

ΕΡΓΑΣΙΑ ΡΟΜΠΟΤΙΚΗΣ 2018-2019

Ελευθεριάδης Πέτρος

6044

Μέρος Α

1)

Για να αποθηκεύσω τις τιμές που παίρνει το ρομπότ από τους αισθητήρες χρησιμοποιώ τις global μεταβλητές `xi, yi, yaw`, έτσι ώστε να ανανεώνονται με κάθε κλήση της callback και μπορώ να τις χρησιμοποιήσω έπειτα και μες τη main.

Επίσης για να λειτουργήσει ο έτοιμος κώδικας που δίνεται, φτιάχνω το μήνυμα τύπου `nav_msgs::Odometry od_msg` και βαζω τα καταλληλά includes για το `tf` και `nav_msgs`.

2)

Εκμεταλλεύομαι το ήδη έτοιμο callback και τις global μεταβλητές από την 1 και το χρησιμοποιώ για να μαθαίνω τη θέση του ρομπότ κάνοντας `subscribe` στο topic "odom". Επίσης δημιουργώ έναν publisher στο topic `cmd_vel` για να δίνω ταχύτητα στο ρομπότ μέσω του `vel_msg` μηνύματος τύπου `geometry_msgs::Twist`

Το πρόβλημα το έλυσα σε 4 κομμάτια, ένα για κάθε πλευράς του τετράγωνου, μέσα σε μια `for` loop η οποία θα εκτελέσει το κάθε κομμάτι 2 φορές για τα 2 διαφορετικά τετράγωνα. Μόνο η απόσταση αλλάζει οπότε τα υπόλοιπα μένουν ίδια. Κάθε κομμάτι αποτελείται 2 κινήσεις, μια ευθεία `distance` μέτρων και μια στροφή 90 μοιρών.

Στο 1ο έχουμε ως συνθήκη το `x=10` ή 5 και στη στροφή το `yaw=1.55`

Στο 2ο έχουμε ως συνθήκη το `y=10` ή 5 και στη στροφή το `yaw=3.1`

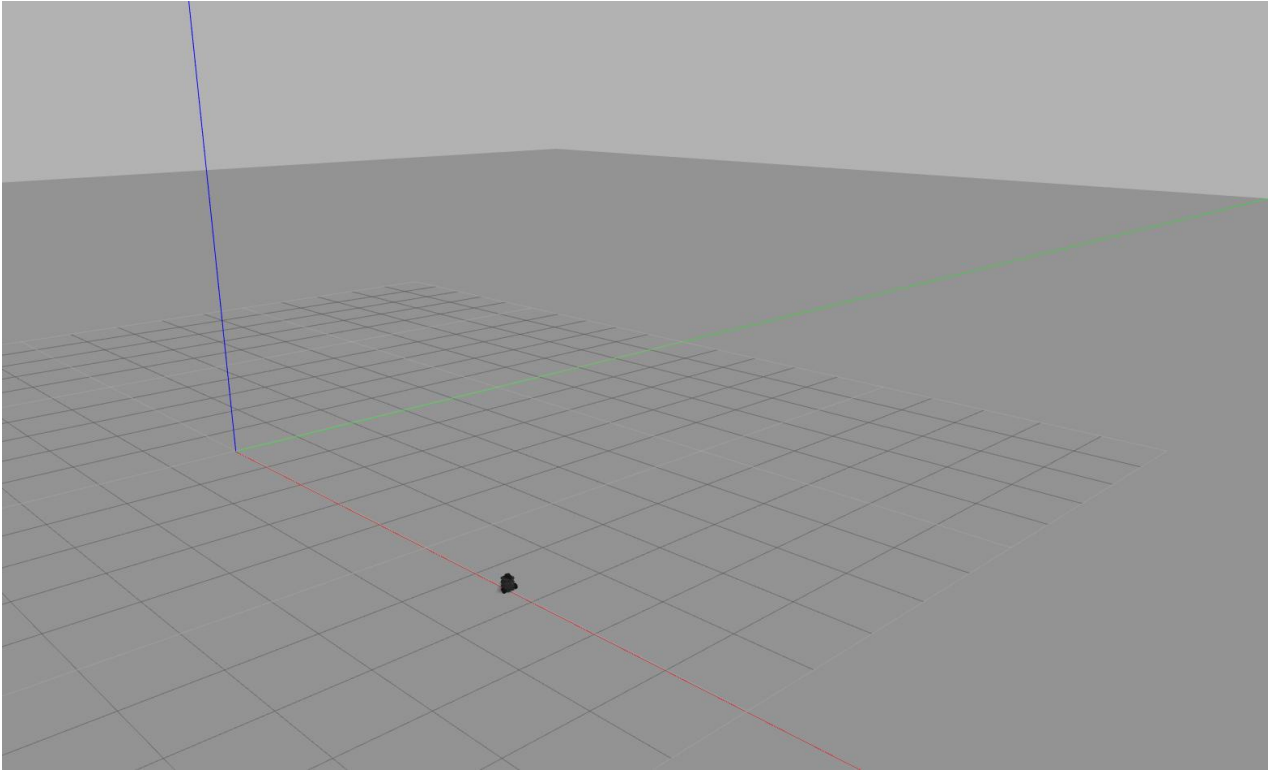
Στο 3ο έχουμε ως συνθήκη το `x=0` και στη στροφή το `yaw=-1.55`

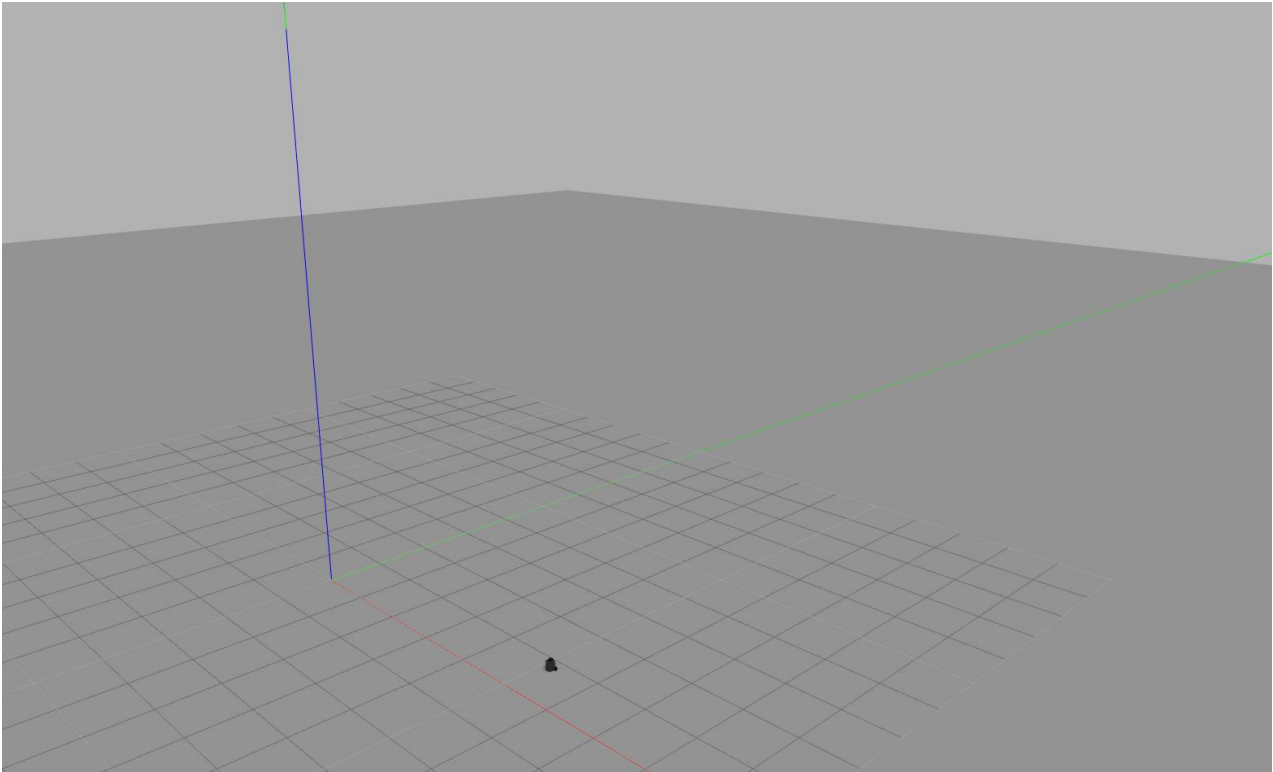
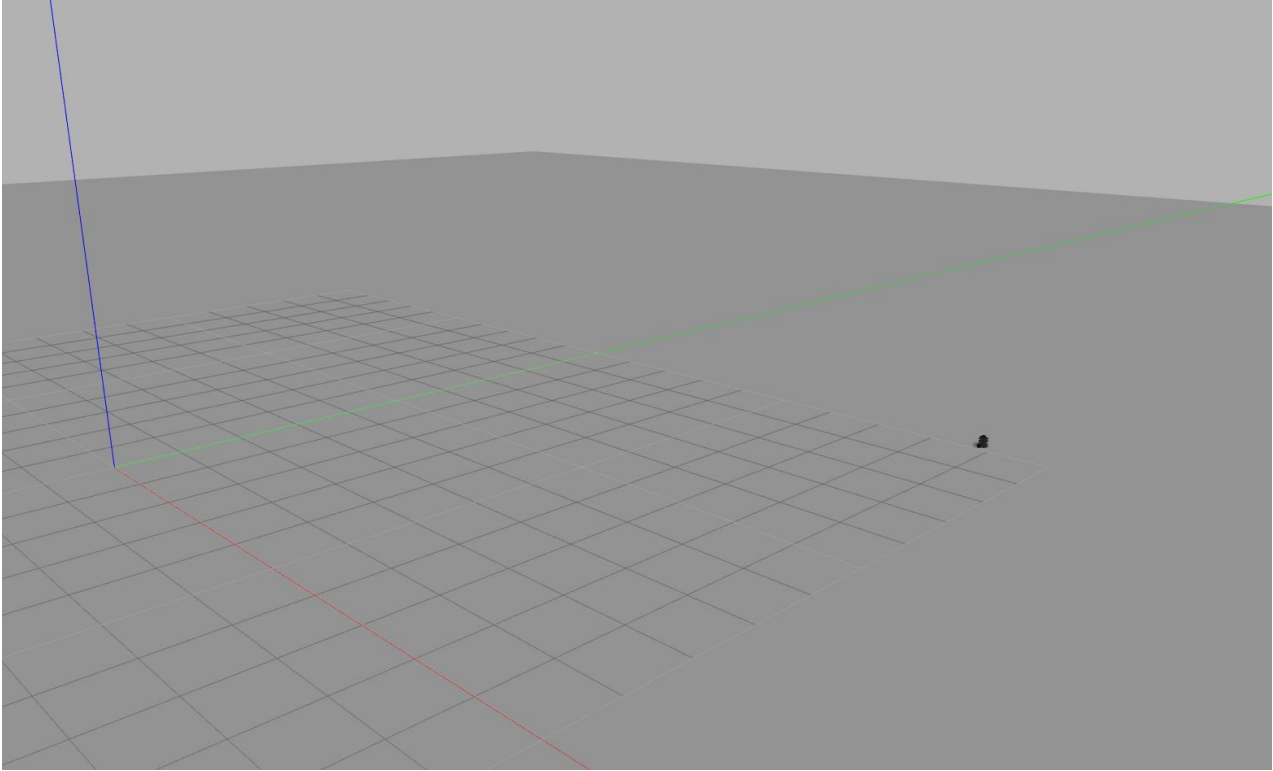
Στο 4ο έχουμε ως συνθήκη το `y=0` και στη στροφή το `yaw=0`

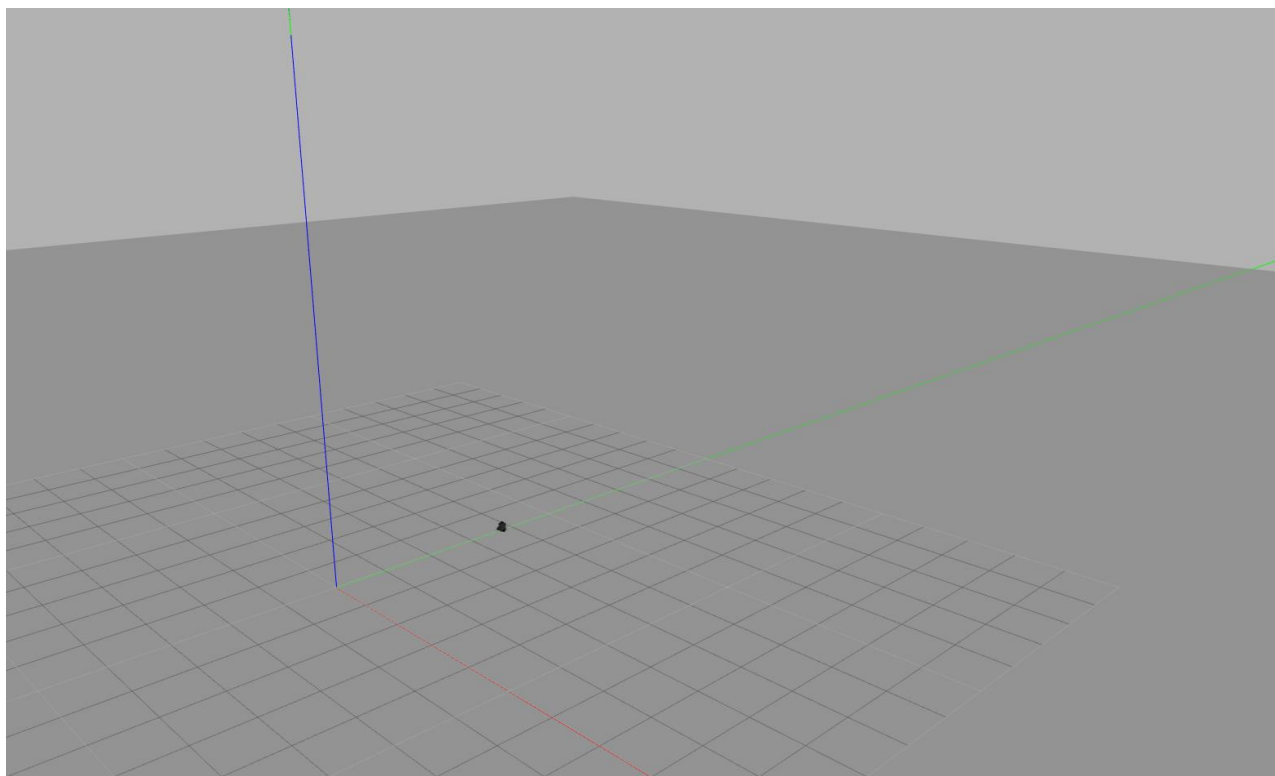
Μετά τη `for` loop, σταματώ το ρομπότ.

Σημείωση: Εξαιτίας της ταχύτητας του ρομπότ, σταματάει λίγο μετρά την θεμιτή απόσταση και η περιστροφική κίνηση δεν είναι ακριβώς 90 μοίρες οπότε φεύγει λίγο από τις γραμμές

Κάποιες εικόνες από τη 2:







Μέρος Β

3)

Κι εδώ χρησιμοποιώ κάποια έτοιμα πράγματα από τα προηγούμενα δυο ερωτήματα, όπως το callback, το subscriber και τον publisher και τις global μεταβλητές.

Αρχικά το πρόγραμμα ζητάει συντεταγμένες από το χρήστη. Έπειτα η στρατηγική επίλυσης είναι απλή. Γυρνάμε το ρομπότ ωσότου να κοιτάει τη θέση στόχο του με μια μικρή απόκλιση 0.1 rad. Όταν τη φτάσει τότε απλά πηγαίνουμε ευθεία στο στόχο.

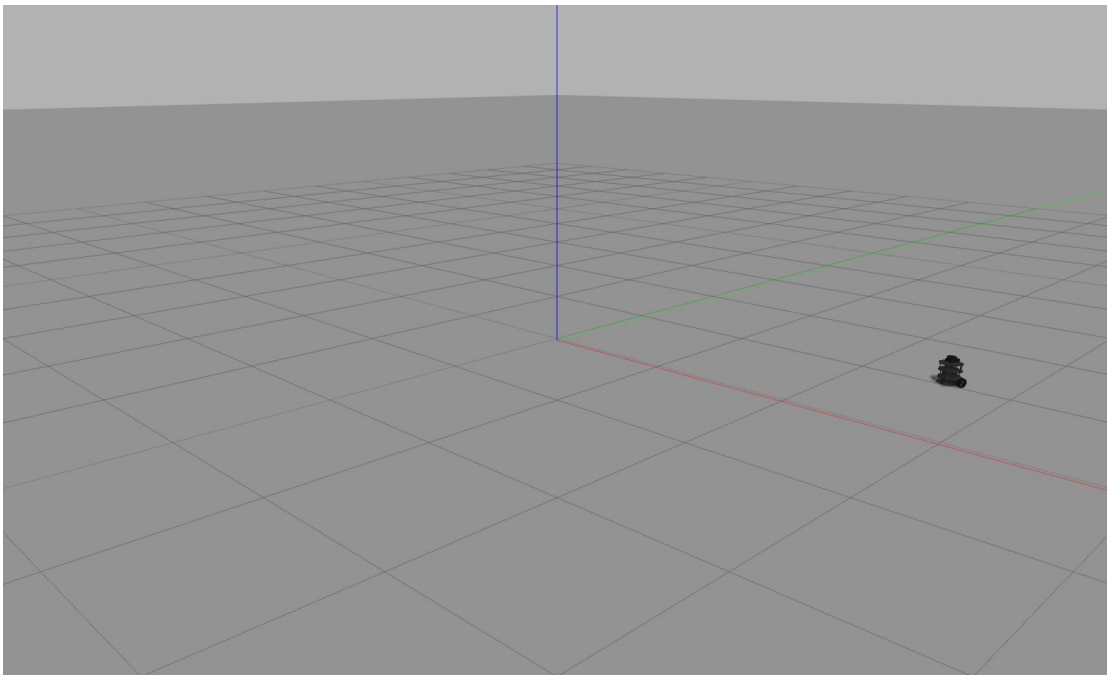
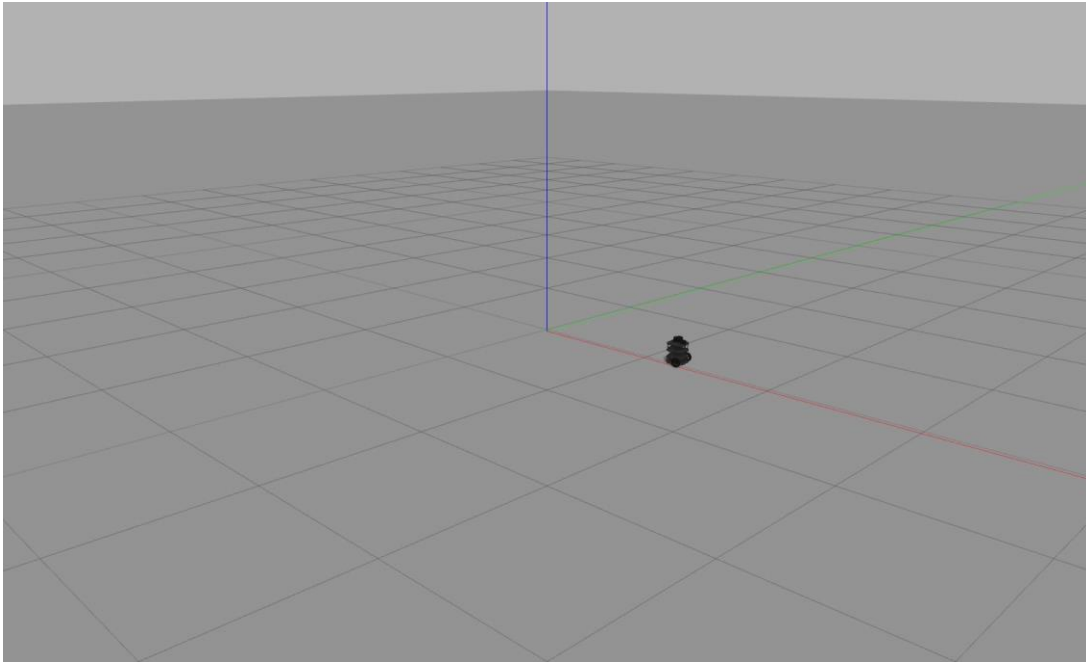
Υπολογίζω κάθε φορά την απόσταση που απομένει αφαιρώντας τη θέση του ρομπότ από τη θέση-στοχο, ώστε να υπολογίζω τη νέα γωνιά

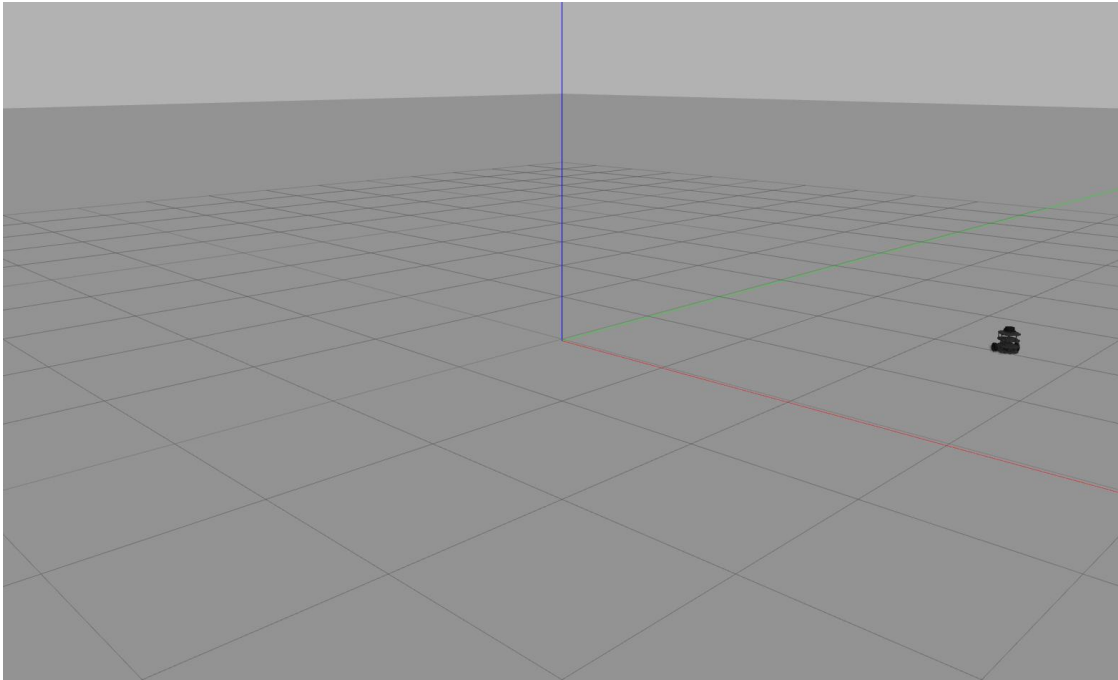
Για να υπολογίσω τη γωνιά που απομένει κάθε φορά χρησιμοποιώ την atan2 με παραμέτρους το απομένων y και x

Σημείωση: δεν έχω εισαγάγει κώδικα για να ελέγχει το σταματημό του ρομπότ, οπότε συνεχίζει να περιστρέφεται.

Κάποιες εικόνες από τη 3:

Θεση στοχος: $x=2, y=2$





F

Τελικη θέση: $x=2, y=2$

