 - l_5^2 + p_y^2}{2l_4 l_5}\right)$$

Το q_3 μαζί με το q_2 επιλύουν το υποπρόβλημα για τις 2 αντίστοιχα στροφικές αρθρώσεις, συνεπώς το q_2 θα έχει την εξής μορφή: $q_2 = \arctan2\left(p_y, \sqrt{(p_x - l_1)^2 + (p_z + l_0)^2 - l_2^2}\right) - \arctan2(l_5 s_3, l_4 + l_5 c_3)$

Υπολογίζουμε τέλος και την πρώτη στροφική άρθρωση. Από τις αρχικές σχέσεις λαμβάνουμε πάλι:

$$\begin{cases} p_x - l_1 = l_2 c_1 + s_1 p_{2x} \\ p_z + l_0 = l_2 s_1 - c_1 p_{2x} \end{cases} \Rightarrow \begin{cases} p_x - l_1 = l_2 c_1 + s_1 p_{2x} \\ s_1 = \frac{p_z + l_0 + c_1 p_{2x}}{l_2} \end{cases} \Rightarrow p_x - l_1 = l_2 c_1 + \frac{p_{2x}(p_z + l_0 + c_1 p_{2x})}{l_2}$$

$$\Rightarrow c_1(p_{2x}^2 + l_2^2) = l_2 p_x - l_1 l_2 - l_0 p_{2x} - p_{2x} p_z \Rightarrow q_1 = \pm \arccos\left(\frac{l_2 p_x - l_1 l_2 - l_0 p_{2x} - p_{2x} p_z}{p_{2x}^2 + l_2^2}\right)$$

Μέρος Β: Κινηματική Προσομίωση

Το τελικό εργαλείο του ρομποτικού βραχίονα καλείται να εκτελέσει περιοδική ευθύγραμμη μετατόπιση μεταξύ 2 σημείων $P_A(x_A, y_A) - P_B(x_B, y_B)$ σε οριζόντιο επίπεδο απόστασης $z = h$ από τη βάση αναφοράς. Το ρομπότ ξεκινάει να κινείται τη χρονική στιγμή $t_0 = 0$ και εκτελεί την προβλεπόμενη κίνηση με περίοδο $T = 20 \text{ sec}$. Για να εξασφαλίσουμε ομαλή κίνηση ως προς την ταχύτητα και ως προς την επιτάχυνση σχεδιάζουμε την επιθυμητή τροχιά μέσω πολυωνύμων παρεμβολής και συγκεκριμένα αξιοποιώντας τη μέθοδο τριών φάσεων για ευθύγραμμη κίνηση. Σύμφωνα με τη μέθοδο, η τροχιά χωρίζεται σε 3 μέρη, το μεσαίο εκ των οποίων συνιστά και την ουσία της κίνησης. Τα 2 ακραία μέρη αναπαρίστανται από πολυώνυμα μεγάλου βαθμού και χρησιμεύουν στο να εξασφαλίσουν την ομαλότητα της κίνησης στις οριακές θέσεις της τροχιάς. Στην περίπτωση μας, όπου απαιτείται συνέχεια ταχυτήτων και επιταχύνσεων, τα πολυώνυμα αυτά θα είναι πέμπτου βαθμού και θα εφαρμόζονται για ένα ορισμένο διάστημα γύρω από τις ακραίες θέσεις, ενώ το κύριο μέρος της κίνησης θα αναπαρίσταται από πολυώνυμο-ευθεία δεύτερου βαθμού. Συνοπτικά η σχεδιαζόμενη τροχιά:

$$x(t) = \begin{cases} x_1(t) = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0, & t < \Delta \\ x_2(t) = b_1 t + b_0, & t < T - \Delta \\ x_3(t) = c_5 t^5 + c_4 t^4 + c_3 t^3 + c_2 t^2 + c_1 t + c_0, & t < T \end{cases}$$

Ο πλήρης προσδιορισμός της τροχιάς απαιτεί την εύρεση των 14 συντελεστών που εμφανίζονται στην παραπάνω εξίσωση. Θα χρειαστούμε 14 αρχικές συνθήκες, οι οποίες είναι:

- Συνέχεια Θέσης: $x_1(\Delta) = x_2(\Delta)$, $x_2(T - \Delta) = x_3(T - \Delta)$
- Συνέχεια Ταχύτητας: $v_1(\Delta) = v_2(\Delta)$, $v_2(T - \Delta) = v_3(T - \Delta)$
- Συνέχεια Επιτάχυνσης: $\gamma_1(\Delta) = \gamma_2(\Delta)$, $\gamma_2(T - \Delta) = \gamma_3(T - \Delta)$
- Αρχική & Τελική Κατάσταση: $x_1(t_0) = x_A$, $x_3(T) = x_B$, $v_1(t_0) = v_A$, $v_3(T) = v_B$, $\gamma_1(t_0) = \gamma_A$, $\gamma_3(T) = \gamma_B$
- Συνθήκες Συσχέτισης Ταχύτητας: $x_{1/3}(\Delta/T) - x_{1/3}(0/T - \Delta) = \Delta * 0.5 * (v_{1/3}(0/T - \Delta) + v_{1/3}(\Delta/T))$

Λύνοντας το σύστημα 14X14 γραμμικών εξισώσεων για την ανάθεση τιμών που παραθέσαμε προηγουμένως, λαμβάνουμε τους συντελεστές των πολυωνύμων παρεμβολής και έχουμε πλέον πλήρη προσδιορισμό της τροχιάς κίνησης. Ανάλογη διαδικασία ακολουθούμε και για την εύρεση της τροχιάς $y(t)$, καθώς και για την εύρεση των τροχιών για την επιστροφή από την τελική στην αρχική θέση σε χρόνο $T \rightarrow 2T$. Ο κώδικας της αντίστοιχης συνάρτησης υπολογισμού των συντελεστών:

```
syms a0 a1 a2 a3 a4 a5 b0 b1 c0 c1 c2 c3 c4 c5
t1 = t0+(tf-t0)*0.1; t2 = t0+(tf-t0)*0.9;

eqn1 = a0 + a1*t0 + a2*t0^2 + a3*t0^3 + a4*t0^4 + a5*t0^5 - xA == 0;
eqn2 = a1 + 2*a2*t0 + 3*a3*t0^2 + 4*a4*t0^3 + 5*a5*t0^4 == 0;
eqn3 = 2*a2 + 6*a3*t0 + 12*a4*t0^2 + 20*a5*t0^3 - g == 0;
eqn4 = a0 + a1*t1 + a2*t1^2 + a3*t1^3 + a4*t1^4 + a5*t1^5 - (b0 + b1*t1) == 0;
eqn5 = a1 + 2*a2*t1 + 3*a3*t1^2 + 4*a4*t1^3 + 5*a5*t1^4 - b1 == 0;
eqn6 = 2*a2 + 6*a3*t1 + 12*a4*t1^2 + 20*a5*t1^3 == 0;
eqn7 = c0 + c1*t2 + c2*t2^2 + c3*t2^3 + c4*t2^4 + c5*t2^5 - (b0 + b1*t2) == 0;
eqn8 = c1 + 2*c2*t2 + 3*c3*t2^2 + 4*c4*t2^3 + 5*c5*t2^4 - b1 == 0;
eqn9 = 2*c2 + 6*c3*t2 + 12*c4*t2^2 + 20*c5*t2^3 == 0;
eqn10 = c0 + c1*tf + c2*tf^2 + c3*tf^3 + c4*tf^4 + c5*tf^5 - xB == 0;
eqn11 = c1 + 2*c2*tf + 3*c3*tf^2 + 4*c4*tf^3 + 5*c5*tf^4 == 0;
eqn12 = 2*c2 + 6*c3*tf + 12*c4*tf^2 + 20*c5*tf^3 + g == 0;
eqn13 = b0+b1*t1 - (a0 + a1*t0 + a2*t0^2 + a3*t0^3 + a4*t0^4 + a5*t0^5) - (0.5*(t1-t0)*b1) == 0;
eqn14 = (c0 + c1*tf + c2*tf^2 + c3*tf^3 + c4*tf^4 + c5*tf^5) - (b0+b1*t2) - (0.5*(t1-t0)*b1) == 0;

[A,B] = equationsToMatrix([ eqn1 eqn2 eqn3 eqn4 eqn5 eqn6 eqn7 eqn8 eqn9 eqn10 eqn11 eqn12 eqn13 eqn14 ], [ a0 a1 a2 a3 a4 a5 b0 b1 c0 c1 c2 c3 c4 c5 ]);
X = vpa(linsolve(A,B));
```

Ακολουθώς ορίζουμε ένα διάνυσμα χρόνου και για κάθε τιμή $t: 0 \rightarrow 2T$ υπολογίζουμε την τιμή της τροχιάς στους άξονες x, y (συνάρτηση polyval) ενώ δημιουργούμε και ένα σταθερό διάνυσμα z (τιμή ίση με το ύψος του επιπέδου κίνησης). Ταυτόχρονα υπολογίζουμε και τις τιμές των αντίστοιχων ταχυτήτων (συνάρτηση polyder):

```
xd = double(trajectory(0,duration,xA,xB,gx));
yd = double(trajectory(0,duration,yA,yB,gy));
rx = double(trajectory(duration,2*duration,xB,xA,-gx));
ry = double(trajectory(duration,2*duration,yB,yA,-gy));

kmax = duration/sample + 1;
x = zeros(length(t),1); y=x; ux=x; uy=x; uz=x; z=x;
for k = 1:2*kmax-1
    time = (k-1)*sample;
    if (time<duration*0.1)
        x(k) = polyval(xd(6:-1:1),time);
        ux(k) = polyval(polyder(xd(6:-1:1)),time);
        y(k) = polyval(yd(6:-1:1),time);
        uy(k) = polyval(polyder(yd(6:-1:1)),time);
    elseif (time<duration*0.9)
        x(k) = polyval(xd(8:-1:7),time);
        ux(k) = polyval(polyder(xd(8:-1:7)),time);
        y(k) = polyval(yd(8:-1:7),time);
        uy(k) = polyval(polyder(yd(8:-1:7)),time);
    elseif (time<duration)
        x(k) = polyval(xd(8:-1:7),time);
        ux(k) = polyval(polyder(xd(8:-1:7)),time);
        y(k) = polyval(yd(8:-1:7),time);
        uy(k) = polyval(polyder(yd(8:-1:7)),time);
    end
end
```

```

x(k) = polyval(xd(14:-1:9),time);
ux(k) = polyval(polyder(xd(14:-1:9)),time);
y(k) = polyval(yd(14:-1:9),time);
uy(k) = polyval(polyder(yd(14:-1:9)),time);
elseif (time<duration*1.1)
x(k) = polyval(rxd(6:-1:1),time);
ux(k) = polyval(polyder(rxd(6:-1:1)),time);
y(k) = polyval(ryd(6:-1:1),time);
uy(k) = polyval(polyder(ryd(6:-1:1)),time);
elseif (time<duration*1.9)
x(k) = polyval(rxd(8:-1:7),time);
ux(k) = polyval(polyder(rxd(8:-1:7)),time);
y(k) = polyval(ryd(8:-1:7),time);
uy(k) = polyval(polyder(ryd(8:-1:7)),time);
else
x(k) = polyval(rxd(14:-1:9),time);
ux(k) = polyval(polyder(rxd(14:-1:9)),time);
y(k) = polyval(ryd(14:-1:9),time);
uy(k) = polyval(polyder(ryd(14:-1:9)),time);
end
z(k) = -13-14;
end

```

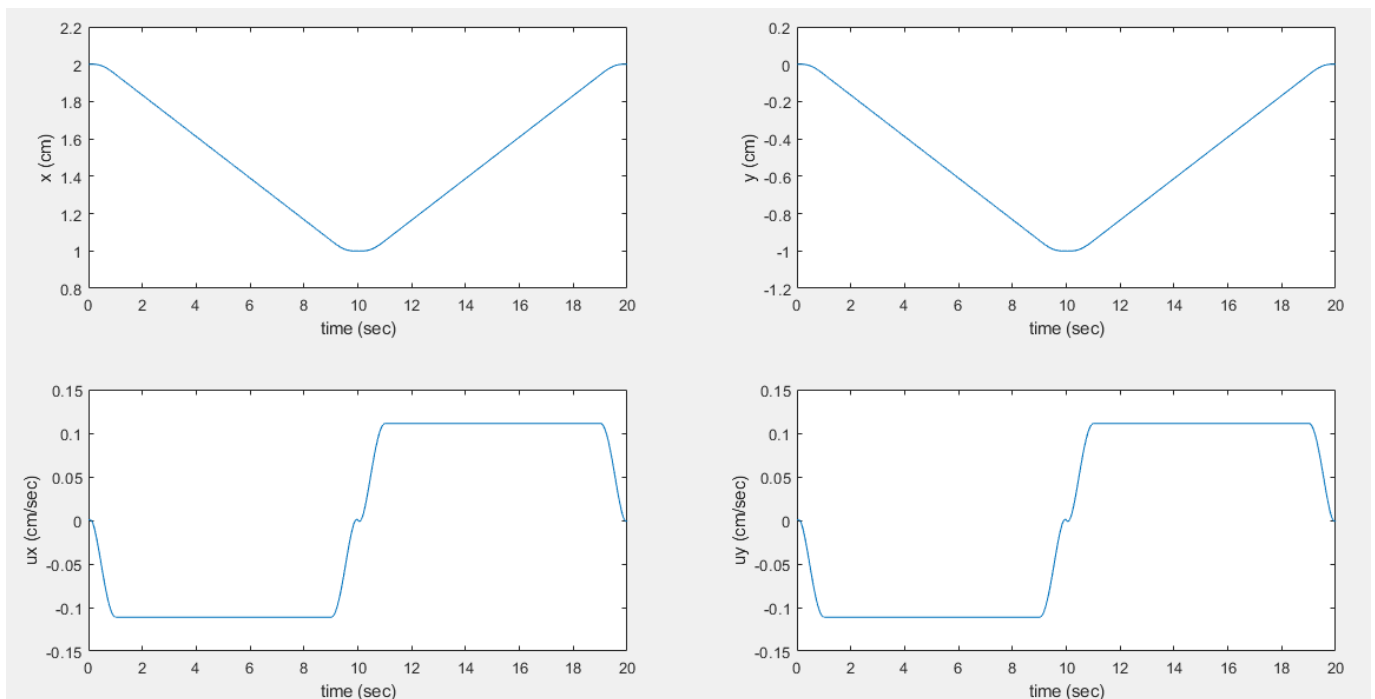
Έχοντας τώρα πλήρως καθορισμένη την επιθυμητή κίνηση, υπολογίζουμε τις απαιτούμενες γωνίες q των αρθρώσεων και τις γωνιακές τους ταχύτητες για κάθε t μέσω του αντίστροφου γεωμετρικού και διαφορικού κινηματικού μοντέλου αντίστοιχα, όπως αυτά αναλύθηκαν παραπάνω:

```

px2 = sqrt(x.^2+z.^2-12^2);
q3 = -acos((y.^2+(x-11).^2+(z+10).^2-12^2-14^2-15^2)./(2*14*15));
q2 = atan2(y,px2)-atan2(15*sin(q3),(14+15*cos(q3)));
q1 = -acos((x*12-px2.*z)./(px2.^2+12^2));
qd1 = zeros(length(t),1); qd2=qd1; qd3=qd1;
for k = 1:length(t)
j = inverse(10,11,12,13,14,15,q1(k),q2(k),q3(k));
qd1(k) = j(1,1)*ux(k) + j(1,2)*uy(k);
qd2(k) = j(2,1)*ux(k) + j(2,2)*uy(k);
qd3(k) = j(3,1)*ux(k) + j(3,2)*uy(k);
end

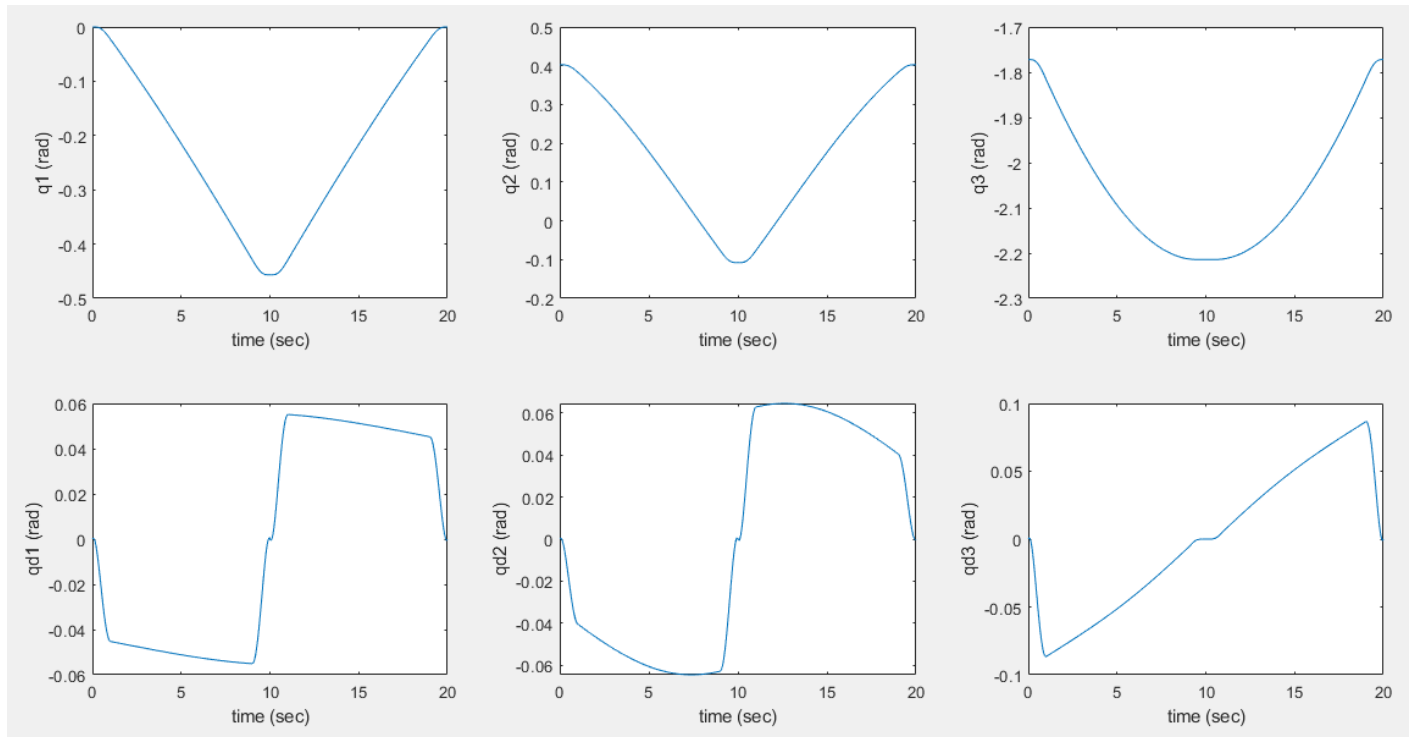
```

α. Γραφική Παράσταση Επιθυμητής Τροχιάς & Ταχύτητας



Παραθέτουμε τα διαγράμματα των παραπάνω σχέσεων για μία περίοδο. Παρατηρούμε πως η μετατόπιση σε κάθε άξονα είναι η προβλεπόμενη, ενώ στις ακραίες θέσεις, με την επίδραση των πολυωνύμων τροχιάς 5^{ου} βαθμού, επιτυγχάνουμε την ομαλότητα τόσο στη θέση, όσο και στην ταχύτητα, ιδίως στα κρίσιμα σημεία. Αυτό υποδηλώνει συνέχεια θέσης, ταχύτητας και επιτάχυνσης, όπως είναι και το ζητούμενο. Παρατηρούμε επίσης πως η κίνηση είναι πράγματι ευθύγραμμη και η ταχύτητα σταθερή στο μεγαλύτερό της μέρος (ευθεία).

b. Γραφική Παράσταση Γωνιών Στροφής & Ταχυτήτων των Αρθρώσεων



Στα συγκεκριμένα διαγράμματα για τις αρθρώσεις παρατηρούμε πως οι μετατοπίσεις και γενικά οι στροφές που απαιτεί η εκτέλεση της προβλεπόμενης κίνησης είναι ομαλές και επιτεύξιμες από πραγματικό ρομποτικό μηχανισμό.

c. Animation Κίνησης

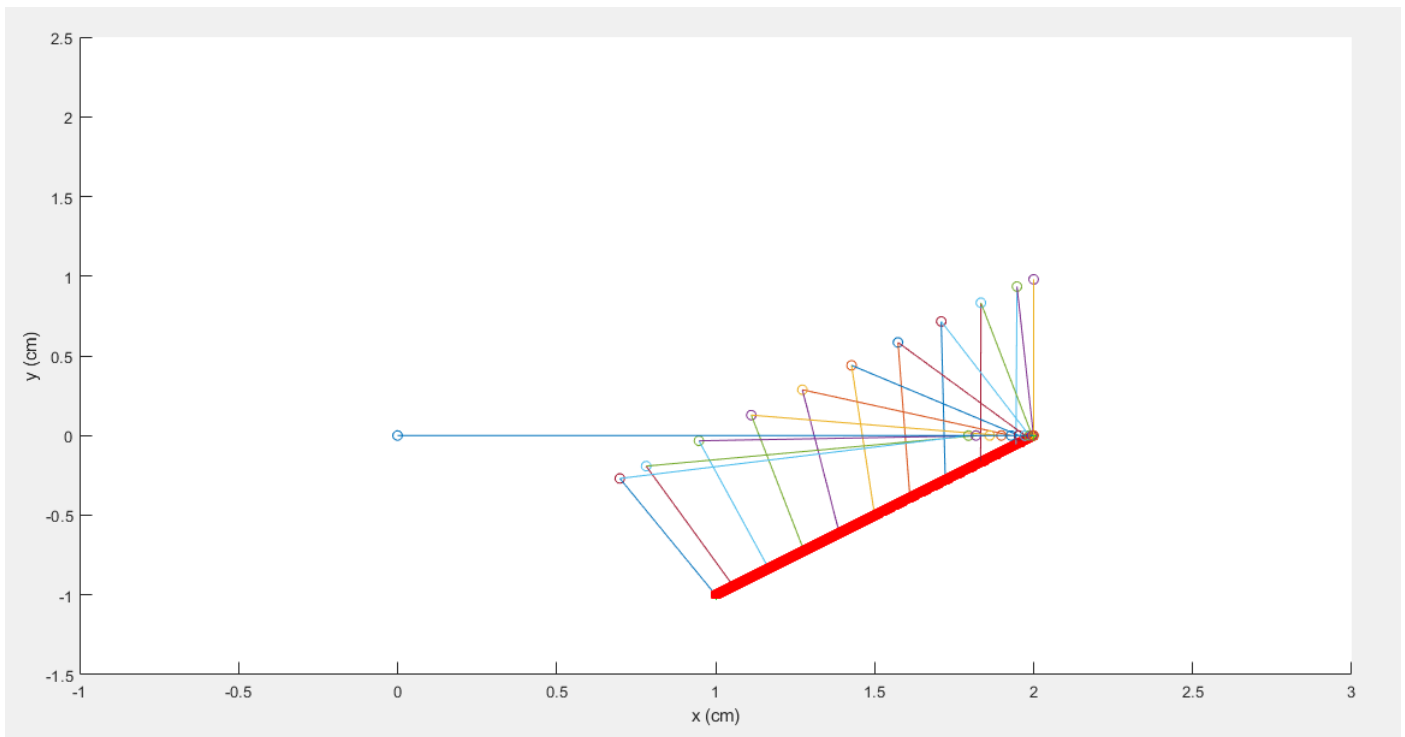
Για την απεικόνιση του συνολικού ρομποτικού μηχανισμού (σύνδεσμοι, αρθρώσεις και κινήσεις αυτών) απαιτείται επιπλέον ο υπολογισμός των θέσεων των συνδέσμων, που επιτυγχάνουμε χρησιμοποιώντας το ευθύ κινηματικό μοντέλο. Επειδή ωστόσο δεν έχουμε πλαίσιο πάνω στη δεύτερη άρθρωση, υπολογίζουμε γεωμετρικά τη θέση του άκρου του πρώτου συνδέσμου και άρα της δεύτερης άρθρωσης:

```
x1 = 12*cos(q1);
y1 = zeros(length(t),1);
z1 = 12.*sin(q1);
```

```
x2 = 11 + 12*cos(q1) + 14*cos(q2).*sin(q1);
y2 = 14*sin(q2);
z2 = 12.*sin(q1) - 10 - 14.*cos(q1).*cos(q2);
```

```
xe = 11 + 12*cos(q1) + 14*cos(q2).*sin(q1) + 15*cos(q2).*cos(q3).*sin(q1) -
15*sin(q1).*sin(q2).*sin(q3);
ye = 15*sin(q2 + q3) + 14*sin(q2);
ze = 12.*sin(q1) - 10 - 14.*cos(q1).*cos(q2) - 15.*cos(q1).*cos(q2).*cos(q3) +
15.*cos(q1).*sin(q2).*sin(q3);
```

Αφού έχουμε υπολογισμένα όλα τα απαραίτητα στοιχεία του ρομπότ, απεικονίζουμε για κάθε t τον συνδυασμό τους, που οδηγεί στην παρακάτω προσομοίωση:



Ο κώδικας παραγωγής της προσομοίωσης:

```
xlabel('x (cm)');
ylabel('y (cm)');
hold on
plot3([0],[0],[0],'o'); T=5;
for tkn=1:100:T*2*kmax
    tk = mod(tkn,2*kmax-2);
    pause(0.2);
    %hold off % uncomment this in order to see the actual robot motion
    plot3(0,0,0);
    hold on
    axis([-1 3 -1.5 2.5 -4 1])
    axis on
    plot3(x,y,z,'rs');
    plot3([0,x1(tk)],[0,y1(tk)],[0,z1(tk)]);
    plot3([x1(tk)],[y1(tk)],[z1(tk)],'o');
    plot3([x1(tk),x2(tk)],[y1(tk),y2(tk)],[z1(tk),z2(tk)]);
    plot3([x2(tk)],[y2(tk)],[z2(tk)],'o');
    plot3([x2(tk),xe(tk)],[y2(tk),ye(tk)],[z2(tk),ze(tk)]);
    plot3([xe(tk)],[ye(tk)],[ze(tk)],'y*');
    plot3([x(tk)],[y(tk)],[ze(tk)],'g+');
end
```

Παρατηρούμε πως σχηματίζεται ένα σκαρίφημα του ρομποτικού μηχανισμού που εξετάζουμε, πιστό στα υπολογισμένα μεγέθη (αριθμός αρθρώσεων, μήκος και κίνηση συνδέσμων). Μάλιστα, βλέπουμε πως όλη παραπάνω ανάλυση οδήγησε στο να προσομοιώσουμε την επιθυμητή κίνηση του τελικού εργαλείου, με αντίστοιχα ομαλές και συνεκτικές μεταξύ τους μετατοπίσεις των διάφορων μερών του μηχανισμού. Το animation επιτυγχάνεται με συνεχές plot() εμφωλευμένο σε δομή for loop. Με βάση τα βοηθητικά αρχεία της άσκησης, παρουσιάζουμε στο ίδιο πλαίσιο όλες τις ενδιαμέσες κινήσεις του ρομπότ. Απενεργοποιούμε αυτή τη δυνατότητα με την εντολή hold off.

ΤΕΛΟΣ ΑΝΑΦΟΡΑΣ

Σημείωση: Το σύνολο του κώδικα MATLAB που χρησιμοποιήθηκε στην παρούσα εργασία εμπεριέχεται στο φάκελο του παραδοτέου. Για το Μέρος Α αντιστοιχεί το αρχείο *helper.m* ενώ για το Μέρος Β το *simulation.m* και οι συναρτήσεις *trajectory.m* (για τον υπολογισμό των πολυωνύμων τροχιάς) και *inverse.m* (για τον αριθμητικό υπολογισμό της αντίστροφης Ιακωβιανής μήτρας).