

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ



ΡΟΜΠΟΤΙΚΗ Ι: ΑΝΑΛΥΣΗ - ΕΛΕΓΧΟΣ - ΕΡΓΑΣΤΗΡΙΟ

(2021 – 2022)

1^η Εργαστηριακή Άσκηση: Ρομποτικό Κύτταρο

Υπεύθυνος Εργαστηρίου:

- Κ. Τζαφέστας

Μεταπτυχιακοί Συνεργάτες:

- Παρασκευάς Οικονόμου

Ημερομηνία διεξαγωγής:

- 27/10/2021 18:15 – 19:00

Ομάδα:

- B4-1

Ονοματεπώνυμο, Α.Μ. & Στοιχεία Επικοινωνίας Μελών:

- | | | |
|----------------------|--------------|--|
| ▪ Χάρης Αλετράρης | (031 18 700) | el18700@mail.ntua.gr |
| ▪ Παναγιώτης Γκότσης | (031 18 010) | el18010@mail.ntua.gr |
| ▪ Ειρηναίος Μιχαήλ | (031 18 705) | el18705@mail.ntua.gr |
| ▪ Χρήστος Τσούφης | (031 17 176) | el17176@mail.ntua.gr |

Περιγραφή Πειράματος

Η 1^η Εργαστηριακή Άσκηση με τίτλο “Ρομποτικό Κύτταρο” εκτελείται στην εργαστηριακή διάταξη MASH και αφορά την λειτουργία ενός ρομποτικού βραχίονα, τύπου IR 52 c, του Εργαστηρίου Ρομποτικής, ο οποίος φαίνεται στην συνέχεια.

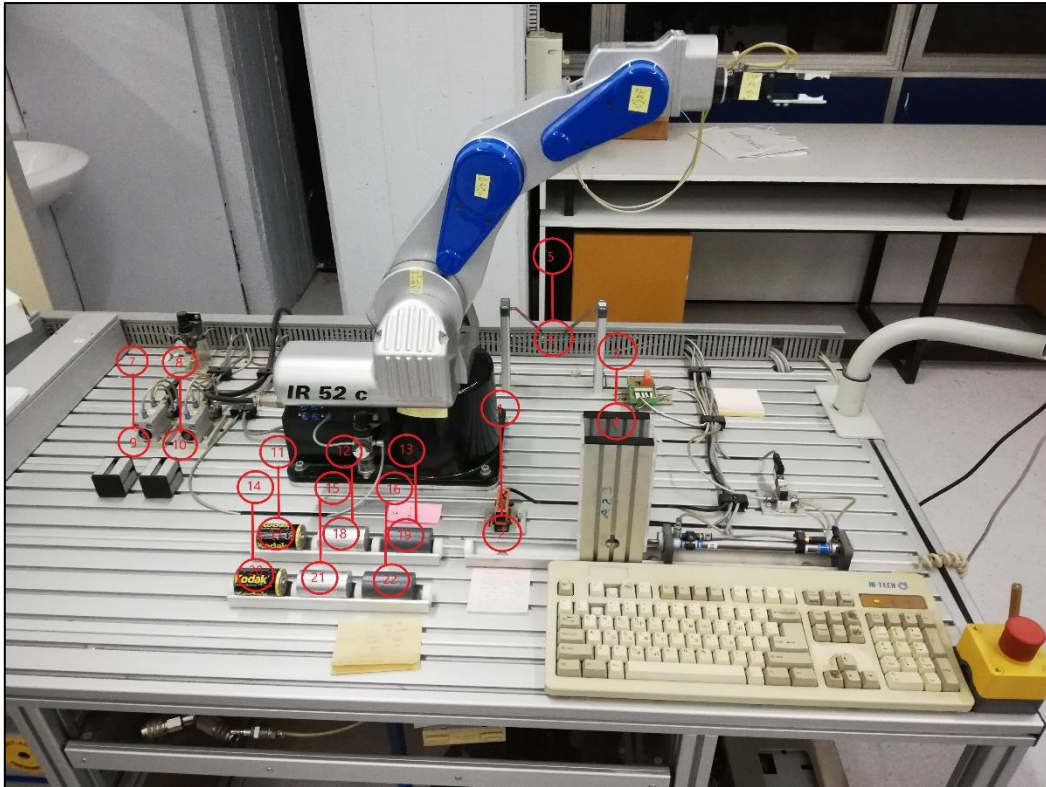


Σκοπός του πειράματος ήταν η εξοικείωση με τον ρομποτικό βραχίονα καθώς και το περιβάλλον γύρω του. Τα μέλη της ομάδας κλήθηκαν να προγραμματίσουν τον ρομποτικό βραχίονα ώστε εκείνος να μεταβεί σε συγκεκριμένα σημεία στο χώρο αλλά ήρθαν και σε επαφή με ηλεκτροπνευματικές διατάξεις και τεχνολογίες αισθητήρων. Σημειώνεται ότι ο βραχίονας έχει 5 βαθμούς ελευθερίας κίνησης οπότε δύναται να προσεγγίσει τα σημεία που τον περικλείουν και αφορούν το πείραμα. Επιπλέον, διαθέτει και τελικό στοιχείο δράσης με δαγκάνα (gripper) που του επιτρέπει να αρπάζει και να μετακινεί μικρά αντικείμενα γύρω του. Επομένως, συνολικά τα παραπάνω στοιχεία μαζί με τα αντικείμενα γύρω τους αποτελούν ένα πλήρες μικρο-περιβάλλον που δικαιολογούν και την ονομασία της άσκησης. Τέλος, εν συντομία, η λειτουργία που εκτελεί η πειραματική διάταξη είναι να δέχεται αντικείμενα από μια στοίβα και να τα διακρίνει σε τρεις κατηγορίες (πλαστικά, μεταλλικά αφόρτιστα, μεταλλικά φορτισμένα) και να τα τοποθετεί σε κατάλληλες θέσεις και να τους επισυνάπτει τις αντίστοιχες ετικέτες.

Χαρακτηριστικά Πειραματικής Διάταξης

Η πειραματική διάταξη διαθέτει τα ακόλουθα στοιχεία, όπως φαίνεται και από την εικόνα:

- Ρομποτικός Βραχίονας IR 52 c: Ο βραχίονας εκτελεί όλες τις μετακινήσεις των αντικειμένων.
- Χωρητικός Αισθητήρας Φόρτισης (σημείο 6): Ο αισθητήρας αναπτύσσει τάση εάν το αντικείμενο είναι φορτισμένο και τοποθετηθεί με τη σωστή πολικότητα (είναι μονής πολικότητας) οπότε ανάβει το LED και μεταφέρει την πληροφορία στον ελεγκτή. Η αρχή λειτουργίας του βασίζεται σε ένα κύκλωμα ταλάντωσης RC.
- Επαγωγικός Αισθητήρας (σημείο 4 – στην βάση της στοίβας): Ο αισθητήρας αναπτύσσει εναλλασσόμενο Η/Μ πεδίο μπροστά από την ενεργό επιφάνειά του και έτσι μπορεί να αναγνωρίσει αν το αντικείμενο είναι μεταλλικό διότι τότε προκαλείται ισχυρή απόσβεση ταλάντωσης και αλλάζει η κατάσταση της εξόδου του αισθητήρα. Στο ύψος του 2^{ου} αντικειμένου ανιχνεύει την ύπαρξη μεταλλικού αντικειμένου.
- Οπτικός Αισθητήρας (σημείο 2): Ο αισθητήρας διαθέτει φωτοκύτταρο εύρους 10 cm το οποίο μέσω αντανάκλασης (σύστημα πομπού – δέκτη) διακρίνει εάν υπάρχει αντικείμενο μπροστά του μόλις αυτό εξέλθει από το κάτω μέρος της στοίβας.
- Κυλινδρικές Μπαταρίες
- Μεταλλικοί Κύλινδροι
- Πλαστικοί Κύλινδροι
- Εργαλείο Επισύναψης Ετικέτας (σημεία 9, 10): Αυτά τα έμβολα συνεισφέρουν στην επικόλληση ετικετών πριν τα αντικείμενα τοποθετηθούν στην θέση τους.
- Στοίβα Εναπόθεσης Αντικειμένων (σημείο 4)
- Υπολογιστής προγραμματισμού βραχίονα: Ο υπολογιστής έχει εγκατεστημένο το περιβάλλον προγραμματισμού PSI. Επιπλέον, διαθέτει μια κάρτα ISA που χρησιμεύει στην επέκταση του επεξεργαστή στο χώρο διευθυνσιοδότησης.
- Ηλεκτροπνευματικό έμβολο: Το έμβολο βρίσκεται δίπλα στην βάση της στοίβας και όταν βρίσκεται στην μόνιμη κατάσταση προωθεί το πρώτο αντικείμενο μπροστά από τον οπτικό αισθητήρα και μετά συσπειρώνεται πάλι στην αρχική του θέση.
- Ελεγκτής: Ο ελεγκτής βρίσκεται κάτω από το σύστημα του βραχίονα και διαθέτει 8 ψηφιακές εισόδους (καταχωρητές) και 8 εξόδους (καταχωρητές) καθώς και 1 καταχωρητή ανάδρασης.
- Κουμπί Κινδύνου: Είναι τοποθετημένο πάνω στο τραπέζι για την άμεση διακοπή κίνησης του βραχίονα μετά από κάποια ανεξέλεγκτη κίνηση.
- Συμπιεστής αέρα: Με την συσσώρευση και προώθηση αέρα μπορεί να μετακινεί τον βραχίονα.



Επεξήγηση σημείων:

Τα σημεία είναι αριθμημένα έτσι ώστε να δείχνουν την θέση κατακόρυφα πάνω από ένα σημείο καθώς και την τελική επιθυμητή θέση. Για παράδειγμα, το ζεύγος 7 – 9 σημαίνει ότι το ρομπότ φτάνει στην θέση 7 η οποία είναι κατακόρυφα πάνω από την θέση 9 που είναι η τελική θέση.

Ο Ρομποτικός βραχίονας αποτελείται από 5 στροφικές αρθρώσεις οπότε έχει 5 βαθμούς ελευθερίας καθώς και μια αρπαγή ως τελικό σημείο δράσης. Το περιβάλλον του αποτελείται από το σύνολο των σημείων που μπορεί να προσεγγίσει από το end-effector, λαμβάνοντας υπόψιν τα σχετικά singularities. Σε κάθε άρθρωση του βραχίονα υπάρχουν εγκατεστημένοι σερβοκινητήρες (servo-set point) υψηλής απόδοσης οι οποίοι προσφέρουν πλήρη έλεγχο της θέσης και της ταχύτητας της εκάστοτε άρθρωσης αφού διαθέτουν ανάλυση ακρίβειας 415.000 βημάτων ανά άξονα περιστροφής. Αξίζει να σημειωθεί ότι σε ρομποτικά συστήματα προτιμώνται οι σερβοκινητήρες, λόγω αξιοπιστίας που διαθέτουν ως προς τον έλεγχο της κίνησης. Επιπλέον, ο gripper ανοιγοκλείνει με χρήση πεπιεσμένου αέρα ώστε να δίνει την δυνατότητα στον βραχίονα να αρπάζει και να απελευθερώνει τα αντικείμενα. Τέλος, ο ελεγκτής του ρομπότ βρίσκεται στην βάση του. Ο προγραμματισμός του βραχίονα γίνεται μέσω κατάλληλου υπολογιστή προσαρμοσμένο στο εκάστοτε ρομπότ. Αυτός ο υπολογιστής δίνει εντολές στον βραχίονα, δέχεται πληροφορίες για την θέση του βραχίονα και για τους 3 αισθητήρες, οι οποίοι είναι απαραίτητοι για την σωστή εκτέλεση του πειράματος. Για την εκτέλεση μιας κίνησης απαιτείται να του δοθεί ως είσοδος το σημείο στο οποίο πρέπει να φτάσει και το είδος της κίνησης που θα πρέπει να εκτελέσει. Επομένως, ο έλεγχος της θέσης γίνεται μέσω αυξητικών κωδικοποιητών για την κάθε άρθρωση, ενώ για την αυτοματοποίηση χρησιμοποιείται το προγραμματιστικό περιβάλλον PSI.

Αρχή Λειτουργίας Ρομποτικού Βραχίονα

Για την εκτέλεση των επιθυμητών λειτουργιών, θα πρέπει το ρομποτικό σύστημα να κινηθεί με προδιαγεγραμμένο τρόπο μέσα στον χώρο. Για να είναι κάτι τέτοιο εφικτό θα πρέπει ο υπολογιστής να εκφράσει με σαφώς ορισμένο τρόπο την τοποθέτηση του βραχίονα στον χώρο. Έτσι, γίνεται χρήση των 4 διαφορετικών συστημάτων συντεταγμένων, αναλόγως με την περίπτωση:

- World Coordinates: Η μεταβολή εκφράζεται με το καρτεσιανό σύστημα (x, y, z) που είναι σταθερά τοποθετημένο στον χώρο του ρομπότ. Οι συντεταγμένες βρίσκονται σε κάθετους ανά δύο άξονες οι οποίοι ορίζουν ένα ορθοκανονικό σύστημα συντεταγμένων. Στον υπολογιστή εισάγει κανείς είτε προσημασμένες μεταβολές των κατευθύνσεων αυτών (δηλαδή σχετικές μετατοπίσεις) και το ρομπότ κινείται προς αυτές τις θέσεις χωρίς να αλλάζει τον προσανατολισμό του gripper.
- Axis Coordinates: Η μεταβολή είναι σχετική και η κατεύθυνση (πρόσημο) της κίνησης είναι καθορισμένη από τον άξονα της κάθε άρθρωσης. Ουσιαστικά, χρησιμοποιούνται όλοι οι άξονες κίνησης του ρομπότ για τον προσδιορισμό της θέσης και των σημείων του χώρου. Στον υπολογιστή εισάγει κανείς τις μοίρες που θα κινηθεί η κάθε άρθρωση του ρομπότ από την τωρινή θέση της. Οι μεταβλητές που αναπαριστούν τις αρθρώσεις είναι οι A_1, A_2, A_3, A_4, A_5 .
- Tool Coordinates: Η μεταβολή εκφράζεται με το καρτεσιανό σύστημα συντεταγμένων (x, y, z) και χρησιμοποιείται όπως και με τις world coordinates, με την μόνη διαφορά ότι το σύστημα συντεταγμένων είναι προσαρτημένο στο εργαλείο (έχει τον άξονα z στην προέκταση της λαβίδας). Είναι προτιμότερο για την εκμάθηση του ρομπότ σε μετακινήσεις όπου τα υπόλοιπα συστήματα δυσκολεύονται. Οι μετατοπίσεις είναι σχετικές και ο προσανατολισμός του τελικού στοιχείου δράσης διατηρείται. Κατά την κίνηση το σύστημα συντεταγμένων του εργαλείου δεν μένει σταθερό στο χώρο αλλά ακολουθεί το εργαλείο.
- Tool Orientation: Η μεταβολή εκφράζεται αντίστοιχα με το tool coordinates και αφορά τοπικά το τελικό εργαλείο του ρομπότ. Σημειώνεται ότι δεν εκφράζει μετατοπίσεις αλλά στροφικές κινήσεις που συνθέτουν τον προσανατολισμό του (roll, pitch, yaw).

Με την χρήση του κατάλληλου κάθε φορά συστήματος, μπορεί κανείς να κατευθύνει ένα ρομπότ ώστε να φτάσει στην επιθυμητή θέση και να αποθηκεύσει στην μνήμη του υπολογιστή το εκάστοτε σημείο καταγράφοντας τις συντεταγμένες του. Ως σημείο έναρξης θεωρείται το σημείο που βρίσκεται ήδη το ρομπότ. Μάλιστα, σημειώνεται ότι κατά την πρώτη κίνηση απαιτείται calibration ώστε να γνωρίζει την θέση του. Έτσι, μπορεί κανείς να προσδιορίσει την θέση είτε με συντεταγμένες (x, y, z) του end-effector είτε με τις γωνίες (a, b, c) ώστε στην επίλυση του αντίστροφου κινηματικού μοντέλου, το ρομπότ να μπορεί να βρεθεί στην ίδια θέση με διαφορετική διάταξη των αρθρώσεων του (Roll-Pitch-Yaw). Εν προκειμένω, το ρομπότ μπορεί να κινηθεί με τους εξής 4 τρόπους: (i) Elbow Up, (ii) Elbow Down, (iii) Front & (iv) Back και επισημαίνεται ότι εάν δοθούν 3 επιλογές σε κάθε άξονα, τότε συνολικά υπάρχουν $5^3 - 1 = 242$ επιλογές.

Σε αυτό το σημείο είναι σημαντικό να αναφερθεί ότι το θεωρητικό μοντέλο που επιλύει ο υπολογιστής για την κίνηση του ρομπότ είναι το ορθό και το αντίστροφο κινηματικό μοντέλο. Στο ορθό μοντέλο, είναι γνωστή η θέση κάθε άρθρωσης οπότε ο υπολογιστής υπολογίζει την θέση και τον προσανατολισμό του τελικού στοιχείου δράσης οπότε προκύπτει μοναδική λύση. Από την άλλη, στο δεύτερο, ο υπολογιστής γνωρίζει το σημείο στο οποίο θέλει να φέρει το ρομπότ οπότε επιλύει το αντίστροφο κινηματικό μοντέλο ώστε να υπολογίσει τις γωνίες της κάθε άρθρωσης που δίνει το επιθυμητό αποτέλεσμα οπότε υπάρχουν πολλές λύσεις και τελικά επιλέγεται εκείνη που βρίσκεται το ρομπότ την τρέχουσα χρονική στιγμή.

Αναλυτικότερα, για την μετακίνηση του ρομπότ από το σημείο A στο σημείο B, υπάρχουν 3 διαφορετικές μεθόδους, αναλόγως με την επιθυμητή τροχιά που θα ακολουθήσει το ρομπότ. Αυτές οι μέθοδοι είναι οι εξής:

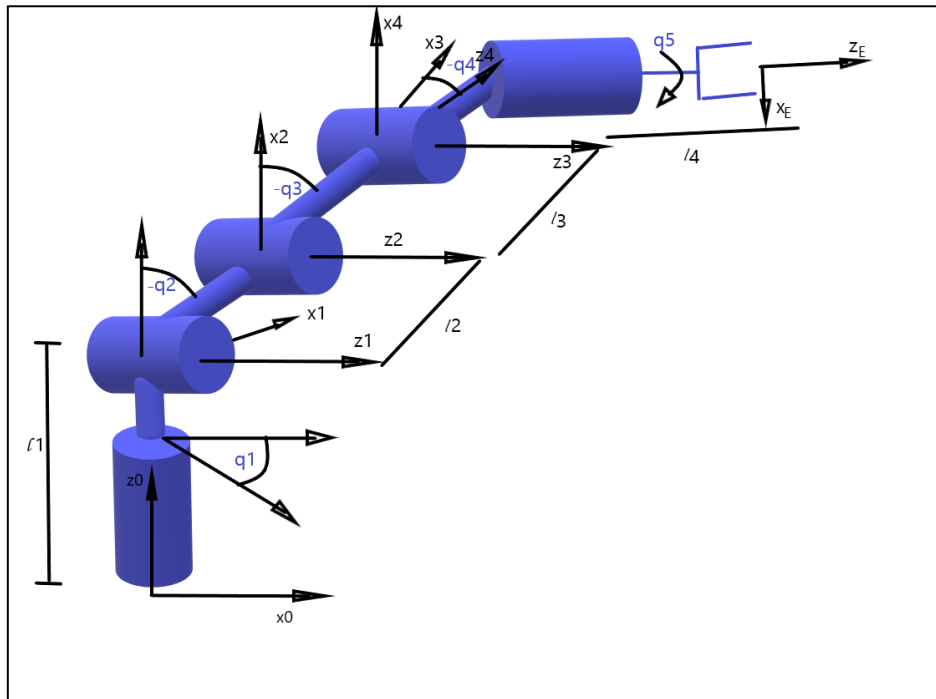
- **Synchro Point-to-Point**: Σύμφωνα με αυτή τη μέθοδο, όλοι οι άξονες κινούνται ταυτόχρονα προς το σημείο προορισμού και σταματούν συγχρόνως την κίνησή τους. Ειδικότερα, το ρομπότ συγκρίνει τις συντεταγμένες των σημείων A, B και ελέγχει το πόσο θα κουνηθεί κάθε άρθρωση ώστε από την κατάσταση της θέσης A να βρεθεί στην θέση B επιλύοντας το ανάστροφο κινηματικό μοντέλο μόνο για αυτές τις δύο θέσεις. Έτσι, μεταβάλλονται όλες οι αρθρώσεις ταυτόχρονα ώστε να φτάσει στην επιθυμητή θέση. Σημειώνεται ότι με αυτή την μέθοδο είναι άγνωστη η διαδρομή που θα ακολουθήσει το ρομπότ αλλά εξασφαλίζεται με μικρή υπολογιστική πολυπλοκότητα (βέλτιστη) η κίνησή του.
- **Line**: Σύμφωνα με αυτή τη μέθοδο, το ρομπότ κινείται έτσι ώστε το εργαλείο του να κινείται σε ευθεία τροχιά μεταξύ των A, B. Μάλιστα, εάν ο προσανατολισμός του εργαλείου διαφέρει από την αρχική προς την τελική θέση, τότε οι γωνίες (a, b, c) μεταβάλλονται κατάλληλα. Αυτό πραγματοποιείται με επαναληπτική χρήση της μεθόδου Point-to-Point. Ο προγραμματιστής επιλέγει το resolution της κίνησης, δηλαδή τα σημεία στα οποία θα χωριστεί η ευθεία, και όσο μεγαλύτερο είναι, τόσο μεγαλύτερη ακρίβεια επιτυγχάνεται για την κίνηση του εργαλείου. Έτσι, πλέον το ρομπότ θα ακολουθήσει μια ακριβή τροχιά αλλά αυτό κοστίζει υπολογιστικά.
- **Arc**: Σύμφωνα με αυτή τη μέθοδο, το εργαλείο ακολουθεί τροχιά κυκλικού τόξου που συνδέει τα A, B. Διευκρινίζεται ότι είναι απαραίτητος ο προσδιορισμός ακόμα ενός σημείου που σχετίζεται με την περιφέρεια του κύκλου ή με το κέντρο του ώστε το ρομπότ να εκτελέσει το μικρότερο τόξο και η τροχιά του να είναι τμήμα έλλειψης. Κι εδώ η κίνηση πάνω από το επίπεδο αναφοράς γίνεται με χρήση Point-to-Point αλλά η μετάβαση κάτω από το επίπεδο αναφοράς με Line.

Το ρομπότ κινείται μεταξύ των σημείων μεταξύ των σημείων με τις μεθόδους Point-to-Point και Line. Πιο συγκεκριμένα, το ρομπότ κινείται με Line όταν από ένα σημείο ψηλά κατεβαίνει σε αυτό που βρίσκεται κατακόρυφα από κάτω του. Επιλέγεται αυτή η μέθοδος διότι έχει ακρίβεια στην κίνηση του ρομπότ και στον τρόπο με τον οποίο προσεγγίζει το επιθυμητό σημείο ώστε να αποφύγει τυχόν εμπόδια και το εργαλείο να έχει τον κατάλληλο προσανατολισμό. Για τις υπόλοιπες κινήσεις, το ρομπότ χρησιμοποιεί την Point-to-Point ώστε να πετύχει μεγαλύτερη ταχύτητα εκτέλεσης εφόσον ο βραχίονας βρίσκεται ψηλά και δεν πρόκειται να πέσει σε κάποιο εμπόδιο.

Επιπλέον, υπογραμμίζεται ότι ο έλεγχος του τρόπου διάσχισης ενός μονοπατιού μπορεί να επιτευχθεί μέσω των παραμέτρων στο PSI (της ταχύτητας διάσχισης, της ταχύτητας των αξόνων, της επιτάχυνσης, της ακρίβειας μονοπατιού και των επικαλύψεων. Τέλος, θα πρέπει να ληφθούν υπόψιν και τυχόν σφάλματα που μπορεί να προκύψουν από σφάλματα οδήγησης PID, από οδόντωση μειωτήρα αλλά και από μοτέρ και ρουλεμάν.

Η Μέθοδος Denavit – Hartenberg

Παρακάτω φαίνεται μια γεωμετρική αναπαράσταση του ρομποτικού βραχίονα του πειράματος σε τυχαία θέση αρθρώσεων. Τα συστήματα συντεταγμένων σε κάθε σύνδεσμο έχουν οριστεί σύμφωνα με την μέθοδο Denavit – Hartenberg.



Σύμφωνα με το ορθό κινηματικό μοντέλο, αναλύεται η αλυσίδα που αναπαριστά τον ρομποτικό βραχίονα με την μέθοδο Denavit – Hartenberg. Ειδικότερα, στο σχήμα φαίνεται η τοποθέτηση των επιμέρους συστημάτων συντεταγμένων που αντιστοιχούν σε κάθε σύνδεσμο του ρομπότ, σύμφωνα με τους κανόνες της μεθόδου.

Αρχικά, τοποθετείται ο άξονας $z_i, \forall i = 0, 1, \dots, 4$ στην γενικευμένη διεύθυνση μετατόπισης της άρθρωσης $i+1$.

Μετά, τοποθετείται ο άξονας $z_5 \equiv z_E$ του στοιχείου δράσης (end-effector) στην προέκταση του τελικού συνδέσμου.

Ύστερα, τοποθετούνται οι άξονες $x_{i+1}, \forall i = 0, 1, \dots, 4$ έτσι ώστε να βρίσκονται στην κοινή κάθετο των αξόνων z_i, z_{i+1} .

Υπάρχουν περιπτώσεις όπου αφήνεται μεγαλύτερη ελευθερία και τοποθετούνται οι άξονες x με επιλογή του προγραμματιστή.

Από το σχήμα φαίνεται ότι οι άξονες x_2, x_3 τοποθετήθηκαν στην προέκταση των συνδέσμων l_2, l_3 , ενώ ο x_E τοποθετήθηκε παράλληλα με τον x_4 στην θέση του τελικού στοιχείου. Στο σχήμα ο x_E έχει αντίθετη φορά διότι έχει στραφεί κατά γωνία q_5 .

Τέλος, οι άξονες y_i τοποθετούνται έτσι ώστε τα επιμέρους καρτεσιανά συστήματα να είναι ορθοκανονικά απλώς δεν έχουν συμπεριληφθεί διότι δεν συμμετέχουν στην ανάλυση κατά DH.

Συνεπώς, ο πίνακας παραμέτρων D-H διαμορφώνεται όπως φαίνεται παρακάτω.

i	θ_i	d_i	a_i	α_i
1	q_1	l_1	0	90°
2	$q_2 + 90^\circ$	0	l_2	0
3	q_3	0	l_3	0
4	$q_4 + 90^\circ$	0	0	90°
$5 \equiv E$	q_5	l_4	0	0

Οπότε, πλέον μπορεί να υπολογιστεί το μητρώο μετατροπής συντεταγμένων του τελικού στοιχείου δράσης ως προς την βάση του ρομπότ.

Άρα, με βάση το μοντέλο ορθής κινηματικής ανάλυσης για το γεωμετρικό μοντέλο του σχήματος, χωρίς τους πρακτικούς περιορισμούς του ρομπότ, θα ισχύει θεωρητικά ότι:

$$A_E^0(q_1, q_2, q_3, q_4, q_5) = A_1^0(q_1)A_2^1(q_2)A_3^2(q_3)A_4^3(q_4)A_5^4(q_5) =$$

$$\left(R_z(q_1)T_z(l_1)R_x\left(\frac{\pi}{2}\right)\right)\left(R_z\left(\frac{\pi}{2} + q_2\right)T_x(l_2)\right)\left(R_z(q_3)T_x(l_3)\right)\left(R_z\left(\frac{\pi}{2} + q_4\right)R_x\left(\frac{\pi}{2}\right)\right)\left(R_z(q_5)T_z(l_4)\right)$$

Όπου,

$$A_1^0(q_1) = R_z(q_1)T_z(l_1)R_x\left(\frac{\pi}{2}\right) = \begin{pmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_3^1(q_2, q_3) = R_z\left(\frac{\pi}{2} + q_2\right)T_x(l_2)R_z(q_3)T_x(l_3) =$$

$$= \begin{pmatrix} -s_2 & -c_2 & 0 & 0 \\ c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_3 & -s_3 & 0 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} -s_2 & -c_2 & 0 & -s_2 l_2 \\ c_2 & -s_2 & 0 & c_2 l_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_3 & -s_3 & 0 & c_3 l_3 \\ s_3 & c_3 & 0 & s_3 l_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} -s_{23} & -c_{23} & 0 & -s_2 l_2 - s_{23} l_3 \\ c_{23} & -s_{23} & 0 & c_2 l_2 + c_{23} l_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_5^3(q_4, q_5) = R_z\left(\frac{\pi}{2} + q_4\right)R_x\left(\frac{\pi}{2}\right)R_z(q_5)T_z(l_4) = \begin{pmatrix} -s_4 & -c_4 & 0 & 0 \\ c_4 & -s_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_5 & -s_5 & 0 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 1 & l_4 \\ 0 & 0 & 0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} -s_4 & 0 & c_4 & 0 \\ c_4 & 0 & s_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_5 & -s_5 & 0 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 1 & l_4 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} -s_4 c_5 & s_4 s_5 & c_4 & c_4 l_4 \\ c_4 c_5 & -c_4 s_5 & s_4 & s_4 l_4 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Οπότε,

$$\begin{aligned} \mathbf{A}_5^0(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5) &= \begin{pmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -s_{23} & -c_{23} & 0 & -s_2 l_2 - s_{23} l_3 \\ c_{23} & -s_{23} & 0 & c_2 l_2 + c_{23} l_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -s_4 c_5 & s_4 s_5 & c_4 & c_4 l_4 \\ c_4 c_5 & -c_4 s_5 & s_4 & s_4 l_4 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \\ & \begin{pmatrix} -c_1 s_{23} & -c_1 c_{23} & s_1 & -c_1 s_2 l_2 - c_1 s_{23} l_3 \\ -s_1 s_{23} & -s_1 c_{23} & -c_1 & -s_1 s_2 l_2 - s_1 s_{23} l_3 \\ c_{23} & -s_{23} & 0 & c_2 l_2 + c_{23} l_3 + l_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -s_4 c_5 & s_4 s_5 & c_4 & c_4 l_4 \\ c_4 c_5 & -c_4 s_5 & s_4 & s_4 l_4 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \\ & \begin{pmatrix} c_1 s_{23} s_4 c_5 - c_1 c_{23} c_4 c_5 + s_1 s_5 & -c_1 s_{23} s_4 s_5 + c_1 c_{23} c_4 s_5 + s_1 c_5 & -c_1 s_{234} & -c_1 s_2 l_2 - c_1 s_{23} l_3 - c_1 s_{234} l_4 \\ s_1 s_{23} s_4 c_5 - s_1 c_{23} c_4 c_5 - c_1 s_5 & -s_1 s_{23} s_4 s_5 + s_1 c_{23} c_4 s_5 - c_1 c_5 & -s_1 s_{234} & -s_1 s_2 l_2 - s_1 s_{23} l_3 - s_1 s_{234} l_4 \\ -s_{234} c_5 & s_{234} s_5 & c_{234} & c_2 l_2 + c_{23} l_3 + l_1 + c_{234} l_4 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

Συνεπώς, θα ισχύει για τον προσανατολισμό και την θέση του συστήματος συντεταγμένων του end-effector ως προς το σύστημα συντεταγμένων της βάσης του μηχανισμού ότι:

$$\mathbf{R}_E^0(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5) = \begin{pmatrix} c_1 s_{23} s_4 c_5 - c_1 c_{23} c_4 c_5 + s_1 s_5 & -c_1 s_{23} s_4 s_5 + c_1 c_{23} c_4 s_5 + s_1 c_5 & -c_1 s_{234} \\ s_1 s_{23} s_4 c_5 - s_1 c_{23} c_4 c_5 - c_1 s_5 & -s_1 s_{23} s_4 s_5 + s_1 c_{23} c_4 s_5 - c_1 c_5 & -s_1 s_{234} \\ -s_{234} c_5 & s_{234} s_5 & c_{234} \end{pmatrix}$$

Και

$$\mathbf{p}_E^0(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5) = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} = \begin{pmatrix} -c_1 s_2 l_2 - c_1 s_{23} l_3 - c_1 s_{234} l_4 \\ -s_1 s_2 l_2 - s_1 s_{23} l_3 - s_1 s_{234} l_4 \\ c_2 l_2 + c_{23} l_3 + l_1 + c_{234} l_4 \end{pmatrix}$$

Όπου, $c_{i...j} = \cos(q_i + \dots + q_j)$, $s_{i...j} = \sin(q_i + \dots + q_j)$

Και $\cos\left(q_i + \frac{\pi}{2}\right) = -\sin(q_i)$, $\sin\left(q_i + \frac{\pi}{2}\right) = \cos(q_i)$

Σχολιασμός:

Παρατηρείται ότι μόνο τα x , y εξαρτώνται από την γωνία q_1 της πρώτης άρθρωσης, το οποίο είναι αναμενόμενο αφού το ρομπότ στρέφεται κατά τον άξονα z . Επιπλέον, είναι φανερό ότι κάθε άρθρωση επηρεάζει κάθε επόμενη της οπότε κάθε αλλαγή σε μια άρθρωση μεταβάλλει και τις υπόλοιπες. Όμως, η τελευταία άρθρωση δεν επηρεάζει την θέση στο World Coordinates σύστημα αφού μεταβάλλει μόνο τον προσανατολισμό και όχι τη θέση του ρομπότ στο χώρο. Τέλος, στην αντίστροφη κινηματική ανάλυση, επειδή προκύπτουν περισσότερες από μια λύσεις, και επειδή ο βραχίονας μπορεί να προσεγγίσει μια θέση με τους 4 διαφορετικούς τρόπους που αναλύθηκαν, η ενδεδειγμένη επίλυση του συστήματος έχει παραληφθεί λόγω μεγάλης πολυπλοκότητας.

Ανάλυση & Περιγραφή Αλγορίθμου

Εδώ θα περιγραφεί ο αλγόριθμος που χρησιμοποιήθηκε για να εκτελεστεί το πείραμα στο εργαστήριο. Υπενθυμίζεται ότι στόχος του βραχίονα ήταν να μετακινήσει ορισμένα αντικείμενα και έπειτα να τους τοποθετήσει ετικέτες ανάλογα με τον τύπο του αντικειμένου. Ειδικότερα, υπήρχαν διαθέσιμα 6 κυλινδρικά αντικείμενα (2 μη-φορτισμένα, 2 φορτισμένα, 2 πλαστικά) και με την βοήθεια αισθητήρων (οπτικός, επαγωγικός, χωρητικός) γινόταν η αναγνώριση από τον υπολογιστή και τελικά ξεχώριζε τα αντικείμενα μεταξύ τους ώστε να τους βάλει τις σωστές ετικέτες και να τα τοποθετήσει σε κατάλληλες θέσεις έως ότου να μην υπάρχουν άλλα αντικείμενα στην στοίβα.

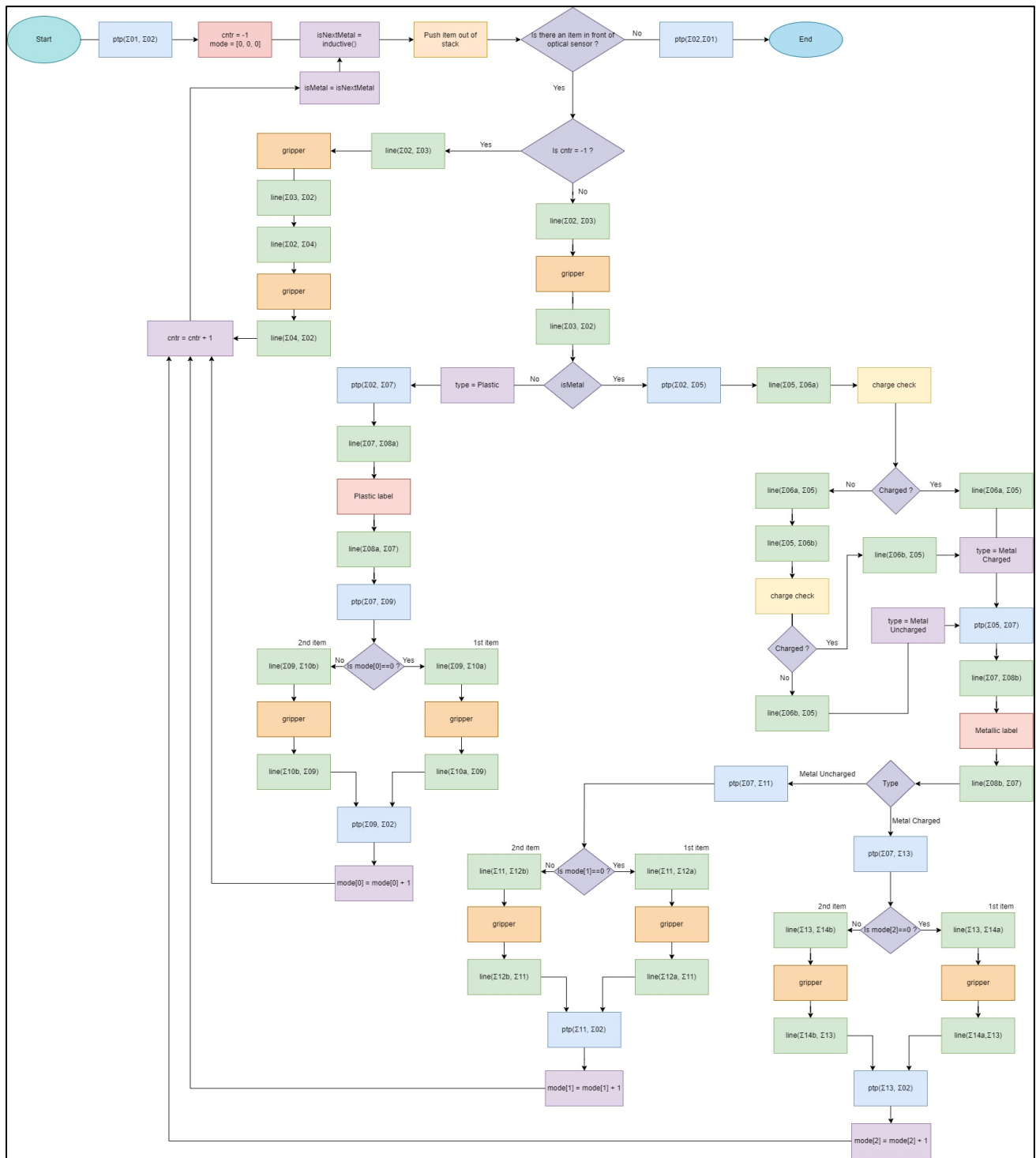
Συνοπτικά, ο αλγόριθμος πραγματοποιεί τα εξής βήματα:

Αρχικά, παίρνει το πρώτο αντικείμενο και το τοποθετεί στην κορυφή της στοίβας, αφού ο επαγωγικός αισθητήρας που αναγνωρίζει αν το στοιχείο είναι μεταλλικό ή όχι κοιτάει πάντα το δεύτερο στοιχείο.

Έπειτα, μέχρι να αδειάσει η στοίβα, εκτελούνται επαναληπτικά τα εξής βήματα.

Αν το αντικείμενο είναι πλαστικό, τότε τοποθέτησε ετικέτα και μετά τοποθέτησέ το στην κατάλληλη θέση με τα πλαστικά στοιχεία αλλιώς, αν είναι μεταλλικό τότε έλεγξε αν είναι φορτισμένο ή όχι, βάλε του ετικέτα και τέλος, τοποθέτησέ το στην κατάλληλη θέση. Σημειώνεται μάλιστα ότι στο σημείο 6 είτε κινείται με την λαβίδα από την κανονική μεριά είτε την αντίστροφη ώστε να εξεταστεί η πολικότητα του μεταλλικού αντικειμένου και από τις δύο μεριές, αφού ο χωρητικός αισθητήρας έχει καθορισμένη πολικότητα και για αυτό τον λόγο στο πρόγραμμα θα υπάρχουν τα σημεία 6a, 6b τα οποία σχετίζονται με τον προσανατολισμό του εργαλείου του βραχίονα.

Στην συνέχεια, φαίνεται το διάγραμμα ροής:



Παρακάτω παρατίθεται ο ψευδοκώδικας:

```
# Robotics I
# Lab #1: Robotic cell

# Pseudo-code

# below there are the implementations of the necessary functions

# this function picks up the items from in front of the optical sensor

func pick():
    # ptp() is the function that implements the point-to-point method
    # line() is the function that implements the line method
    line( $\Sigma 02$ ,  $\Sigma 03$ )
    # gripper() the function that implements the picking/releasing of the gripper
    gripper()
    line( $\Sigma 03$ ,  $\Sigma 02$ )

# this function checks if a metal object is charged

func isCharged():
    line( $\Sigma 05$ ,  $\Sigma 06a$ )
    # capacitive sensor; True if charged, else False
    if capacitive():
        line( $\Sigma 06a$ ,  $\Sigma 05$ )
        return True
    line( $\Sigma 06a$ ,  $\Sigma 05$ )
    line( $\Sigma 05$ ,  $\Sigma 06b$ )
    if capacitive():
        line( $\Sigma 06b$ ,  $\Sigma 05$ )
        return True
    line( $\Sigma 06b$ ,  $\Sigma 05$ )
    return False

# this function puts labels on the items according to their type

func label(type):
    if type == 0:
        line( $\Sigma 07$ ,  $\Sigma 08a$ )
        plastic_label()
        line( $\Sigma 08a$ ,  $\Sigma 07$ )
    else:
        line( $\Sigma 07$ ,  $\Sigma 08b$ )
        metallic_label()
        line( $\Sigma 08b$ ,  $\Sigma 07$ )
```

```
# this function places items in the correct order
```

```
func place(type, mode):  
    # plastic  
    if type == 0:  
        ptp( $\Sigma$ 07,  $\Sigma$ 09)  
        if mode[type] == 0:  
            line( $\Sigma$ 09,  $\Sigma$ 10a)  
            gripper()  
            line( $\Sigma$ 10a,  $\Sigma$ 09)  
        elif mode[type] == 1:  
            line( $\Sigma$ 09,  $\Sigma$ 10b)  
            gripper()  
            line( $\Sigma$ 10b,  $\Sigma$ 09)  
        ptp( $\Sigma$ 09,  $\Sigma$ 02)  
  
    # non-charged  
    if type == 1:  
        ptp( $\Sigma$ 07,  $\Sigma$ 11)  
        if mode[type] == 0:  
            line( $\Sigma$ 11,  $\Sigma$ 12a)  
            gripper()  
            line( $\Sigma$ 12a,  $\Sigma$ 11)  
        elif mode[type] == 1:  
            line( $\Sigma$ 11,  $\Sigma$ 12b)  
            gripper()  
            line( $\Sigma$ 12b,  $\Sigma$ 11)  
        ptp( $\Sigma$ 11,  $\Sigma$ 02)  
  
    # charged  
    if type == 2:  
        ptp( $\Sigma$ 07,  $\Sigma$ 13)  
        if mode[type] == 0:  
            line( $\Sigma$ 13,  $\Sigma$ 14a)  
            gripper()  
            line( $\Sigma$ 14a,  $\Sigma$ 13)  
        elif mode[type] == 1:  
            line( $\Sigma$ 13,  $\Sigma$ 14b)  
            gripper()  
            line( $\Sigma$ 14b,  $\Sigma$ 13)  
        ptp( $\Sigma$ 13,  $\Sigma$ 02)  
  
    mode[type] = mode[type] + 1  
  
    return mode
```

```

# main function

func main():
    # initialization
    ptp( $\Sigma$ 01,  $\Sigma$ 02)
    cntr = -1
    #Type 0: plastic, 1: non-charged, 2: charged
    mode = [0, 0, 0]
    isNextMetal = inductive() # Describes inductive sensor; True if metal,else False
    push_item() # Pushes bottom item from stack

    # optical() describes the optical sensor; True if there is an object, else False
    while optical():
        # initially, pick the first item and put it back in the stack
        if cntr = -1:
            pick()
            line( $\Sigma$ 02,  $\Sigma$ 04)
            gripper()
            line( $\Sigma$ 04,  $\Sigma$ 02)

        # then handle the rest of the items
        else:
            pick()
            if isMetal:
                ptp( $\Sigma$ 02,  $\Sigma$ 05)
                if isCharged():
                    type = 2
                else:
                    type = 1
                ptp( $\Sigma$ 05,  $\Sigma$ 07)
            else:
                type = 0
                ptp( $\Sigma$ 02,  $\Sigma$ 07)
            label(type)
            mode = place(type, mode)

            cntr = cntr + 1
            isMetal = isNextMetal
            isNextMetal = inductive()
            push_item()

    ptp( $\Sigma$ 02,  $\Sigma$ 01)
    print("Items processed: ", cntr)

```

Resources

- [1] Υλικό & Σημειώσεις μαθήματος
- [2] John J. Craig, Introduction to Robotics: Mechanics and Control (2009)