Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Ρομποτική 1

Εργαστηριακή Άσκηση 4

Cobot

Ομάδα Α3-4	Νικόλαος Αδαμόπουλος	03122074
	Λεωνίδας Λαγομιτζής	03121061
	Ηλίας-Στυλιανός Ηλιάδης	03122140
	Ηλίας Πουλόπουλος	03122160
	Ανδρέας Αγγελόπουλος	03122049

Περιεχόμενα

1 Εισαγωγή

1.1 Setup της εργαστηριακής άσκησης

Το setup που χρησιμοποιήθηκε στην άσκηση αποτελείται από τα εξής υποσυστήματα.

1.1.1 Ο ρομποτικός βραχίονας – MyCobot 280 Jetson Nano

Πρόκειται για τον συνεργατικό βραχίονα myCobot 280 Jetson Nano, έναν μικροριμποτικό βραχίονα 6 βαθμών ελευθερίας με ενσωματωμένο NVIDIA Jetson Nano και μικροελεγκτή ATOM για διπλή επεξεργασία. Σύμφωνα με το συνοδευτικό PDF (σελ. 3) το ρομπότ ζυγίζει περίπου 1030 g, σηκώνει μέχρι 250 g, έχει ακτίνα χεριού 280 mm και επαναληψιμότητα της τάξης των ± 5 mm. Είναι τοποθετημένο πάνω σε σταθερή βάση στο εργαστήριο.

1.1.2 Υπολογιστής χειρισμού

Το setup περιλαμβάνει έναν desktop PC συνδεδεμένο μέσω USB με το ρομπότ και εκτελεί ειδικό πρόγραμμα σε Python για την άσκηση. Ο υπολογιστής στέλνει εντολές σειριακά προς το myCobot, διαβάζει αρχεία `.txt` με θέσεις και εντολές gripper και εκτελεί τα scripts του εργαστηρίου (move, pick–place, kinesthetic teaching).

1.1.3 Λογισμικό και βιβλιοθήκες Python

Στον κώδικα χρησιμοποιείται η βιβλιοθήκη `pymycobot` για την επικοινωνία με το ρομπότ· οι βασικές λειτουργίες υλοποιούνται μέσω συναρτήσεων όπως `send_angles`/`sync_send_angles`, `send_coords` και `set_gripper_value`.

1.1.4 Χώρος εργασίας – αντικείμενα

Στο πρώτο μέρος (pick & place) χρησιμοποιήσαμε έναν μικρό πλαστικό κύβο και καθορισμένες θέσεις λήψης και εναπόθεσης στην επιφάνεια εργασίας, με ιδιαίτερη προσοχή στους περιορισμούς του workspace ώστε να αποφευχθούν συγκρούσεις.

1.1.5 Διεπαφή Kinesthetic Teaching (Μέρος 3)

Το GUI είναι υλοποιημένο σε `tkinter` και επιτρέπει στον χρήστη να καθοδηγεί χειροκίνητα το ρομπότ, να αποθηκεύει σημεία αναφοράς, να ελέγχει τον gripper και να εκτελεί αυτόνομα την αποθηκευμένη διαδικασία.

1.2 Βασικές διαφορές μεταξύ Robot και Cobot

Τα κλασικά βιομηχανικά ρομπότ χαρακτηρίζονται από υψηλές ταχύτητες και ακρίβεια, απαιτούν ειδικούς χώρους ασφαλείας και εξειδικευμένο προγραμματισμό, και σχεδιάζονται για βαριά, επαναλαμβανόμενα καθήκοντα. Αντίθετα, τα συνεργατικά ρομπότ διαθέτουν αισθητήρες ροπής και μηχανισμούς ασφάλειας που επιτρέπουν άμεση συνεργασία με τον άνθρωπο· σταματούν σε περίπτωση αντίστασης, έχουν μικρότερη ισχύ αλλά μεγαλύτερη προσαρμοστικότητα και διευκολύνουν τον προγραμματισμό (π.χ. drag & teach / kinesthetic teaching). Ο myCobot 280 ανήκει στη δεύτερη κατηγορία και προορίζεται για καθοδήγηση με το χέρι και χρήση σε εκπαιδευτικό περιβάλλον.

1.3 Χαρακτηριστικά του MyCobot 280 Jetson Nano

Μηχανικά, ο myCobot έχει 6 βαθμούς ελευθερίας, ακτίνα εργασίας 280 mm, επανάληψη περίπου ± 5 mm, μέγιστο φορτίο 250 g και βάρος περίπου 1030 g· τα μήκη συνδέσμων και ο πλήρης πίνακας Denavit–Hartenberg περιλαμβάνονται στο διάγραμμα (PDF, σελ. 4–5). Η αρχιτεκτονική περιλαμβάνει onboard Jetson Nano για επεξεργασία εικόνας/AI και μικροελεγκτή ATOM για real-time low-level control. Λογισμικά υποστηριζόμενα περιλαμβάνουν Python API (`pymycobot`), ROS και συμβατότητα με Linux/Windows, ενώ υπάρχει δυνατότητα χρήσης διάφορων end-effectors όπως gripper, suction cup ή camera module.

1.4 Πεδία εφαρμογών

Ο myCobot 280 προορίζεται κυρίως για εκπαιδευτικά εργαστήρια και πανεπιστημιακά μαθήματα ρομποτικής, όπου χρησιμοποιείται για διδασκαλία κινηματικής και πειράματα με vision/AI. Επιπλέον, μπορεί να εξυπηρετήσει ελαφριές βιομηχανικές εργασίες χαμηλού βάρους, απλές αυτοματοποιήσεις και ερευνητικές εφαρμογές σε HRI, έλεγχο αρθρώσεων και collaborative robotics.

1.5 Συνοπτική επιστημονική περιγραφή

Η άσκηση στηρίζεται στη χρήση του συνεργατικού myCobot 280, συνδεδεμένου με υπολογιστή που εκτελεί Python κώδικα. Ο χρήστης δίνει εντολές είτε μέσω αρχείων `.txt` είτε μέσω GUI, και το ρομπότ εκτελεί καρτεσιανές κινήσεις, κινήσεις άρθρωσης και εντολές gripper· η χαμηλή αδράνεια και οι αισθητήρες θέσης καθιστούν το σύστημα ασφαλές για άμεση ανθρώπινη αλληλεπίδραση.

2 1ο Μέρος - Pick and Place

2.1 Υλοποίηση Διαδικασίας Pick-and-Place με Καρτεσιανές Συντεταγμένες

Στην παρούσα ενότητα υλοποιήθηκε η διαδικασία λήψης-και-εναπόθεσης (pick-and-place) κύβων, προγραμματίζοντας τις συντεταγμένες των ενδιάμεσων θέσεων (waypoints) του ρομποτικού βραχίονα.

Για τον έλεγχο της κίνησης και της λαβής χρησιμοποιήθηκαν οι ακόλουθες εντολές:

- `set_coords` (X, Y, Z): Ορισμός ενδιάμεσων θέσεων κίνησης, όπου X, Y, Z οι καρτεσιανές συντεταγμένες σε χιλιοστά (mm).
- `set_gripper_state` (S): Καθορισμός κατάστασης της αρπάγης/λαβής. Για $S = 0$ η αρπάγη ανοίγει, ενώ για $S = 1$ η αρπάγη κλείνει.
- `set_coords_pick_pos` (X, Y, Z): Ορισμός της θέσης λήψης (pick position) σε καρτεσιανές συντεταγμένες (mm).
- `set_coords_place_pos` (X, Y, Z): Ορισμός της θέσης εναπόθεσης (place position) σε καρτεσιανές συντεταγμένες (mm).

2.1.1 Διαδικασία Εκτέλεσης

Η ακολουθία των κινήσεων που σχεδιάστηκε και εκτελέστηκε περιλαμβάνει τα εξής βήματα:

1. **Σημείο Προσέγγισης (Approach Point):** Μετακίνηση σε θέση ασφαλείας, κατακόρυφα πάνω από τον κύβο, με ανοιχτή λαβή.
2. **Θέση Λήψης:** Κάθοδος στη θέση του κύβου και εντολή κλεισίματος της λαβής ($S = 1$).
3. **Ανύψωση:** Επιστροφή στο σημείο προσέγγισης (Θέση 1) για ασφαλή μεταφορά.
4. **Σημείο Εναπόθεσης (Approach Place Point):** Μετακίνηση σε θέση ασφαλείας, κατακόρυφα πάνω από τον στόχο εναπόθεσης.
5. **Θέση Εναπόθεσης:** Κάθοδος στο επιθυμητό σημείο και εντολή ανοίγματος της λαβής ($S = 0$).
6. **Απομάκρυνση:** Επιστροφή στο σημείο εναπόθεσης (Θέση 4) πριν την επαναφορά στην αρχική θέση.

Η παραπάνω διαδικασία φαίνεται σε αυτό το βίντεο.

2.1.2 Παρατηρήσεις & Μέτρα Ασφαλείας

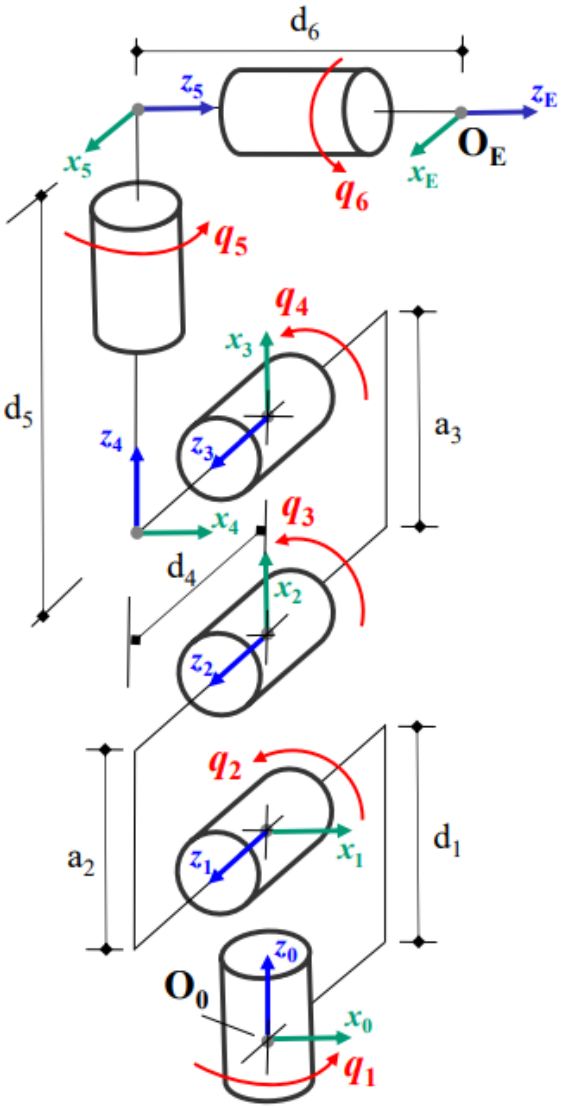
Κρίσιμη παράμετρος για την επιτυχία του πειράματος ήταν η χρήση των ενδιάμεσων σημείων προσέγγισης και απομάκρυνσης (Βήματα 1 και 4/6).

- Η παρεμβολή της **Θέσης 1** μεταξύ της αρχικής θέσης και της θέσης λήψης εξασφαλίζει κάθετη προσέγγιση. Χωρίς αυτή, το ρομπότ θα εκτελούσε διαγώνια κίνηση, αυξάνοντας τον κίνδυνο ανατροπής του κύβου πριν τη λήψη.
- Αντίστοιχα, η χρήση της **Θέσης 6** μετά την εναπόθεση είναι απαραίτητη ώστε ο βραχίονας να απομακρυνθεί κατακόρυφα. Σε αντίθετη περίπτωση, μια διαγώνια κίνηση επιστροφής θα μπορούσε να χτυπήσει και να ρίξει τον τοποθετημένο κύβο.

Τέλος, κατά τον προγραμματισμό των συντεταγμένων (X, Y, Z) δόθηκε ιδιαίτερη προσοχή στους περιορισμούς του χώρου εργασίας (workspace constraints), ώστε να αποφευχθούν συγκρούσεις (collisions) του ρομπότ με τον εαυτό του (self-collision), τη βάση στήριξης ή την επιφάνεια εργασίας.

3 2ο Μέρος - Ευθεία Κινηματική

Το κινηματικό διάγραμμα με τις κινηματικές παραμέτρους του βραχίονα καθώς και τα ενδιάμεσα πλαίσια αναφοράς (coordinate frames) τοποθετημένα με βάση τη μέθοδο Denavit–Hartenberg (D–H) παρατίθεται ακολούθως.



Σχήμα 1: Κινηματικό διάγραμμα του myCobot 280 Jetson Nano όπου τα ενδιάμεσα πλαίσια αναφοράς είναι τοποθετημένα σύμφωνα με τη σύμβαση Denavit–Hartenberg (D–H)

i	θ_i (rad)	d_i (mm)	a_i (mm)	α_i
1	q_1	131.22	0	$\frac{\pi}{2}$
2	$q_2 + \frac{\pi}{2}$	0	110.4	0
3	q_3	0	96	0
4	$q_4 - \frac{\pi}{2}$	63.4	0	$-\frac{\pi}{2}$
5	$q_5 - \frac{\pi}{2}$	75.05	0	$-\frac{\pi}{2}$
6	q_6	45.6	0	0

Πίνακας 1: Πίνακας παραμέτρων Denavit–Hartenberg

Σύμφωνα με τη μήτρα Denavit–Hartenberg:

$$A_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Συνεπώς:

$$A_1^0 = \begin{bmatrix} \cos q_1 & 0 & \sin q_1 & 0 \\ \sin q_1 & 0 & -\cos q_1 & 0 \\ 0 & 1 & 0 & 131.22 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^1 = \begin{bmatrix} -\sin q_2 & -\cos q_2 & 0 & -110.4 \sin q_2 \\ \cos q_2 & -\sin q_2 & 0 & 110.4 \cos q_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} \cos q_3 & -\sin q_3 & 0 & 96 \cos q_3 \\ \sin q_3 & \cos q_3 & 0 & 96 \sin q_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4^3 = \begin{bmatrix} \sin q_4 & 0 & \cos q_4 & 0 \\ -\cos q_4 & 0 & \sin q_4 & 0 \\ 0 & -1 & 0 & 63.4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5^4 = \begin{bmatrix} \sin q_5 & 0 & \cos q_5 & 0 \\ -\cos q_5 & 0 & \sin q_5 & 0 \\ 0 & -1 & 0 & 75.05 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6^5 = \begin{bmatrix} \cos q_6 & -\sin q_6 & 0 & 0 \\ \sin q_6 & \cos q_6 & 0 & 0 \\ 0 & 0 & 1 & 45.6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Οπότε:

$$A_2^0 = A_1^0 A_2^1 = \begin{bmatrix} \cos q_1 & 0 & \sin q_1 & 0 \\ \sin q_1 & 0 & -\cos q_1 & 0 \\ 0 & 1 & 0 & 131.22 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -\sin q_2 & -\cos q_2 & 0 & -110.4 \sin q_2 \\ \cos q_2 & -\sin q_2 & 0 & 110.4 \cos q_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} -c_1 s_2 & -c_1 c_2 & s_1 & -110.4\, c_1 s_2 \\ -s_1 s_2 & -s_1 c_2 & -c_1 & -110.4\, s_1 s_2 \\ c_2 & -s_2 & 0 & 131.22 + 110.4\, c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^0 = A_2^0 A_3^2 = \begin{bmatrix} -c_1 s_2 & -c_1 c_2 & s_1 & -110.4\, c_1 s_2 \\ -s_1 s_2 & -s_1 c_2 & -c_1 & -110.4\, s_1 s_2 \\ c_2 & -s_2 & 0 & 131.22 + 110.4\, c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_3 & -s_3 & 0 & 96\, c_3 \\ s_3 & c_3 & 0 & 96\, s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} -c_1 s_{23} & -c_1 c_{23} & s_1 & -110.4\, c_1 s_2 - 96\, c_1 s_{23} \\ -s_1 s_{23} & -s_1 c_{23} & -c_1 & -110.4\, s_1 s_2 - 96\, s_1 s_{23} \\ c_{23} & -s_{23} & 0 & 131.22 + 110.4\, c_2 + 96\, c_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4^0 = A_3^0 A_4^3 = \begin{bmatrix} -c_1 s_{23} & -c_1 c_{23} & s_1 & -110.4\, c_1 s_2 - 96\, c_1 s_{23} \\ -s_1 s_{23} & -s_1 c_{23} & -c_1 & -110.4\, s_1 s_2 - 96\, s_1 s_{23} \\ c_{23} & -s_{23} & 0 & 131.22 + 110.4\, c_2 + 96\, c_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_4 & 0 & c_4 & 0 \\ -c_4 & 0 & s_4 & 0 \\ 0 & -1 & 0 & 63.4 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} c_1 c_{234} & -s_1 & -c_1 s_{234} & -110.4\, c_1 s_2 - 96\, c_1 s_{23} + 63.4\, s_1 \\ s_1 c_{234} & c_1 & -s_1 s_{234} & -110.4\, s_1 s_2 - 96\, s_1 s_{23} - 63.4\, c_1 \\ s_{234} & 0 & c_{234} & 131.22 + 110.4\, c_2 + 96\, c_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5^0 = A_4^0 A_5^4$$

$$= \begin{bmatrix} c_1 c_{234} & -s_1 & -c_1 s_{234} & -110.4\, c_1 s_2 - 96\, c_1 s_{23} + 63.4\, s_1 \\ s_1 c_{234} & c_1 & -s_1 s_{234} & -110.4\, s_1 s_2 - 96\, s_1 s_{23} - 63.4\, c_1 \\ s_{234} & 0 & c_{234} & 131.22 + 110.4\, c_2 + 96\, c_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_5 & 0 & c_5 & 0 \\ -c_5 & 0 & s_5 & 0 \\ 0 & -1 & 0 & 75.05 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_5 s_1 + c_{234} c_1 s_5 & c_1 s_{234} & c_{234} c_1 c_5 - s_1 s_5 & 63.4\, s_1 - 75.05\, c_1 s_{234} - 96\, c_1 s_{23} - 110.4\, c_1 s_2 \\ c_{234} s_1 s_5 - c_1 c_5 & s_1 s_{234} & c_1 s_5 + c_{234} c_5 s_1 & -63.4\, c_1 - 75.05\, s_1 s_{234} - 96\, s_1 s_{23} - 110.4\, s_1 s_2 \\ s_{234} s_5 & -c_{234} & s_{234} c_5 & 131.22 + 96\, c_{23} + 110.4\, c_2 + 75.05\, c_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6^0 = A_5^0 A_6^5$$

$$= \begin{bmatrix} c_5 s_1 + c_{234} c_1 s_5 & c_1 s_{234} & c_{234} c_1 c_5 - s_1 s_5 & 63.4\, s_1 - 75.05\, c_1 s_{234} - 96\, c_1 s_{23} - 110.4\, c_1 s_2 \\ c_{234} s_1 s_5 - c_1 c_5 & s_1 s_{234} & c_1 s_5 + c_{234} c_5 s_1 & -63.4\, c_1 - 75.05\, s_1 s_{234} - 96\, s_1 s_{23} - 110.4\, s_1 s_2 \\ s_{234} s_5 & -c_{234} & s_{234} c_5 & 131.22 + 96\, c_{23} + 110.4\, c_2 + 75.05\, c_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & 45.6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} R & p_E \\ 0 & 1 \end{bmatrix}$$

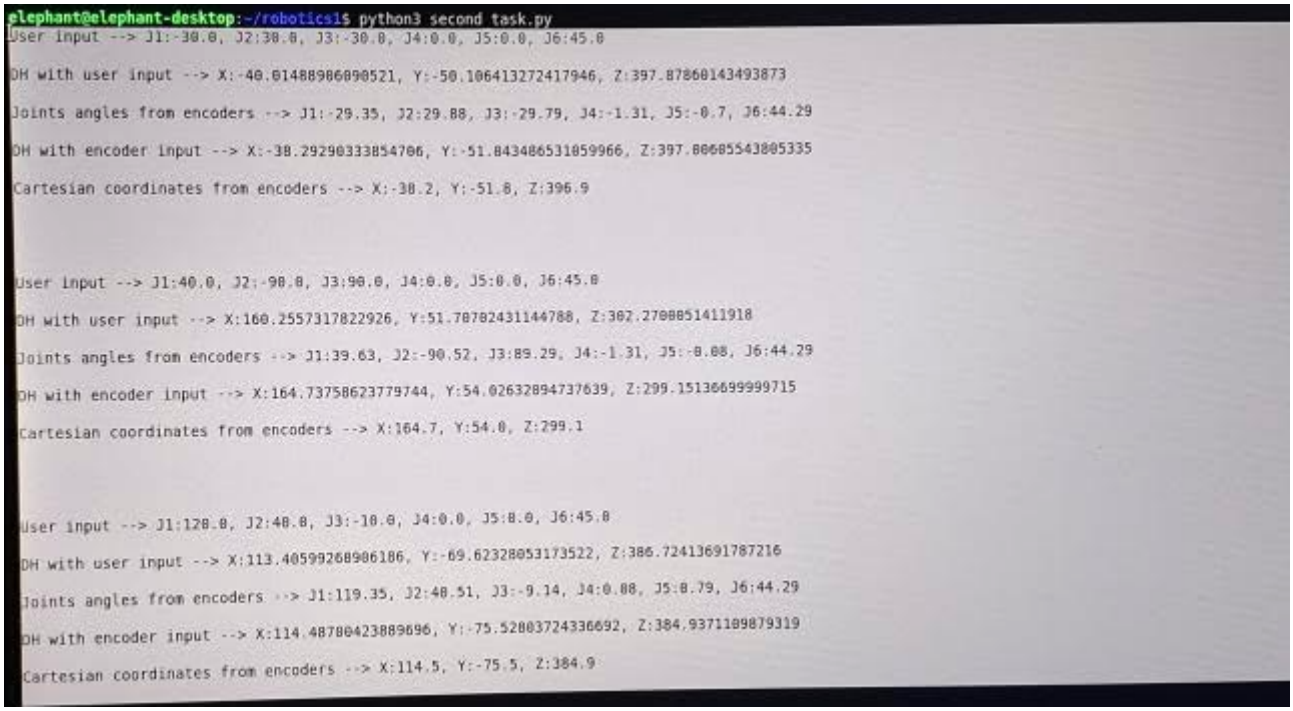
$$R = \begin{bmatrix} c_6(c_5 s_1 + c_{234} c_1 s_5) + s_{234} c_1 s_6 & s_{234} c_1 c_6 - s_6(c_5 s_1 + c_{234} c_1 s_5) & c_{234} c_1 c_5 - s_1 s_5 \\ s_{234} s_1 s_6 - c_6(c_1 c_5 - c_{234} s_1 s_5) & s_6(c_1 c_5 - c_{234} s_1 s_5) + s_{234} c_6 s_1 & c_1 s_5 + c_{234} c_5 s_1 \\ s_{234} c_6 s_5 - c_{234} s_6 & -c_{234} c_6 - s_{234} s_5 s_6 & s_{234} c_5 \end{bmatrix}$$

$$p_E = \begin{bmatrix} s_1(63.4 - 45.6 s_5) - c_1(75.05 s_{234} + 96 s_{23} + 110.4 s_2 - 45.6 c_{234} c_5) \\ c_1(45.6 s_5 - 63.4) - s_1(75.05 s_{234} + 96 s_{23} + 110.4 s_2 - 45.6 c_{234} c_5) \\ 75.05 c_{234} + 96 c_{23} + 110.4 c_2 + 45.6 s_{234} c_5 + 131.2 \end{bmatrix}$$

Τελικά,

$$p_E = \begin{bmatrix} s_1(63.4 - 45.6 s_5) - c_1(75.05 s_{234} + 96 s_{23} + 110.4 s_2 - 45.6 c_{234} c_5) \\ c_1(45.6 s_5 - 63.4) - s_1(75.05 s_{234} + 96 s_{23} + 110.4 s_2 - 45.6 c_{234} c_5) \\ 75.05 c_{234} + 96 c_{23} + 110.4 c_2 + 45.6 s_{234} c_5 + 131.2 \end{bmatrix}$$

Στην παρακάτω εικόνα απεικονίζονται οι γωνίες των αρθρώσεων που έδωσε ο χρήστης και αυτές που υπολογίστηκαν από τους encoders. Επιπλέον, για κάθε συνδυασμό γωνιών δίνονται οι τελικές θέσεις που υπολογίστηκαν στον κώδικα της Python.



Σχήμα 2: Γωνίες χρήστη, γωνίες encoders και οι αντίστοιχες καρτεσιανές συντεταγμένες του τελικού εργαλείου δράσης

Αρκεί να υπολογίσουμε τις συντεταγμένες του end-effector χρησιμοποιώντας ως γωνίες των αρθρώσεων αυτές των encoders.

- $q_1 = -29.35°,\ q_2 = 29.88°,\ q_3 = -29.79°,\ q_4 = -1.31°,\ q_5 = -0.7°,\ q_6 = 44.29°$

$$p_E = \begin{bmatrix} -38.291859143472900 \\ -51.843077768995656 \\ 397.0066583177501 \end{bmatrix}$$

- $q_1 = 39.63°,\ q_2 = -90.52°,\ q_3 = 89.29°,\ q_4 = -1.31°,\ q_5 = -0.08°,\ q_6 = 44.29°$

$$p_E = \begin{bmatrix} 164.7375363728268 \\ 54.026940109758270 \\ 299.1513541312829 \end{bmatrix}$$

- $q_1 = 119.35°,\ q_2 = 40.51°,\ q_3 = -9.14°,\ q_4 = 0.08°,\ q_5 = 0.79°,\ q_6 = 44.29°$

$$p_E = \begin{bmatrix} 114.4860319110242 \\ 75.527988772924320 \\ 384.9379322035499 \end{bmatrix}$$

Παρατηρούμε ότι τα παραπάνω αποτελέσματα διαφέρουν σημαντικά από τα αντίστοιχα που υπολόγισε ο κώδικας Python (DH with encoder input). Μάλιστα, σε μερικές περιπτώσεις διαφέρουν από το τρίτο δεκαδικό ψηφίο, απόκλιση που δεν μπορεί να θεωρηθεί αμελητέα.

Η συμπεριφορά αυτή οφείλεται στο ότι στον κώδικα Python χρησιμοποιήθηκε η προσέγγιση $\frac{\pi}{2} = 1.5708$ για τον υπολογισμό των παραμέτρων DH, όπως φαίνεται στην ακόλουθη εικόνα.

```python
import time
import numpy as np
from pymycobot.mycobot import MyCobot

# Mycobot280 robot DH parameters
# (theta, d, a, alpha offset # mm and radians)
# pi/2 = 1.5708
mycobot_dh_parameters = [
    [0.0, 131.22, 0.0, 1.5708, 0.0],
    [0.0, 0.0, 110.4, 0.0, 1.5708],
    [0.0, 0.0, 96.0, 0.0, 0.0],
    [0.0, 63.4, 0.0, -1.5708, -1.5708],
    [0.0, 75.05, 0.0, -1.5708, -1.5708],
    [0.0, 45.6, 0.0, 0.0, 0.0]
]

# Define transformation (theta, d, a, alpha)
def transform(theta, d, a, alpha):
    return np.array([[np.cos(theta), -np.sin(theta)*np.cos(alpha), np.sin(theta)*np.sin(alpha), a*np.cos(theta)],
                     [np.sin(theta), np.cos(theta)*np.cos(alpha), -np.cos(theta)*np.sin(alpha), a*np.sin(theta)],
                     [0, np.sin(alpha), np.cos(alpha), d],
                     [0, 0, 0, 1]])

# Define forward kinematics (dh_params, joint_angles)
def forward_kinematics(dh_params, joint_angles):
    # Function to compute overall transformation from the base to the end-effector
    T = np.eye(4)
    for i in range(len(joint_angles)):
        theta, d, a, alpha, offset = dh_params[i]  # Unpack DH parameters
        theta += joint_angles[i]
        T = T @ transform(theta + offset, d, a, alpha)  # Compute transformation
    return T  # Return the end-effector position (x, y, z)

```

Σχήμα 3: Προσέγγιση $\frac{\pi}{2} = 1.5708$ στον κώδικα Python

Προκειμένου να δείξουμε ότι πράγματι η απόκλιση που παρατηρείται οφείλεται αποκλειστικά σε αυτή την προσέγγιση, δημιουργήσαμε το ακόλουθο πρόγραμμα Matlab.

```m
% ================================================
% FORWARD KINEMATICS WITH π/2 = 1.5708
% ================================================
clear all
clc
format long g
pi_2 = 1.5708;          % Approximation for π/2
% DH parameters table [θ_offset, d, a, α]
dh_params = [
    [0,           131.22, 0,   pi_2];
    [pi_2,     0,      110.4, 0];
    [0,           0,      96,    0];
    [-pi_2,    63.4,   0,    -pi_2];
    [-pi_2,    75.05,  0,    -pi_2];
    [0,           45.6,   0,     0]
];
syms q1 q2 q3 q4 q5 q6 real
dh_T = @(theta, d, a, alpha) [
    cos(theta), -sin(theta)*cos(alpha),  sin(theta)*sin(alpha), a*cos(theta);
    sin(theta),  cos(theta)*cos(alpha), -cos(theta)*sin(alpha), a*sin(theta);
    0,           sin(alpha),            cos(alpha),            d;
    0,           0,                      0,                     1
];
A = cell(6,1);
A{1} = dh_T(q1 + dh_params(1,1), dh_params(1,2), dh_params(1,3), dh_params(1,4));
A{2} = dh_T(q2 + dh_params(2,1), dh_params(2,2), dh_params(2,3), dh_params(2,4));
A{3} = dh_T(q3 + dh_params(3,1), dh_params(3,2), dh_params(3,3), dh_params(3,4));
A{4} = dh_T(q4 + dh_params(4,1), dh_params(4,2), dh_params(4,3), dh_params(4,4));
A{5} = dh_T(q5 + dh_params(5,1), dh_params(5,2), dh_params(5,3), dh_params(5,4));
A{6} = dh_T(q6 + dh_params(6,1), dh_params(6,2), dh_params(6,3), dh_params(6,4));
A06 = A{1} * A{2} * A{3} * A{4} * A{5} * A{6};
A06_simplified = simplify(A06);
p_E = A06_simplified(1:3, 4);
```

```m
angle_sets = {
    [-29.35,  29.88, -29.79, -1.31, -0.7,   44.29];
    [ 39.63, -90.52,  89.29, -1.31, -0.08,  44.29];
    [119.35,  40.51,  -9.14,  0.08,  0.79,  44.29]
};
results_python = {
  [-38.29290333854706, -51.843486531059966, 397.00605543805335];
  [164.73758623779744, 54.02632094737639, 299.15136699999715];
  [114.48700423889696, -75.52803724336692, 384.9371109879319];
};
for i = 1:length(angle_sets)
    angles_deg = angle_sets{i};
    angles_rad = angles_deg * (pi / 180);
    python_result = results_python{i}';
    p_E_val = double(subs(p_E, ...
        {q1, q2, q3, q4, q5, q6}, ...
        num2cell(angles_rad)));
    fprintf('q1 = %.2f°, q2 = %.2f°, q3 = %.2f°, q4 = %.2f°, q5 = %.2f°, q6 = %.2f°\n', ...
        angles_deg(1), angles_deg(2), angles_deg(3), angles_deg(4), angles_deg(5), angles_deg(6));
    fprintf('p_E=\n');
    fprintf('    [%.16f;\n', p_E_val(1));
    fprintf('     %.16f;\n', p_E_val(2));
    fprintf('     %.16f]\n\n', p_E_val(3));
    diff = p_E_val - python_result;
    fprintf('p_E-p_{Epython}=\n');
    fprintf('    [%.16f;\n', diff(1));
    fprintf('     %.16f;\n', diff(2));
    fprintf('     %.16f]\n\n', diff(3));
    nn=sqrt(sum(diff.^2));
    fprintf('norm: ||p_E - p_{Epython}|| = %.16f\n', nn);
    fprintf('-------------------------------------------------------------------------------------------');
    fprintf('\n');
end
```

Σχήμα 4: Κώδικας Matlab με την προσέγγιση $\frac{\pi}{2} = 1.5708$

Η έξοδος του προγράμματος δίνεται ακολούθως.

```
q1 = -29.35°, q2 = 29.88°, q3 = -29.79°, q4 = -1.31°, q5 = -0.70°, q6 = 44.29°
p_E=
    [-38.2929033385471129;
     -51.8434865310599307;
     397.0060554380533517]

p_E-p_{Epython}=
    [-0.0000000000000497;
     0.0000000000000355;
     0.0000000000000000]

norm: ||p_E - p_{Epython}|| = 0.0000000000000611
-------------------------------------------------------------------------------------------
q1 = 39.63°, q2 = -90.52°, q3 = 89.29°, q4 = -1.31°, q5 = -0.08°, q6 = 44.29°
p_E=
    [164.7375862377975011;
     54.0263209473764121;
     299.1513669999970944]

p_E-p_{Epython}=
    [0.0000000000000568;
     0.0000000000000213;
     -0.0000000000000568]

norm: ||p_E - p_{Epython}|| = 0.0000000000000832


-------------------------------------------------------------------------------------------
q1 = 119.35°, q2 = 40.51°, q3 = -9.14°, q4 = 0.08°, q5 = 0.79°, q6 = 44.29°
p_E=
    [114.4870042388970006;
     -75.5280372433670095;
     384.9371109879318169]

p_E-p_{Epython}=
    [0.0000000000000426;
     -0.0000000000000853;
     -0.0000000000000568]

norm: ||p_E - p_{Epython}|| = 0.0000000000001110
-------------------------------------------------------------------------------------------
```

Σχήμα 5: Έξοδος κώδικα Matlab

Παρατηρούμε πλέον ότι οι αποκλίσεις από τις τιμές που υπολόγισε ο κώδικας Python είναι αμελητέες. Δείξαμε συνεπώς ότι οι συντεταγμένες που προσδιορίσαμε με το κινηματικό μοντέλο διαφέρουν από εκείνες που υπολογίζει ο κώδικας Python αποκλειστικά λόγω της προσέγγισης $\frac{\pi}{2} = 1.5708$.

4 3ο Μέρος - Kinesthetic Teaching