



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

Εξαμηνιαία Εργασία

στο μάθημα «Ρομποτική II: Ευφυή Ρομποτικά Συστήματα»

των φοιτητών

Ηλιόπουλος Γεώργιος, Α.Μ.: 03118815
Σερλής Εμμανουήλ-Αναστάσιος, Α.Μ.: 03118125
Ομάδα 26

Εξαμηνιαία Εργασία 2Α:

Αυτοκινούμενα ρομπότ: Παρακολούθηση εμποδίου
(Mobile robots: Wall Following)

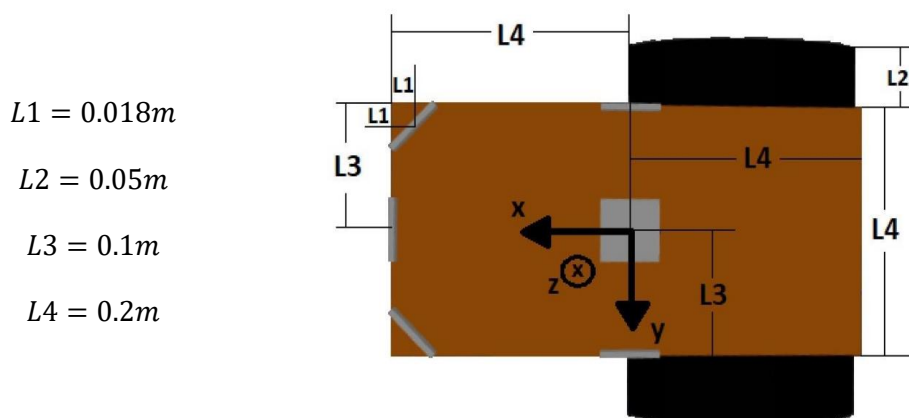
Διδάσκων: Κων/νος Τζαφέστας

Αθήνα, Ιούνιος 2022

A. Θεωρητική Ανάλυση

Σκοπός της εργασίας 2Α είναι ο προγραμματισμός ενός ρομπότ διαφορικής οδήγησης, έτσι ώστε να πραγματοποιεί παρακολούθηση εμποδίου (wall following). Συγκεκριμένα, η δοθείσα ρομποτική διάταξη αρχικά καλείται να πλησιάσει τα εμπόδια ξεκινώντας από την αρχική θέση (στο κέντρο της διάταξης), ενώ στην συνέχεια πρέπει να πραγματοποιήσει τουλάχιστον μία πλήρη περιφορά, κατά την οποία θα διατηρεί σταθερή απόσταση από τα εμπόδια.

Η ρομποτική διάταξη διαφορικής κίνησης διαθέτει 2 τροχούς διαμέτρου 20cm. Επιπλέον, διαθέτει σύνολο 6 αισθητήρες: α) 5 αισθητήρες υπερήχων σόναρ, οι οποίοι μετρούν απόσταση από εμπόδια, καθώς και β) ένα IMU (Inertial Measurement Unit) 9 βαθμών ελευθερίας, το οποίο μετράει στροφικές ταχύτητες, γραμμικές επιταχύνσεις, καθώς και περιστροφή γύρω από κάθε άξονα. Παρακάτω ακολουθούν τόσο τα τοπικά πλαίσια αναφοράς όσο και οι τιμές των αναγραφόμενων μεγεθών:

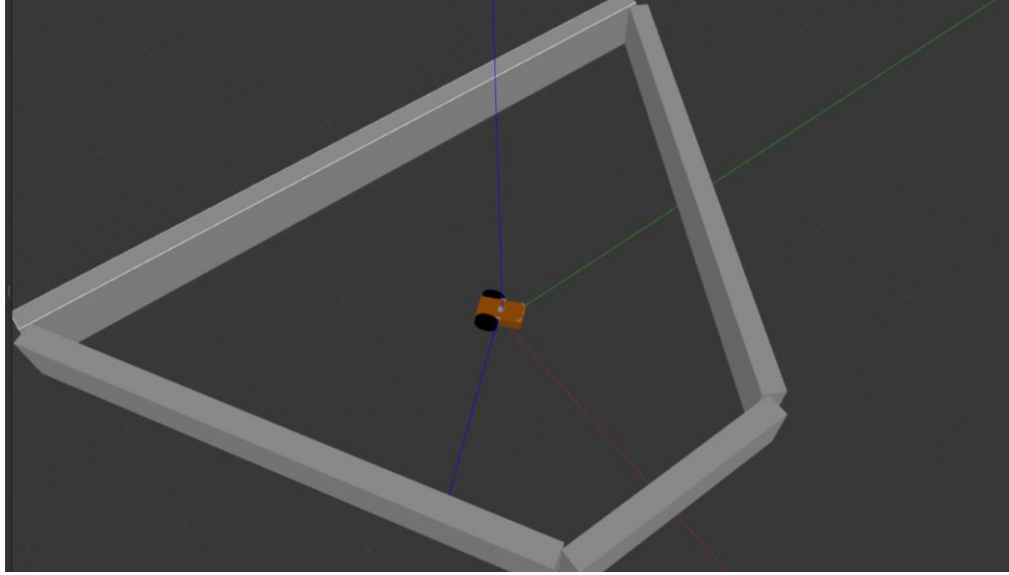


Εικόνα 1. Ρομπότ διαφορικής κίνησης

Ο ζητούμενος κωδικός αριθμός της ομάδας είναι $X = X1 + X2 = 5 + 5 \Rightarrow X = 10$, γεγονός που μας οδηγεί στα εξής αποτελέσματα:

- Αρχικός προσανατολισμός: $angle = mod(X, \pi) = 10 mod 3.14 = 0.575$
- Φορά περιστροφής: $mod(X, 2) = 10 mod 2 = 0 \Rightarrow CW$

Οπότε, έπειτα από την κατάλληλη τροποποίηση του αρχείου `mymobibot_world_wf.launch`, μας οδήγησε στην κάτωθι διάταξη αρχικοποίησης:



Εικόνα 2: Διάταξη αρχικοποίησης wall following task

Ο αλγόριθμος που θα πραγματοποιεί το εν λόγω task, αρχικά, καλείται να μετακινήσει το ρομπότ κοντά σε κάποιο από τα εμπόδια. Έτσι, δίνεται $V_x = 0.5 \text{ m/s}$ και $\omega_z = 0 \text{ rad/s}$.

Συγκεκριμένα, η εν λόγω υποεργασία διατηρείται μέχρις ότου έχουμε μέτρηση από τον μπροστινό αισθητήρα (F-sensor) μικρότερη των 0.3m .

Στην συνέχεια, το ρομπότ καλείται να πραγματοποιήσει στροφή ώστε να έρθει σε alignment με το εμπόδιο-τοίχο. Για αυτό, και μιας και η φορά περιστροφής είναι ωρολογιακή, λαμβάνονται υπόψη οι αποστάσεις των αριστερά αισθητήρων (L και FL-sensors). Συγκεκριμένα, το ρομπότ στρίβει κατά $\omega_z = \pi \text{ rad/sec}$ με γραμμική ταχύτητα $v_x = 0.05 \text{ m/s}$, ώστε να μην σταματήσει τελείως η γραμμική του κίνηση κατά την περιστροφή. Ωστόσο, δεν γινόταν να δοθεί $v_x > 1 \text{ m/s}$, αφού κάτι τέτοιο θα οδηγούσε σύγκρουση με τα εμπόδια. Η εν λόγω περιστροφή διαρκεί όσο η απόσταση των αριστερών αισθητήρων μένει μεγαλύτερη από την απόσταση του μπροστινού αισθητήρα, δηλαδή:

$$(dist_{FL} + 0.25 \geq dist_F) \vee (dist_L + 0.25 \geq dist_F),$$

Ο όρος 0.25 προστίθεται ώστε να κρατήσει λίγο περισσότερο χρονικά η διαδικασία περιστροφής, γεγονός που θα οδηγήσει σε λιγότερες επαναλήψεις του εν λόγω βήματος ευθυγράμμισης με το εκάστοτε εμπόδιο.

Μόλις τελειώσει το βήμα αυτό, ο αλγόριθμος περνά στην εκτέλεση ενός PD ελεγκτή. Αναφορικά με τον PD controller, ο ρόλος του είναι να προσδίδει σταθερή $v_x = 0.5 \text{ m/s}$, ενώ ταυτόχρονα να ρυθμίζει την γωνιακή ταχύτητα ω_z έτσι ώστε να έχουμε επιθυμητές αποστάσεις από τους FL και L sensors ίσες με:

$$safe_dist = 0.2\text{m}$$

$$L_dist_d = safe_dist + L2 = 0.25m$$

$$FL_dist_d = \sqrt{L_1^2 + (L_1 + L_{dist})^2} \cong 0.27m$$

Επιπλέον, θέσαμε κέρδη ελεγκτή **Kp=30** και **Kd=10** μιας και το πρωτεύον μέγεθος προς έλεγχο και σταθεροποίηση είναι η θέση του οχήματος ως προς τα εμπόδια

Τελικά, ο ελεγκτής περιγράφεται από τις κάτωθι αναλυτικές εξισώσεις:

- > $error_L = (L_dist_d - L_dist_sensor)$
- > $error_FL = (FL_dist_d - FL_dist_sensor)$
- > $p_error = error_L + error_FL$
- > $d_error_L = (error_L - error_L_prev)/dt$
- > $d_error_FL = (error_FL - error_FL_prev)/dt$
- > $d_error = d_error_L + d_error_FL$
- > $\omega z = Kp * p_error + Kd * d_error$

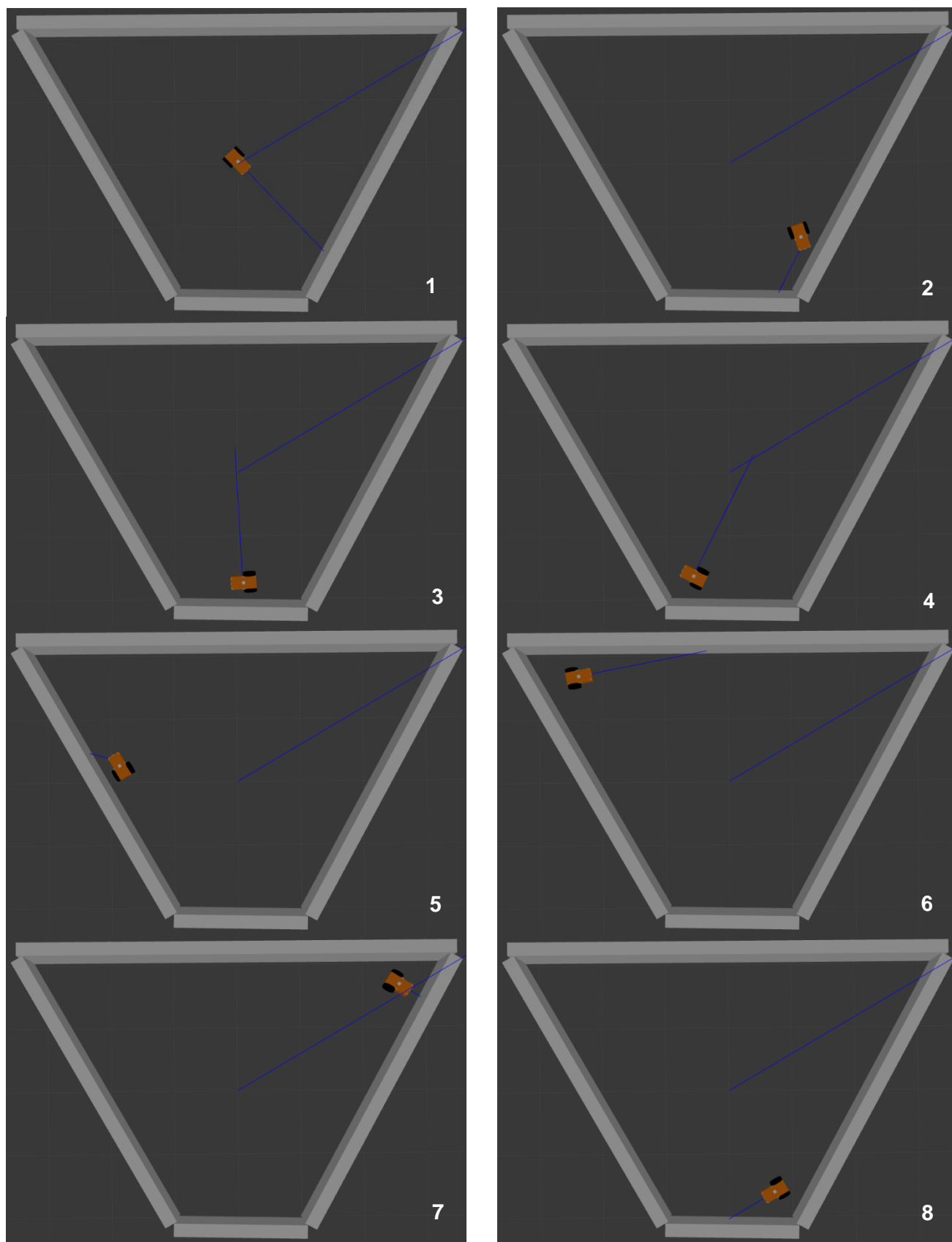
Στην λογική του βήματος περιστροφής, η λειτουργία του PD controller διαρκεί όσο η απόσταση του F_sensor είναι μεγαλύτερη από τις αποστάσεις των αριστερά αισθητήρων, δηλαδή πρέπει να ισχύει:

$$(dist_{FL} + 0.25 < dist_F) \wedge (dist_L + 0.25 < dist_F)$$

Ειδικά, ο PD Controller είναι προγραμματισμένος έτσι ώστε να μεταβαίνει εκ νέου στην κατάσταση περιστροφής η οποία περιεγράφηκε παραπάνω. Ο άνωθι αλγόριθμος-πλην του προκαταρκτικού βήματος μετάβασης κοντά στα εμπόδια-επαναλαμβάνεται περιοδικά, μέχρι το τέλος της προσομοίωσης.

B. Προσομοίωση

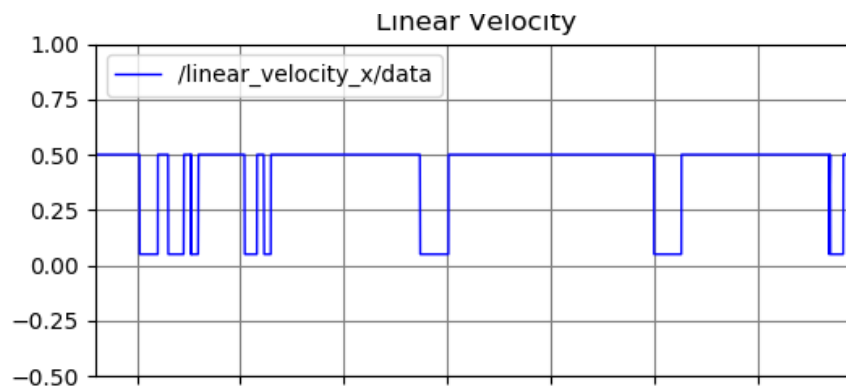
Στην εικόνα 3 φαίνεται η πορεία που ακολουθεί το ρομπότ από την θέση αρχικοποίησης έως ότου κάνει μία πλήρη περιστροφή. Πιο αναλυτικά, στην εικόνα 3.1 κινείται με ταχύτητα $0.5 m/s$ μέχρι να συναντήσει εμπόδιο. Στην εικόνα 3.2 συναντά το εμπόδιο και στρίβει CW. Στην εικόνα 3.5 ακολουθεί το εμπόδιο με βοήθεια του PD controller.



Εικόνα 3: Η πορεία του ρομπότ

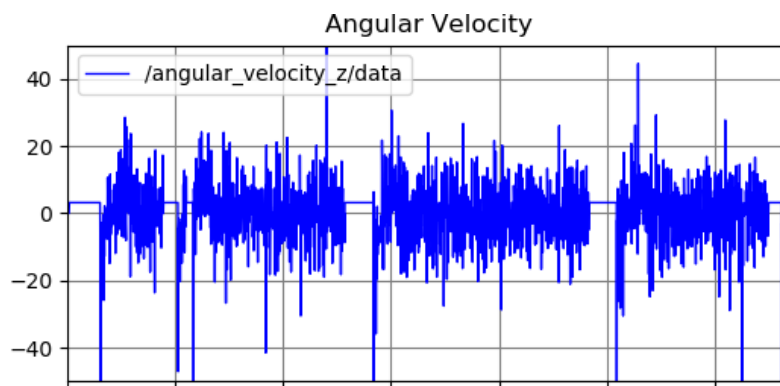
Στη συνέχεια παραθέτουμε ορισμένα χρήσιμα γραφήματα μαζί με σύντομες επεξηγήσεις.

Στην εικόνα 4, παρατίθεται η γραμμική ταχύτητα του οχήματος συναρτήσει του χρόνου. Παρατηρούμε πως η γραμμική ταχύτητα παίρνει τιμές 0.5 m/s είτε 0.05 m/s όπως άλλωστε έχουμε ρυθμίσει τον αλγόριθμό μας να κάνει. Για το διάστημα όπου η ταχύτητα είναι σταθερή και ίση με 0.5 m/s , το ρομπότ κάνει απλή ακολούθηση εμποδίου, ενώ για τα διαστήματα όπου η ταχύτητα είναι 0.05 m/s το ρομπότ εκτελεί την υποεργασία στροφής.



Εικόνα 4: Γραμμική ταχύτητα x

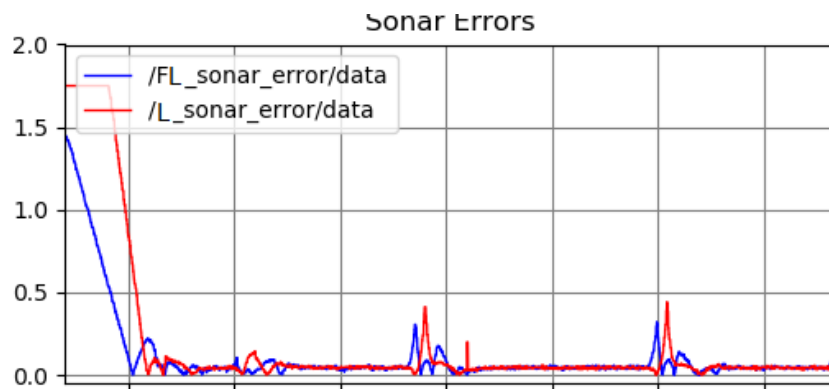
Στην εικόνα 5, παρατίθεται η γωνιακή ταχύτητα του οχήματος συναρτήσει του χρόνου. Παρατηρούμε πως η γωνιακή ταχύτητα παρουσιάζει πολλές διακυμάνσεις. Συγκεκριμένα, στα διαστήματα όπου το ρομπότ ακολουθεί τον τοίχο η γωνιακή ταχύτητα αυξομειώνεται με απότομο τρόπο λόγω της επενέργειας του ελεγκτή. Όταν το ρομπότ εκτελεί στροφή, τότε η γωνιακή ταχύτητα παίρνει σταθερή τιμή ίση με $\pi \text{ rad/s}$.



Εικόνα 5: Γωνιακή ταχύτητα z

Για να δούμε πόσο καλά εκτελεί την επιθυμητή εργασία το ρομπότ χρειάζεται να μελετήσουμε τα σφάλματα των αισθητήρων που χρησιμοποιούμε. Στην εικόνα 6 βλέπουμε τα σφάλματα του αριστερού και του μπροστά αριστερού αισθητήρα. Ως σφάλμα ορίζουμε την απόσταση που μετρά ο αισθητήρας μείον την επιθυμητή απόσταση που θέλουμε να έχουμε από τον τοίχο. Παρατηρούμε

ότι αρχικά το σφάλμα είναι μεγάλο, αφού το ρομπότ κινείται από την θέση αρχικοποίησης προς τον τοίχο. Όσο το ρομπότ κινείται παράλληλα με τον τοίχο το σφάλμα είναι σταθερό και σχεδόν ίσο με το μηδέν ενώ όταν το ρομπότ πλησιάζει σε γωνίες το σφάλμα παρουσιάζει απότομη αύξηση, μιας και τότε ενεργοποιείται η υποεργασία στροφής.



Εικόνα 6: Σφάλμα αισθητήρων - θέσης