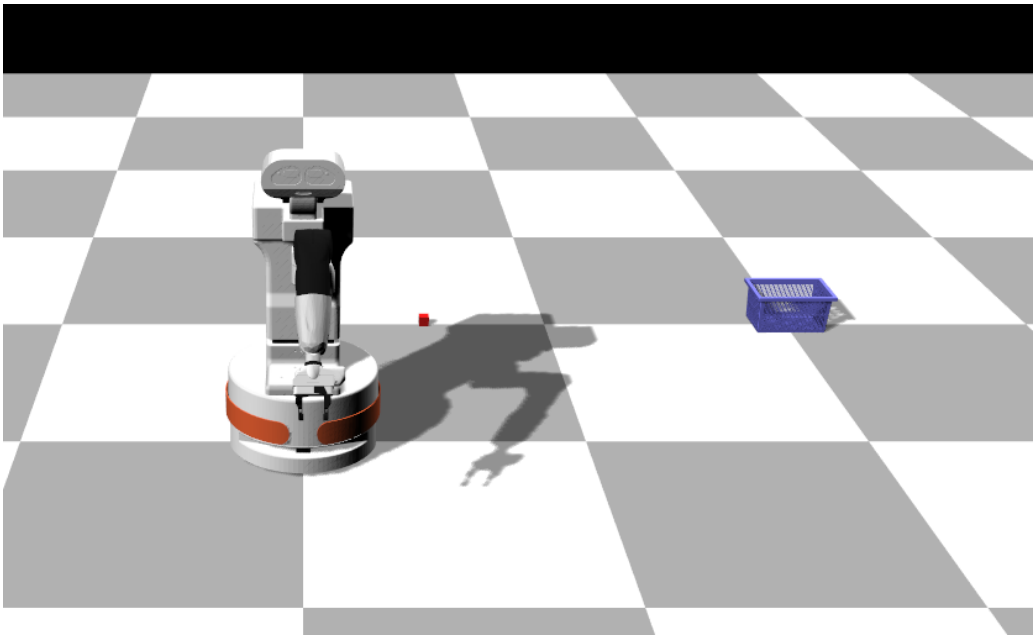
Τεχνολογίες Ευφυών Συστημάτων και Ρομποτική 2021-22
Εργασία Ρομποτικής

Διδάσκων: Κωνσταντίνος Χατζηλυγερούδης (costashatz@upatras.gr)
Κώδικας: Κωνσταντίνος Τσίγγανος (kontsinganos@gmail.com)

Προθεσμία υποβολής: 1 Ιουλίου 2022

1 Γενικά

Σκοπός της εργασίας είναι να δημιουργήσετε ένα ελεγκτή που θα επιτρέπει στο ρομπότ Tiago (από την PAL Robotics) να λύνει ένα "Pick and Place" σενάριο όπως φαίνεται στην παρακάτω εικόνα. Το ρομπότ θα πρέπει να είναι σε θέση να πιάνει τον κύβο από οποιαδήποτε αρχική θέση και να το τοποθετεί στο καλάθι, το οποίο μπορεί να βρίσκεται οπουδήποτε στο χώρο.



Σχήμα 1: "Pick and Place" σενάριο

Θα σας δοθούν συγκεκριμένες θέσεις για τον κύβο και το καλάθι που καλύπτουν ένα μεγάλο χώρο. Σκοπός είναι να υλοποίησετε έναν ελεγκτή που θα συνδυάζει Behavior Trees με έναν low-level task

space controller. Υπάρχουν οι παρακάτω περιορισμοί:

- Θα σας δοθούν συγκεκριμένες αρχικές θέσεις του ρομπότ και των αντικειμένων, οι οποίες δεν θα πρέπει να τροποποιηθούν
- Θα σας δοθούν περιγραφές του κόσμου και των αντικειμένων ως URDF αρχεία, τα οποία δεν θα πρέπει να τροποποιηθούν
- Το ρομπότ θα ελέγχεται είτε με κινητήρες servo είτε με κινητήρες torque. Εξαίρεση αποτελεί η βάση του ρομπότ (δείτε οδηγίες παρακάτω)
- Δεν επιτρέπεται η χρήση οποιασδήποτε βιβλιοθήκης που να υπολογίζει ελεγκτές (θα πρέπει να φτιάξετε τον ελεγκτή μόνοι σας). Για τα Behavior Trees επιτρέπεται η χρήση των βιβλιοθηκών (προαιρετικά): `py_trees` (python) και `BehaviorTree.CPP` (C++).

2 Περιβάλλον Προσομοίωσης

Θα πρέπει να χρησιμοποιήσετε το περιβάλλον προσομοίωσης robot_dart. **Βεβαιωθείτε ότι έχετε το τελευταίο commit (τουλάχιστον 11/05/2022).** Εφόσον έχετε αρχικοποιήσει σωστά τα αντικείμενα θα πρέπει να εμφανίζονται όπως στο Σχήμα 1. Σας δίνονται αρχεία κώδικα τόσο σε C++ όσο και σε Python τα οποία μπορείτε να χρησιμοποιήσετε ως βάση για την λύση σας.

Δίνονται τα παρακάτω αρχεία κώδικα που περιέχουν τις απαραίτητες συναρτήσεις:

- `utils.py` ή `utils.hpp`
  - Ο κώδικας περιέχει την συνάρτηση `box_into_basket(box_translation, basket_translation, basket_angle)`
  - Η συνάρτηση δέχεται ως είσοδο τις θέσεις του κύβου και του καλαθιού και επιτρέφει True αν ο κύβος βρίσκεται μέσα στο καλάθι
- `tiago_pick_place.py` ή `tiago_pick_place.cpp`
  - Ο κώδικας περιέχει μία βάση για να ξεκινήσετε να λύνετε το πρόβλημα
  - Δεν είναι απαραίτητο να τον ακολουθήσετε πιστά

Αν επιθυμείτε να χρησιμοποιήσετε `C++`, έχουμε φτιάξει ένα περιβάλλον που θα κάνει compile τον κώδικα. Πρέπει να τρέξετε τις εντολές: `./waf configure` (μία φορά) και `./waf` κάθε φορά που αλλάζετε ένα αρχείο για να ξανα-τρέξει το compilation.

Η προσομοίωση θα πρέπει να έχει υποχρεωτικά τα παρακάτω χαρακτηριστικά:

- Βήμα προσομοίωσης (timestep): `0.001 s`
- Μηχανή ανίχνευσης συγκρούσεων (collision engine): `bullet`
- Μέγεθος κύβου: $0.04 \times 0.04 \times 0.04$
- Όλοι οι κινητήρες εκτός από τον βραχίονα θα πρέπει να είναι servo (ακόμα και οι πρώτοι 6 βαθμοί ελευθερίας). Για τον βραχίονα μπορείτε να χρησιμοποιήσετε είτε servo είτε torque
- Από τους πρώτους 6 βαθμούς ελευθερίας επιτρέπεται να ελέγχετε μόνο τους `['rootJoint_rot_z', 'rootJoint_pos_x', 'rootJoint_pos_y']`

3 Παραδοτέα

Θα πρέπει να παραδοθούν τουλάχιστον τα παρακάτω:

- Κώδικας σε `python` ή `C++` που θα υλοποιεί τον ελεγκτή και θα τον εφαρμόζει για 50 τυχαίες αρχικές θέσεις (από αυτές που δίνονται)
- Αναφορά με λεπτομερή περιγραφή του ελεγκτή και της υλοποίησης