

Εργασία

Θέμα 1 (50%)

Ο στόχος είναι ο σχεδιασμός επιθυμητών τροχιών και ο προγραμματισμός της κίνησης ενός τροχοφόρου ρομπότ διαφορετικής κίνησης, το Jackal της Clearpath, (α) σε περιβάλλον προσομοίωσης και (β) σε πραγματικές συνθήκες. Το περιβάλλον προσομοίωσης, θα δημιουργηθεί με την χρήση των λογισμικών πακέτων ROS, RViz ή/και Gazebo. Το Jackal απεικονίζεται στο περιβάλλον του Gazebo στο Σχήμα 1(α'), ενώ το πραγματικό ρομπότ απεικονίζεται στο εργαστήριο στο Σχήμα 1(β'). Ακολουθούν οδηγίες εγκατάστασης για ROS 1.



(α')



(β')

Σχήμα 1: (α') Το τροχοφόρο ρομπότ Jackal στο Gazebo. (β') Το τροχοφόρο ρομπότ Jackal με τον ρομποτικό βραχίονα Gen3 Lite της Kínova στο εργαστήριο.

Οδηγίες εγκατάστασης (ROS 1 Noetic)

Καταρχάς, υποθέτουμε ότι έχετε εγκαταστήσει τα Ubuntu 20.04 και στη συνέχεια το ROS 1, συγκεκριμένα το πακέτο `ros-noetic-desktop-full`, και ότι ο φάκελος εργασίας είναι ο `~/catkin_ws`. Αναλυτικές οδηγίες και tutorials θα βρείτε στην ιστοσελίδα του ROS. Επίσης, πρέπει να εγκαταστήσετε (αν δεν είναι εγκατεστημένα) κάποια απαραίτητα πακέτα με τις εντολές

- `sudo apt install ros-noetic-jackal-simulator`
- `sudo apt install ros-noetic-jackal-desktop`
- `sudo apt install ros-noetic-jackal-navigation`



Αν οι παραπάνω εντολές έχουν εκτελεσθεί χωρίς σφάλματα, μπορείτε να δείτε το ρομπότ στο Gazebo με την εντολή

- `roslaunch jackal_gazebo empty_world.launch`

και με το λογισμικό RViz (εντολή `rviz`). Στο RViz θα πρέπει να επιλέξετε ως Fixed Frame το `odom`, και να προσθέσετε (Add) το `RobotModel`.

Το ρομπότ μπορεί να κινηθεί δίνοντας εντολές ταχύτητας μέσω του topic `/jackal_velocity_controller/cmd_vel`.

Η θέση και ο προσανατολισμός του, καθώς και η ταχύτητά του, υπάρχουν στο topic `/odometry/filtered`. Δοκιμάστε να κινήσετε το ρομπότ με την εντολή

```
rostopic pub /jackal_velocity_controller/cmd_vel geometry_msgs/Twist -r 3 '[0.0,0.0,0.0]' '[0.0, 0.0, 0.3]'
```

η οποία περιστρέφει το ρομπότ με περιστροφική ταχύτητα 0.3 rad/s . Βεβαιωθείτε ότι μπορείτε να κινήσετε το ρομπότ δίνοντας διάφορες εντολές ταχύτητας μέσω του κατάλληλου topic.

Προσοχή : Ενώ υπάρχουν πολλά έτοιμα προγράμματα οδήγησης και ελέγχου του ρομπότ (το ROS δίνει αρκετές δυνατότητες), η εργασία θα υλοποιηθεί δίνοντας εντολές ταχύτητας στο ρομπότ μέσω του topic `*/cmd_vel` **αποκλειστικά**. Ο λόγος είναι ότι θα πρέπει εσείς να σχεδιάσετε και να γράψετε τον κώδικα κίνησης του ρομπότ. **Έτοιμοι controllers δεν γίνονται δεκτοί.**

Προβλήματα θέματος 1

1. Γράψτε πρόγραμμα (node), το οποίο να κινεί το τροχοφόρο ρομπότ με αυθαίρετη επιθυμητή γραμμική, v_d , και περιστροφική ταχύτητα, ω_d , εκφρασμένες ως προς το σωματόδετο ΣΣ. Αποθηκεύστε (χρησιμοποιήστε το πακέτο `rosbag`) τις τροχιές των μετατοπίσεων/περιστροφών και των ταχυτήτων (γραμμική και γωνιακή) για τις τιμές του Πίνακα 1.

Πίνακας 1: Τιμές για το ερώτημα 1.

Ταχύτητα $[v_d, \omega_d]^T$	Διάρκεια
$[0 \text{ m/s}, -15^\circ/\text{s}]^T$	10 s
$[0.1 \text{ m/s}, 10^\circ/\text{s}]^T$	15 s

2. Η αρχική θέση, (x_0, y_0) , και προσανατολισμός, θ_0 , του ρομπότ είναι δοσμένα. Σχεδιάστε τρεις διαφορετικές τροχιές:

- (α') Η πρώτη τροχιά θα περιγράφει την περιστροφή του ρομπότ, έτσι ώστε να "κοιτάξει" μια δοσμένη τελική θέση, (x_f, y_f) .
- (β') Στη συνέχεια, η δεύτερη τροχιά θα περιγράφει την γραμμική κίνηση του ρομπότ, έτσι ώστε να φτάσει στην δοσμένη τελική θέση.
- (γ') Αφού το ρομπότ φτάσει στην τελική θέση, η τρίτη τροχιά θα περιγράφει την περιστροφή του, έτσι ώστε να αποκτήσει έναν δοσμένο τελικό προσανατολισμό, θ_f .



Χρησιμοποιήστε μια μέθοδο και αντίστοιχες παραμέτρους, σύμφωνα με τον Πίνακα 2, όπου AM, είναι ο **μεγαλύτερος AM** των μελών της κάθε ομάδας. Επιλέξτε τη διάρκεια κάθε τμήματος, t_f , έτσι ώστε η ταχύτητα να μην υπερβαίνει την μέγιστη τιμή της. Για την μέθοδο γραμμικών συναρτήσεων με παραβολικά τμήματα, υποθέστε ότι το t_b είναι ίσο με το 10% του t_f .

Πίνακας 2: Μέθοδοι και παράμετροι σχεδιασμού τροχιάς για το θέμα 1.

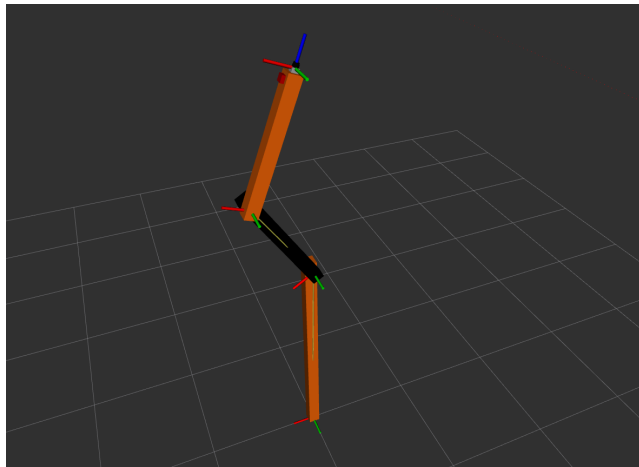
Μέθοδος/Παράμετρος	Τιμή
Κυβικά πολυώνυμα	Άρτιος AM (ο μεγαλύτερος της ομάδας)
Γραμμικές συναρτήσεις με παραβολικά τμήματα	Περιττός AM (ο μεγαλύτερος της ομάδας)
Αρχική θέση και προσανατολισμός	$\mathbf{q}_0 = \begin{bmatrix} x_0 \\ y_0 \\ \vartheta_0 \end{bmatrix} = \begin{bmatrix} 0 \text{ m} \\ 0 \text{ m} \\ 0 \text{ rad} \end{bmatrix}$
Τελική θέση και προσανατολισμός	$\mathbf{q}_f = \begin{bmatrix} x_f \\ y_f \\ \vartheta_f \end{bmatrix} = \begin{bmatrix} \text{round}\left(\frac{\text{AM}}{1000}\right) \text{ m} \\ \text{round}\left(\frac{\text{AM}}{2000}\right) \text{ m} \\ \left(\frac{\text{AM}}{3500}\right) \text{ rad} \end{bmatrix}$
Μέγιστη γραμμική ταχύτητα	0.2 m/s
Μέγιστη γωνιακή ταχύτητα	30°/s

3. Γράψτε πρόγραμμα (node), το οποίο να υλοποιεί τις τροχιές που σχεδιάσατε στο προηγούμενο ερώτημα. Στον κώδικα που θα γράψετε, η κίνηση του ρομπότ θα υλοποιείται δίνοντας απευθείας εντολές ταχύτητας στο ρομπότ μέσω του topic `*/cmd_vel`. **Καμιά άλλη μέθοδος δεν θα γίνει δεκτή.** Δοκιμάστε με διάφορα σημεία για να εξετάσετε την ορθότητα του κώδικα.

Αποθηκεύστε (χρησιμοποιήστε το πακέτο `rosbag`) διαγράμματα με τις επιθυμητές τροχιές θέσης/προσανατολισμού και ταχύτητας που σχεδιάσατε, καθώς και αυτές που πραγματοποίησε το ρομπότ.

Θέμα 2 (50%)

Ο στόχος είναι ο προγραμματισμός της κίνησης ενός ρομποτικού βραχίονα σε περιβάλλον προσομοίωσης, δες Σχήμα 2), έτσι ώστε να κινηθεί σύμφωνα με προσχεδιασμένες τροχιές. Το περιβάλλον προσομοίωσης, θα δημιουργηθεί με την χρήση των λογισμικών πακέτων ROS, RViz ή/και Gazebo. Ακολουθούν οδηγίες εγκατάστασης για ROS 1.



Σχήμα 2: Ο βραχίονας rrbot (Revolute-Revolute Manipulator Robot) στο RViz.

Οδηγίες εγκατάστασης (ROS 1 Noetic)

Καταρχάς, υποθέτουμε ότι έχετε εγκαταστήσει τα Ubuntu 20.04 και στη συνέχεια το ROS 1, συγκεκριμένα το πακέτο `ros-noetic-desktop-full`, και ότι ο φάκελος εργασίας είναι ο `~/catkin_ws`. Αναλυτικές οδηγίες και tutorials θα βρείτε στην ιστοσελίδα του ROS. Επίσης, πρέπει να εγκαταστήσετε (αν δεν είναι εγκατεστημένα) κάποια απαραίτητα πακέτα με τις εντολές

- `sudo apt-get install ros-noetic-ros-control`
- `sudo apt-get install ros-noetic-ros-controllers`
- `sudo apt-get install ros-noetic-gazebo-ros-control`

Θα χρησιμοποιήσουμε τον ρομποτικό βραχίονα δύο βαθμών ελευθερίας, και το αντίστοιχο πακέτο, που περιγράφεται εδώ. Μπείτε στον φάκελο `~/catkin_ws/src` και εκτελέστε

- `git clone https://github.com/lyleokoth/rrbot`

Τέλος, μπείτε στον φάκελο `~/catkin_ws` και εκτελέστε την εντολή

- `catkin_make`

Αν οι παραπάνω εντολές έχουν εκτελεσθεί χωρίς σφάλματα, μπορείτε να δείτε το ρομπότ με την εντολή

• `roslaunch rrbot_control rrbot_control.launch`

Δοκιμάστε να κινήσετε το ρομπότ με την εντολή

```
rostopic pub /rrbot/joint1_position_controller/command std_msgs/Float64  
"data: 0.7"
```

η οποία περιστρέφει την πρώτη άρθρωση του βραχίονα κατά 0.7 rad. Βεβαιωθείτε ότι μπορείτε να κινήσετε και τις δύο αρθρώσεις του ρομπότ δίνοντας διάφορες εντολές περιστροφής των αρθρώσεων μέσω των κατάλληλων topics. Οι γωνίες και οι γωνιακές ταχύτητες των αρθρώσεων, υπάρχουν στο topic `*/joint_states`.

Σημείωση: Η τοποθέτηση των συστημάτων συντεταγμένων δεν ακολουθεί τους κανόνες που έχουμε πει στο μάθημα. Για παράδειγμα, οι άξονες περιστροφής είναι οι Y και όχι οι Z. Επίσης, όταν οι αρθρώσεις έχουν και οι δύο τιμή 0, ο βραχίονας είναι σε κατακόρυφη θέση. Αυτό όμως δεν αλλάζει σε κάτι την ουσία της υλοποίησης.

Προβλήματα θέματος 2

1. Γράψτε πρόγραμμα (node), το οποίο να περιστρέφει τις αρθρώσεις του ρομποτικού βραχίονα, q_1 και q_2 , σε επιθυμητές γωνίες. Αποθηκεύστε (χρησιμοποιήστε το πακέτο **rosbag**) διαγράμματα με τις τροχιές των μεταβλητών των αρθρώσεων, καθώς και των ταχυτήτων τους, για τις τιμές του Πίνακα 3.

Πίνακας 3: Τιμές για το ερώτημα 1.

Μεταβλητή άρθρωσης	Αρχική γωνία	Τελική επιθυμητή γωνία
q_1	0°	25°
q_2	0°	-30°

2. Σχεδιάστε τροχιές για τις αρθρώσεις, έτσι ώστε ταυτόχρονα να ξεκινούν από τις αρχικές γωνίες και να σταματούν σε κάποιες τελικές επιθυμητές γωνίες. Χρησιμοποιήστε μια μέθοδο και αντίστοιχες παραμέτρους, σύμφωνα με τον Πίνακα 4, όπου AM, είναι ο **μεγαλύτερος AM** των μελών της κάθε ομάδας. Επιλέξτε τη διάρκεια της κίνησης, t_f , ίδια και για τις δύο αρθρώσεις, έτσι ώστε η γωνιακή ταχύτητα κάθε άρθρωσης να μην υπερβαίνει την μέγιστη τιμή της. Για την μέθοδο γραμμικών συναρτήσεων με παραβολικά τμήματα, υποθέστε ότι το t_b είναι ίσο με το 10% του t_f .
3. Γράψτε πρόγραμμα (node), το οποίο να κινεί το ρομπότ σύμφωνα με τις τροχιές που σχεδιάσατε στο προηγούμενο πρόβλημα. Στον κώδικα που θα γράψετε, η κίνηση του ρομπότ θα υλοποιείται δίνοντας εντολές γωνίας στις αρθρώσεις μέσω των topics που αναφέρονται στις οδηγίες εγκατάστασης. **Καμιά άλλη μέθοδος δεν θα γίνει δεκτή.** Δοκιμάστε με διάφορες τελικές γωνίες για να εξετάσετε την ορθότητα του κώδικα.

Αποθηκεύστε (χρησιμοποιήστε το πακέτο **rosbag**) διαγράμματα με τις επιθυμητές τροχιές **γωνιών** και **γωνιακών ταχυτήτων** που σχεδιάσατε, καθώς και αυτές που πραγματοποίησε το ρομπότ.

4. Με την χρήση turnin θα παραδώσετε (α) το πακέτο με όλους τους κώδικες ώστε να μπορεί να εκτελεστεί, και (β) αναλυτική αναφορά. Σύντομα θα υπάρξουν οδηγίες στην ιστοσελίδα του μαθήματος.

Πίνακας 4: Μέθοδοι και παράμετροι σχεδιασμού τροχιών των αρθρώσεων για τα προβλήματα 2.2 και 2.3.

Μέθοδος/Παράμετρος	Τιμή
Κυβικά πολυώνυμα	Περιττός AM (ο μεγαλύτερος της ομάδας)
Γραμμικές συναρτήσεις με παραβολικά τμήματα	Άρτιος AM (ο μεγαλύτερος της ομάδας)
Αρχικές γωνίες των αρθρώσεων	$\mathbf{q}_0 = \begin{bmatrix} q_{1,0} \\ q_{2,0} \end{bmatrix} = \begin{bmatrix} 0^\circ \\ 0^\circ \end{bmatrix}$
Τελικές γωνίες των αρθρώσεων	$\mathbf{q}_f = \begin{bmatrix} q_{1,f} \\ q_{2,f} \end{bmatrix} = \begin{bmatrix} \text{round}\left(\frac{AM}{80}\right)^\circ \\ -\text{round}\left(\frac{AM}{120}\right)^\circ \end{bmatrix}$
Μέγιστη γωνιακή ταχύτητα άρθρωσης 1	$\dot{q}_{1,max} = 9^\circ/s$
Μέγιστη γωνιακή ταχύτητα άρθρωσης 2	$\dot{q}_{2,max} = 11^\circ/s$