

Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Εξαμηνιαία Εργασία 1

**Κινηματικός έλεγχος ρομποτικού χειριστή με πλεονάζοντες βαθμούς ελευθερίας:
Παρακολούθηση τροχιάς και αποφυγή εμποδίου**

Μάθημα: Ρομποτική II: Ευφυή Ρομποτικά Συστήματα

Εξάμηνο: 8^ο

Ονοματεπώνυμο: Αλεξοπούλου Γεωργία, Γκενάκου Ζωή

Εισαγωγή

A. Το Ρομπότ

Τα collaborative ρομπότ, ή cobots, έχουν σχεδιαστεί για άμεση και ασφαλή αλληλεπίδραση με τον άνθρωπο, σε αντίθεση με τα παραδοσιακά βιομηχανικά ρομπότ που λειτουργούν αυτόνομα, συνήθως σε περιβάλλοντα απομονωμένα από την ανθρώπινη παρουσία. Αυτή η ικανότητα διευκολύνεται από διάφορα χαρακτηριστικά, όπως εξοπλισμένους αισθητήρες, περιορισμένα όρια ταχύτητας και δύναμης και συγκεκριμένες ιδιότητες υλικού που ενσωματώνονται στην κατασκευή τους. Επιπλέον, η προσιτή τιμή τους τα καθιστά βιώσιμες λύσεις για μικρομεσαίες επιχειρήσεις.

Στο Σχήμα 1 απεικονίζεται το xArm 7, που ανήκει στην κατηγορία cobot, ικανό να χειρίζεται ωφέλιμα φορτία έως 3,5 kg με εύρος εργασίας που εκτείνεται 70 cm από τη βάση του. Μπορεί να επιτύχει μέγιστη ταχύτητα 1 m/s στον τελικό του τελεστή, με 7 βαθμούς ελευθερίας (DOFs) όπως υποδηλώνει το όνομά του. Επί του παρόντος, βρίσκει εφαρμογές όχι μόνο στην ακαδημαϊκή έρευνα αλλά και σε βιομηχανικά περιβάλλοντα,



Εικόνα 1 – Cobot xArm 7

ενσωματωμένα ως αυτόνομα συστήματα σε γραμμές παραγωγής. Ένας σημαντικός παράγοντας που συμβάλλει στον αναμενόμενο αντίκτυπό του στο εγγύς μέλλον, εκτός από την οικονομική του απόδοση, είναι η συμβατότητά του με το ευρέως χρησιμοποιούμενο περιβάλλον ROS (Robot Operating System) για την ανάπτυξη εφαρμογών ρομποτικής.

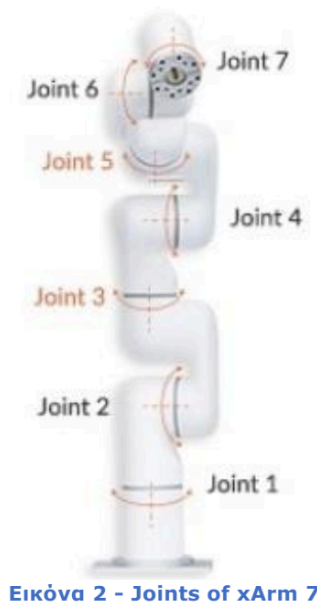
Στόχος της εργασίας είναι η υλοποίηση ενός αλγορίθμου για την παρακολούθηση τροχιάς από το τελικό στοιχείο δράσης του xArm 7 με ταυτόχρονη αποφυγή εμποδίων.

Στην συνέχεια θα παρατίθεται η θεωρητική ανάλυση που πραγματοποιήθηκε για το κομμάτι της προσομοίωσης.

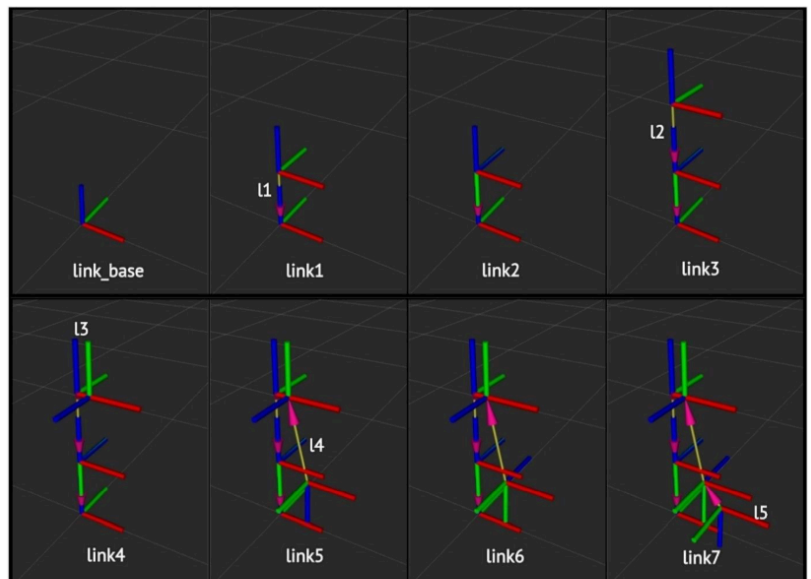
B. Θεωρητική Ανάλυση

Ευθεία Κινηματική Ανάλυση

Το πρώτο βήμα της θεωρητικής ανάλυσης είναι η ευθεία κινηματική ανάλυση του ρομποτικού χειριστή. Το Σχήμα 2 απεικονίζει τις θέσεις των αρθρώσεων του ρομποτικού βραχίονα και τις αντίστοιχες κατευθύνσεις περιστροφής τους. Με βάση την εικόνα αυτή και τα πλαίσια αναφοράς που δίνονται στην εκφώνηση της άσκησης (Εικόνα 3), παράγουμε τους πίνακες της ευθείας κινηματικής ανάλυσης.



Εικόνα 2 - Joints of xArm 7



Εικόνα 3 – Reference frames and links of xArm 7 (by Rviz)

Οι παρακάτω διαδοχικοί μετασχηματισμοί και η τοποθέτηση των πλαισίων αναφοράς στους συνδέσμους ακολουθεί ουσιαστικά την εναλλακτική σύμβαση D-H, όπως αυτή περιγράφεται στο βιβλίο του John Craig.

$$A_1^0(q_1) = Rot(z, q_1) \cdot Tra(z, l_1) \rightarrow A_1^0(q_1) = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^1(q_2) = Rot\left(x, -\frac{\pi}{2}\right) \cdot Rot(z, q_2) \rightarrow A_2^1(q_2) = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2(q_3) = Rot\left(x, \frac{\pi}{2}\right) \cdot Rot(z, q_3) \cdot Tra(z, l_2) \rightarrow A_3^2(q_3) = \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ 0 & 0 & -1 & -l_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4^3(q_4) = Rot\left(x, \frac{\pi}{2}\right) \cdot Tra(x, l_3) \cdot Rot(z, q_4) \rightarrow A_4^3(q_4) = \begin{bmatrix} c_4 & -s_4 & 0 & l_3 \\ 0 & 0 & -1 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5^4(q_5) = Rot\left(x, \frac{\pi}{2}\right) \cdot Tra(x, l_4 \sin \theta_1) \cdot Rot(z, q_5) \cdot Tra(z, l_4 \cos \theta_1) \rightarrow A_5^4(q_5) = \begin{bmatrix} c_5 & -s_5 & 0 & l_4 \cdot \sin \theta_1 \\ 0 & 0 & -1 & -l_4 \cdot \cos \theta_1 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6^5(q_6) = Rot\left(x, \frac{\pi}{2}\right) \cdot Rot(z, q_6) \rightarrow A_6^5(q_6) = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_7^6(q_7) = Rot\left(x, -\frac{\pi}{2}\right) \cdot Tra\left(x, l_5 \sin \theta_2\right) \cdot Rot\left(z, q_7\right) \cdot Tra\left(z, l_5 \cos \theta_2\right) \rightarrow A_7^6(q_7) = \begin{bmatrix} c_7 & -s_7 & 0 & l_5 \cdot \sin \theta_2 \\ 0 & 0 & 1 & l_5 \cdot \cos \theta_2 \\ -s_7 & -c_7 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Όπου:

$$l_1 = 26.7cm, l_2 = 29.3cm, l_3 = 5.25cm, l_4 = 35.12cm, l_5 = 12.32cm, \\ \theta_1 = 0.2225 rad, \theta_2 = 0.6646 rad$$

Τους πίνακες αυτούς χρησιμοποιούμε στον κώδικα του kinematics.py για να υπολογίσουμε με την σειρά τους του πίνακες $A_1^0, A_2^0, A_3^0, A_4^0, A_5^0, A_6^0, A_7^0$.

Διαφορική Κινηματική Ανάλυση

Το δεύτερο βήμα της θεωρητικής ανάλυσης είναι η Διαφορική Κινηματική Ανάλυση του ρομποτικού χειριστή. Εξάλλου θα χρειαστεί στον κώδικα kinematics.py να υπολογίσουμε την Ιακωβιανή Μήτρα. Επομένως στο βήμα αυτό υπολογίζουμε όλους τους υποπίνακες J_{L_i} με $i = 1, 2, 3, \dots, 7$, όπου ο καθένας από αυτούς αντιστοιχεί σε μία άρθρωση του ρομπότ και δείχνει την συνεισφορά της στην γραμμική ταχύτητα που αποκτά ο end-effector του ρομπότ. Η διαδικασία υπολογισμού των υποπινάκων υποδεικνύεται παρακάτω:

- Ο τύπος υπολογισμού των υποπινάκων: $J_{L_i} = b_{i-1} \times r_E^{i-1}$, όπου b_{i-1} ο άξονας περιστροφής της άρθρωσης q_i ως προς το πλαίσιο αναφοράς O_0 και r_E^{i-1} η απόσταση του κέντρου O_i της άρθρωσης q_i από το σημείο E όπου αποτελεί το σημείο του τελικού στοιχείου δράσης. Ο άξονας b_{i-1} της εκάστοτε άρθρωσης i θα είναι κάθε φορά ο z άξονας.
- Επομένως το b_{i-1} υπολογίζεται ως εξής: $b_{i-1} = A_i^0[1:3, 3]$, δηλαδή τα τρία πρώτα στοιχεία της τρίτης στήλης του πίνακα A_i^0 για $i = 2, 3, \dots, 7$, ενώ για τον άξονα περιστροφής της πρώτης άρθρωσης ισχύει $b_0 = [0, 0, 1]^T$.

- Τέλος, η απόσταση r_E^{i-1} υπολογίζεται: $r_E^{i-1} = r_E^0 - r_{i-1}^0 = A_7^0[1:3,4] - A_{i-1}^0[1:3,4]$, δηλαδή τα τρία πρώτα στοιχεία της τέταρτης στήλης για του πίνακες A_7^0, A_{i-1}^0 .

Από την Ιακωβιανή μήτρα θα παραλείψουμε να υπολογίσουμε τους υποπίνακες J_{A_i} , καθώς στην συγκεκριμένη εργασία δεν μας ενδιαφέρει η γωνιακή ταχύτητα ή να ελέγξουμε τον προσανατολισμό του συστήματος, αλλά η γραμμική ταχύτητα του τελικού στοιχείου δράσης και ο έλεγχος της θέσης του end-effector.

Επομένως, μετά από πολλές πράξεις, υπολογίζουμε την (ελαττωμένη) Ιακωβιανή μήτρα η οποία έχει την μορφή:

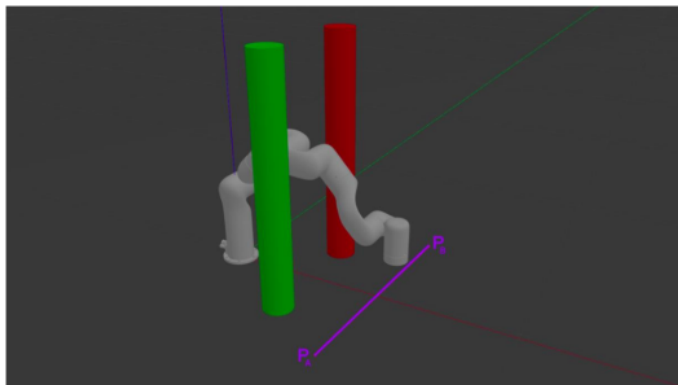
$$J = \begin{bmatrix} J_{L_1} & J_{L_2} & J_{L_3} & J_{L_4} & J_{L_5} & J_{L_6} & J_{L_7} \end{bmatrix}, \text{ όπου } J_{L_i} = \begin{bmatrix} J_{1i} \\ J_{2i} \\ J_{3i} \end{bmatrix},$$

Άρα συνολικά έχουμε:

$$J = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} & J_{15} & J_{16} & J_{17} \\ J_{21} & J_{22} & J_{23} & J_{24} & J_{25} & J_{26} & J_{27} \\ J_{31} & J_{32} & J_{33} & J_{34} & J_{35} & J_{36} & J_{37} \end{bmatrix}$$

Απαιτούμενες Διεργασίες

Όπως αναφέραμε και παραπάνω, ο στόχος της εργασίας είναι η υλοποίηση ενός αλγορίθμου για την παρακολούθηση τροχιάς από το τελικό στοιχείο δράσης της διάταξης με ταυτόχρονη αποφυγή



Εικόνα 4 - xArm 7 within obstacles

εμποδίων. Πιο συγκεκριμένα, το τελικό στοιχείο δράσης του βραχίονα καλείται να εκτελέσει ευθύγραμμη περιοδική κίνηση μεταξύ των θέσεων P_A και P_B , αποφεύγοντας το κόκκινο και πράσινο εμπόδιο, όπως φαίνεται στην Εικόνα 4. Οπότε θα χωρίσουμε την εργασία σε δύο επιμέρους υποεργασίες και θα τις αναλύσουμε ξεχωριστά:

1η Υποεργασία - Σχεδιασμός Τροχιάς End Effector

Στην πρώτη διεργασία, θα ασχοληθούμε με τον σχεδιασμό της τροχιάς. Οι δύο ακραίες θέσεις που ορίζουν το ευθύγραμμο τμήμα κίνησης του τελικού στοιχείου δράσης του ρομπότ είναι συμμετρικές ως προς τον x-άξονα, βρίσκονται στην ευθεία $\{x = 0.617 | z = 0.199\}$, ενώ απέχουν απόσταση 40 cm.

Επομένως, υπολογίζουμε τις συντεταγμένες των σημείων:

$$P_A(0.617, -0.2, 0.199) \text{ και } P_B(0.617, 0.2, 0.199)$$

Αφού επιθυμούμε περιοδική κίνηση μεταξύ των θέσεων αυτών, μπορούμε να χρησιμοποιήσουμε τριγωνομετρικές συναρτήσεις για την έκφραση της συντεταγμένης y. Πιο συγκεκριμένα, στον κώδικα του controller.py ορίζουμε την συντεταγμένη y ως εξής:

$$y = 0.2 \cdot \sin\left(\frac{2\pi}{T} \cdot t\right), \text{ όπου } T \text{ η περίοδος της ταλάντωσης.}$$

Αντίστοιχα, για την ταχύτητα της κίνησης του end-effector, θα παραγωγίσουμε την παραπάνω έκφραση:

$$y = 0.2 \cdot \frac{2\pi}{T} \cdot \cos\left(\frac{2\pi}{T} \cdot t\right)$$

Καθώς το γραμμικό μοντέλο μας είναι μη γραμμικό χρειάζεται να χρησιμοποιήσουμε έναν ελεγκτή θέσης, ώστε να διορθώνονται οι αποκλίσεις που προκύπτουν. Για αυτό τον λόγο, εισάγουμε τον παρακάτω περιορισμό για τις ταχύτητες του end-effector.

$$q1_dot = \text{pinvJ} @ (\text{desired_velocity} + K * (\text{desired_position} - \text{real_position}))$$

Όπου έχουμε ορίσει $\text{pinvJ} = \text{pinv}(J)$ την Moore - Penrose ψευδοαντίστροφη της Ιακωβιανής μήτρας.

2η Υποεργασία - Αποφυγή Εμποδίων

Στην δεύτερη διεργασία επιδιώκεται αποφυγή των εμποδίων, θεωρώντας συνεχώς γνωστές τις θέσεις τους στο χώρο. Σε αυτό το σημείο να επισημαίνεται ότι η διάμετρος της διατομής κάθε συνδέσμου του βραχίονα είναι 12.6 cm, ενώ οι δύο κύλινδροι-εμπόδια είναι πανομοιότυποι με διάμετρο διατομής 10 cm, τα κέντρα τους απέχουν απόσταση 40 cm, και βρίσκονται πάνω στο επίπεδο $x = 0.3$.

Παρατηρούμε πως οι αρθρώσεις οι οποίες έρχονται σε επαφή με τα εμπόδια είναι οι αρθρώσεις q_3, q_4, q_5 . Για την υλοποίηση του αλγορίθμου ορίζουμε έναν πίνακα που θα κρατάει την απόσταση των αρθρώσεων από το κέντρο.

```
distance = np.matrix([[ (1/2)*Kc*((self.A03[1,3] - center)**2)], \
                        [(1/2)*Kc*((self.A04[1,3] - center)**2)], \
                        [(1/2)*Kc*((self.A05[1,3] - center)**2)])
```

Συγκεκριμένα χρησιμοποιούμε την μέθοδο ελαχιστοποίησης κόστους, όπου η συνάρτηση συνάρ κóστους ??? έχει την μορφή:

$$dist = K_1 \cdot (A_3^0[1:3] - center)^2 + K_2 \cdot (A_4^0[1:3] - center)^2 + K_3 \cdot (A_5^0[1:3] - center)^2$$

Και στον κώδικα του controller.py:

```
weights = np.array([15,30,15])

dist = weights[0]*dist3 + weights[1]*dist4 + weights[2]*dist5
```

Όπου weights πρόκειται για τα K_1, K_2, K_3 . Δίνουμε μεγαλύτερο βάρος στην 4η άρθρωση καθώς αυτή αποτελεί την κεντρική άρθρωση του βραχίονα η οποία είναι πιο επίφοβη να συγκρουστεί με τα εμπόδια, επομένως είναι πιο σημαντικό να ελαχιστοποιηθεί η απόστασή της.

Η ελαχιστοποίηση της παραπάνω συνάρτησης αντιστοιχεί στον υπολογισμό του διανύσματος αναφοράς ξ , με βάση την Gradient Descent μέθοδο.

$$\xi = -K \frac{\partial dist(q_i)}{\partial q_i}$$

Όπου K το συνολικό κέρδος για την υποεργασία 2 και $\frac{\partial dist(q_i)}{\partial q_i}$ είναι ένα 7×1 διάνυσμα με κάθε γραμμή i να έχει παράγωγο της συνάρτησης ως προς το q_i διάνυσμα.

Ακολουθώντας τη θεωρία του μαθήματος, ορίζουμε τη συνεισφορά της δεύτερης υποεργασίας στην συνολική κίνηση του βραχίονα ως εξής:

$$q_2 \dot{=} (I - J^+ J) * \xi$$

```
q2_dot = (np.eye(7) - np.dot(pinvJ, J)) @ weights
```

Όπου I ο μοναδιαίος πίνακας 7 διαστάσεων, J^+ είναι η Moore - Penrose pseudo-inverse Jacobian, J η Jacobian και ξ το gradient descent της συνάρτησης κόστους που προηγουμένως

υπολογίσαμε. Το τελικό διάνυσμα των γωνιών των αρθρώσεων του βραχίονα δίνεται από την υπέρθεση των δύο υποεργασιών και έχει τη μορφή:

$$\dot{q} = \dot{q}_1 + \dot{q}_2$$

```
for i in range(7):  
    self.joint_angvel[i] = q1_dot[i,0] + q2_dot[i,0]
```

C. Προσομοίωση

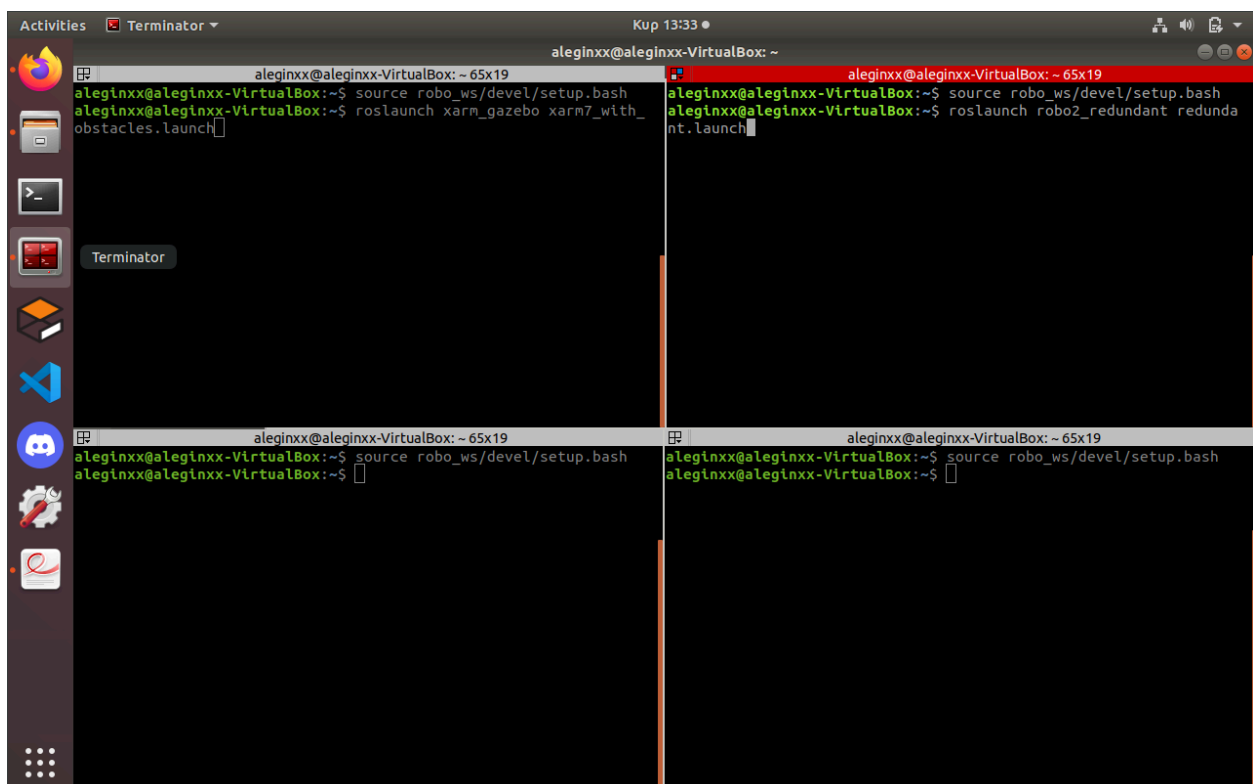
Για το περιβάλλον προσομοίωσης της εργασίας, εργαστήκαμε με τα εργαλεία ROS Melodic και Gazebo.

Αρχικά, δημιουργήσαμε ένα περιβάλλον εργασίας με το όνομα `robo_ws`, και στη διεύθυνση `~/robo_ws/src` εγκαθιστούμε το `gazebo_ros_pkgs` (το οποίο επιτρέπει στα δύο εργαλεία να επικοινωνούν), τον φάκελο `robo2_redundant` που μας δίνεται και κλωνοποιούμε το github directory με link https://github.com/oikonpar/xarm_ros.git. Σύμφωνα με την παραπάνω θεωρητική ανάλυση, κατασκευάζουμε τους κώδικες controller και kinematics. Χάρη σε αυτούς τους δύο κώδικες καθορίζεται τόσο η τροχιά του ρομποτ, όσο και τα εμπόδια που τοποθετούνται στο planar field του Gazebo.

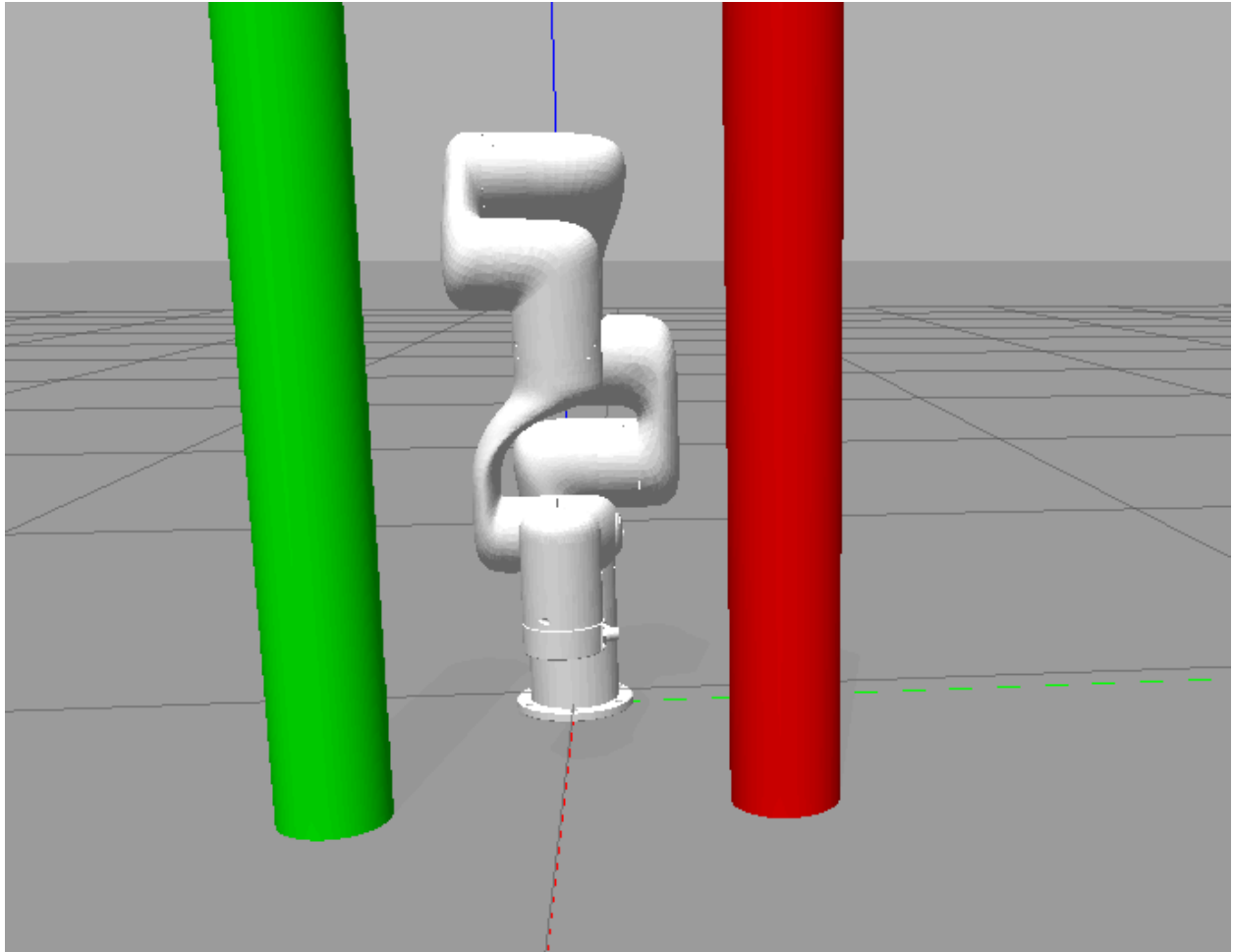
Για την εκτέλεση του ros πακέτου μας, οφείλουμε να το αρχικοποιούμε με την παρακάτω εντολή, κάθε φορά που ανοίγουμε καινούριο terminal:

```
source robo_ws/devel/setup.bash
```

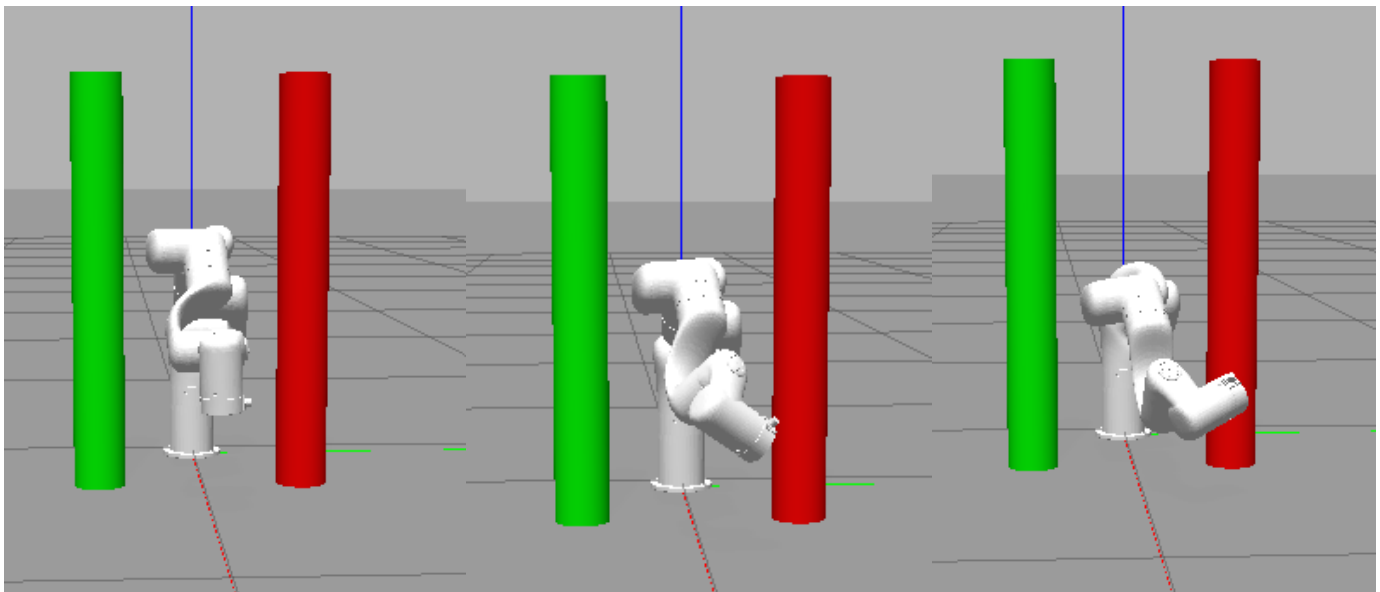
Για την εκτέλεση του project, χρησιμοποιούμε επίσης την εφαρμογή terminator, η οποία εμφανίζει πολλά terminal windows ταυτόχρονα και προτείνεται για την εκτέλεση ros packages. Πριν την εκτέλεση του κώδικα στο gazebo, βρισκόμαστε σε αυτό το σημείο:

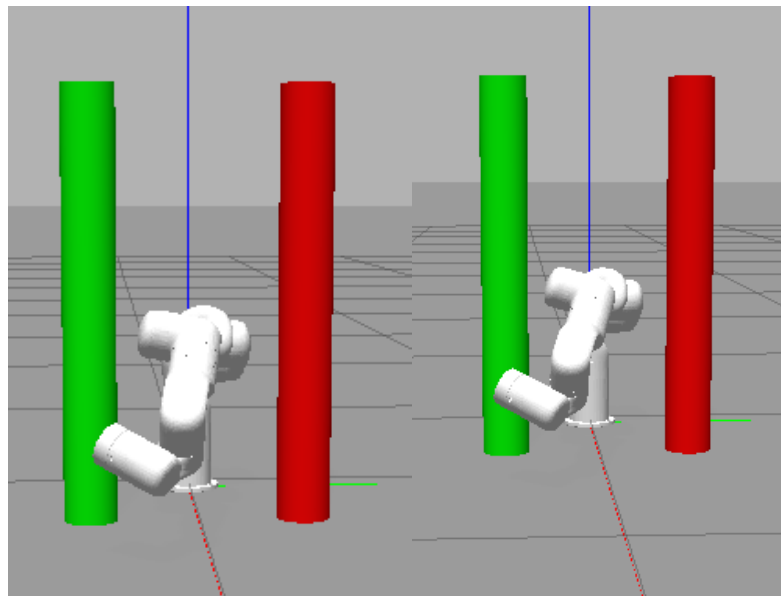
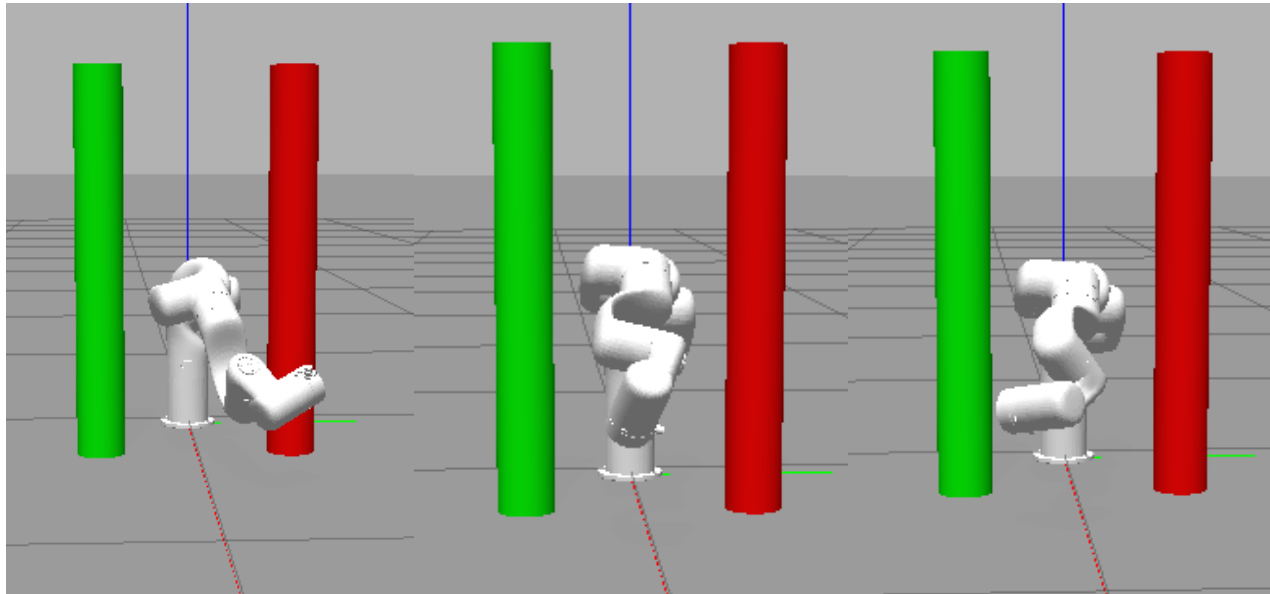


Η εκτέλεση της εντολής `roslaunch xarm_gazebo xarm7_with_obstacles.launch` εκκινεί το πρόγραμμα Gazebo, στο οποίο υπάρχει ο στατικός βραχίονας που έχουμε κατασκευάσει, με εμφανείς τους άξονες x,y,z, καθώς και τα δύο εμπόδια εκατέρωθέν του :

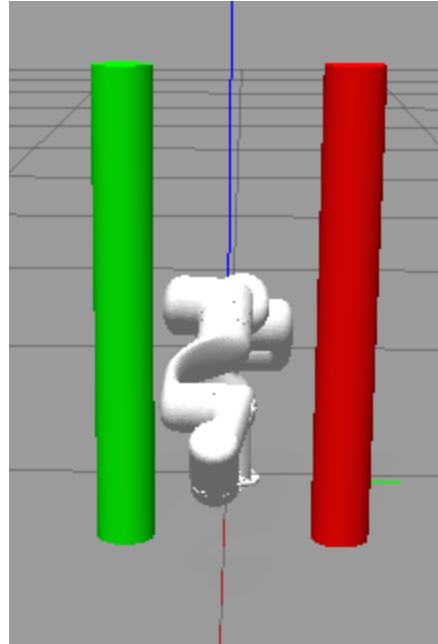


Εκτελώντας την εντολή `roslaunch robo2_redundant redundant.launch`, το ρομπότ αρχίζει να κινείται, εκτελώντας παλινδρομική κίνηση μεταξύ των δύο εμποδίων.

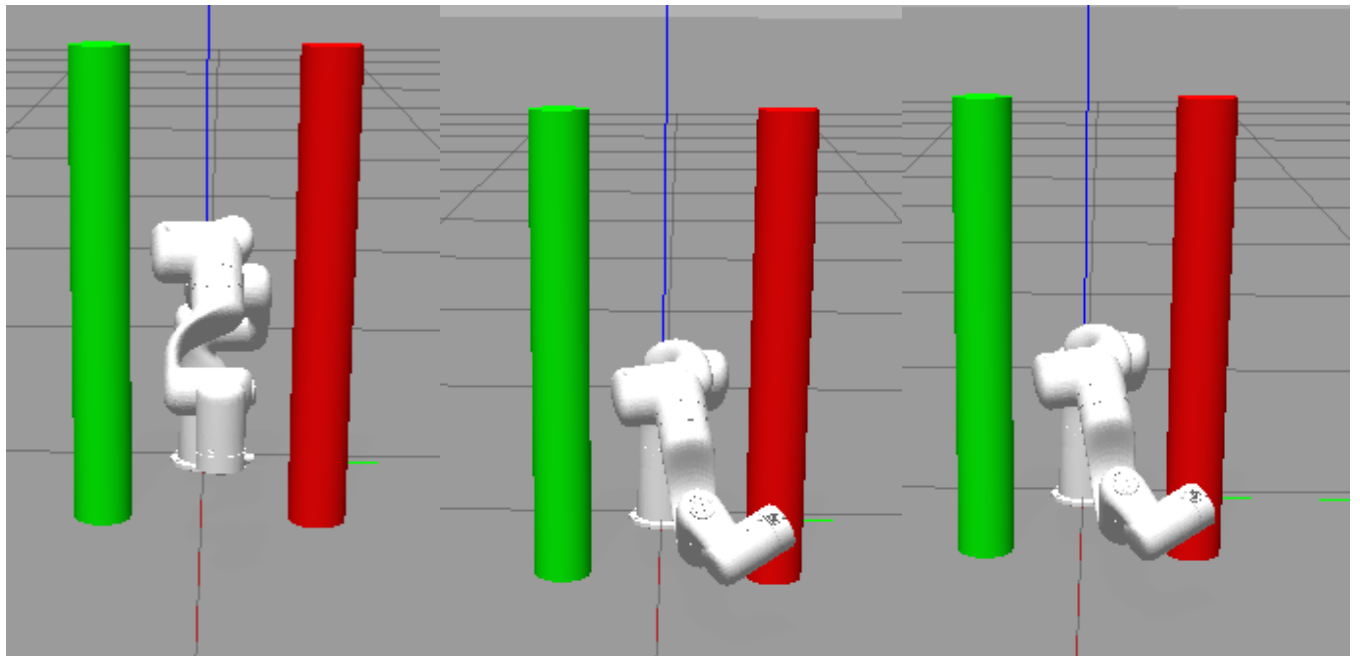


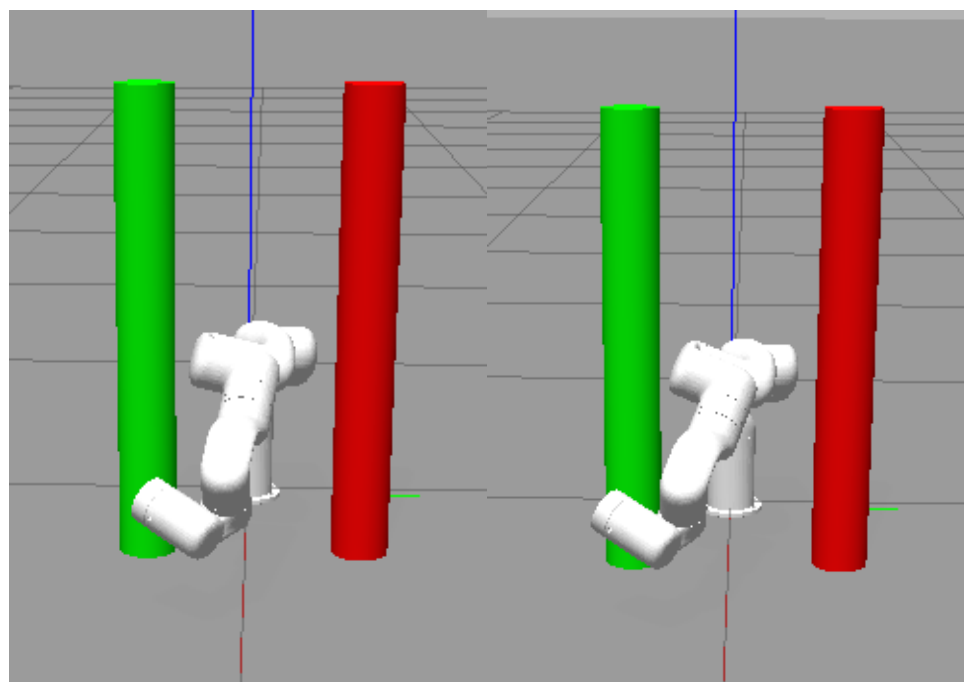
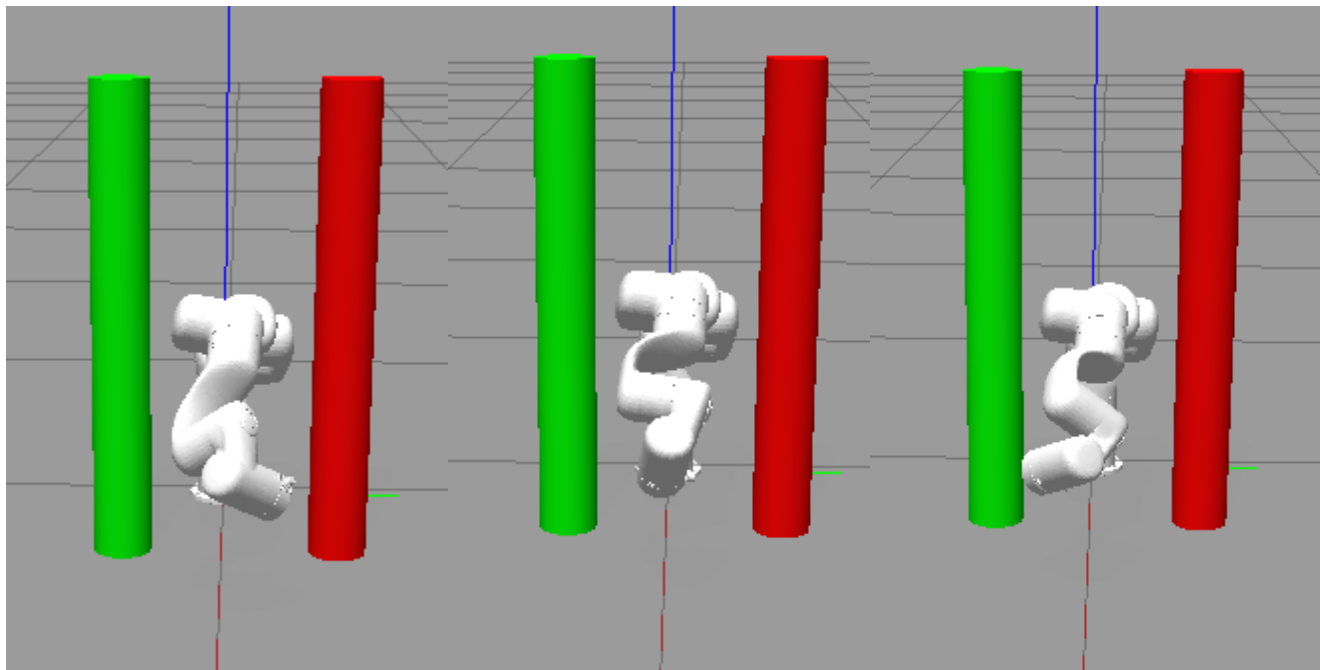


Μετά την εκτέλεση της παραπάνω κίνησης, μετατοπίζουμε το πράσινο εμπόδιο πιο κοντά στον βραχίονα, ώστε να ελέγξουμε την αποτελεσματικότητα του αλγορίθμου αποφυγής εμποδίων. Αυτή είναι η νέα αρχική θέση του ρομπότ:



Εκτελώντας εκ νέου την εντολή `roslaunch robo2_redundant redundant.launch`, το ρομπότ ξεκινά και πάλι να κινείται.

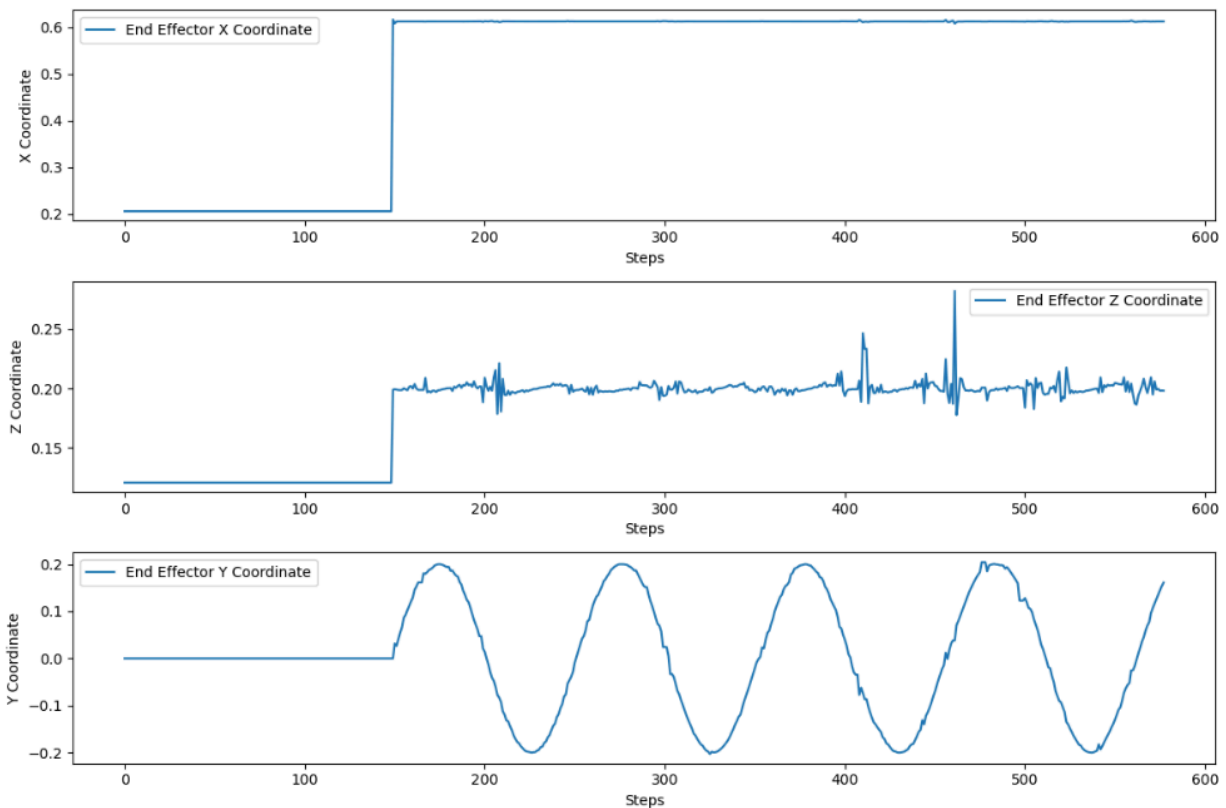




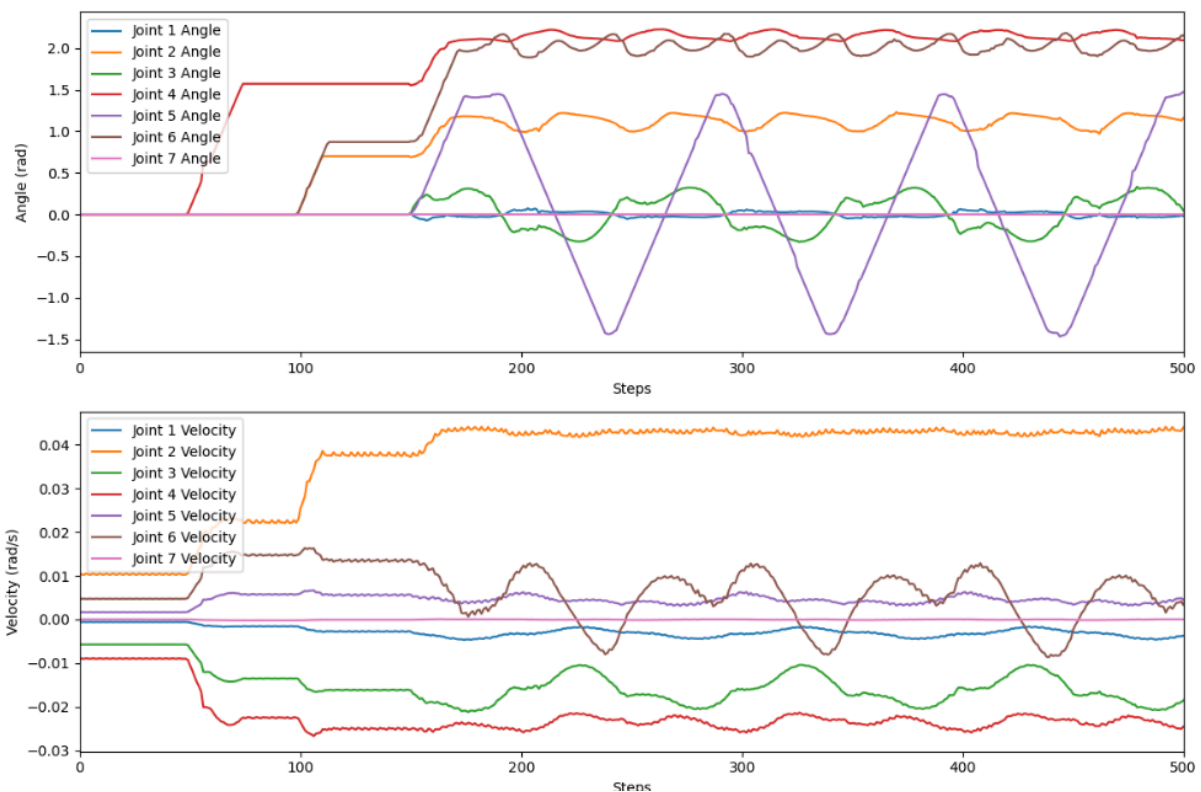
Παρατηρούμε πως, παρά τη μετακίνηση του εμποδίου, ο βραχίονας εκτελεί αποτελεσματικά την κίνησή του, αλλάζοντας την τροχιά που ακολουθεί προκειμένου να αποφύγει το εμπόδιο, ενώ ταυτόχρονα διατηρεί τη μικρότερη δυνατή απόσταση από το κέντρο των δύο εμποδίων.

Αξίζει, σε αυτό το σημείο, να απεικονίσουμε γραφικά την κίνηση στους άξονες x, y, z του end effector, καθώς και όλων των joints. Για σκοπούς παρατήρησης, πειραματιστήκαμε με τις τιμές των κερδών, της περιόδου και των steps, ώστε να καταλήξουμε σε ένα όσο το δυνατόν πιο ομαλό αποτέλεσμα. Οι τρεις περιπτώσεις στις οποίες θεωρήσαμε πως αξίζει να εμβαθύνουμε, είναι αυτές όπου $T = 1 \text{ sec}$ και $\text{Steps} = \{10, 30\}$.

- **T = 1 sec:**



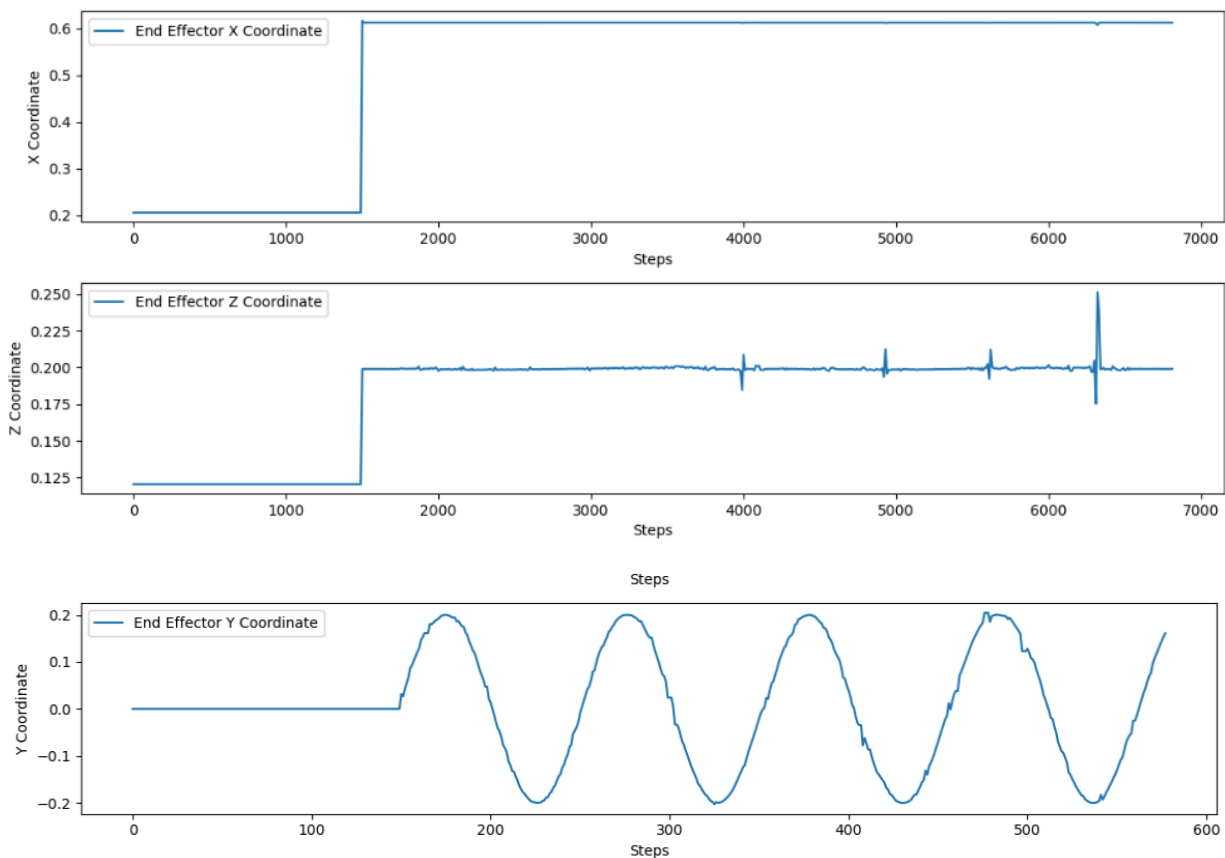
Παρατηρούμε πως ο χρόνος εκκίνησης του βραχίονα είναι περίπου ίσος με 150 Steps. Η άνοδος του end-effector (X-Coordinate) είναι ομαλή τόσο κατά τη διάρκεια της εκκίνησης, όσο και καθ'όλη τη διάρκεια της κίνησης του Cobot. Στον άξονα Z παρατηρούμε πως υπάρχουν ταλαντώσεις κατά τη διάρκεια της κίνησης του βραχίονα. Ο άξονας Z στις παραπάνω φωτογραφίες είναι αυτός που χρωματίζεται με κόκκινο (παράλληλος με το έδαφος). Κατά συνέπεια, είναι λογικές οι ταλαντώσεις στη δεύτερη γραφική παράσταση, καθώς ο end-effector εκτελεί Translation κίνηση κατά μήκος του άξονα Z. Αντίστοιχα, η κίνηση που πραγματοποιεί ο end-effector κατά μήκος του άξονα Y είναι Rotational. Αυτό εξηγεί την ταλάντωση της γραφικής παράστασης στον τρίτο και τελευταίο άξονα.



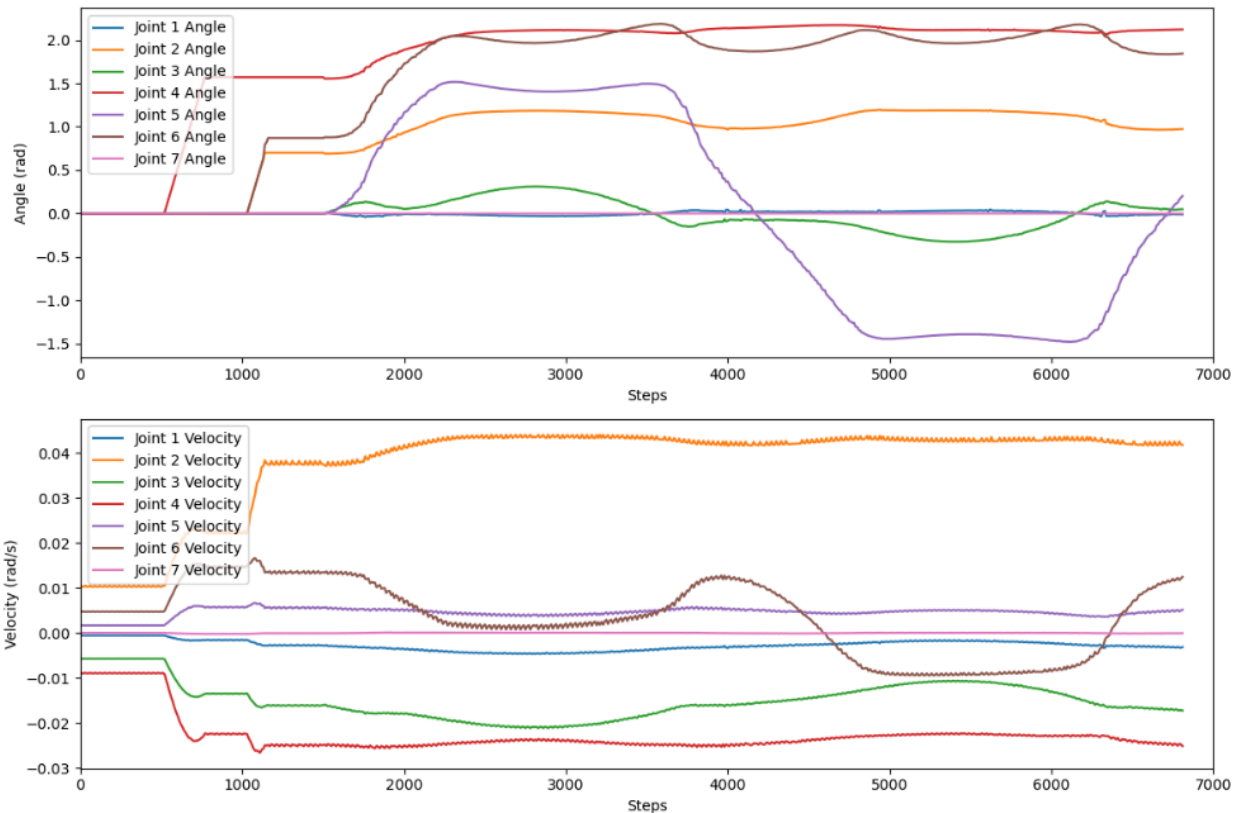
Αρχικά, αξίζει να σχολιάσουμε το γράφημα των γωνιών των αρθρώσεων. Παρατηρούμε πως όλες οι αρθρώσεις, με εξαίρεση το joint 7 που αναφέρεται στον end-effector, εμφανίζουν περιοδική συμπεριφορά σε σχέση με τις γωνίες στις οποίες περιστρέφονται οι αρθρώσεις. Ανάλογα με το πόσο κομβικό ρόλο διαθέτει μια άρθρωση στην εκτέλεση της προκαθορισμένης τροχιάς το πλάτος της ταλάντωσης αυξομειώνεται. Πιο συγκεκριμένα, η άρθρωση 1 (Joint 1) που αναφέρεται στη βάση του Cobot, εκτελεί ταλάντωση αμελητέου πλάτους, καθώς η μόνη περιστροφή που κάνει είναι μεταξύ των δύο εμποδίων και δεν μετακινείται στον χώρο. Αντίθετα, το joint 5 εκτελεί ταλάντωση με μεγαλύτερο πλάτος απ' όλες, καθώς ο ρόλος της άρθρωσης στην αποφυγή εμποδίων είναι ο πιο σημαντικός. Σημαντικό ρόλο στην αποφυγή εμποδίων έχουν, επίσης, οι αρθρώσεις 2, 4 και 6.

Σχετικά με την ταχύτητα των αρθρώσεων, παρατηρούμε πως οι αρθρώσεις που εκτελούν τη μεγαλύτερη μετακίνηση μέσα στον χώρο (joint 6) έχουν το μεγαλύτερο πλάτος ταλάντωσης, ενώ αρθρώσεις που συμμετέχουν πολύ στην περιστροφική κίνηση, διατηρούν μεν μεγάλη τιμή (θετική ή αρνητική) καθ'όλη τη διάρκεια της κίνησης, ωστόσο με μικρό πλάτος ταλάντωσης.

- $T = 5$, $K = 150$, $K_c = 100$:



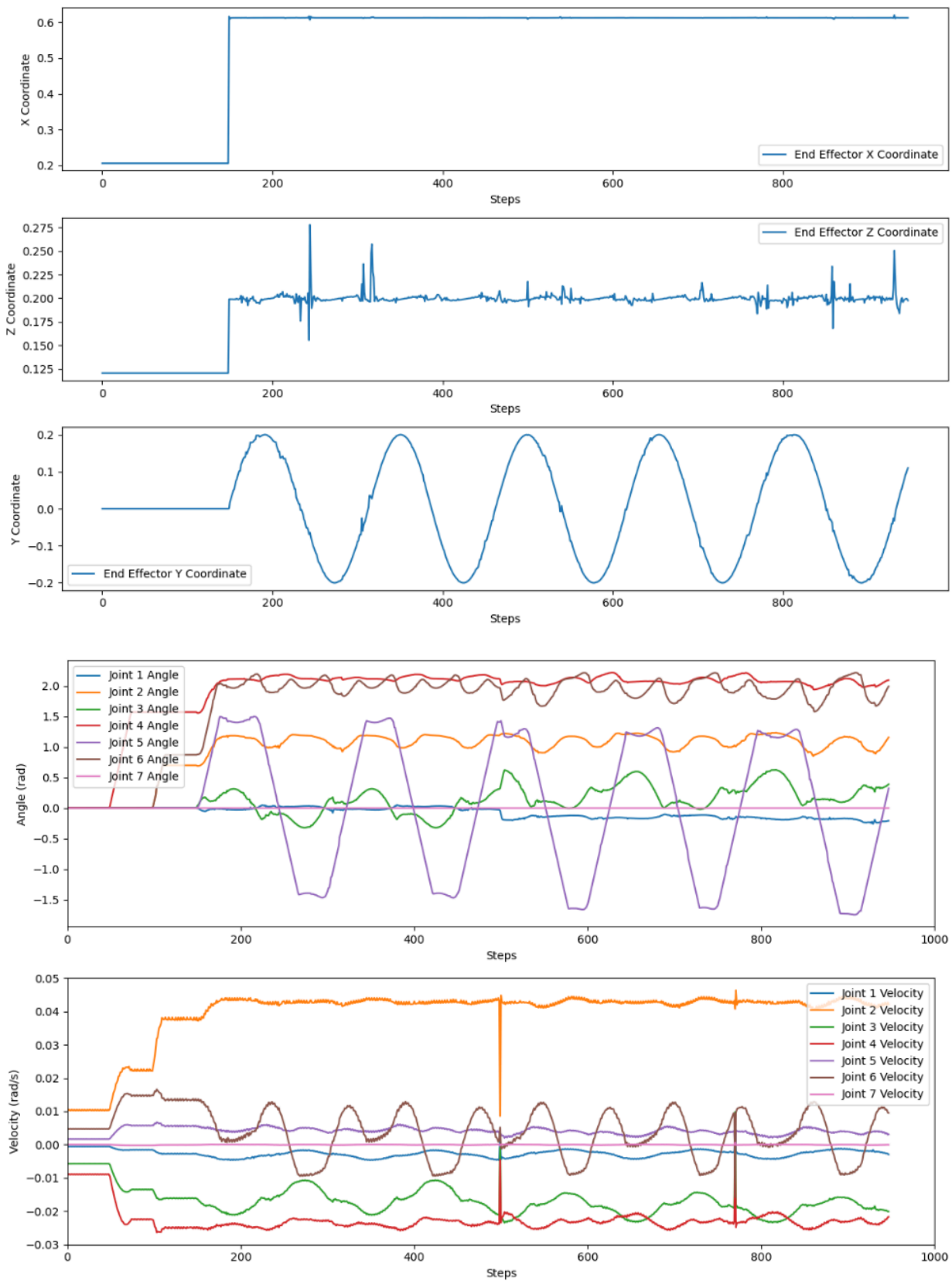
Σύμφωνα με τα παραπάνω διαγράμματα, γίνεται εμφανές πως, καθ' όλη τη διάρκεια της κίνησης, ο end effector ακολουθεί την προκαθορισμένη τροχιά με συνέπεια, ενώ οι ασυνέχειες (spikes) που παρουσιάζονται μετά το Step-4000 οφείλονται στη μετακίνηση του πράσινου εμποδίου. Παρ' όλα αυτά, παρατηρούμε πως η κίνηση που ακολουθεί τελικά ο βραχίονας ακολουθεί την αρχική τροχιά που έχουμε θέσει στον κώδικα.



Τα παραπάνω διαγράμματα μαρτυρούν μια αν-ομαλότητα στην κίνηση του βραχίονα, κάτι το οποίο ήταν αναμενόμενο δεδομένης της πολύ αργής κίνησής του. Στο πρώτο διάγραμμα φαίνονται οι ταχύτητες των αρθρώσεων κατά τη διάρκεια της κίνησης. Κατά την εκτέλεση της τροχιάς, το γράφημα των ταχυτήτων είναι ένας συνεχής παλμός, το πλάτος του οποίου αυξομειώνεται. Συγκεκριμένα, το πλάτος του joint6 μετά το Step-4000 υπόκειται μια απότομη αύξηση, γεγονός το οποίο οφείλεται, όπως προηγουμένως, στη μετακίνηση του πράσινου εμποδίου στο βήμα αυτό.

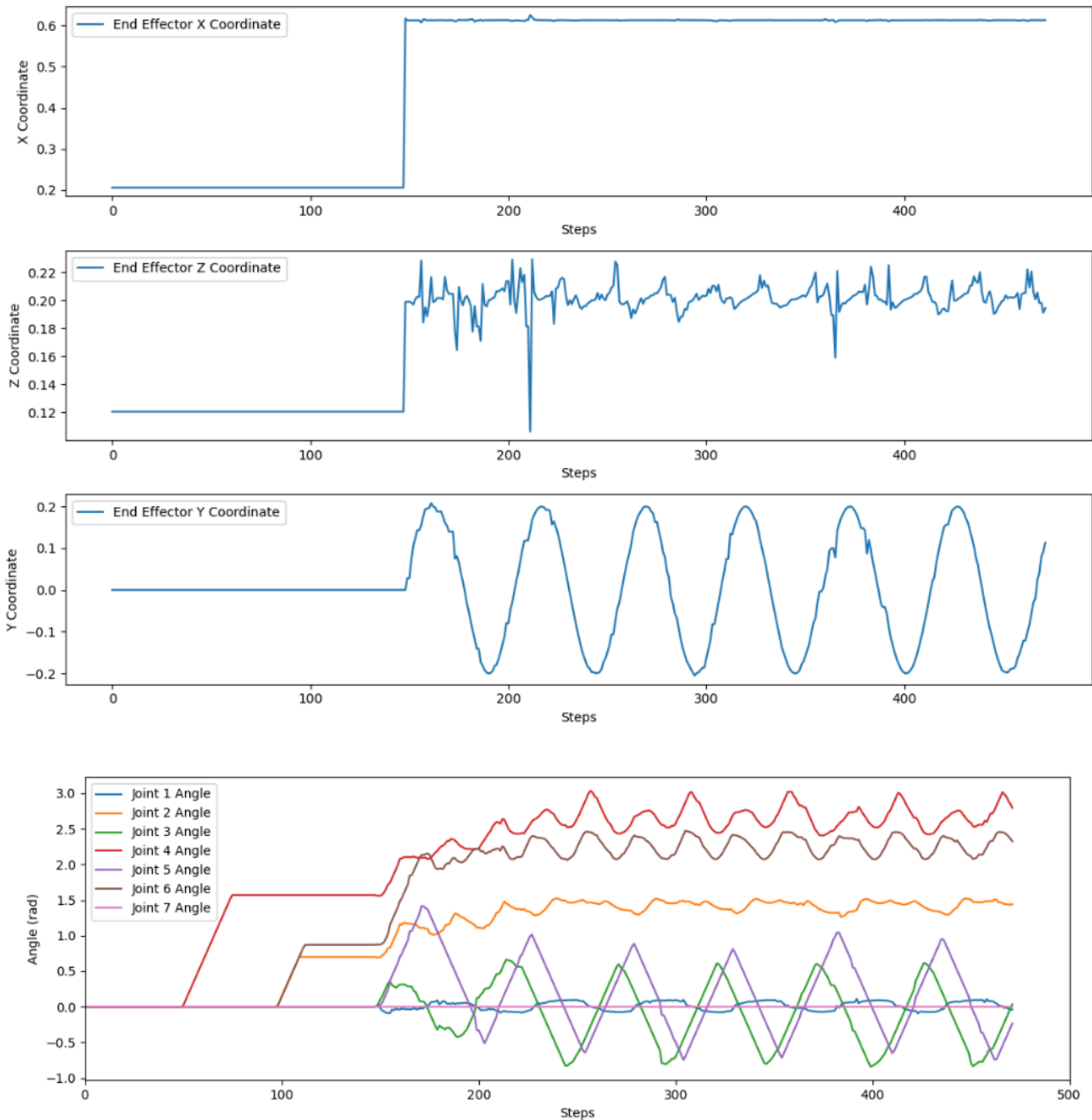
Στο δεύτερο διάγραμμα, φαίνονται οι γωνίες στροφής των αρθρώσεων κατά τη διάρκεια της κίνησης. Οι γωνίες των αρθρώσεων λαμβάνουν μη μηδενικές τιμές σε διαφορετικές χρονικές στιγμές, κάτι το οποίο είναι λογικό καθώς έτσι ορίζει η κίνηση του cobot. Συγκεκριμένα μετά το Step-4000, το joint5 μειώνεται δραματικά μετά το Step-4000, κάτι το οποίο είναι αναμενόμενο. Όπως προαναφέρθηκε, κατά τη μετακίνηση του πράσινου εμποδίου σε μια καινούρια θέση, το cobot επιχειρεί να εκτελέσει μια τροχιά όσο το δυνατόν πιο “κοντινή” σε αυτή που έχουμε ορίσει μέσω του κώδικα, ενώ ταυτόχρονα αποφεύγει τα εμπόδια που έχουμε ορίσει στο planar field. Συνεπώς, στην προσπάθεια αποφυγής εμποδίου, η ομαλότητα της εκτέλεσης της τροχιάς διαταράσσεται, προκειμένου να αποφευχθεί η σύγκρουση.

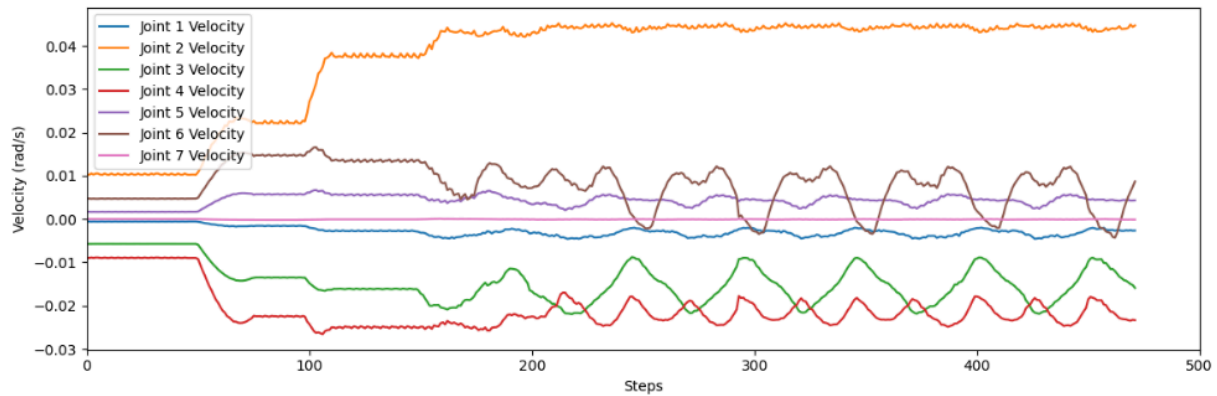
- **Step = 10:**



Στο παραπάνω set αποτελεσμάτων, παρατηρούμε πως η μορφή των γραφικών παραστάσεων για X-Y-Z Coordinates παραμένει ίδια. Σε ότι αφορά ωστόσο τις γραφικές παραστάσεις των γωνιών και ταχυτήτων, παρατηρούμε έντονα spikes στο step 500 καθώς εκείνη τη χρονική στιγμή μετακινήσαμε το κόκκινο εμπόδιο προς τον βραχίονα, ενώ παρόμοιο spike παρατηρείται κοντά στο step 800, όπου μετακινήσαμε το πράσινο εμπόδιο. Και στις δύο αυτές περιπτώσεις, παρατηρούμε πως ο βραχίονας εξακολουθεί να εκτελεί την προκαθορισμένη τροχιά του, προσαρμόζοντας τις ταχύτητες και τις γωνίες των αρθρώσεων, προκειμένου να αποφευχθούν τα εμπόδια.

- **Step = 30:**





D. Συμπεράσματα

Σύμφωνα με τα παραπάνω, καταλήγουμε στα εξής συμπεράσματα σχετικά με την ομαλότητα της κίνησης του Cobot:

- Η αύξηση της περιόδου κίνησης T οδηγεί σε μεγαλύτερη ευστάθεια στον άξονα Z , καθώς η κίνηση του βραχίονα γίνεται πιο αργά (άρα και πιο ομαλά).
- Αύξηση του αριθμού των Steps οδηγεί σε πιο έντονες ταλαντώσεις στις γραφικές παραστάσεις των ταχυτήτων και των γωνιών των αρθρώσεων, καθώς η περίοδος της δειγματοληψίας μεγαλώνει, επομένως οι διαφορές μεταξύ των τιμών γίνεται πιο αισθητή.
- Η αλλαγή των τιμών των κερδών K και K_c μπορεί να επηρεάσει την προσομοίωση με ποικίλους τρόπους. Όταν η αύξηση γίνεται εντός ορίων, η προσομοίωση μπορεί να γίνεται με μεγαλύτερη ακρίβεια, ενώ μεγάλη αύξηση των κερδών καθιστά την προσομοίωση ανακριβή ή και ασταθή. Αυτό συμβαίνει διότι οι τιμές αυτές ευθύνονται για τον έλεγχο του συστήματος, επομένως πειραματικά καταλήγουμε στις τιμές που φαίνονται στον κώδικα.