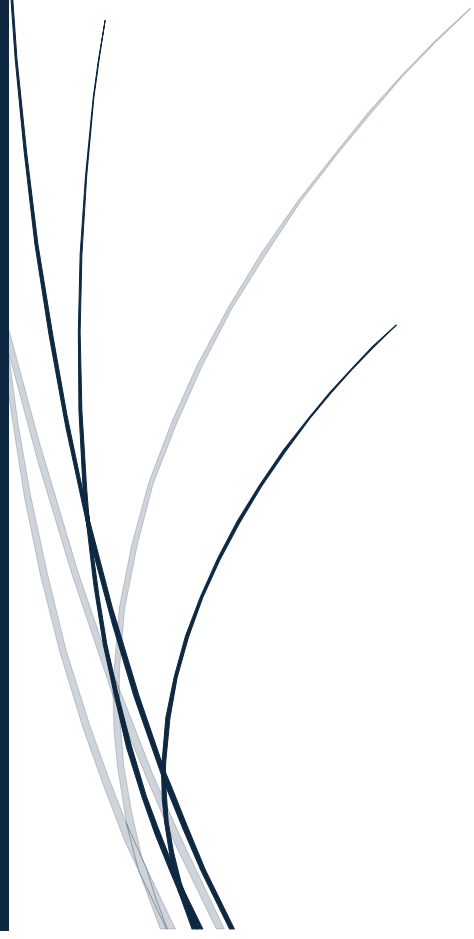
2023 - 2024

Εργασία Ρομποτικής

Δημητριάδης Γεώργιος – ΑΜ: 5209
Δημητρίου Αριστοτέλης – ΑΜ: 5211



MYE031-Ρομποτική
UNIVERSITY OF IOANNINA

Πίνακας περιεχομένων

Θέμα 1ο

Σε αυτό το κομμάτι της άσκησης μας ζητείται ο σχεδιασμός επιθυμητών τροχιών και ο προγραμματισμός της κίνησης ενός τροχοφόρου ρομπότ διαφορικής κίνησης, το Jackal της Clearpath.

Figure 1: Το τροχοφόρο ρομπότ της Jackal

Χαρακτηριστικά του ρομπότ

- Διαστάσεις: 508 x 430 x 250 mm
- Βάρος: 17Kg
- Μέγιστο ωφέλιμο βάρος: 20Kg
- Μέγιστη ταχύτητα: 2.0 m/s
- Μέγιστη διάρκεια λειτουργίας: 4 hours
- Τάση και ρεύμα: 5V - 5A, 12V - 10A, 24V - 20A
- Drivers and APIs: ROS, Mathworks

Απαντήσεις προβλημάτων θέματος 1

Ερώτημα 1ο

Σε αυτό το ερώτημα μας ζητείται να συντάξουμε ένα 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, και να αποθηκεύσουμε τις τροχιές των μετατοπίσεων/περιστροφών και των ταχυτήτων.

```python
#!/usr/bin/env python3
import rospy
from geometry_msgs.msg import Twist
import time
import math
import rosbag
def move_robot():
    rospy.init_node('custom_movement_node', anonymous=True)
    rate = rospy.Rate(10)  # 10 Hz
    cmd_vel_pub = rospy.Publisher('/cmd_vel', Twist, queue_size=10)
    twist = Twist()
    start_time = time.time()
    bag = rosbag.Bag('robot_movement1.bag', 'w')

    while not rospy.is_shutdown():
        current_time = time.time()
        elapsed_time = current_time - start_time

        if elapsed_time <= 10:
            twist.angular.z = math.radians(-15)
            twist.linear.x = 0
        elif 10 < elapsed_time <= 25:
            twist.angular.z = math.radians(10)
            twist.linear.x = 0.1
        else:
            break

        cmd_vel_pub.publish(twist)
        bag.write('/cmd_vel', twist)

if __name__ == '__main__':
        move_robot()
```

Figure 2: Κώδικας 1ου ερωτήματος

Εισαγωγή των απαραίτητων βιβλιοθηκών:

- rospy: Βιβλιοθήκη του ROS για την Python.
- Twist: Μήνυμα του ROS που χρησιμοποιείται για τον έλεγχο της κίνησης του ρομπότ.
- time: Βιβλιοθήκη για χειρισμό χρόνου.
- rosbag: Βιβλιοθήκη για εγγραφή και ανάγνωση των μηνυμάτων του ROS σε αρχεία.

Στη συνάρτηση move_robot():

- Η γραμμή rospy.init_node('custom_movement_node', anonymous = True) αρχικοποιεί έναν κόμβο με το όνομα 'custom_movement_node'.
- Η γραμμή rate = rospy.Rate(10) ορίζει το ρυθμό εκτέλεσης των εντολών στα 10 Hz.
- Ορίζεται το αντικείμενο cmd_vel_pub για την δημοσίευση μηνυμάτων ελέγχου κίνησης.
- Αρχικοποιείται ένα αντικείμενο Twist() για τον έλεγχο των ταχυτήτων.
- Δημιουργείται ένα αρχείο bag (robot_movement1.bag) για την καταγραφή των μηνυμάτων του ρομπότ.

Στη συνέχεια, εκτελούνται οι παρακάτω ενέργειες:

- Υπολογίζεται ο χρόνος που έχει παρέλθει από την έναρξη του προγράμματος.
- Βάσει του περασμένου χρόνου, ρυθμίζονται οι γωνιακές και γραμμικές ταχύτητες του ρομπότ.
- Δημοσιεύονται οι εντολές κίνησης στον κόμβο /cmd_vel.
- Καταγράφονται οι εντολές κίνησης στο αρχείο bag.

Ερώτημα 2ο

Σε αυτό το ερώτημα μας ζητείται να σχεδιάσουμε 3 διαφορετικές τροχιές χρησιμοποιώντας γραμμικές συναρτήσεις με παραβολικά τμήματα.

$$q_0 = \begin{bmatrix} x_0 \\ y_0 \\ \theta_0 \end{bmatrix} = \begin{bmatrix} 0\ m \\ 0\ m \\ 0\ rad \end{bmatrix} \qquad q_f = \begin{bmatrix} x_f \\ x_f \\ \theta_f \end{bmatrix} = \begin{bmatrix} 5\ m \\ 3\ m \\ 1\ rad \end{bmatrix}$$

1η τροχιά

Το ρομπότ μας πρέπει να κοιτάξει προς την τελική θέση.

Άρα $\theta_f = atan^2(y_f - y_0, x_f - x_0) = 0{,}54\, rad$

Στη μέση της διαδρομής: $\theta_h = \frac{(\theta_f + \theta_0)}{2} = 0{,}27\, rad$

Επίσης ισχύει ότι στην μέση της διαδρομής:

$$\theta_h = \theta_0 + \frac{1}{2}\ddot{\theta} t_b^2 + \ddot{\theta} t_b (t_h - t_b)$$

$$0{,}27 = 0{,}045 \ddot{\theta} t_f^2$$

$$\ddot{\theta} t_f^2 = 6 \text{ (σχέση 1)}$$

Ισχύει ότι την μέγιστη ταχύτητα θα την αποκτήσει πρώτη φορά την $t = t_b$:

$$\dot{\theta} = \ddot{\theta} t_b$$

$$\ddot{\theta} t_f \cdot 0{,}1 \leq 0{,}5236 \ (\max Velocity\ \sigma\varepsilon\ rad)$$

$$\ddot{\theta} t_f \leq 5{,}236$$

$$\frac{6}{t_f} \leq 5{,}236 \text{ (από σχέση 1)}$$

$$t_f \geq 1{,}14\, sec$$

Έστω ότι επιλέγουμε: $t_f = 1{,}2\, sec$

Επομένως από σχέση 1: $\ddot{\theta} = 4{,}16$ rad/s^2

Ας ελέγξουμε την τροχιά. Θα υπολογίσουμε το που βρίσκεται στο παραβολικό μέρος της επιταχυνόμενης κίνησης, στο γραμμικό μέρος και στο παραβολικό μέρος της επιβραδυνόμενης κίνησης:

$$\theta_b = \theta_0 + \frac{1}{2}\ddot{\theta} t_b^2 = 0{,}029952\, rad$$

$$\theta_{t_f - t_b} = \theta_0 + \frac{1}{2}\ddot{\theta} t_b^2 + \ddot{\theta} t_b (t_f - 2t_b) = 0{,}509184\, rad$$

Έπειτα θα στρίψει επιβραδυνόμενα όσο έστριψε και στην αρχή, άρα φτάνει στον τελικό στόχο.

Επομένως η τροχιά είναι αποδεκτή και από $\dot{\theta} = \ddot{\theta} t_b$ προκύπτει ότι:

$$\dot{\theta} = 0{,}4992\, r/s.$$

2η τροχιά

Η 2η τροχιά θα περιγράφει την γραμμική κίνηση του ρομπότ έτσι ώστε να φτάσει στην δοσμένη τελική θέση. Από το (x_0, y_0) το ρομπότ πρέπει να φθάσει στο (x_f, y_f).

Η απόσταση (ευκλείδεια) των 2 σημείων είναι :

$$d_f = \sqrt{(x_f - x_0)^2 - (y_f - y_0)^2} = 5{,}8 \text{ m}$$

Όπως και στον προηγούμενο σχεδιασμό τροχιάς:

$$d_h = \frac{d_f + d_0}{2} = 2{,}9 \text{ m}$$

$$2{,}9 = d_0 + \frac{1}{2}\ddot{d}t_b^2 + \ddot{d}\,th(t_h - t_b)$$

$$2{,}9 = 0{,}045\ddot{d}t_f^2$$

$$\ddot{d}t_f^2 = 64{,}4 \text{ (σχέση 2)}$$

Ισχύει ότι την μέγιστη ταχύτητα θα την αποκτήσει πρώτη φορά την $t = t_b$

$$\dot{d} \leq 0{,}2$$

$$\ddot{d}t_f \leq 2$$

$$\frac{64{,}4}{t_f} \leq 2 \text{(από σχέση 2)}$$

$$t_f \geq 32{,}2$$

Έστω ότι επιλέγουμε: $t_f = 40sec$

Επομένως από σχέση 2: $\ddot{d} = 0.04025 \text{ m/s}^2$

Ας ελέγξουμε την τροχιά. Θα υπολογίσουμε το που βρίσκεται στο παραβολικό μέρος της επιταχυνόμενης κίνησης, στο γραμμικό μέρος και στο παραβολικό μέρος της επιβραδυνόμενης κίνησης:

$$db = d_0 + \frac{1}{2}\ddot{d}t_b^2 = 0{,}322\ m$$

$$d_{tf-t_b} = d_0 + \frac{1}{2}\ddot{d}t_b^2 + \ddot{d}t_b(t_f - 2t_b) = 5{,}474\ m$$

Έπειτα θα κινηθεί με επιβράδυνση όσο κινήθηκε και στην αρχή, άρα φτάνει στον τελικό στόχο.

Επομένως η τροχιά είναι αποδεκτή και από $\dot{d} = \ddot{d}t_b$ προκύπτει

$$\dot{d} = 0{,}161\ m/s.$$

3η τροχιά

Στην 3η τροχιά εφόσον το ρομπότ φτάσει στην τελική θέση, θα περιγράψουμε την περιστροφή του, έτσι ώστε να αποκτήσει έναν δοσμένο τελικό προσανατολισμό:

$$\theta_f = 1\ rad$$

Σε αυτό το σημείο θα κινηθούμε με την ίδια ακριβώς διαδικασία όπως και στην πρώτη τροχιά αλλά με διαφορά τώρα ότι το $\theta_0 = 0.54rad$

Στη μέση της διαδρομής: $\theta_h = \frac{(\theta_f + \theta_0)}{2} = 0{,}77\ rad$

Επίσης ισχύει ότι στην μέση της διαδρομής:

$$\theta_h = \theta_0 + \frac{1}{2}\ddot{\theta}t_b^2 + \ddot{\theta}t_b(t_h - t_b)$$

$$0{,}23 = 0{,}045\ddot{\theta}t_f^2$$

$$\ddot{\theta}t_f^2 = 5{,}1 \text{(σχέση 3)}$$

Επίσης ισχύει ότι: $\dot{\theta} = \ddot{\theta}t_b$

$$\ddot{\theta}t_f \cdot 0{,}1 \leq 0{,}5236\ (\max Velocity\ \sigma\varepsilon\ rad)$$

$$\ddot{\theta}t_f \leq 5{,}236$$

$$\frac{5.1}{t_f} \leq 5{,}236 \text{ (από σχέση 3)}$$

$$t_f \geq 0{,}97\ sec$$

Έστω ότι επιλέγουμε: $t_f = 1\ sec$

Επομένως από σχέση 3: $\ddot{\theta} = 5{,}1$ rad/s^2

Ας ελέγξουμε την τροχιά. Θα υπολογίσουμε το που βρίσκεται στο παραβολικό μέρος της επιταχυνόμενης κίνησης, στο γραμμικό μέρος και στο παραβολικό μέρος της επιβραδυνόμενης κίνησης:

$$\theta_b = \theta_0 + \frac{1}{2}\ddot{\theta}t_b^2 = 0{,}5655\ rad$$

$$\theta_{t_{f-t_b}} = \theta_0 + \tfrac{1}{2}\ddot{\theta}t_b^2 + \ddot{\theta}t_b\left(t_f - 2t_b\right) = 0{,}9735 \text{ rad}$$

Έπειτα θα στρίψει επιβραδυνόμενα όσο έστριψε και στην αρχή ($\frac{1}{2}\ddot{\theta}t_b^2 = 0{,}0255$), άρα φτάνει στον τελικό στόχο.

Επομένως η τροχιά είναι αποδεκτή και από $\dot{\theta} = \ddot{\theta}t_b$ προκύπτει

$$\dot{\theta} = 0{,}51\ r\text{ad}/s.$$

Ερώτημα 3

Τέλος παρακάτω παραθέτουμε τον κώδικα του ολικού σχεδιασμού τροχιάς.

```python
#!/usr/bin/env python3
import rospy
from geometry_msgs.msg import Twist
import time
import rosbag

def moverobot():
    rospy.init_node('custom_angular_movement_node', anonymous=True)
    rate = rospy.Rate(10)
    cmd_vel_pub = rospy.Publisher('/cmd_vel', Twist, queue_size=20)
    twist = Twist()
    bag = rosbag.Bag('robot_movement2.bag', 'w')

    start_time = time.time()
    while not rospy.is_shutdown():
        current_time = time.time()
        elapsed_time = current_time - start_time

        if elapsed_time <= 0.12:
            twist.linear.x = 0
            twist.angular.z = 4.16 * elapsed_time
        elif 0.12 < elapsed_time <= 1.08:
            twist.linear.x = 0
            twist.angular.z = 0.4992
        elif 1.08 < elapsed_time <= 1.2:
            twist.linear.x = 0
            twist.angular.z = 0.4992 - 4.16 * (elapsed_time - 1.08)
        elif 1.2 < elapsed_time <= 5.2:
            twist.angular.z = 0
            twist.linear.x = 0.04025 * (elapsed_time - 1.2)
        elif 5.2 < elapsed_time <= 37.2:
            twist.angular.z = 0
            twist.linear.x = 0.161
        elif 37.2 < elapsed_time <= 41.2:
            twist.angular.z = 0
            twist.linear.x = 0.161 - 0.04025 * (elapsed_time - 37.2)
        elif 41.2 < elapsed_time <= 41.3:
            twist.linear.x = 0
            twist.angular.z = 5.1 * (elapsed_time - 41.2)
        elif 41.3 < elapsed_time <= 42.1:
            twist.linear.x = 0
            twist.angular.z = 0.51
        elif 42.1 < elapsed_time <= 42.2:
            twist.linear.x = 0
            twist.angular.z = 0.51 - 5.1 * (elapsed_time - 42.1)
        else:
            break

        cmd_vel_pub.publish(twist)
        bag.write('/cmd_vel', twist)

    bag.close()

if __name__ == '__main__':
        moverobot()
```

Figure 3: Κώδικας 3ου ερωτήματος

Εισαγωγή των απαραίτητων βιβλιοθηκών:

- rospy: Βιβλιοθήκη του ROS για Python.
- Twist: Μήνυμα του ROS για τον έλεγχο κίνησης.
- time: Βιβλιοθήκη για λειτουργίες χρόνου.
- rosbag: Βιβλιοθήκη για εγγραφή και ανάγνωση αρχείων bag του ROS.

Στη συνάρτηση move_robot():

- Αρχικοποιείται ο κόμβος ROS με το όνομα 'custom_angular_movement_node'.
- Ορίζεται ο ρυθμός εκτέλεσης των εντολών στα 10 Hz.
- Δημιουργείται ένα publisher για το topic /cmd_vel για να στέλνονται μηνύματα τύπου Twist.
- Δημιουργείται ένα αντικείμενο Twist για να καθορίσει τις γραμμικές και γωνιακές ταχύτητες.
- Αρχικοποιείται ο χρόνος εκκίνησης και ένα αρχείο bag ανοίγει για την καταγραφή των εντολών.

Στη συνέχεια, εκτελούνται οι παρακάτω ενέργειες:

- Για τα πρώτα 0.12 δευτερόλεπτα, η γωνιακή ταχύτητα αυξάνεται γραμμικά.
- Από 0.12 έως 1.08 δευτερόλεπτα, η γωνιακή ταχύτητα παραμένει σταθερή.
- Από 1.08 έως 1.2 δευτερόλεπτο, η γωνιακή ταχύτητα μειώνεται γραμμικά.
- Από 1.2 έως 5.2 δευτερόλεπτα, η γραμμική ταχύτητα αυξάνεται γραμμικά.
- Από 5.2 έως 37.2 δευτερόλεπτα, η γραμμική ταχύτητα παραμένει σταθερή.
- Από 37.2 έως 41.2 δευτερόλεπτα, η γραμμική ταχύτητα μειώνεται γραμμικά.
- Από 41.2 έως 41.3 δευτερόλεπτα, η γωνιακή ταχύτητα αυξάνεται γραμμικά.
- Από 41.3 έως 42.1 δευτερόλεπτα, η γωνιακή ταχύτητα παραμένει σταθερή.
- Από 42.1 έως 42.2 δευτερόλεπτα, η γωνιακή ταχύτητα μειώνεται γραμμικά.
- Μετά από 42.2 δευτερόλεπτα, το πρόγραμμα σταματά.
- Δημοσιεύονται οι ταχύτητες στο topic /cmd_vel.
- Καταγράφονται τα μηνύματα σε αρχείο bag.

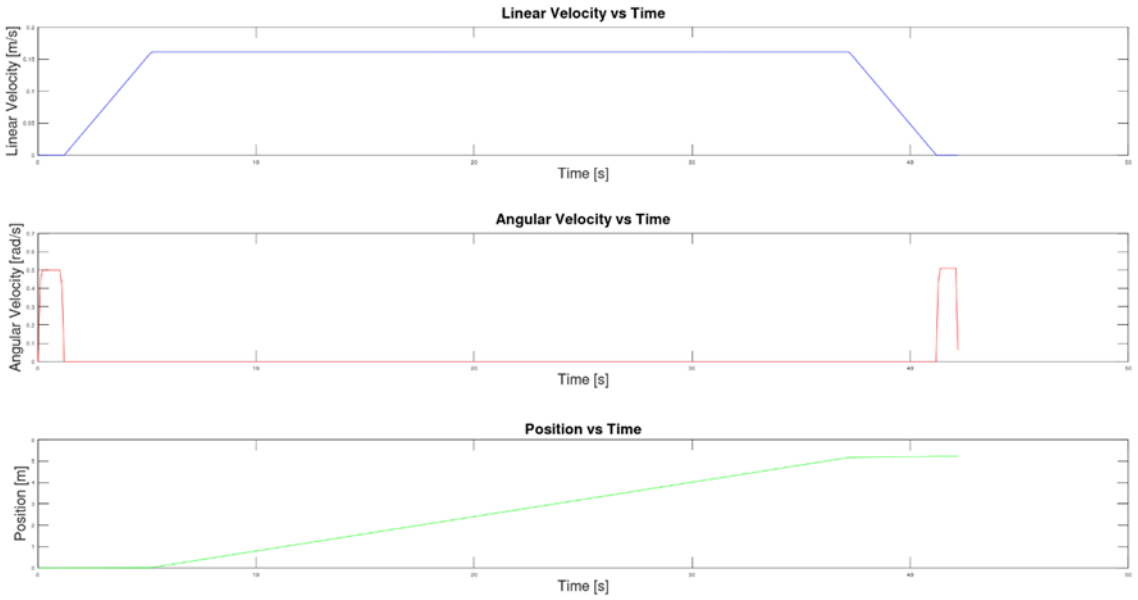


Figure 4: Διάγραμμα rosebag

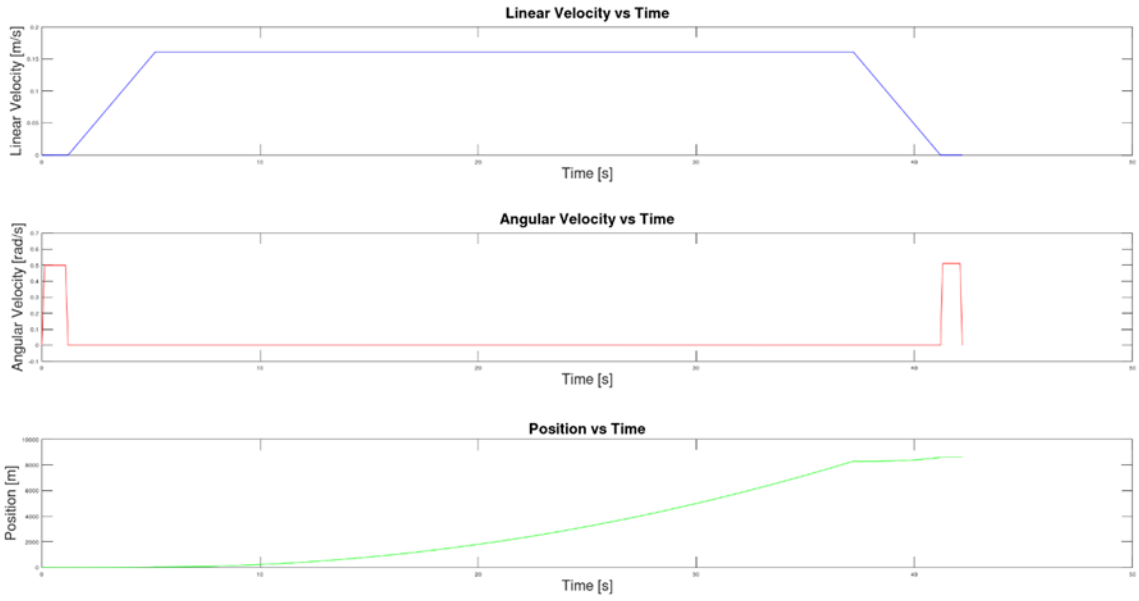


Figure 5: Θεωρητικό διάγραμμα

Εδώ παρατηρούμε σχεδόν την κίνηση την οποία αναμέναμε, δηλαδή το γράφημα προκύπτει από γραμμικό σχεδιασμό τροχιάς με παραβολικά τμήματα. Προφανώς στην αρχή της τροχιάς δεν έχουμε αρχική ταχύτητα καθώς το τροχοφόρο ρομπότ περιστρέφεται προς το τελικό σημείο. Μετά την περιστροφή αρχίζει να αποκτάει ταχύτητα (επιταχυνόμενη κίνηση). Έπειτα η ταχύτητα σταθεροποιείται στο γραμμικό μέρος και τέλος η ταχύτητα καταλήγει να μηδενίζεται (επιβραδυνόμενη κίνηση). Τέλος παρατηρούμε ότι το τροχοφόρο ρομπότ φτάνει στον τελικό του στόχο.

Όπως αναλύσαμε και παραπάνω η γωνιακή ταχύτητα θα είναι διάφορη του μηδενός στην αρχή και το τέλος της κίνησης. Όπως και πριν παρατηρούμε την αύξηση της γωνιακής ταχύτητας, την σταθεροποίηση της στο γραμμικό μέρος και τέλος την μείωσή της.

Θέμα 2ο

Σε αυτό το κομμάτι της άσκησης μας ζητείται ο προγραμματισμός της κίνησης ενός ρομποτικού βραχίονα σε περιβάλλον προσομοίωσης, έτσι ώστε να κινηθεί σύμφωνα με προσχεδιασμένες τροχιές.

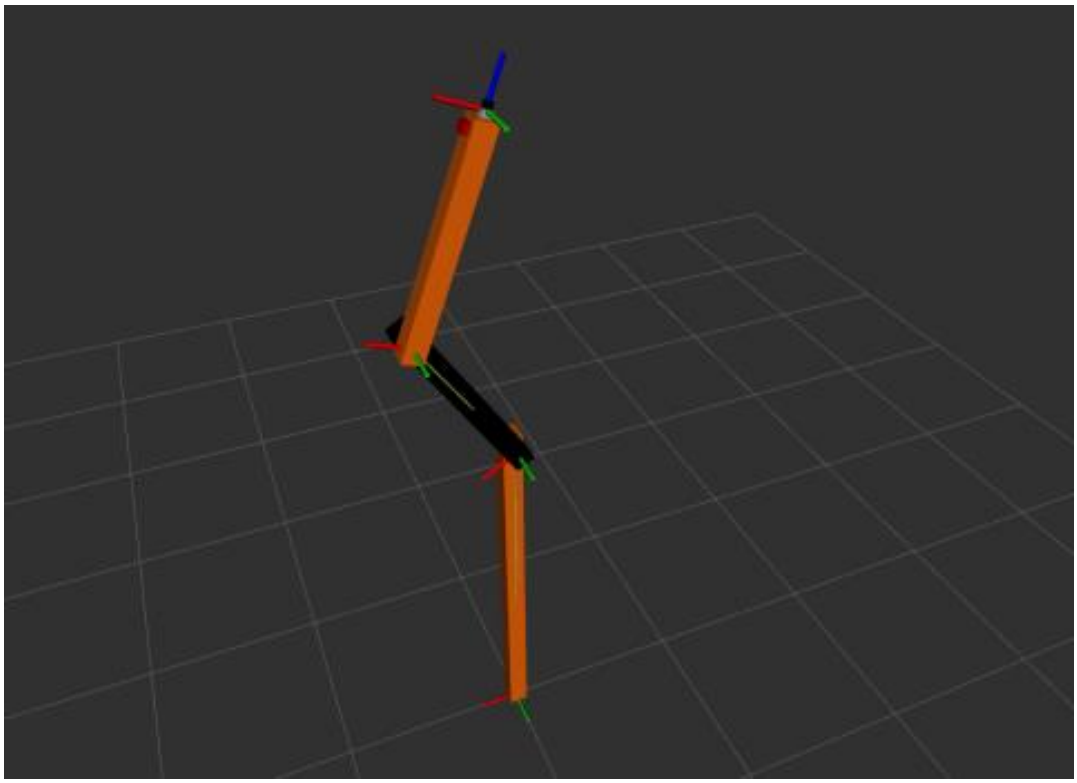


Figure 6: Ο βραχίονας rrbot

Απαντήσεις προβλημάτων θέματος 2

Ερώτημα 1

Σε αυτό το ερώτημα μας ζητείται να συντάξουμε ένα πρόγραμμα (node), το οποίο να περιστρέφει τις αρθρώσεις του ρομποτικού βραχίονα q1 και q2, σε επιθυμητές γωνίες και να αποθηκεύσουμε διαγράμματα με τις τροχιές των μεταβλητών των αρθρώσεων.

```python
#!/usr/bin/env python3
import rospy
from std_msgs.msg import Float64
import math
import rosbag

def move_joints():
    rospy.init_node('move_joints_node', anonymous=True)
    joint1_pub = rospy.Publisher('/rrbot/joint1_position_controller/command', Float64, queue_size=10)
    joint2_pub = rospy.Publisher('/rrbot/joint2_position_controller/command', Float64, queue_size=10)
    rospy.sleep(1)


    bag = rosbag.Bag('robot_arm_movement1.bag', 'w')

    joint1_command = Float64()
    joint2_command = Float64()
    joint1_command.data = math.radians(25)
    joint2_command.data = math.radians(-30)
    joint1_pub.publish(joint1_command)
    joint2_pub.publish(joint2_command)
    bag.write('/rrbot/joint1_position_controller/command', joint1_command)
    bag.write('/rrbot/joint2_position_controller/command', joint2_command)


if __name__ == '__main__':
        move_joints()
```

Figure 7: Κώδικας 1ου ερωτήματος

Εισαγωγή των απαραίτητων βιβλιοθηκών:

- rospy: Βιβλιοθήκη του ROS για Python.
- Std_msgs.msg: Μήνυμα του ROS για τον έλεγχο κίνησης.
- math: Βιβλιοθήκη για μαθηματικές συναρτήσεις
- rosbag: Βιβλιοθήκη για εγγραφή και ανάγνωση αρχείων bag του ROS.

Στη συνάρτηση move_joints():

- Αρχικοποιείται ο κόμβος ROS με το όνομα 'move_joints_node'.
- Δημιουργείται το publisher για την αποστολή εντολών κίνησης στην άρθρωση joint1.
- Δημιουργείται το publisher για την αποστολή εντολών κίνησης στην αρθρώση joint2.
- Δημιουργείται ένα αρχείο bag με το όνομα 'robot_arm_movement1.bag' για την καταγραφή των δεδομένων.
- Δημιουργείται ένα αντικείμενο Float64 για την αποθήκευση της εντολής για την αρθρώση joint1.

- Δημιουργείται ένα αντικείμενο Float64 για την αποθήκευση της εντολής για την αρθρώση joint2.
- Ορίζεται η γωνία σε radians για την άρθρωση joint1.
- Ορίζεται η γωνία σε radians για την άρθρωση joint2.
- Δημοσιεύεται η εντολή κίνησης για την άρθρωση joint1..
- Δημοσιεύεται η εντολή κίνησης για την άρθρωση joint2.
- Καταγράφεται η εντολή για την άρθρωση joint1 στο αρχείο bag.
- Καταγράφεται η εντολή για την άρθρωση joint1 στο αρχείο bag.

Ερώτημα 2

Σε αυτό το ερώτημα μας ζητείται να σχεδιάσουμε μια τροχιά για τις αρθρώσεις του ρομποτικού βραχίονα χρησιμοποιώντας κυβικά πολυώνυμα έτσι ώστε ταυτόχρονα να ξεκινούν από τις αρχικές γωνίες και να σταματούν σε κάποιες τελικές επιθυμητές γωνίες.

$$q_0 = \begin{bmatrix} q1,0 \\ q2,0 \end{bmatrix} = \begin{bmatrix} 0^o \\ 0^o \end{bmatrix} \qquad q_f = \begin{bmatrix} q1,f \\ q2,f \end{bmatrix} = \begin{bmatrix} +65^o \\ -43^o \end{bmatrix}$$

Για την άρθρωση 1 το κυβικό μας πολυώνυμο είναι το εξής:

$$q_1 = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

Με αρχικές μηδενικές συνθήκες και τελικές συνθήκες στόχου προκύπτουν με τον γνωστό τρόπο τα παρακάτω:

$$a_0 = 0$$

$$a_1 = 0$$

$$a_2 = \frac{3}{tf^2}(q_{1,f} - q_{1,0}) = \frac{3 \cdot 65}{t_f^2}$$

$$a_3 = \frac{-2}{tf^3}(q_{1,f} - q_{1,0}) = \frac{-2 \cdot 65}{t_f^3}$$

Μπορούμε τώρα να πάμε στην μέση της διαδρομής και να πούμε ότι θα έχει μέγιστη ταχύτητα, έτσι προκύπτει η εξής ανίσωση:

$$\dot{q} \leq 9$$

$$a_1 + 2a_2\, t_h + 3a_3 {t_h}^2 \leq 9$$

$$t_{f_1} \geq 10{,}8\ sec$$

Για την άρθρωση 2 το κυβικό μας πολυώνυμο είναι το εξής:

$$q_2 = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

Με αρχικές μηδενικές συνθήκες και τελικές συνθήκες στόχου προκύπτουν με τον γνωστό τρόπο τα παρακάτω:

$$a_0 = 0$$

$$a_1 = 0$$

$$a_2 = \frac{3}{tf^2}(q_{1,f} - q_{1,0}) = \frac{3\cdot(-43)}{t_f^2}$$

$$a_3 = \frac{-2}{tf^3}\left(q_{1,f} - q_{1,0}\right) = \frac{-2\cdot(-43)}{t_f^3}$$

Μπορούμε τώρα να πάμε στην μέση της διαδρομής και να πούμε ότι θα έχει μέγιστη ταχύτητα, έτσι προκύπτει η εξής ανίσωση(προφανώς η παρακάτω ταχύτητα θα είναι με απολυτό διότι στρέφεται προς τις μείον μοίρες):

$$\dot{q} \leq 12$$

$$a_1 + 2a_2\, t_h + 3a_3 {t_h}^2 \leq 9$$

$$t_{f_2} \geq 11{,}7\ sec$$

Επομένως επιλέγουμε το μέγιστο. Έστω $t_f = 12sec$ έτσι για την άρθρωση 1 και 2 προκύπτουν τα παρακάτω κυβικά πολυώνυμα:

$$q_1 = 1.35t^2 - 0.075t^3$$

$$\dot{q}_1 = 2.7t - 0.225t^2$$

$$q_2 = -0.896t^2 + 0.05t^3$$

$$\dot{q}_2 = -1.792t + 0.15t^2$$

Ερώτημα 3

Τέλος παρακάτω παραθέτουμε τον κώδικα του ολικού σχεδιασμού τροχιάς.

```python
#!/usr/bin/env python3
import rospy
from std_msgs.msg import Float64
import time
import math
import rosbag
def move_joints():
    duration=12
    rospy.init_node('move_joints_node', anonymous=True)
    joint1_pub = rospy.Publisher('/rrbot/joint1_position_controller/command', Float64, queue_size=10)
    joint2_pub = rospy.Publisher('/rrbot/joint2_position_controller/command', Float64, queue_size=10)
    bag=rosbag.Bag('robot_arm_movement2.bag', 'w')
    start_time = time.time()
    while not rospy.is_shutdown():
        current_time=time.time()
        elapsed_time=current_time-start_time
        if elapsed_time<=duration:
                current_time = time.time() - start_time
                q1 = 1.35* current_time**2 - 0.075 * current_time ** 3
                q2 = -0.896 * current_time**2 + 0.05 * current_time ** 3
                q1dot=2.7*current_time-0.025*current_time**2
                q2dot=-1.792*current_time+0.15*current_time**2
                joint1vel_command=Float64()
                joint2vel_command=Float64()
                joint1_command = Float64()
                joint2_command = Float64()
                joint1_command.data = math.radians(q1)
                joint2_command.data = math.radians(q2)
                joint1vel_command.data=math.radians(q1dot)
                joint2vel_command.data=math.radians(q2dot)
                joint1_pub.publish(joint1_command)
                joint2_pub.publish(joint2_command)
                bag.write('/rrbot/joint1_position_controller/command', joint1_command)
                bag.write('/rrbot/joint2_position_controller/command', joint2_command)
                #bag.write('/rrbot/joint1_position_controller/command', joint1vel_command)
                #bag.write('/rrbot/joint1_position_controller/command', joint2vel_command)
        else:
                break
        joint1_pub.publish(math.radians(q1))
        joint2_pub.publish(math.radians(q2))

if __name__ == '__main__':
        move_joints()
```

Figure 8: Κώδικας 3ου ερωτήματος

Εισαγωγή Βιβλιοθηκών και Αρχικοποίηση Κόμβων:

- Ο κώδικας ξεκινάει με τις απαραίτητες εισαγωγές βιβλιοθηκών.
- Εισάγεται η βιβλιοθήκη rospy για την επικοινωνία με το ROS, καθώς και τα μηνύματα Float64 από το πακέτο std_msgs.

Στη συνάρτηση move_joints():

- Η συνάρτηση move_joints() ξεκινάει με την αρχικοποίηση των publishers για τις κινητήρες των αρθρώσεων του ρομπότ.
- Ένα αρχείο ROS bag ανοίγει για να καταγράψει τις κινήσεις.
- Ορίζεται ο χρόνος έναρξης της κίνησης.
- Στη συνέχεια, μπαίνει σε έναν βρόχο while που εκτελείται μέχρι να διακοπεί.
- Μέσα στη βρόχο, υπολογίζονται οι γωνίες των αρθρώσεων του ρομπότ βάσει τα κυβικά πολυώνυμα.
- Οι γωνίες αυτές μετατρέπονται σε rad και δημοσιεύονται στα αντίστοιχα topics του ROS.
- Τα δεδομένα των γωνιών και των ταχυτήτων καταγράφονται επίσης στο ROS bag.
- Η βρόχος εξέρχεται όταν περάσει ο ορισμένος χρόνος διάρκειας.

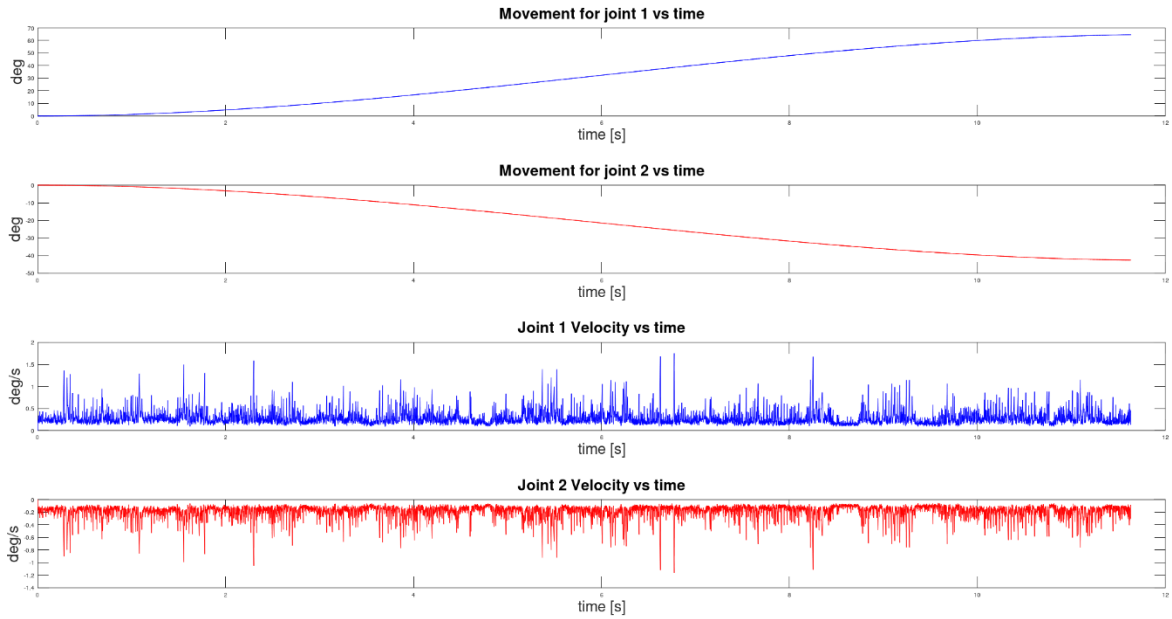


Figure 9: Διάγραμμα rosbag

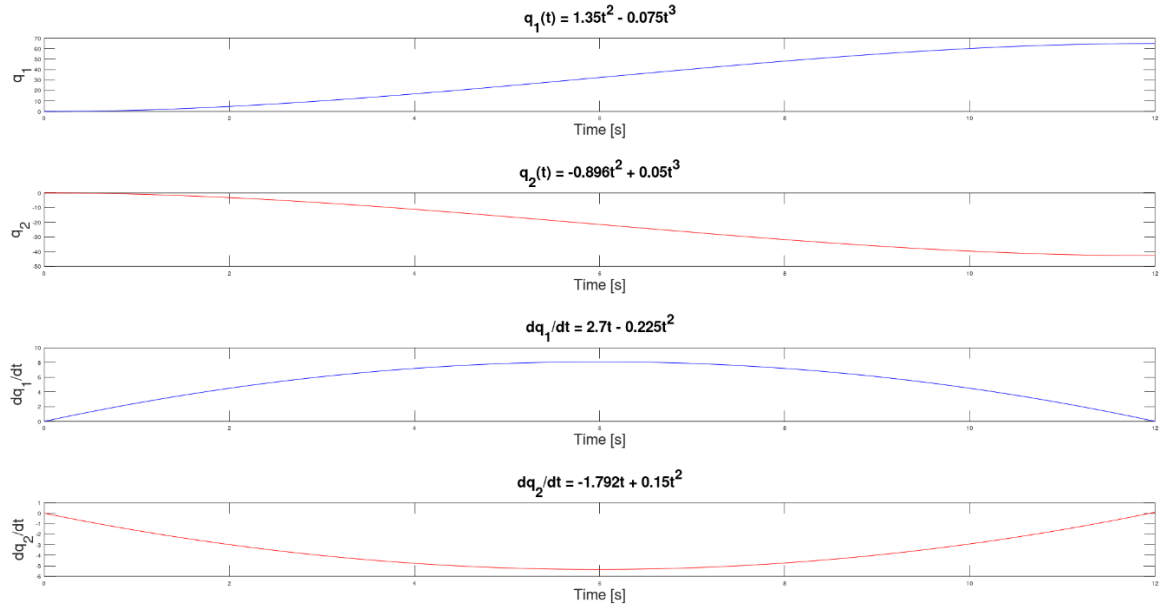


Figure 10: Θεωρητικό διάγραμμα

Σε αυτό το σημείο παρατηρούμε ότι και οι δύο οι αρθρώσεις φτάνουν στις αναμενόμενες τελικές θέσεις.

Σημείωση: Εμφανίστηκε ένα πρόβλημα με την κατασκευή του διαγράμματος ταχύτητας αρθρώσεων – χρόνου από το rosbag, καθώς οι διαταραχές είναι έντονες.