

Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

## Ρομποτική 1

### Εργαστηριακή Άσκηση 5

Niryo Ned2

Ομάδα A3-4	Νικόλαος Αδαμόπουλος	03122074
	Λεωνίδας Λαγομιτζής	03121061
	Ηλίας-Στυλιανός Ηλιάδης	03122140
	Ηλίας Πουλόπουλος	03122160
	Ανδρέας Αγγελόπουλος	03122049

## Περιεχόμενα

<b>1 Εισαγωγή</b>	<b>2</b>
<b>2 1o Μέρος</b>	<b>2</b>
<b>3 2o Μέρος (α) - Pick and Place</b>	<b>2</b>
<b>4 2o Μέρος (β) - Color Based Pick and Place</b>	<b>4</b>
4.1 Σκοπός της Άσκησης . . . . .	4
4.2 Μεθοδολογία και Αλγόριθμος Όρασης . . . . .	4
4.3 Διαδικασία Calibration και Εκτέλεσης . . . . .	4
4.4 Σχολιασμός Κώδικα . . . . .	5

## 1 Εισαγωγή

## 2 1o Μέρος

## 3 2o Μέρος (α) - Pick and Place

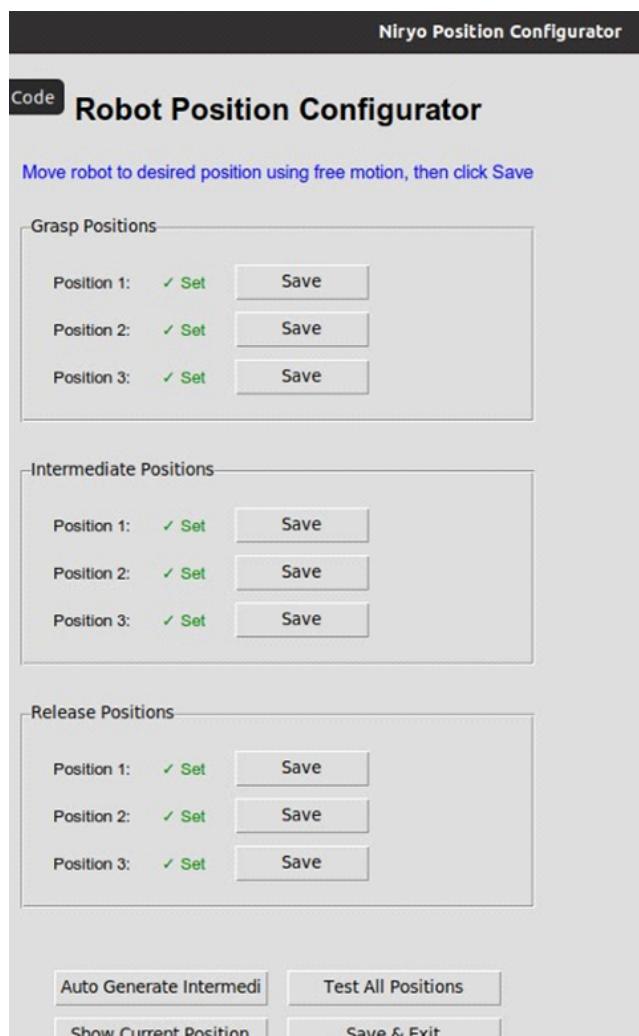
Σε αυτό το βήμα εργαστήκαμε με το free motion του ρομποτικού βραχίονα, με σκοπό να ορίσουμε τρεις βασικές θέσεις για καθένα από τα τρία αντικείμενα. Η πρώτη θέση αντιστοιχεί στο σημείο όπου ξεκινά το άνοιγμα του βραχίονα, η δεύτερη στο σημείο όπου αρχίζει το κλείσιμό του για την αρπαγή του αντικειμένου και η τρίτη στη θέση όπου ο ρομποτικός βραχίονας αφήνει το αντικείμενο. Για τη διευκόλυνση της διαδικασίας αυτής, έχει δημιουργηθεί ένα βοηθητικό αρχείο που επιτρέπει την αποθήκευση ενδιάμεσων θέσεων του ρομποτικού βραχίονα, προσφέροντας μεγαλύτερη ακρίβεια και ευελιξία στον χειρισμό του. Επιπλέον, η άσκηση παρέχει τη δυνατότητα χρήσης ενός αλγορίθμου αναγνώρισης σχημάτων, ο οποίος επιτρέπει τον εντοπισμό αντικειμένων με κυκλική μορφή. Οι χρήστες έχουν τη δυνατότητα να αλλάξουν τη σειρά με την οποία ο ρομποτικός βραχίονας συλλέγει τα αντικείμενα, καθώς και να ορίσουν τα σημεία στα οποία αυτά τοποθετούνται. Τέλος, υποστηρίζεται η πλήρως αυτοματοποιημένη εκτέλεση της διαδικασίας, γεγονός που καθιστά την άσκηση ιδιαίτερα ευέλικτη και κατάλληλη για εκπαιδευτικούς σκοπούς και πειραματισμό στον προγραμματισμό κινήσεων και στον χειρισμό αντικειμένων.

Η πρώτη εικόνα παρουσιάζει το κεντρικό μενού ελέγχου του ρομποτικού συστήματος Niryo, μέσω τερματικού. Ο χρήστης μπορεί να επιλέξει μεταξύ διαφορετικών λειτουργιών, όπως ζωντανή προβολή κάμερας, ανίχνευση κυκλικών αντικειμένων, χειροκίνητη ή αυτοματοποιημένη διαδικασία pick and place, καθώς και έξοδο από το πρόγραμμα.

Η δεύτερη εικόνα απεικονίζει το γραφικό περιβάλλον διαμόρφωσης θέσεων του ρομποτικού βραχίονα (Robot Position Configurator). Μέσα από αυτό το περιβάλλον, ο χρήστης μπορεί να αποθηκεύσει θέσεις σύλληψης, ενδιάμεσες θέσεις και θέσεις απελευθέρωσης αντικειμένων, χρησιμοποιώντας free motion. Επιπλέον, παρέχονται επιλογές για έλεγχο των αποθηκευμένων θέσεων και αυτόματη δημιουργία ενδιάμεσων κινήσεων.

```
=====
NIRYO ROBOT MENU
=====
1. Live Camera View
2. Detect Circles (Single Shot)
3. Test Pick and Place (Manual)
4. Automated Pick and Place
5. Exit
=====
Enter your choice (1-5): 
```

Σχήμα 1: Κεντρικό μενού ελέγχου του ρομποτικού συστήματος Niryo (τερματικό): επιλογές για ζωντανή προβολή κάμερας, ανίχνευση κυκλικών αντικειμένων, χειροκίνητη ή αυτοματοποιημένη διαδικασία pick and place και έξοδο.



Σχήμα 2: Γραφικό περιβάλλον «Robot Position Configurator» για αποθήκευση θέσεων σύλληψης, ενδιάμεσων θέσεων και θέσεων απελευθέρωσης, με επιλογές ελέγχου και αυτόματης δημιουργίας ενδιάμεσων κινήσεων.

## 4 2o Μέρος (β) - Color Based Pick and Place

### 4.1 Σκοπός της Άσκησης

Στο τελικό μέρος της εργαστηριακής άσκησης, στόχος ήταν η υλοποίηση μιας εφαρμογής διαλογής αντικειμένων (sorting) βασισμένης στο χρώμα (Color Based Pick-and-Place). Σε αντίθεση με το προηγούμενο μέρος όπου οι θέσεις ήταν σταθερές και προκαθορισμένες, εδώ ο ρομποτικός βραχίονας καλείται να εντοπίσει δυναμικά τα αντικείμενα στον χώρο εργασίας μέσω της κάμερας, να αναγνωρίσει το χρώμα τους και να τα τοποθετήσει στις αντίστοιχες θέσεις εναπόθεσης.

### 4.2 Μεθοδολογία και Αλγόριθμος Όρασης

Η διαδικασία της όρασης βασίστηκε στην επεξεργασία εικόνας που λαμβάνεται από την κάμερα του Niryo Ned2. Συγκεκριμένα, ακολουθήθηκαν τα εξής βήματα:

- Λήψη και Διόρθωση Εικόνας:** Ο βραχίονας μετακινείται σε θέση παρατήρησης (observation pose) για να λάβει εικόνα του χώρου εργασίας. Εφαρμόζεται διόρθωση για τις παραμορφώσεις του φακού της κάμερας.
- Χρωματικός Διαχωρισμός (HSV):** Για την ανίχνευση των αντικειμένων (Κόκκινο, Πράσινο, Μπλε) χρησιμοποιήθηκε ο χρωματικός χώρος HSV (Hue, Saturation, Value). Μέσω ειδικού εργαλείου με μπάρες κύλισης (trackbars), ρυθμίστηκαν τα κατώφλια (thresholds) για κάθε χρώμα ώστε να δημιουργηθούν οι κατάλληλες μάσκες απομόνωσης.
- Ανίχνευση Σχήματος:** Πέρα από το χρώμα, ο αλγόριθμος φίλτράρει τα αντικείμενα με βάση το εμβαδόν (area) και την κυκλικότητα (circularity) για να επιβεβαιώσει ότι πρόκειται για τους επιθυμητούς δίσκους.

### 4.3 Διαδικασία Calibration και Εκτέλεσης

Για τη σύνδεση του συστήματος όρασης με τον φυσικό χώρο, χρησιμοποιήθηκε μια διαδικασία βαθμονόμησης (calibration). Τοποθετήθηκε ένα πρότυπο (template) με τρεις κύκλους στον χώρο εργασίας για να καθοριστούν οι ονομαστικές θέσεις αρπαγής (nominal grasp positions).

Κατά την εκτέλεση του σεναρίου:

- Ο αλγόριθμος αντιστοιχίζει κάθε ανιχνευμένο κύκλο σε μια από τις προκαθορισμένες θέσεις αρπαγής.
- Το ρομπότ εκτελεί την ακολουθία κινήσεων: *Ενδιάμεση θέση → Αρπαγή → Ενδιάμεση θέση → Απελευθέρωση*, τοποθετώντας κάθε αντικείμενο στη θέση που αντιστοιχεί στο χρώμα του.
- Η διαδικασία επαναλαμβάνεται μέχρι να ταξινομηθούν όλα τα αντικείμενα που εντοπίστηκαν στον χώρο.

Αξίζει να σημειωθεί ότι το σύστημα διαθέτει ευελιξία, επιτρέποντας την αλλαγή των θέσεων εναπόθεσης ή τη διαχείριση διαφορετικού αριθμού αντικειμένων, προσαρμόζοντας τη συμπεριφορά του ρομπότ δυναμικά.

#### 4.4 Σχολιασμός Κώδικα

Η υλοποίηση της άσκησης βασίστηκε σε δύο κύρια προγράμματα Python (`color_mask_test.py` και `task3.py`). Από την ανάλυση του κώδικα προκύπτουν οι εξής παρατηρήσεις:

- **Δυναμική Ρύθμιση Μάσκας:** Για τον προσδιορισμό των βέλτιστων ορίων HSV, χρησιμοποιήθηκε το βοηθητικό σκριπτ `color_mask_test.py`. Αυτό παρέχει γραφικό περιβάλλον με trackbars για τη ρύθμιση των τιμών  $H, S, V$  (Low/High) και των παραμέτρων μορφολογίας (Kernel, Iterations) σε πραγματικό χρόνο, αποθηκεύοντας τα αποτελέσματα στο αρχείο `color_mask_config.json`.
- **Αλγόριθμος Φιλτραρίσματος:** Για την εξάλειψη των false positives, ο αλγόριθμος στο `task3.py` εφαρμόζει τα δύο γεωμετρικά κριτήρια που αναφέραμε και στην μεθοδολογία:
  - **Εμβαδόν (Area):** Τα αντικείμενα γίνονται δεκτά μόνο αν καλύπτουν περιοχή μεταξύ 200 και 12.000 pixels.
  - **Κυκλικότητα (Circularity):** Ελέγχεται η σχέση  $\frac{4\pi \cdot \text{Area}}{\text{Perimeter}^2} > 0.55$ , ώστε να απορρίπτονται αντικείμενα που δεν είναι επαρκώς κυκλικά.
- **Ταξινόμηση Χρώματος:** Η διαδικασία αναγνώρισης ακολουθεί προσέγγιση δύο σταδίων. Αρχικά εφαρμόζεται η γενική μάσκα για τον εντοπισμό όλων των υποψήφιων αντικειμένων. Στη συνέχεια, για κάθε εντοπισμένο αντικείμενο, λαμβάνεται ένα δείγμα (patch  $5 \times 5$  pixels) από το κέντρο του και υπολογίζεται η μέση τιμή HSV, η οποία συγκρίνεται με προκαθορισμένα εύρη (`self.color_ranges`) για την τελική κατάταξη σε Κόκκινο, Πράσινο ή Μπλε.
- **Αντιστοίχιση Θέσεων (Positional Mapping):** Κατά τη διαδικασία Matching, εφαρμόζεται αντιστροφή της σειράς (`position_mapping = [2, 1, 0]`), ώστε να εναρμονιστεί η διάταξη των αντικειμένων όπως τα αντιλαμβάνεται η κάμερα (αριστερά-προς-δεξιά) με την αρίθμηση των θέσεων αρπαγής του ρομπότ (`grasp_1, grasp_2, grasp_3`).