

ΡΟΜΠΟΤΙΚΗ II

8ο ΕΞΑΜΗΝΟ 2019-2020

1η Εξαμηνιαία Εργασία

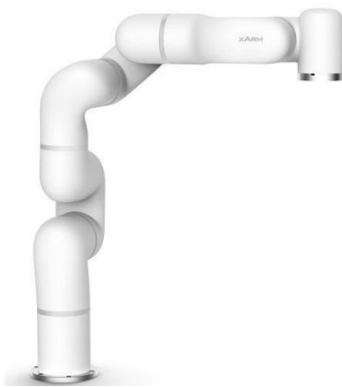
Καθηγητής: Κωνσταντίνος Τζαφέστας

Φοιτητές: Γεώργιος-Ταξιάρχης Γιαννιός, Α.Μ.:03116156
gianniosgeorgios45@gmail.com

Δημήτριος Κρασιάς, Α.Μ.:03116030
dimitriskranias@gmail.com

Περιεχόμενα:

Μέρος Α: Θεωρητική Ανάλυση



Στόχος της Εργασίας

Επεξήγηση του Αλγορίθμου

Kinematics.py

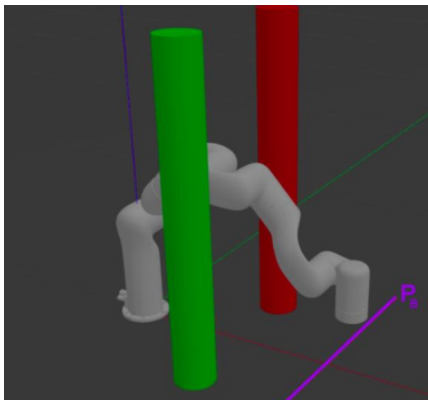
Controller.py

1η Υποεργασία

2η Υποεργασία

Ρύθμιση Κερδών

Μέρος Β: Προσομοίωση



Υπολογισμός Αρχικής Θέσης Βραχίονα

Γραφικές Παραστάσεις¹

Αρχική Θέση Εμποδίων

Πράσινο στη θέση -0.1

Πράσινο στη θέση -0.05

Κόκκινο στη θέση 0.1

Κόκκινο στη θέση 0.05

¹ Οι γραφικές παραστάσεις αφορούν α) Σφάλματα τροχιάς - Θέσεις εμποδίων, β) Θέση (x,y,z) end-effector, γ) Αποστάσεις σημείων που χρησιμοποιήθηκαν στο κριτήριο απο εμποδία

Μέρος Α: Θεωρητική Ανάλυση

Στόχος της Εργασίας

Στόχος της εργασίας αυτής είναι η υλοποίηση ενός αλγορίθμου στο περιβάλλον ROS και η προσομοίωση του στο Gazebo, για το xArm 7, που να μπορεί να πραγματοποιεί μια περιοδική κίνηση πάνω στην ευθεία $\{x=0.6043|z=0.1508\}$, μήκους 0.4m και συμμετρική ως προς τον άξονα y (δηλαδή από το -0.2 έως το +0.2), αλλά και να αποφεύγει τα εμπόδια-κυλίνδρους τα οποία απέχουν 0.4m μεταξύ τους και κινούνται στο επίπεδο $x=0.3$.

Επεξήγηση του Αλγορίθμου

Kinematics.py

Ο βραχίονας περιγράφεται μαθηματικά από τον ακόλουθο πίνακα D.H (Σύμβαση βιβλίου John Craig):

α_{i-1} $Rot(x_{i-1})$	a_{i-1} $Tra(x_{i-1})$	d_i $Tra(z_i)$	θ_i $Rot(z_i)$
0	0	l_1	q_1
$-\pi/2$	0	0	q_2
$\pi/2$	0	l_2	q_3
$\pi/2$	l_3	0	q_4
$\pi/2$	x	y	q_5
$\pi/2$	0	0	q_6
$-\pi/2$	z	w	q_7

Όπου: $x = l_4 \sin(q_1)$, $y = l_4 \cos(q_1)$, $z = l_5 \sin(q_2)$, $w = l_5 \cos(q_2)$

Αρχικά, λοιπόν υπολογίσαμε με τη χρήση *Matlab* τα ομογενή μητρώα μετασχηματισμού από το πλαίσιο i-1 στο πλαίσιο i. Αυτό έγινε εφικτό δημιουργώντας μια συνάρτηση **Transfer.m** η οποία δέχόταν 4 ορίσματα ($\alpha_{i-1}, a_{i-1}, d_i, \theta_i$) και επέστρεφε έναν πίνακα 4x4 που αντιστοιχούσε στον πίνακα A_{i-1}^i σύμφωνα με τη σχέση:

$$A_{i-1}^i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \sin(\theta_i) \cdot \cos(a_{i-1}) & \cos(\theta_i) \cdot \cos(a_{i-1}) & -\sin(a_{i-1}) & -\sin(a_{i-1}) \cdot d_i \\ \sin(\theta_i) \cdot \sin(a_{i-1}) & \cos(\theta_i) \cdot \sin(a_{i-1}) & \cos(a_{i-1}) & \cos(a_{i-1}) \cdot d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(Στο αρχείο *Transfer.m*, όπου υπήρχε ο όρος $\cos()$ χρησιμοποιήσαμε και την $\text{round}()$ ώστε να εμφανιστεί μηδενικό αποτέλεσμα)

Στη συνέχεια πραγματοποιήθηκε η εύρεση της Ιακωβιανής (**Jacobian.m**) με παραγωγή. Πιο ειδικά αφού πολλαπλασιάσαμε διαδοχικά όλα τα επιμέρους μητρώα, υπολογίσαμε τον πίνακα A^0_7 και απομονώσαμε τα τρία πρώτα στοιχεία της τέταρτης στήλης που αντιστοιχούσαν στις μεταβλητές p_{ex}, p_{ey}, p_{ez} του end-effector (αντίστοιχα). Έπειτα σύμφωνα με τη σχέση:

$$J = \begin{bmatrix} \frac{\partial p_{ex}}{\partial q_1} & \frac{\partial p_{ex}}{\partial q_2} & \cdots & \frac{\partial p_{ex}}{\partial q_7} \\ \frac{\partial p_{ey}}{\partial q_1} & \frac{\partial p_{ey}}{\partial q_2} & \cdots & \frac{\partial p_{ey}}{\partial q_7} \\ \frac{\partial p_{ez}}{\partial q_1} & \frac{\partial p_{ez}}{\partial q_2} & \cdots & \frac{\partial p_{ez}}{\partial q_7} \end{bmatrix} (3 \times 7)$$

υπολογίστηκε η Ιακωβιανή Ορίζουσα.

Όλα τα παραπάνω συμπεριλαμβάνονται στα αρχεία .m (*Transfer.m*, *Jacobian.m*)

Controller.py

Αρχικά πραγματοποιήθηκε **path planning** (offline) ώστε να προσδιοριστεί η θέση $y(t)$ του τελικού σημείου (για κίνηση περιόδου T). Μιας και ήταν επιθυμητή τόσο η συνέχεια στην ταχύτητα μεταξύ Αρχικής (y_0) και Τελικής Θέσης (y_f) αλλά και ο μηδενισμός της ταχύτητας στα ακραία σημεία, χρησιμοποιήθηκε το ακόλουθο πολυώνυμο 3^{ου} βαθμού:

$$y(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

όπου,

$$a_0 = y_0$$

$$a_1 = \dot{y}_0$$

$$a_2 = \frac{3}{T^2}(y_f - y_0) - \frac{2}{T}\dot{y}_0 - \frac{1}{T}\dot{y}_f$$

$$a_3 = -\frac{2}{T^3}(y_f - y_0) + \frac{1}{T^2}(\dot{y}_0 + \dot{y}_f)$$

Έπειτα για να πραγματοποιήσουμε τα ζητούμενα, χρειάστηκε να διασπάσουμε την ρομποτική εργασία σε δύο επιμέρους υποεργασίες. Η **πρώτη** υποεργασία έχει σαν στόχο την εκτέλεση της περιοδικής κίνησης, ενώ η **δεύτερη** την αποφυγή εμποδίων.

1η Υποεργασία

Πιο συγκεκριμένα για την πρώτη υποεργασία χρησιμοποιήσαμε την παρακάτω εξίσωση:

$$task_1 = J^+(v_{1d} + K_1(p_{1d} - f_1(q)))$$

- ❑ J^+ : Η ψευδοαντίστροφη της Jacobian
- ❑ v_{1d} : Η επιθυμητή ταχύτητα του end-effector για την επόμενη χρονική στιγμή (που έχουμε υπολογίσει από το path planning)
- ❑ K_1 : Το κέρδος που προκύπτει πειραματικά ώστε να μην παρεκκλίνει από την τροχιά ο end-effector του βραχίονα
- ❑ p_{1d} : Η επιθυμητή θέση του end-effector για την επόμενη χρονική στιγμή (ομοίως προκύπτει από το path planning)
- ❑ $f_1(q)$: Η πραγματική θέση του end-effector

Το κέρδος K_1 και η διαφορά μεταξύ desired και real position τοποθετήθηκαν για να μην παρεκκλίνει ο end-effector (λόγω σφαλμάτων) από την επιθυμητή τροχιά.

2η Υποεργασία

Στόχος της δεύτερης υπερεργασίας ήταν η αποφυγή των κινούμενων εμποδίων. Για να γίνει αυτό έπρεπε ο ελεγκτής να στείλει στον ρομποτικό βραχίονα

ένα διάνυσμα \dot{q}_0 το οποίο θα είναι το οποίο ισούται με :

$$\dot{q}_0 = k \cdot \nabla_q[V(q)]$$

Όπου $V(q)$ συνάρτηση κριτηρίου που επιθυμούμε να βελτιστοποιήσουμε. Εμείς για κάθε joint επιλέξαμε την:

$$V_i(q) = \frac{1}{2} K_i [y_i(q) - y_{obs}]^2$$

- ❑ K_i : Κέρδος που υπολογίστηκε πειραματικά
- ❑ y_i : y-συνιστώσα του joint i

- y_{obs} : y -συνιστώσα του **κέντρου** μεταξύ των εμποδίων (δεν είναι ακριβώς το κέντρο αλλά λίγο μετατοπισμένες τιμές ανάλογα με την κατεύθυνση κίνησης του ρομπότ για καλύτερη αποφυγή εμποδίων).

Οπότε η εξίσωση για τη 2^η υποεργασία είναι σύμφωνα με τη θεωρία:

$$\text{task}_2 = (I - J^+ \cdot J) \cdot \xi_3 + (I - J^+ \cdot J) \cdot \xi_4 \text{ στην οποία:}$$

$$\xi_3 = - \left[\frac{\partial V_3(q)}{\partial q_1} \quad \frac{\partial V_3(q)}{\partial q_2} \quad \frac{\partial V_3(q)}{\partial q_3} \quad \frac{\partial V_3(q)}{\partial q_4} \quad \frac{\partial V_3(q)}{\partial q_5} \quad \frac{\partial V_3(q)}{\partial q_6} \quad \frac{\partial V_3(q)}{\partial q_7} \right]^T = -K_3 \begin{bmatrix} l_2 \cos(q_1) \cdot \sin(q_2) \\ l_2 \cos(q_2) \cdot \sin(q_1) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

και

$$\xi_4 = - \left[\frac{\partial V_4(q)}{\partial q_1} \quad \frac{\partial V_4(q)}{\partial q_2} \quad \frac{\partial V_4(q)}{\partial q_3} \quad \frac{\partial V_4(q)}{\partial q_4} \quad \frac{\partial V_4(q)}{\partial q_5} \quad \frac{\partial V_4(q)}{\partial q_6} \quad \frac{\partial V_4(q)}{\partial q_7} \right]^T = -K_4 \begin{bmatrix} l_2 \cos(q_1) \sin(q_2) - l_3 [\sin(q_1) \sin(q_3) - \cos(q_1) \cos(q_2) \cos(q_3)] \\ l_2 \cos(q_2) \sin(q_1) - l_3 \cos(q_3) \sin(q_1) \sin(q_2) \\ l_3 [\cos(q_1) \cos(q_3) - \cos(q_2) \sin(q_1) \sin(q_3)] \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Ο υπολογισμός για παράδειγμα του $\frac{\partial V_4}{\partial q_3}$ πραγματοποιήθηκε ως εξής :

$$\frac{\partial V_4}{\partial q_3} = \frac{\partial}{\partial q_3} \left(\frac{1}{2} (y_3(q) - y_{\text{obs}})^2 \right) \xrightarrow{y_{\text{obs}} \text{ ανεξάρτητο } q}$$

$$\frac{\partial V_4}{\partial q_3} = y_3(q) \cdot \frac{\partial y_3(q)}{\partial q_3} = A_3^0[1,3] \cdot \frac{\partial A_3^0[1,3]}{\partial q_3}, \text{ όπου } A_3^0[1,3] \text{ το στοιχείο } p_{y_3}$$

(Ομοίως και τα υπόλοιπα)²

Με το παραπάνω κριτήριο εξασφαλίζουμε ότι οι αρθρώσεις **3** και **4** θα βρίσκονται πάνω στην ευθεία η οποία:

- 1.Είναι παράλληλη στο επίπεδο z
- 2.Είναι κάθετη στην ευθεία που συνδέει τα εμπόδια
- 3.Διέρχεται από το σημείο τομής τους

Ρύθμιση Κερδών

Βέβαια, για να γίνει αυτό, χρειάζεται να βάλουμε και τα κατάλληλα κέρδη. Για **μεγάλα κέρδη**, έχουμε πιο καλή ανταπόκριση στην αποφυγή των εμποδίων, αλλά στις ακραίες θέσεις (δηλαδή όταν το κόκκινο εμπόδιο βρίσκεται σε τιμή του y μικρότερη του 0.05 ή το πράσινο σε τιμή του y μεγαλύτερη του -0.05) τείνει να μην κάνει σωστά την ταλάντωση (για να αποφύγει τυχούσες ιδιομορφίες). Για **μικρά κέρδη**, για γρήγορη μεταβολή της κατεύθυνσης κίνησης των εμποδίων, δεν προλαβαίνει ο ρομποτικός βραχίονας να αντιδράσει και χτυπάει στα εμπόδια.

Οπότε, εμείς μετά από αρκετές δοκιμές, καταλήξαμε στις τιμές που βρίσκονται στον κώδικα έτσι ώστε ανάμεσα στα όρια που είπαμε και παραπάνω (κόκκινο εμπόδιο έως και θέση 0.05 και πράσινο έως -0.05), το ρομπότ να αποφεύγει τα εμπόδια και να ακολουθεί την σωστή τροχιά. Σε ελάχιστες περιπτώσεις (π.χ. αν ο ρομποτικός βραχίονας βρίσκεται αρκετά κοντά σε ένα εμπόδιο και αυτό αλλάξει κατεύθυνση κίνησης και κινείται με μεγάλη ταχύτητα) ίσως χτυπήσει, αν και στις προσομοιώσεις του Gazebo δεν παρατηρήσαμε κάτι τέτοιο.

Να σημειωθεί ότι το task2 δεν εκτελείται συνεχώς, αλλά εξαρτάται από την διαφορά του μεγίστου των κριτηρίων από μια σταθερή διαφορά που επιλέχθηκε πειραματικά (στον κώδικα ορίζεται έτσι: $\max(\text{joint_distance_3}, \text{joint_distance_4}) \geq 0.03$). Αυτό το “**φίλτρο**” υπάρχει για να μην ασκούνται αντίθετες δυνάμεις στον βραχίονα και να τον καταπονούν, όταν αυτό δεν είναι αναγκαίο.

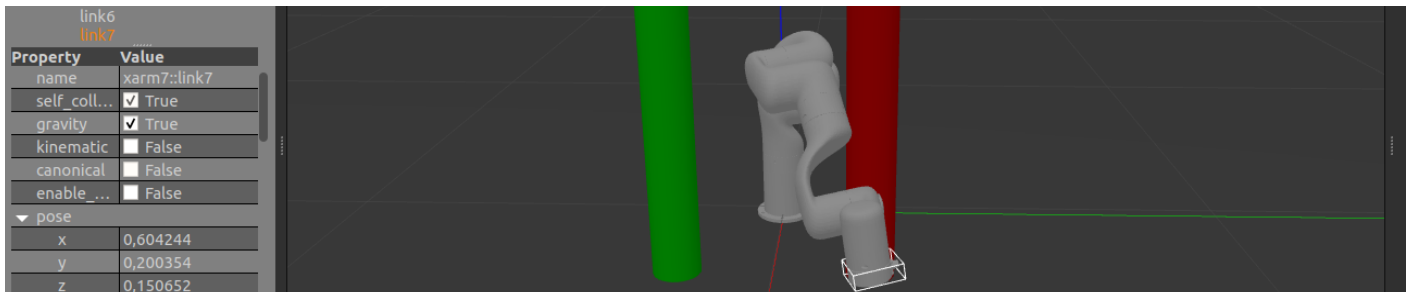
Μέρος B: Προσομοίωση

Υπολογισμός Αρχικής Θέσης Βραχίονα:

Προκειμένου να αποφύγουμε επαναλαμβανόμενο κώδικα (ένα path planning απο την αρχική θέση μέχρι το άκρο και ένα δεύτερο μεταξύ των αρθρώσεων),

² Στο τέλος του κώδικα **Jacobian.m**, υπολογίστηκαν και τα παραπάνω gradients

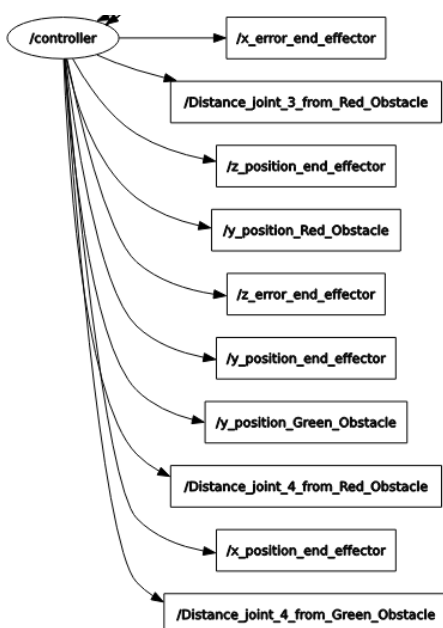
κρίναμε αναγκαίο ο βραχίονας να εκκινεί από την ακόλουθη θέση (οι συντεταγμένες του End-Effector φαίνονται στο αριστερό μέρος της παρακάτω εικόνας).



Για τον υπολογισμό της αρχικής θέσης του βραχίονα (δηλαδή της θέσης από όπου θα ξεκινήσει η ταλάντωση) χρησιμοποιήσαμε το ήδη υπάρχον path-planning (με την διαφορά ότι θέλουμε να πάμε από το **κέντρο** στην **θέση 0.2**, (αντί για την θέση -0.2 από την θέση 0.2) και το κριτήριο αποφυγής εμποδίου. Έτσι προέκυψαν οι τιμές των αρθρώσεων που χρησιμοποιούμε στην αρχή της συνάρτησης publish. Τις τιμές αυτές τις κάναμε activate με την εντολή (για κάθε άρθρωση i):

```
print(self.joint_angpos[i])
```

Γραφικές Παραστάσεις



Προκειμένου να απεικονίσουμε γραφικά κάποιες μεταβλητές του προγράμματος, δημιουργήσαμε topics και κάναμε publish τις τιμές των επιθυμητών μεταβλητών. Αυτό επιβεβαιώθηκε εκτελώντας σε ένα νέο τερματικό την εντολή:

```
$ rosrun rqt_graph rqt_graph
```

,όπου βλέπουμε ότι δημιουργήθηκαν τα νέα topics

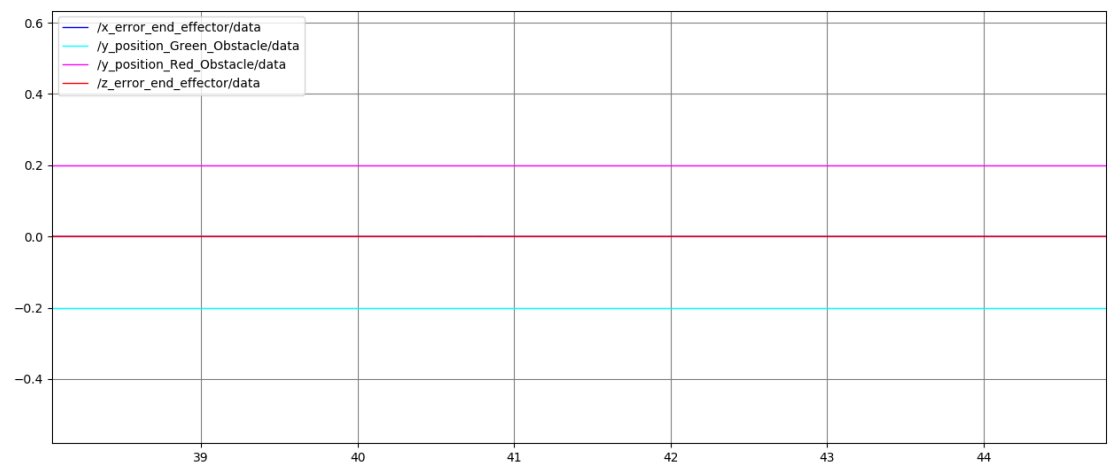
Στη συνέχεια εκτελώντας την εντολή

```
$ rqt_plot rqt_plot
```

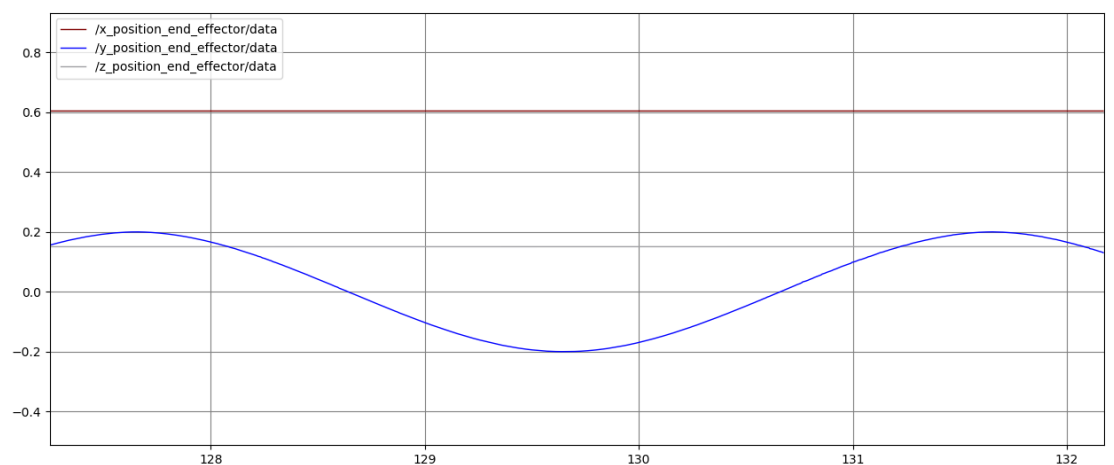
επιλέξαμε τις ακόλουθες γραφικές

❑ Αρχική Κατάσταση Εμποδίων:

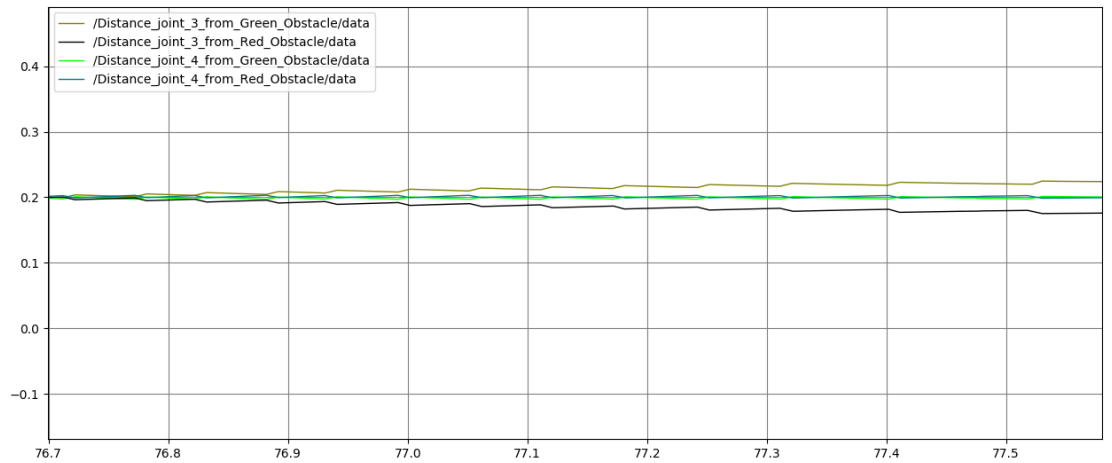
❑ Σφάλματα Τροχιάς και Θέσεις Εμποδίων



❑ Θέσεις End-Effector



❑ Αποστάσεις Αρθρώσεων 3 και 4 από Εμπόδια

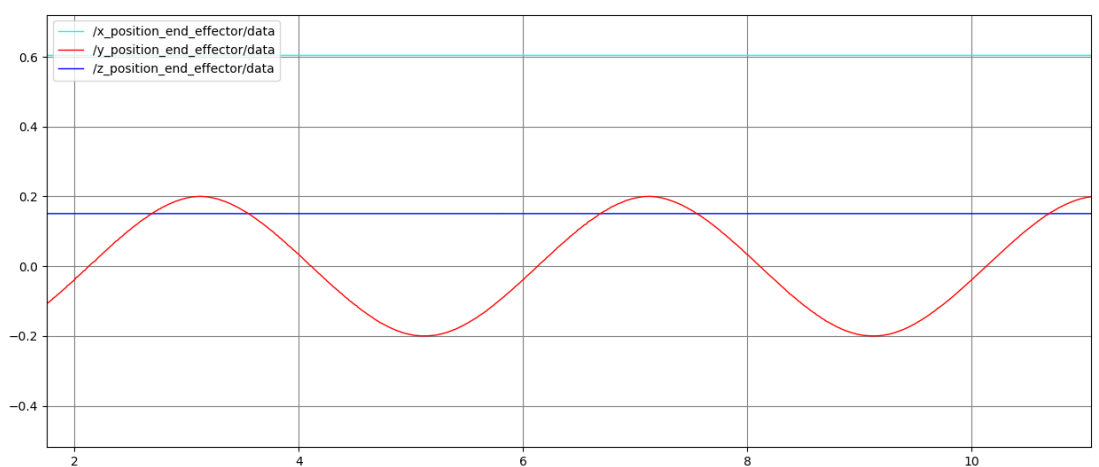


Πράσινο Εμπόδιο στη θέση -0.1

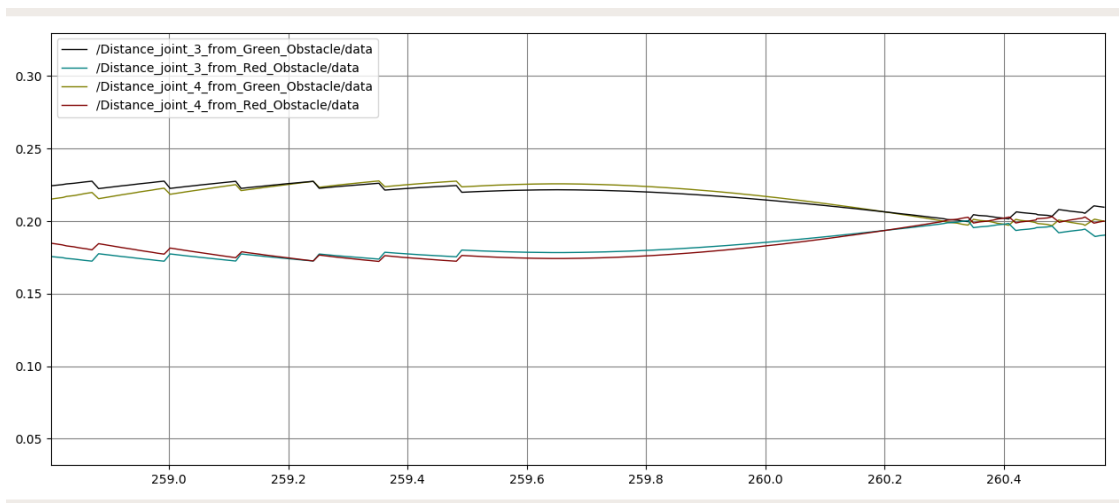
Σφάλματα Τροχιάς και Θέσεις Εμποδίων



Θέσεις End-Effector

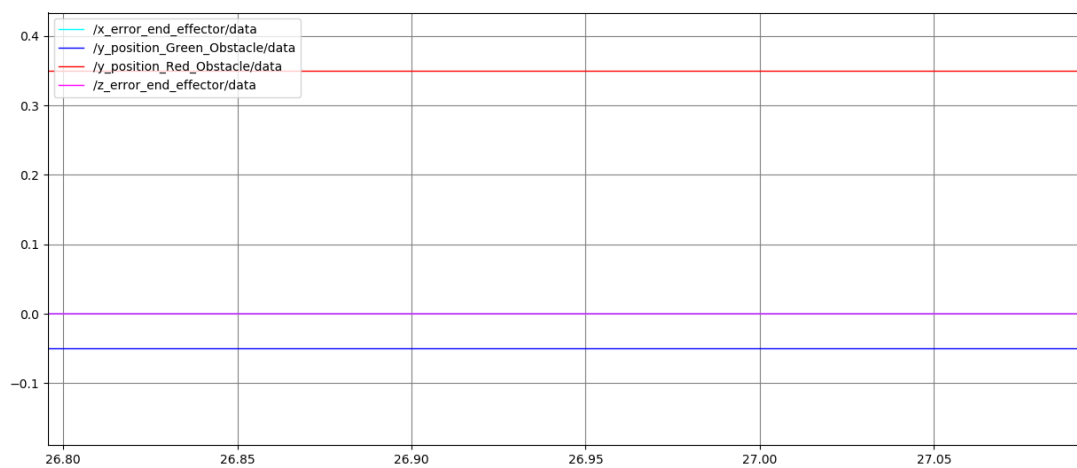


Αποστάσεις Αρθρώσεων 3 και 4 από Εμπόδια

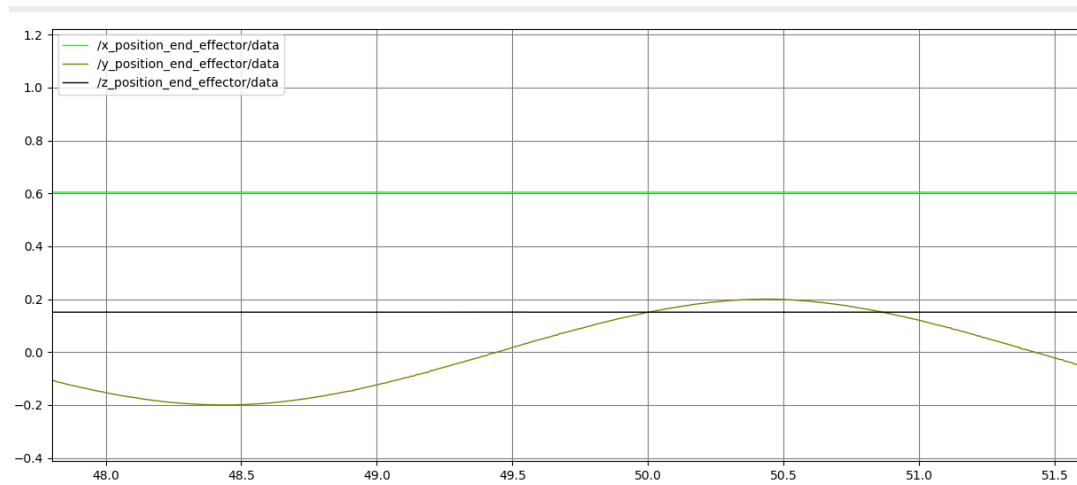


Πράσινο Εμπόδιο στη θέση -0.05

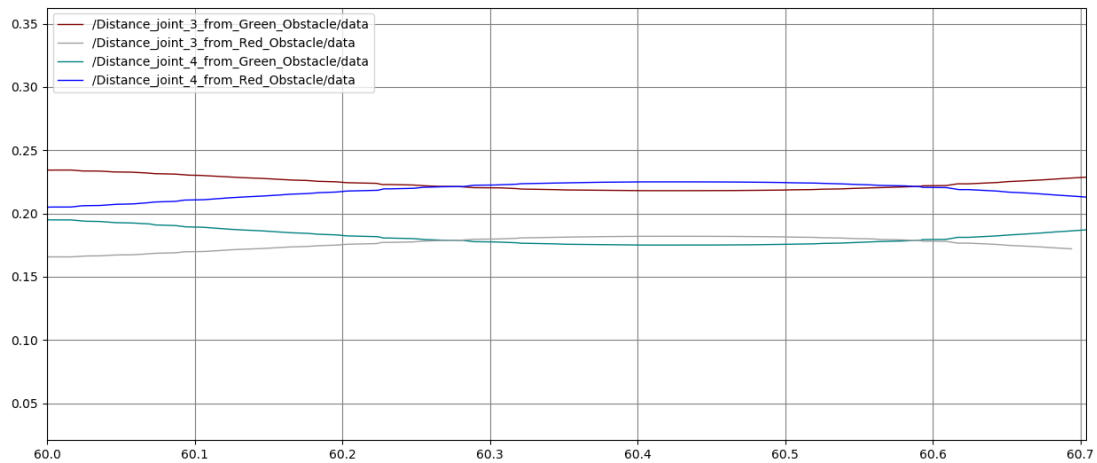
Σφάλματα Τροχιάς και Θέσεις Εμποδίων



Θέσεις End-Effector

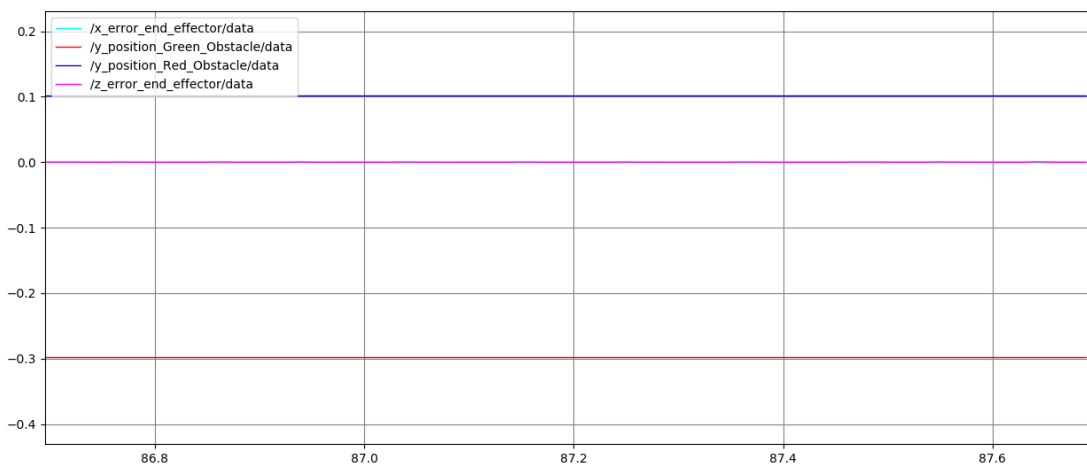


Αποστάσεις Αρθρώσεων 3 και 4 από Εμπόδια

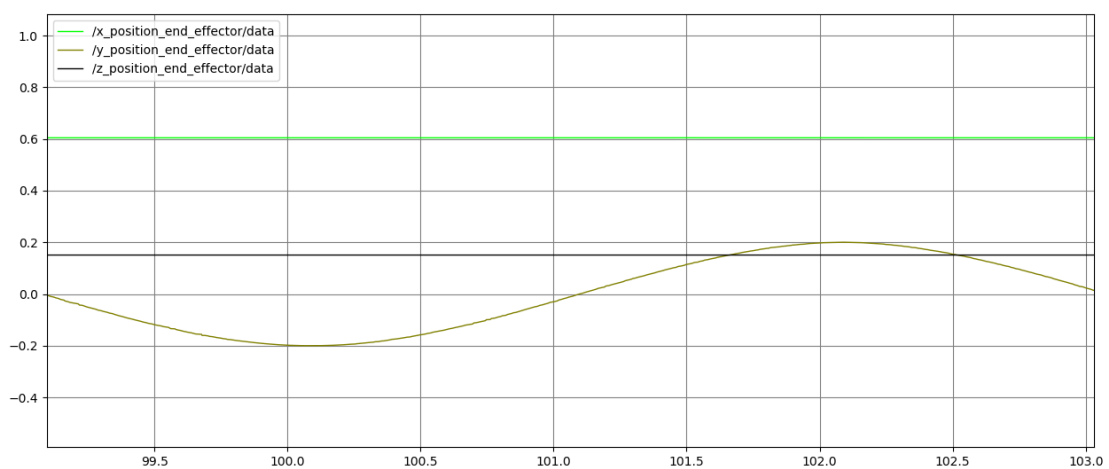


❑ Κόκκινο Εμπόδιο στη θέση +0.1

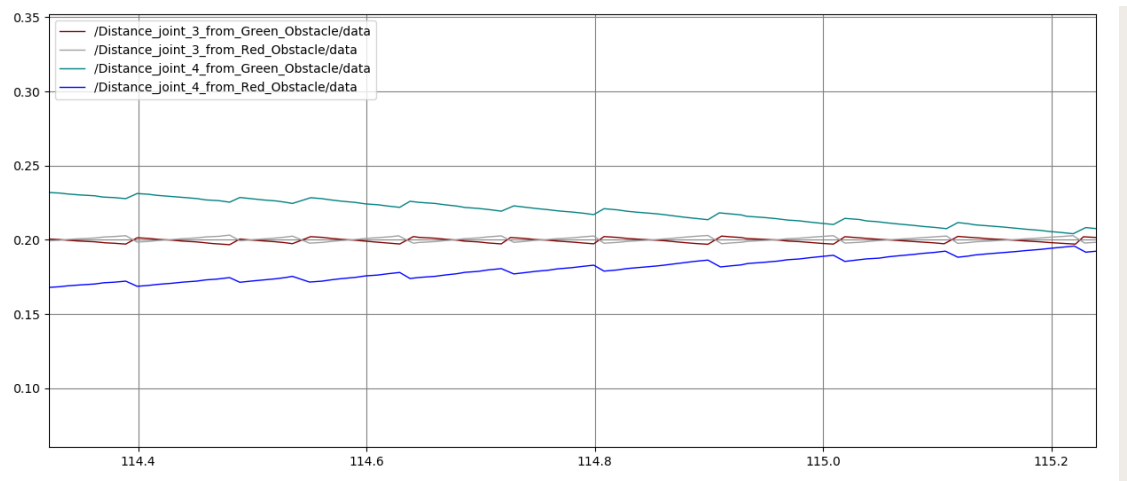
❑ Σφάλματα Τροχιάς και Θέσεις Εμποδίων



❑ Θέσεις End-Effector

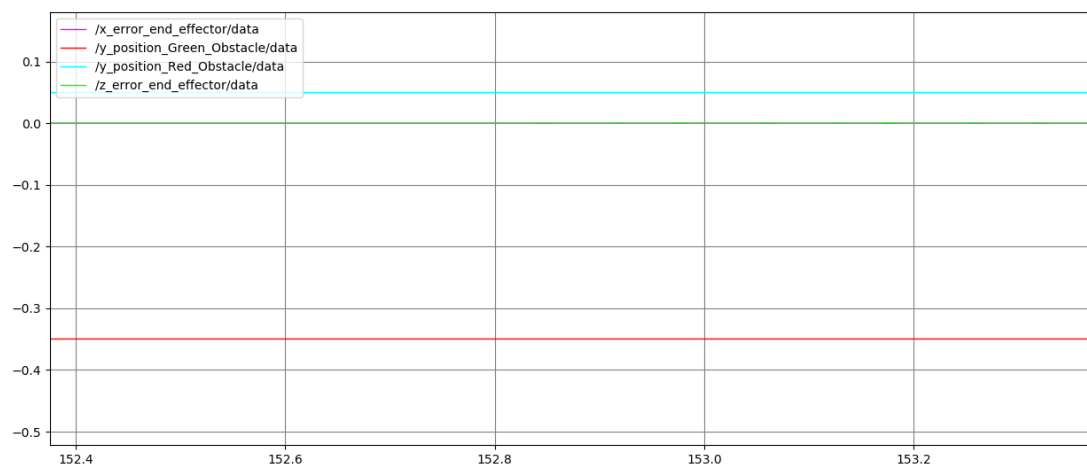


❑ Αποστάσεις Αρθρώσεων 3 και 4 από Εμπόδια

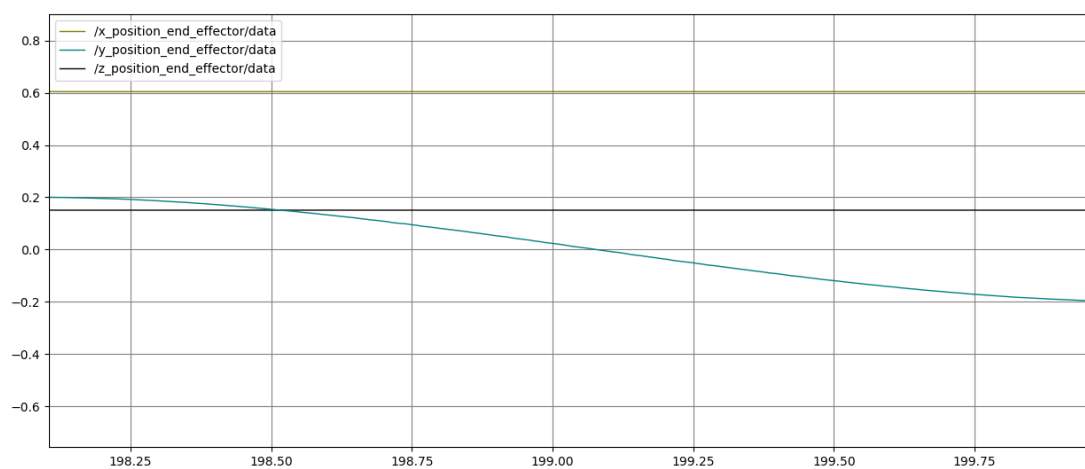


❑ Κόκκινο Εμπόδιο στη θέση +0.05

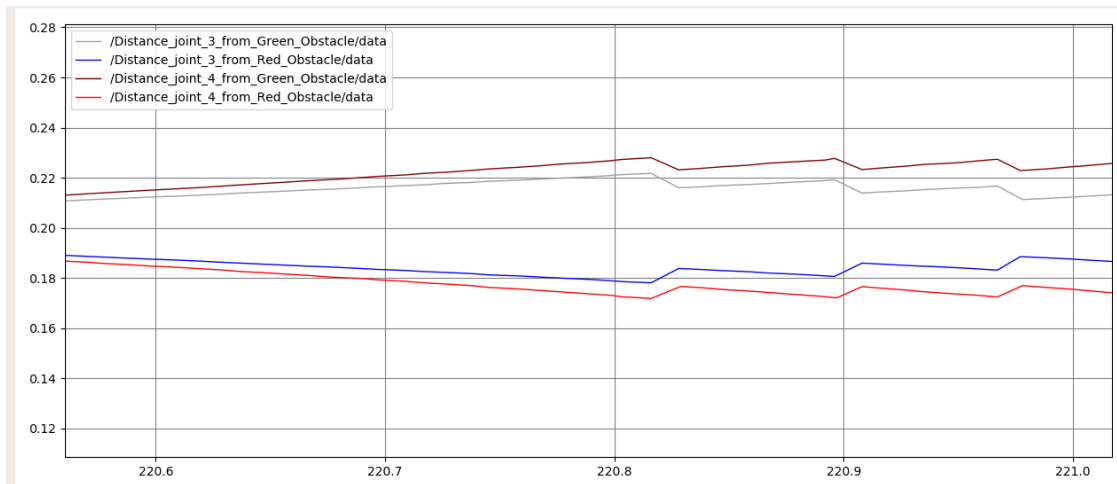
❑ Σφάλματα Τροχιάς και Θέσεις Εμποδίων



❑ Θέσεις End-Effector



❑ Αποστάσεις Αρθρώσεων 3 και 4 από Εμπόδια



Σχολιασμός Γραφικών Αποτελεσμάτων:

Όσον αφορά τα σφάλματα τροχιάς, όπως βλέπετε, είναι σχεδόν μηδενικά (δηλαδή πληρούν τις προδιαγραφές για σφάλμα μισού χιλιοστού). Ίσως σε κάποιες ιδιαίτερες περιπτώσεις (χωρίς να είναι σταθερά τα εμπόδια), να ξεφεύγουν λίγο, αλλά στις προσομοιώσεις μας δεν φάνηκε κάτι τέτοιο.

Οι θέσεις του end-effector (x, y, z συνιστώσες) δείχνουν πάλι ότι δεν υπάρχει σφάλμα στις συνιστώσες x, z, αλλά και φαίνεται η περιοδική κίνηση στον άξονα y.

Τέλος, οι γραφικές παραστάσεις που αφορούν τις αποστάσεις των joint 3 και joint 4 από τα εμπόδια (μόνο στην y-συνιστώσα) είναι λίγο πιο ιδιαίτερες από τις προηγούμενες, διότι φαίνεται να μην πλησιάζουν καθόλου στα εμπόδια (η μικρότερη απόσταση είναι περίπου 0.15). Αυτό όμως οφείλεται στο γεγονός ότι οι αισθητήρες των joints βρίσκονται στο κέντρο των links, ενώ τα links εκτείνονται και προς τα αριστερά και προς τα δεξιά. Αυτό δείχνει την ανάγκη να χρησιμοποιηθεί το κριτήριο μας (δηλαδή ενδιάμεσο των 2 εμποδίων) παρά απόσταση από τα εμπόδια καθεαυτά. Μπορεί σε κάποιες περιπτώσεις να περιορίζει τις κινήσεις των joints, αλλά είναι ένα αναγκαίο κακό για να γίνει καλύτερη και σιγουρότερη η αποφυγή των εμποδίων.

Σημείωση:

Να σημειωθεί ότι τα διαγράμματα έγιναν μόνο σε fixed θέσεις γιατί ήταν πιο εύκολη η αποθήκευση των γραφικών παραστάσεων. Σε περιπτώσεις κίνησης των εμποδίων θα ήταν σχεδόν αδύνατον να αποθηκευτούν ταυτόχρονα 3 γραφικές παραστάσεις για την ίδια χρονική στιγμή. Παρ' όλα αυτά μελετήθηκαν και τέτοιες περιπτώσεις για να παρατηρηθούν τυχόντα σφάλματα ή περίεργα αποτελέσματα.