

Ρομποτική 2

Σμαράγδα Μπενέτου el18048 — Ανδρέας Βατίστας el18020

Εξαμηνιαία εργασία 2Α

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



Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Ακαδημαϊκό Έτος 2021-2022

Α.Θεωρητική Ανάλυση

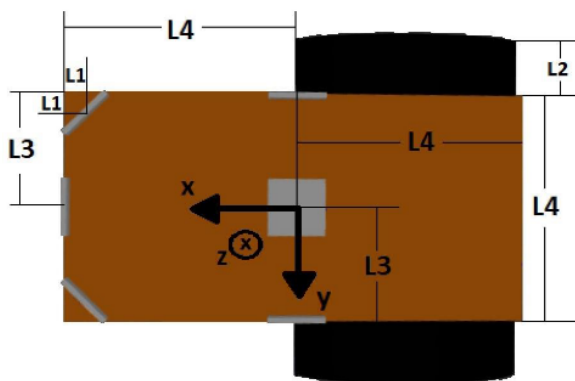
Στόχος

Πλήρη περιφορά κινητού ρομπότ υπό σταθερή απόσταση μέσα σε ένα χώρο που οριοθετείται από τοίχους.

Από τα τελευταία ψηφία των αριθμών μητρώων μας, έχουμε $X=8+0=8$. Επομένως $X \bmod 2 = 0$ και άρα καλούμαστε να υλοποιήσουμε κίνηση του αυτόνομου ρομπότ κατά την ωρολογιακή φορά (CW).

Προδιαγραφές κινητού ρομπότ

Οι διαστάσεις του κινητού ρομπότ διαφορετικής οδήγησης φαίνονται στην παρακάτω εικόνα και οι ρόδες του έχουν ακτίνα 0.1m. Επίσης διαθέτει 5 αισθητήρες σόναρ στις θέσεις που φαίνονται στην παρακάτω εικόνα και έναν IMU στο κέντρο.



Όπου $L1=0.05\text{m}$, $L2=0.018\text{m}$, $L3=0.1\text{m}$ και $L4=0.2\text{m}$

Για λόγους απλότητας στη συνέχεια του ριπορτ οι μονάδες μέτρησης θα παραλείπονται.

Τρόπος Λύσης

Αρχικά το ρομπότ μας ξεκινάει με γωνία αρχικού προσανατολισμού ίση με $\text{angle} = 8 \bmod \pi = 1.717 \text{ rad}$.

Η διαδικασία μας αποτελείται από 1+2 βήματα. Την προσέγγιση σε κάποιο τοίχο και την εφαρμογή ελεγκτή που παραλληλοποιεί και στρίβει ώστε να πραγματοποιείται παρακολούθηση των εμποδίων (τοιχών).

Αρχικά το ρομπότ μας κινείται προς κάποιο τοίχο πηγαίνοντας ευθεία βάσει του τυχαίου αρχικού του προσανατολισμού.

Όταν το μπροστινό σόναρ λάβει κάποιο σήμα κάτω από 2.0 (2.0 λαμβάνει και όταν τα beams είναι out of range), εισέρχεται προσωρινά σε κυκλική τροχιά. Αυτό επιλέχθηκε προκειμένου η τελική κοντινή θέση του ρομπότ στον συγκεκριμένο τοίχο να το βρει με τον τοίχο στα αριστερά του, ώστε ο ελεγκτής μας να συγκλίνει. Αυτό το πρόβλημα δεν υπάρχει σε περιπτώσεις που ο χώρος οριοθετείται από τετράγωνη διάταξη τοίχων.

Οι ταχύτητες για την εν λόγω κυκλική τροχιά επιλέγονται ως γραμμική=0.3 και γωνιακή=0.4. Αυτό θα μας δώσει κυκλική κίνηση ακτίνας 0.75 και συνεπώς διαμέτρου 1.5, που θα καταλήγει και στο κοντινότερο στον τοίχο σημείο (απόσταση 0.5 προφανώς).

Όταν το ρομπότ μας πλησιάσει αρκετά κοντά, εισέρχεται στην ρουτίνα του ελεγκτή τροχιάς που υλοποιήσαμε.

Ο συγκεκριμένος ελεγκτής εκτελεί παραλληλοποίηση αλλά και στρίβει στις στροφές. Προφανώς και ελέγχει μόνο τη γωνιακή ταχύτητα, ενώ η γραμμική επιλέχθηκε σταθερή.

Ο συγκεκριμένος ελεγκτής λαμβάνει υπόψη τα errors για την αριστερή απόσταση (proportional & derivative όροι), έναν proportional όρο κλίσης (με αρκετά μικρότερο k_p - γι' αυτό και δεν επιλέχθηκε και d-όρος) σε σχέση με τον τοίχο και έναν όρο μπροστινής απόστασης (pd), που αξιοποιείται μόνο για το steering κομμάτι στις στροφές.

Για τους όρους απόστασης δίνονται τα επιθυμητά μεγέθη αριστερής (επιλέχθηκε 0.3) και μπροστινής μέτρησης (μόνο όταν βρίσκεται σε κάποια κοντινή σε τοίχο απόσταση λαμβάνεται υπόψη, γιατί όσο μεγαλύτερο τόσο το καλύτερο). Το steering αρχίζει να γίνεται όταν το μπροστινό σόναρ βλέπει τιμές κάτω από ένα κατώφλι (επιλέχθηκε `front_desired_distance=0.6`).

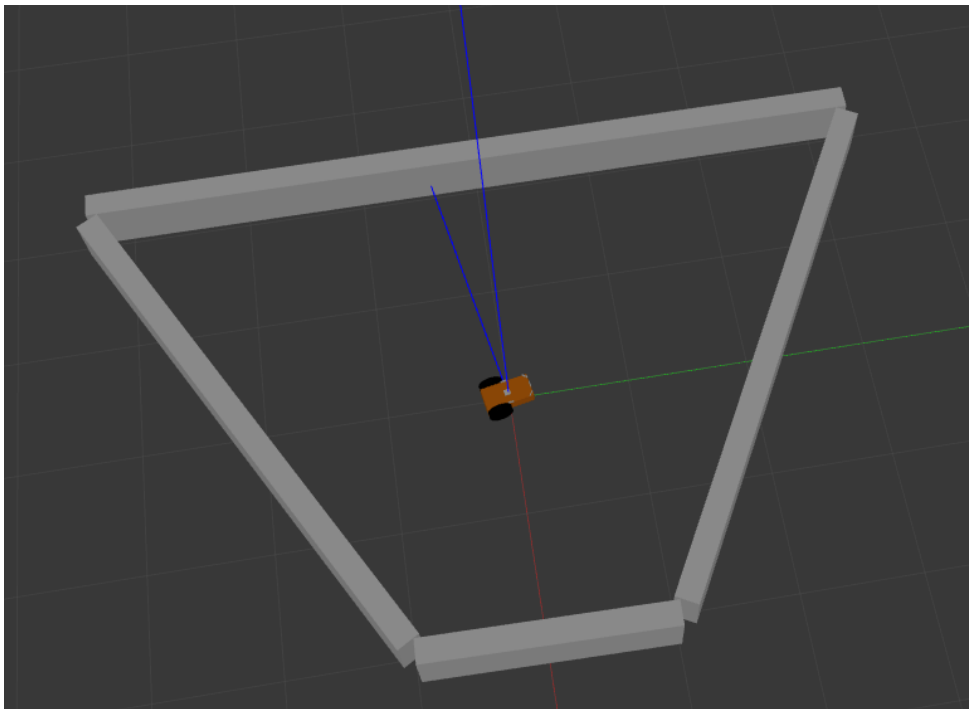
Για τον όρο της κλίσης που διατηρεί την παραλληλία της κίνησης, μπορεί κανείς να δει αναλυτικότερα το "Παράρτημα" στο τέλος του ριπόρτ.

Όλοι οι όροι k_p , k_d για τις διάφορες συνιστώσες επιλέχθηκαν μετά από κατάλληλο tuning πειραματικής μεθόδου και για γραμμική ταχύτητα ίση με 0.1.

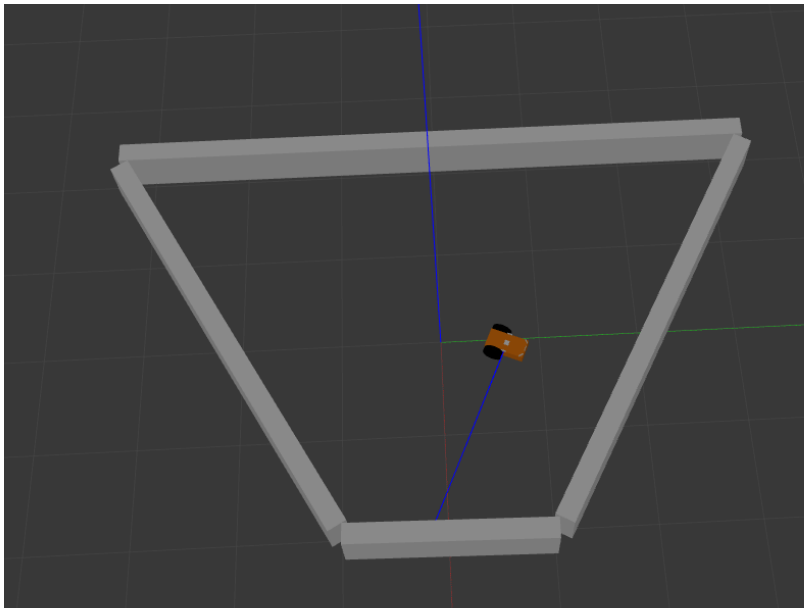
Οι παραπάνω τιμές καθώς και ο τρόπος υλοποίησης του ελεγκτή παρουσιάζονται με αρκετά καθαρό και αρκετά κατανοητό τρόπο στον κώδικα για περαιτέρω λεπτομέρειες.

Β. Προσομοίωση

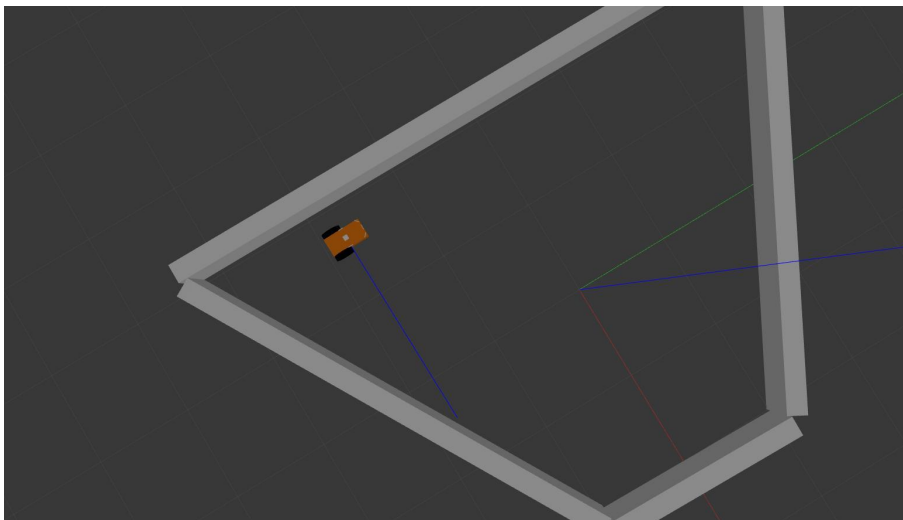
Ο αρχικός προσανατολισμός, όπως προκύπτει από τις προδιαγραφές του προβλήματος.

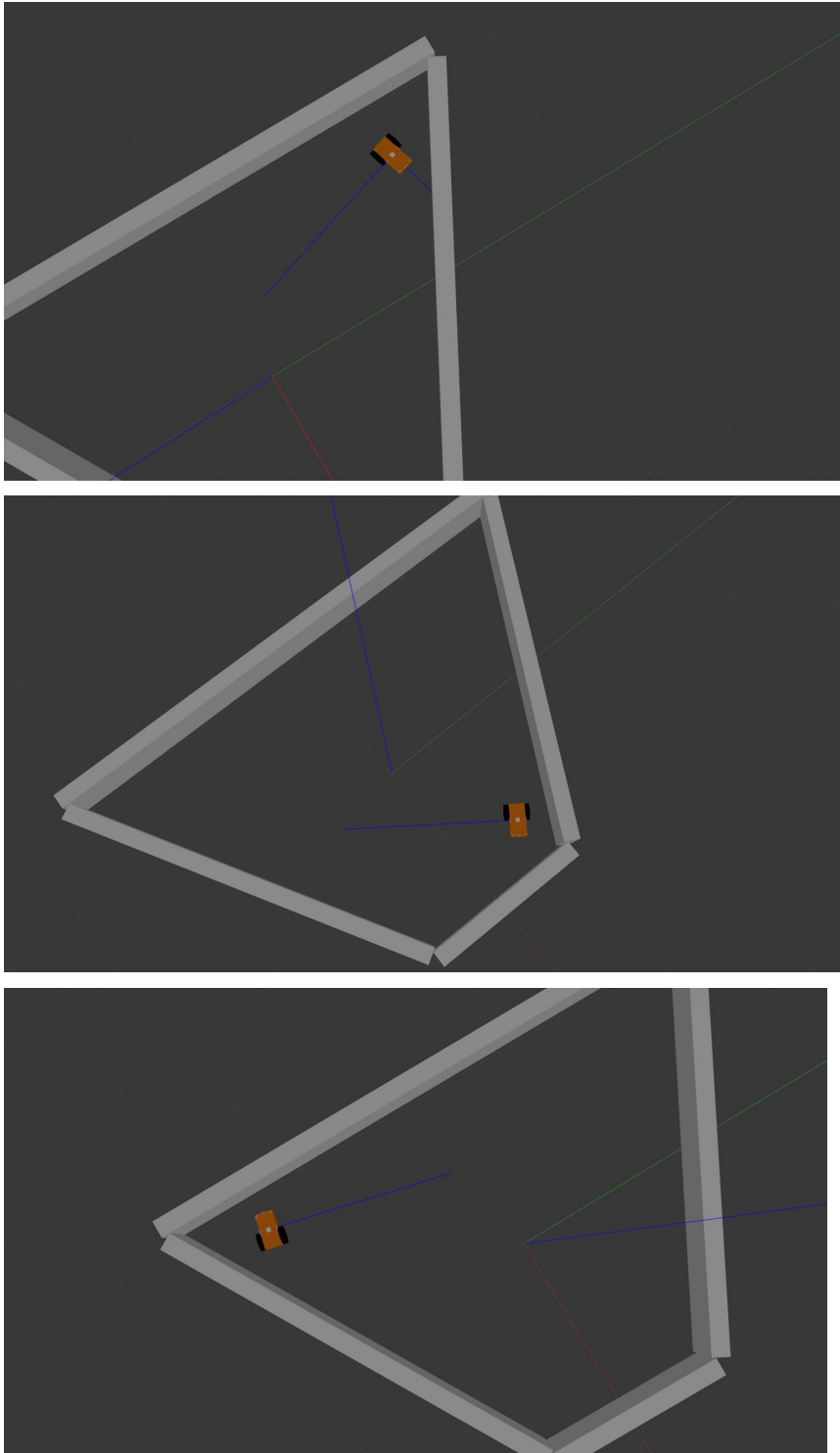


Για το αρχικό στάδιο φαίνεται στο επόμενο καρέ η είσοδος σε μικρή (ημι)κυκλική τροχιά (χωρίς πραγματοποίηση κάποιου είδους ελέγχου σε αυτήν, έως ότου φτάσει κοντά σε τοίχο) προκειμένου να ξεκινήσει wall following με τον τοίχο στα αριστερά του.



Παρατηρούμε ότι στην ευθύγραμμη τροχιά κρατάει σταθερή απόσταση και κινείται παράλληλα από τοίχο ενώ στις γωνίες εκτελεί ομαλή καμπυλόγραμμη τροχιά.

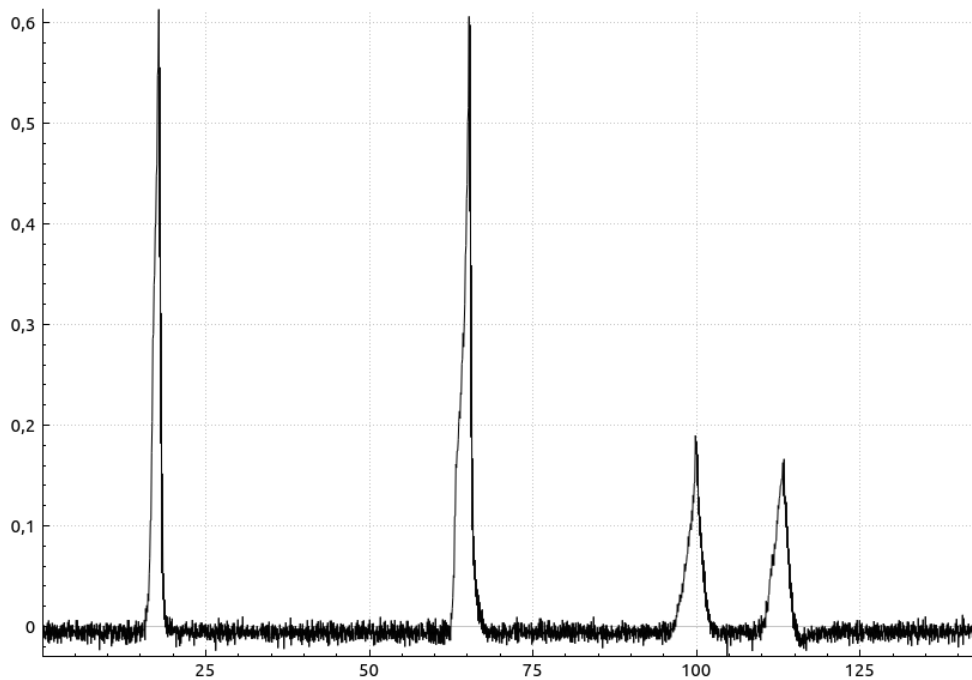




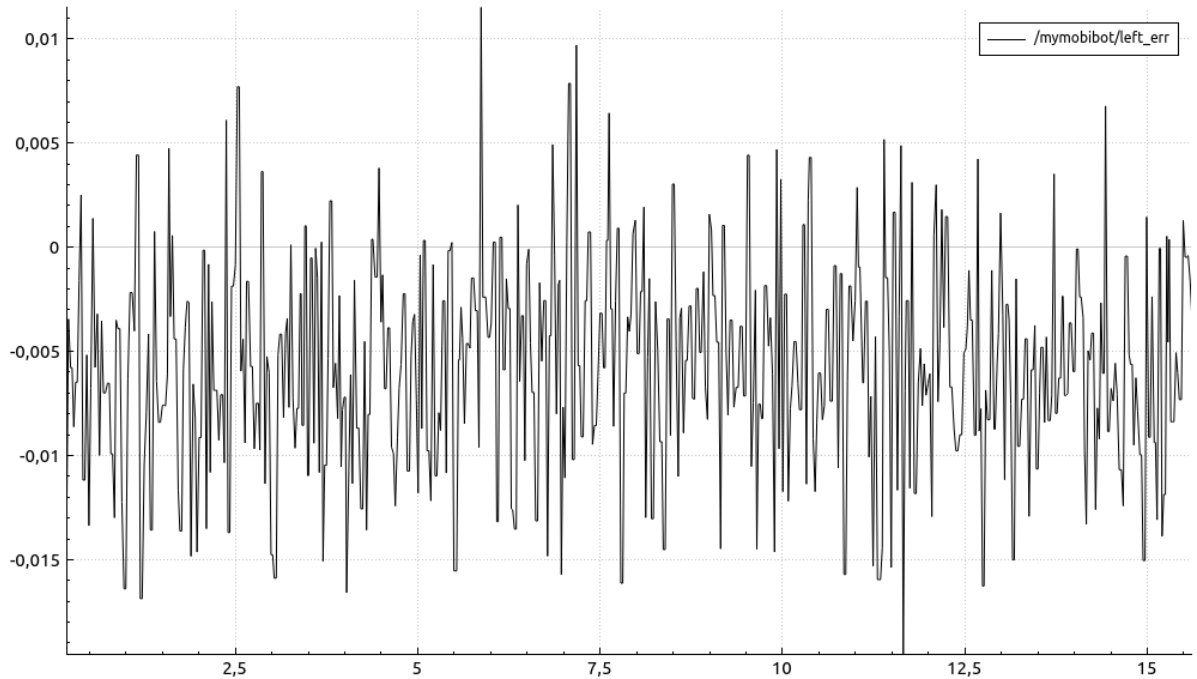
Από τις γραφικές παραστάσεις ελέγχουμε εάν πράγματι κρατάει τις ζητούμενες αποστάσεις.

Η παρακάτω είναι η γραφική παράσταση της μεταβλητής $left_err = sonar_left - desired_distance$ και επομένως μας δίνει το λάθος στην

απόσταση του κινητού ρομπότ από το πλησιέστερο εμπόδιο με βάση τη ζητούμενη απόσταση που είναι τα 0.3m.

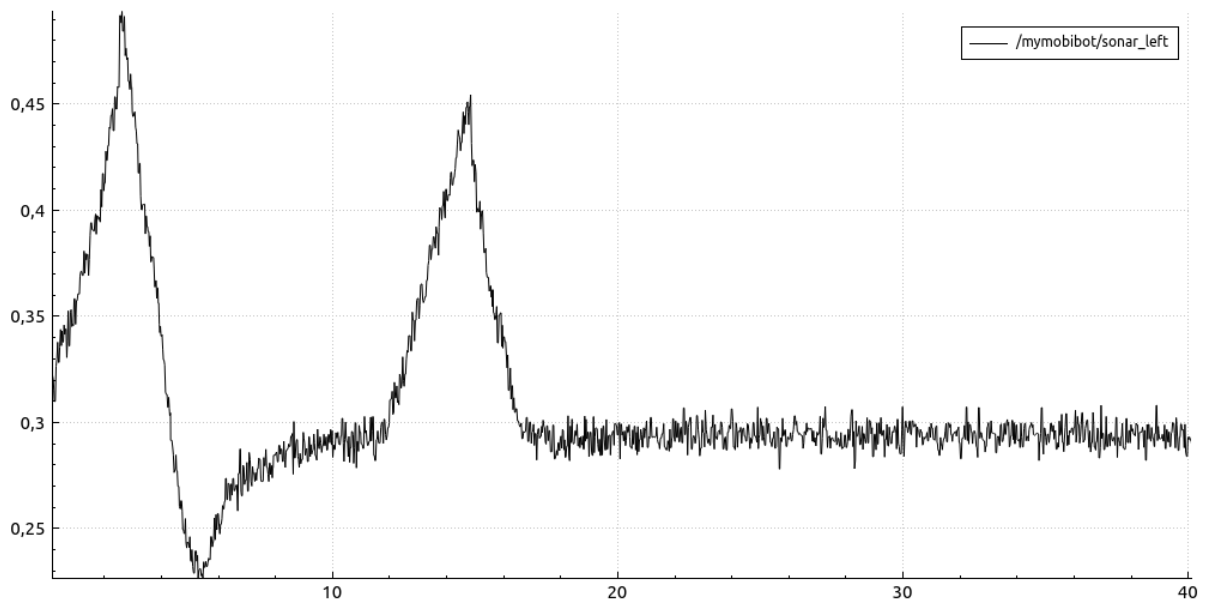


Παρατηρούμε ότι το λάθος είναι πολύ κοντά στο μηδέν με εξαίρεση 4 peaks τα οποία αντιπροσωπεύουν τις 4 γωνίες της διαδρομής που ακολουθεί σε μία περίοδο. Τα δύο peaks είναι πολύ μεγαλύτερα από τα άλλα δύο, τα μέν έχουν λάθος 0.6m ενώ τα δε 0.18m(όπου αφού είναι θετικά το κινητό δεν πλησιάζει τα εμπόδια σε απόσταση λιγότερη των 0.3m(με ένα πολύ μικρό σφάλμα που θα φανεί καλύτερα στην παρακάτω γραφική παράσταση). Τα πρώτα 2 peaks αντιστοιχούν στις κλειστές γωνίες, δηλαδή μικρότερες των 90 μοιρών και τα άλλα δύο στις ανοιχτές, μεγαλύτερες των 90 μοιρών.



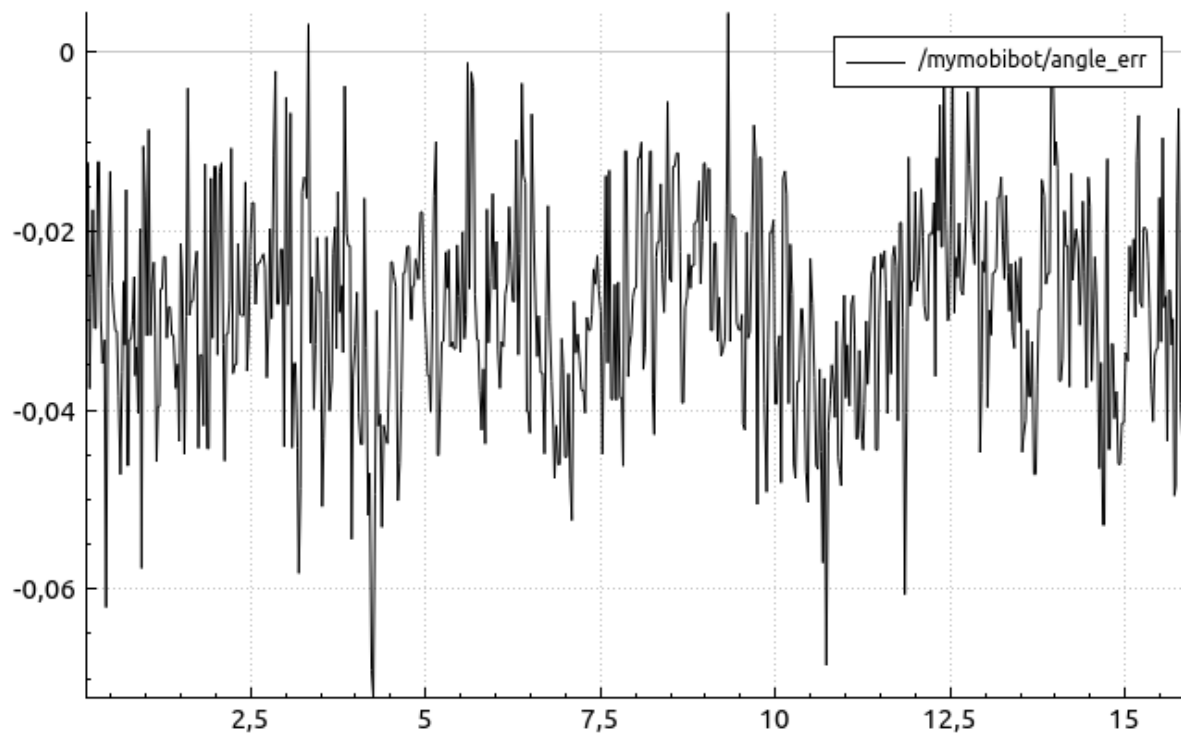
Παρατηρούμε ότι το σφάλμα δεν ξεπερνάει κατά απόλυτη τιμή το 0.02, αποτέλεσμα που κρίνεται ως πολύ ικανοποιητικό.

Γραφική παράσταση κάθετης απόστασης κινητού ρομπότ από εμπόδιο όπως ανιχνεύεται από το αριστερό sonar



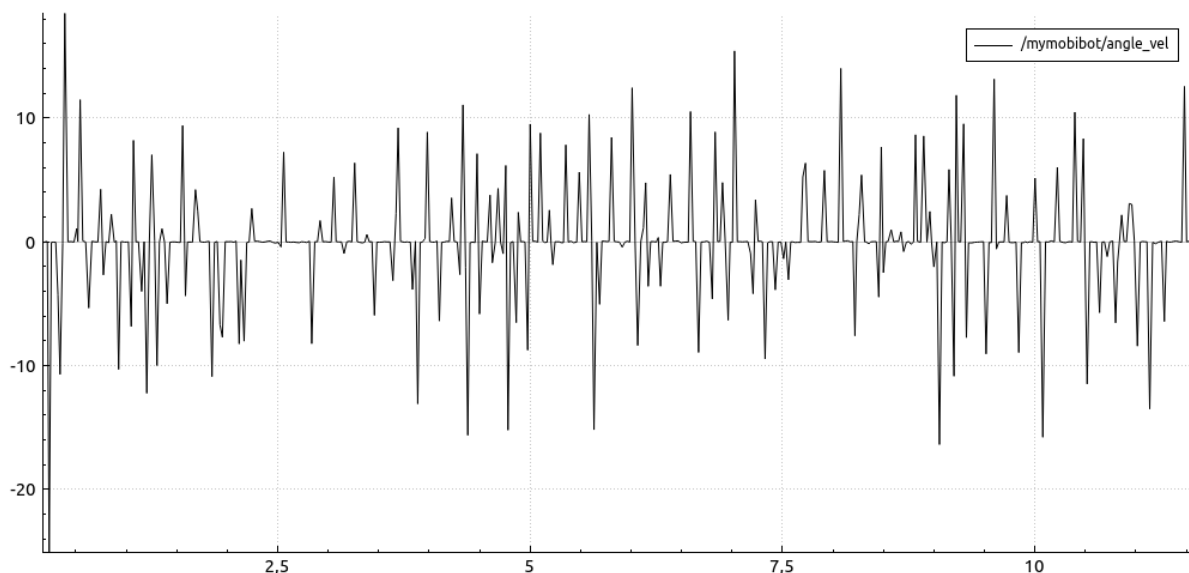
Εδώ παρατηρούμε ότι στην γωνία έφτασε σε μέγιστη απόσταση 0.055m.

Επίσης καταγράφουμε και το angle error το οποίο είναι το σφάλμα προσανατολισμού του κινητού και ευθυγράμμισης του με το εμπόδιο.



Το οποίο δεν ξεπερνάει όπως παρατηρούμε τις 0.08 μοίρες κατά απόλυτη τιμή.

Παραθέτουμε και την γραφική παράσταση της γωνιακής ταχύτητας του κινητού

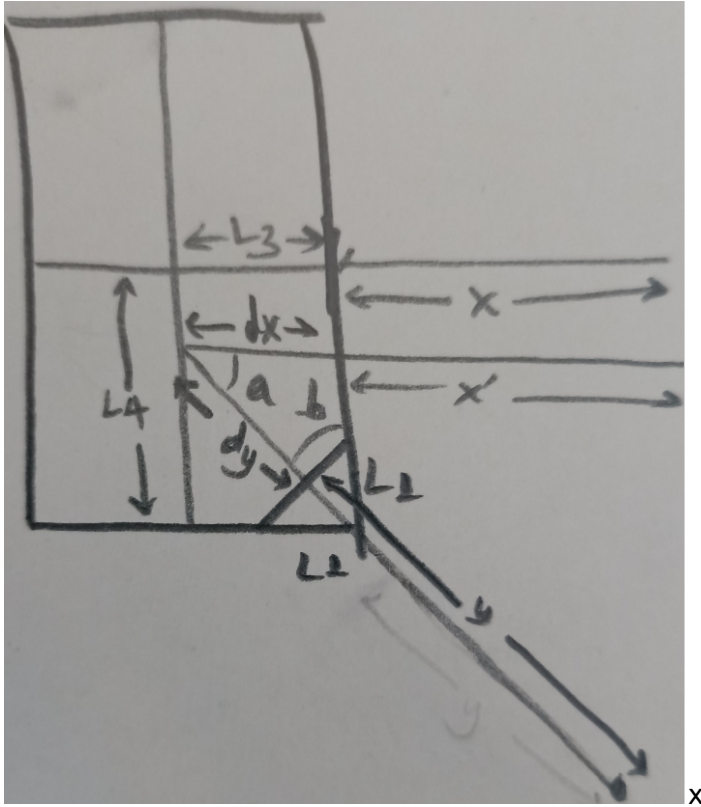


Παρατηρούμε ότι πραγματοποιούνται συνεχώς διορθώσεις, όπως είναι προφανές ώστε να διατηρεί σωστό παράλληλο προσανατολισμό και αντίστοιχα για να πραγματοποιεί τις στροφές.

Παράρτημα

Για το κομμάτι της παράλληλης κίνησης:

Για να ελέγξουμε την κατεύθυνση του ρομπότ (angle velocity) με ευθυγράμμισή του με τον τοίχο (σε σταθερή γνωστή απόσταση) συγκρίνουμε τις αποστάσεις που ανιχνεύουν τα left και front-left sonar ως εξής:



Επιθυμούμε η γωνία b να είναι ορθή επομένως γνωρίζοντας την γωνία a (υπολογίζεται παρακάτω), και γνωρίζοντας το Δx , Δy (οι αποστάσεις από το κέντρο του ρομπότ ώστε να μπορούμε να χρησιμοποιήσουμε το σχηματιζόμενο τρίγωνο, με Δx , Δy να υπολογίζονται παρακάτω) λαμβάνουμε την συνθήκη:

$\cos(\alpha) = (y+\Delta y)/(x+\Delta x) \rightarrow (y+\Delta y)/(x+\Delta x) - \cos(\alpha) = 0$ και στον αλγόριθμο $-0.1 < (y+\Delta y)/(x+\Delta x) - \cos(\alpha) < 0.1$

Αφού το τρίγωνο που δημιουργείται διχοτομεί το τριγωνάκι με πλευρές L1 και αφού το τριγωνάκι αυτό είναι ισοσκελές τότε το $b=45$ μοίρες και άρα το $a = 45$ μοίρες.

Επίσης το $\Delta x = L3 = 0.1\text{m}$ και το $\Delta y = L3/\cos\alpha - (\sqrt{2}/2)L1 = 0.1414 - 0.03536 = 0.106\text{m}$

άρα η συνθήκη γίνεται

$$-0.1 < (y + 0.106) / (x + 0.1) - 0.707 < 0.1$$

(όπου $\text{sqrt}(2)/2 = 0.707$)

Κατά την φάση ευθύγραμμης κίνησης με παρακολούθηση εμποδίου χρησιμοποιούμε PD ελεγκτή για την γωνιακή ταχύτητα διατηρώντας την γραμμική ταχύτητα σταθερή στα 0.1m/s.

Τα σφάλματα που παρακολουθούμε είναι το σφάλμα απόστασης από τον τοίχο αλλά και η παράλληλη κίνηση ως προς τον τοίχο.

Το πρώτο σφάλμα είναι:

$e = \text{sonar_left} - \text{desired_distance}$

και $e' = (e_{\text{prev}} - e)/dt$

Και για το δεύτερο σφάλμα:

$e2 = (\text{sonar_frontleft} + 0.106) / (\text{sonar_left} + 0.1) - 0.707$

με $k_p = 10$, $k_d = 10$, $k_p2 = 2$ άρα η συνάρτηση της γωνιακής ταχύτητας είναι:

$\omega = -(k_p * e + k_d * e') + k_p2 * e2$