Αναφορά εργαστηριακών ασκήσεων

Νικόλαος Μητσόπουλος
TH20591

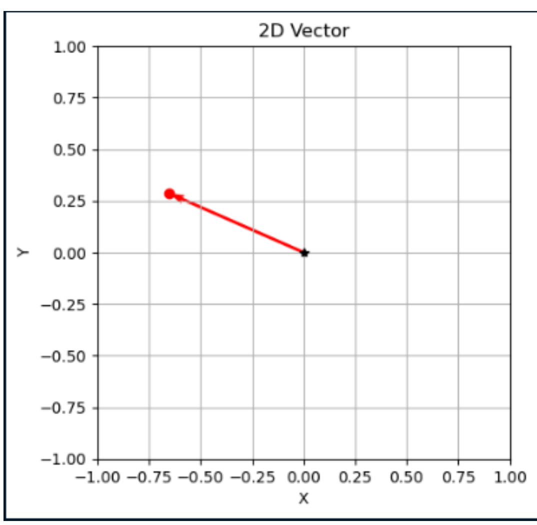
Function plot_free_vec

Η συνάρτηση plot_free_vec(x, c) έχει ως σκοπό να σχεδιάσει ένα διάνυσμα σε δύο ή τρεις διαστάσεις, ανάλογα με το μήκος του διανύσματος εισόδου. Η παράμετρος x αντιπροσωπεύει το ίδιο το διάνυσμα και μπορεί να είναι είτε λίστα είτε numpy array με δύο ή τρία στοιχεία. Η παράμετρος c καθορίζει το χρώμα με το οποίο θα σχεδιαστεί το διάνυσμα, με προεπιλεγμένη τιμή το μπλε.

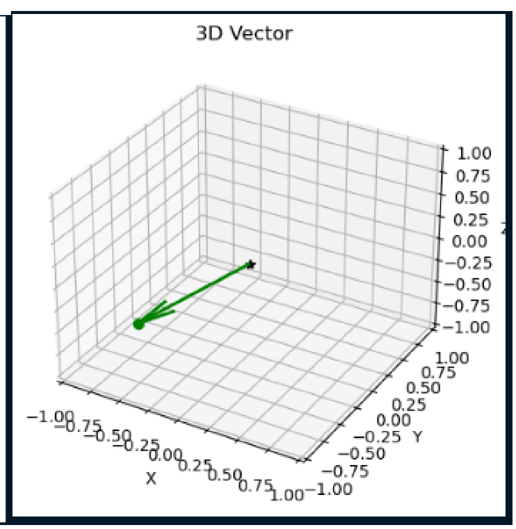
Όταν το διάνυσμα είναι δύο διαστάσεων, η συνάρτηση δημιουργεί ένα γράφημα με άξονες X και Y. Το διάνυσμα σχεδιάζεται από την αρχή των αξόνων μέχρι τη θέση που ορίζεται από τα στοιχεία του διανύσματος. Η σχεδίαση πραγματοποιείται με τη χρήση της συνάρτησης ax.quiver(...). Σημειώνεται η αρχή του διανύσματος με ένα αστεράκι (‘*’) και το άκρο του με έναν κύκλο (‘o’). Το γράφημα περιλαμβάνει πλέγμα για καλύτερη κατανόηση και φέρει τίτλο "2D Vector".

Όταν το διάνυσμα είναι τριών διαστάσεων, χρησιμοποιείται τρισδιάστατη απεικόνιση μέσω της εντολής projection='3d'. Το διάνυσμα σχεδιάζεται από το σημείο (0, 0, 0) προς τη θέση που καθορίζεται από τις συντεταγμένες του διανύσματος. Τα όρια των αξόνων ρυθμίζονται ώστε να κυμαίνονται από -1 έως 1 και στο γράφημα προστίθεται τίτλος "3D Vector". Και σε αυτή την περίπτωση, σημειώνονται η αρχή και το άκρο του διανύσματος με τα αντίστοιχα σύμβολα.

Στα παραδείγματα που ακολουθούν τη δήλωση της συνάρτησης, δημιουργούνται ένα τυχαίο διάνυσμα δύο διαστάσεων και ένα τριών διαστάσεων, με τιμές στο διάστημα από -1 έως 1, και σχεδιάζονται αντίστοιχα με κόκκινο και πράσινο χρώμα.



Εικόνα 1 3D Διάνυσμα



Εικόνα 2 2D Διάνυσμα

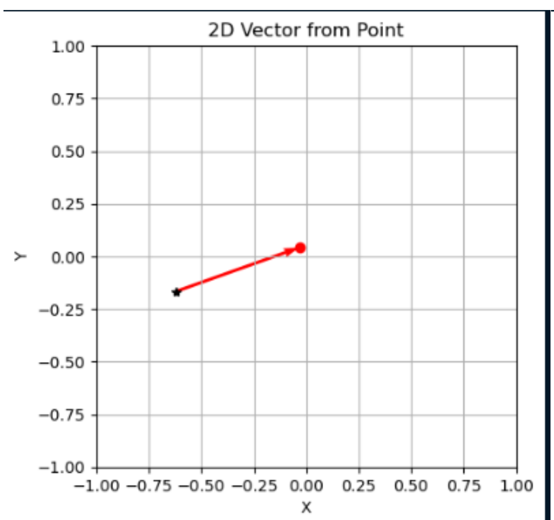
Function plot_vec(a,x,c)

Η δεύτερη συνάρτηση του προγράμματος είναι η plot_vec(x, a=[0, 0], c='b'), η οποία επιτρέπει τη γραφική αναπαράσταση ενός διανύσματος που ξεκινά από ένα αυθαίρετο σημείο του χώρου και όχι απαραίτητα από την αρχή των αξόνων, όπως στην προηγούμενη περίπτωση. Η είσοδος x αντιπροσωπεύει το διάνυσμα, η a είναι το σημείο εκκίνησης του διανύσματος (με προεπιλεγμένη τιμή το [0, 0]) και η c καθορίζει το χρώμα του διανύσματος.

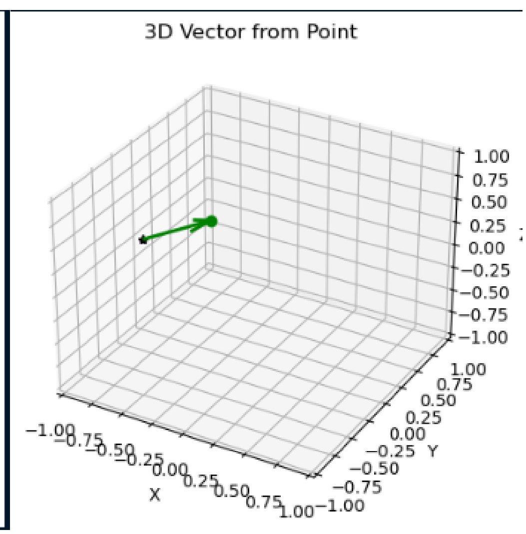
Όταν το διάνυσμα είναι δύο διαστάσεων, η συνάρτηση δημιουργεί ένα 2D γράφημα και σχεδιάζει το διάνυσμα από το σημείο (a[0], a[1]) προς το σημείο (a[0] + x[0], a[1] + x[1]). Το σημείο εκκίνησης επισημαίνεται με έναν αστερίσκο (‘*’) και το άκρο του διανύσματος με έναν κύκλο (‘o’). Ο χώρος του γραφήματος διατηρείται συμμετρικός με όρια από -1 έως 1 και περιλαμβάνει πλέγμα για την καλύτερη κατανόηση της γεωμετρίας. Ο τίτλος του γραφήματος είναι "2D Vector from Point".

Για την περίπτωση τριών διαστάσεων, η συνάρτηση χρησιμοποιεί τρισδιάστατη προβολή και σχεδιάζει το διάνυσμα από το σημείο (a[0], a[1], a[2]) προς το σημείο (a[0] + x[0], a[1] + x[1], a[2] + x[2]). Αντίστοιχα, σημειώνονται το σημείο εκκίνησης και το άκρο του διανύσματος με τα ίδια σύμβολα και τίθεται τίτλος "3D Vector from Point". Τα όρια των αξόνων καθορίζονται επίσης στο διάστημα από -1 έως 1.

Στα παραδείγματα που ακολουθούν, δημιουργούνται δύο διανύσματα και δύο σημεία εκκίνησης (ένα για 2D και ένα για 3D περίπτωση), όλα με τυχαίες τιμές στο διάστημα [-1, 1]. Τα διανύσματα σχεδιάζονται σε κόκκινο και πράσινο χρώμα αντίστοιχα.



Εικόνα 3 2D Διάνυσμα Αυθαίρετης Εκκίνησης



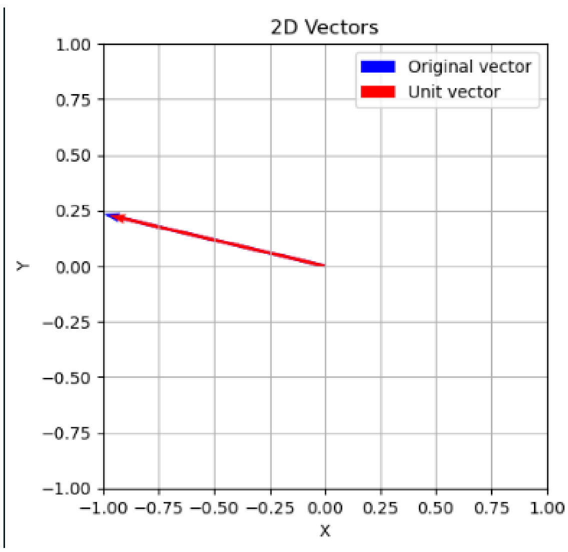
Εικόνα 4 3D Διάνυσμα Αυθαίρετης Εκκίνησης

Function s=make_unit(x)

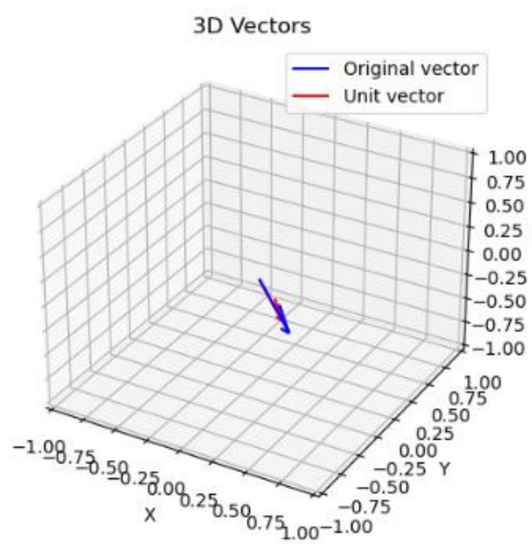
Η τρίτη συνάρτηση του προγράμματος είναι η make_unit(x), η οποία μετατρέπει ένα δοσμένο διάνυσμα σε μοναδιαίο, δηλαδή ένα διάνυσμα με μέτρο ίσο με 1, που έχει την ίδια κατεύθυνση με το αρχικό. Η συνάρτηση δέχεται ως είσοδο το διάνυσμα x και επιστρέφει το κανονικοποιημένο διάνυσμα. Η κανονικοποίηση επιτυγχάνεται διαιρώντας το διάνυσμα με το μέτρο του, το οποίο υπολογίζεται με τη χρήση της np.linalg.norm(x). Αν το διάνυσμα έχει μηδενικό μέτρο, επιστρέφεται το ίδιο το διάνυσμα για αποφυγή διαίρεσης με το μηδέν.

Ακολουθούν παραδείγματα για διάνυσμα δύο και τριών διαστάσεων. Σε κάθε περίπτωση εμφανίζονται στην έξοδο τόσο το αρχικό όσο και το κανονικοποιημένο διάνυσμα.

Για να επιδειχθεί οπτικά η διαφορά μεταξύ του αρχικού και του μοναδιαίου διανύσματος, υλοποιούνται δύο επιπλέον βοηθητικές συναρτήσεις: η plot_vector_2d(x, u) και η plot_vector_3d(x, u). Η plot_vector_2d εμφανίζει το αρχικό διάνυσμα με μπλε και το κανονικοποιημένο με κόκκινο χρώμα σε δισδιάστατο χώρο, ενώ η plot_vector_3d κάνει την ίδια απεικόνιση σε τρισδιάστατο χώρο. Τα γραφήματα περιλαμβάνουν τίτλους, άξονες, πλέγμα και υπόμνημα (legend) για την καλύτερη κατανόηση της διαφοράς μεταξύ των δύο διανυσμάτων.



Εικόνα 5 2D Κανονικοποιημένο Διάνυσμα



Εικόνα 6 3D Κανονικοποιημένο Διάνυσμα

Function s=project_vec(a,b)

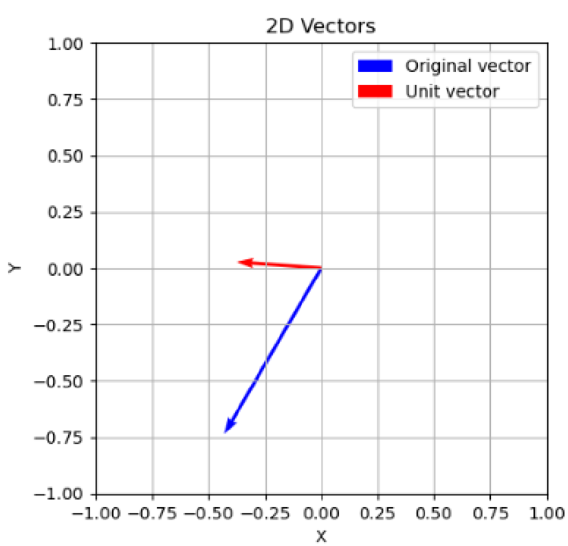
Η τέταρτη συνάρτηση ονομάζεται project_vec(a, b) και υπολογίζει την προβολή του διανύσματος b πάνω στο διάνυσμα a.

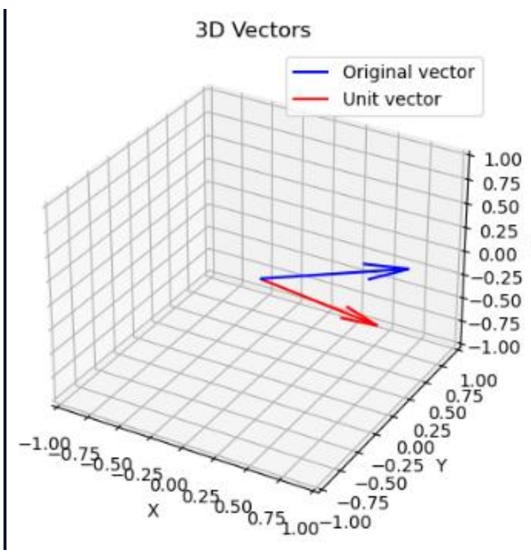
Η συνάρτηση αρχικά μετατρέπει τα διανύσματα a και b σε numpy πίνακες για να διευκολυνθούν οι μαθηματικές πράξεις. Στη συνέχεια, υπολογίζει το μέτρο του διανύσματος a. Αν το μέτρο είναι μηδέν, εγείρεται εξαίρεση (ValueError) διότι δεν έχει νόημα να προβάλλουμε ένα διάνυσμα πάνω σε μηδενικό διάνυσμα.

Αν το διάνυσμα a έχει κανονικό μήκος, υπολογίζεται το μοναδιαίο διάνυσμα με την ίδια κατεύθυνση (unit_a) και η προβολή του b πάνω στο a προκύπτει από τον τύπο: **proj = dot(b, unit_a) * unit_a**, όπου dot είναι το εσωτερικό γινόμενο των διανυσμάτων. Η προβολή επιστρέφεται από τη συνάρτηση και αποτυπώνει το κομμάτι του b που είναι ευθυγραμμισμένο με το a.

Η λειτουργία της συνάρτησης ελέγχεται με δύο παραδείγματα. Στο πρώτο, δύο τυχαία διανύσματα δύο διαστάσεων δημιουργούνται και υπολογίζεται η προβολή. Στο δεύτερο, γίνεται το ίδιο για τριών διαστάσεων διανύσματα. Η προβολή εμφανίζεται γραφικά με χρήση των συναρτήσεων plot_vector_2d και plot_vector_3d.

Στα αντίστοιχα γραφήματα, το διάνυσμα b εμφανίζεται με μπλε και η προβολή του πάνω στο a με κόκκινο.





Εικόνα 7 2D Απεικόνιση Προβολής α Διανύσματος ως προς το β Διάνυσμα

Εικόνα 8 3D Απεικόνιση Προβολής α Διανύσματος ως προς το β Διάνυσμα

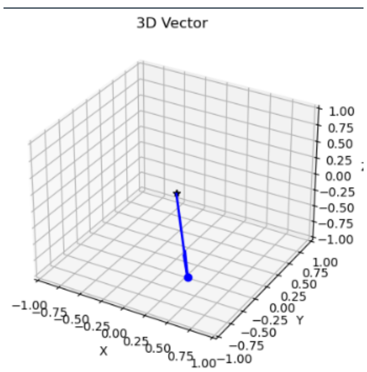
Script cross_demo

Το πέμπτο μέρος αφορά τη χρήση του εξωτερικού γινομένου δύο τρισδιάστατων διανυσμάτων μέσω της συνάρτησης cross_demo(a, b). Η συνάρτηση αυτή υλοποιεί τη μαθηματική πράξη του διανυσματικού γινομένου, που επιστρέφει ένα νέο διάνυσμα κάθετο τόσο στο a όσο και στο b.

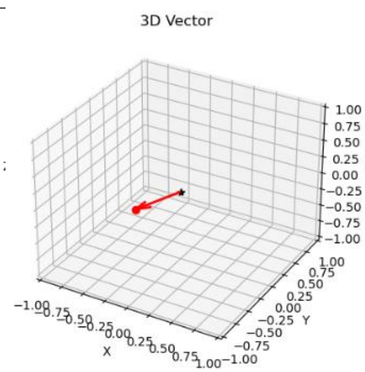
Η συνάρτηση δέχεται δύο διανύσματα εισόδου και καλεί την ενσωματωμένη συνάρτηση np.cross(a, b) της βιβλιοθήκης NumPy για τον υπολογισμό του διανυσματικού γινομένου. Το αποτέλεσμα επιστρέφεται στον χρήστη.

Για να παρουσιαστεί η λειτουργία της συνάρτησης, δημιουργούνται δύο τυχαία διανύσματα τριών διαστάσεων. Υπολογίζεται το γινόμενο a × b, και τα τρία διανύσματα (τα δύο αρχικά και το αποτέλεσμα) απεικονίζονται γραφικά με διαφορετικά χρώματα: το a με μπλε, το b με κόκκινο και το αποτέλεσμα c με πράσινο.

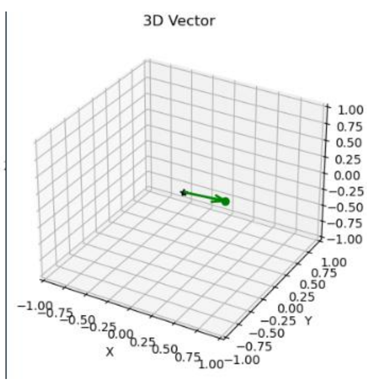
Η απεικόνιση προσφέρει γεωμετρική κατανόηση της σχέσης μεταξύ των διανυσμάτων. Ο χρήστης μπορεί να διαπιστώσει ότι το πράσινο διάνυσμα είναι κάθετο στο επίπεδο που ορίζεται από τα άλλα δύο, κάτι που αποτελεί βασικό χαρακτηριστικό του διανυσματικού γινομένου. Η κατεύθυνση του νέου διανύσματος καθορίζεται από τον κανόνα του δεξιού χεριού.



Εικόνα 9 Διάνυσμα Α



Εικόνα 10 Διάνυσμα Β



Εικόνα 11 Εξωτερικό Γινόμενο (Α Χ Β)

Functions plot_Rot(R)

Η έκτη συνάρτηση του προγράμματος είναι η plot_Rot(R, origin=np.zeros(3), ax=None, name=''), η οποία έχει ως στόχο την απεικόνιση ενός συστήματος συντεταγμένων στον τρισδιάστατο χώρο, όπως αυτό ορίζεται από έναν πίνακα περιστροφής RRR. Η συνάρτηση μπορεί να χρησιμοποιηθεί για να αναπαραστήσει την προσανατολισμένη θέση ενός αντικειμένου στο χώρο, κάτι που είναι εξαιρετικά σημαντικό σε εφαρμογές ρομποτικής και κινηματικής.

Η είσοδος R είναι ένας πίνακας 3x3, ο οποίος περιέχει ως στήλες τα διανύσματα κατεύθυνσης των αξόνων X, Y και Z του νέου συστήματος συντεταγμένων. Η παράμετρος origin καθορίζει τη θέση της αρχής του συστήματος (προεπιλογή: το σημείο (0,0,0)) και η name δίνει έναν τίτλο στο γράφημα. Αν δεν παρέχεται αντικείμενο άξονα (ax), η συνάρτηση δημιουργεί νέο τρισδιάστατο γράφημα.

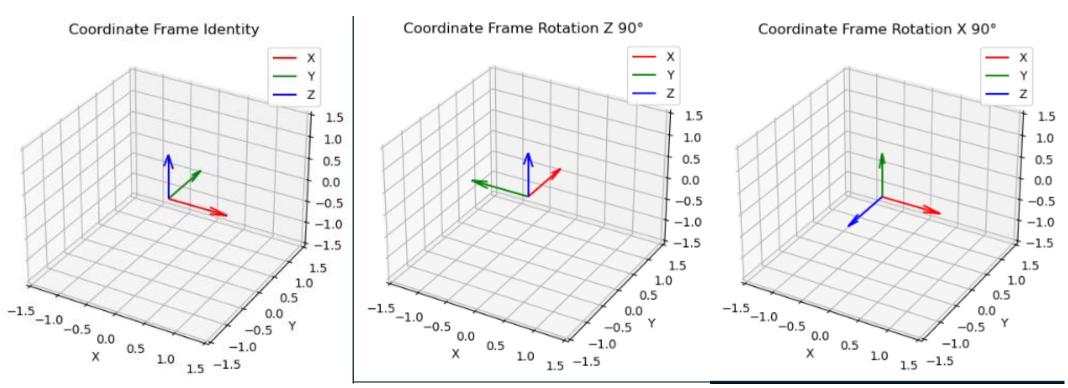
Η συνάρτηση εξάγει τις στήλες του πίνακα R ως ξεχωριστά διανύσματα για τους άξονες X, Y και Z. Στη συνέχεια, χρησιμοποιεί τη συνάρτηση quiver της matplotlib για να σχεδιάσει τα τρία βέλη από το σημείο origin προς τις κατευθύνσεις των αξόνων, με χρώματα κόκκινο (X), πράσινο (Y) και μπλε (Z). Προστίθενται τίτλος και υπόμνημα για καλύτερη κατανόηση του διαγράμματος.

Η συνάρτηση χρησιμοποιείται σε τρία παραδείγματα:

1. Αρχικά εμφανίζεται το σύστημα συντεταγμένων ταυτισμένο με το παγκόσμιο σύστημα, με χρήση της μοναδιαίας ταυτότητας np.eye(3), και φέρει τον τίτλο "Identity".

2. Στη συνέχεια απεικονίζεται ένα σύστημα συντεταγμένων περιστραμμένο κατά 90° ως προς τον άξονα Z.

3. Τέλος, απεικονίζεται ένα σύστημα περιστραμμένο κατά 90° ως προς τον άξονα X.



Εικόνα 12 Αρχικό σύστημα συντεταγμένων χωρίς καμία περιστροφή

Εικόνα 13 Περιστροφή του συστήματος συντεταγμένων κατά 90° γύρω από τον άξονα Z.

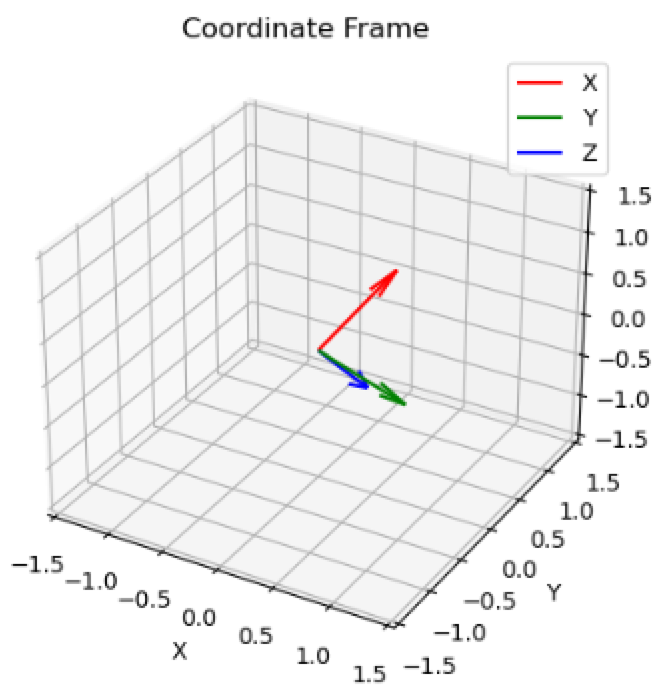
Εικόνα 14 Περιστροφή του συστήματος συντεταγμένων κατά 90° γύρω από τον άξονα X.

Function R=generate_rot()

Η έβδομη συνάρτηση generate_rot() δημιουργεί έναν τυχαίο πίνακα περιστροφής R στον τρισδιάστατο χώρο. Η υλοποίηση αξιοποιεί τη βιβλιοθήκη scipy, και πιο συγκεκριμένα τη Rotation από το υποπακέτο scipy.spatial.transform.

Η λειτουργία της συνάρτησης είναι απλή και αποτελεσματική. Η μέθοδος R.random() παράγει ένα αντικείμενο περιστροφής με τυχαία προσανατολισμό, ενώ με την εντολή as_matrix() η περιστροφή μετατρέπεται σε πίνακα 3×3. Το αποτέλεσμα είναι ένας ορθοκανονικός πίνακας που περιγράφει τον νέο προσανατολισμό ενός συστήματος συντεταγμένων.

Αμέσως μετά την παραγωγή του πίνακα, η μεταβλητή R_matrix εκτυπώνεται στην οθόνη, επιτρέποντας την επιθεώρηση της αριθμητικής μορφής του. Στη συνέχεια, ο πίνακας περνά στη συνάρτηση plot_Rot(R_matrix), η οποία αναλαμβάνει να απεικονίσει το αντίστοιχο σύστημα συντεταγμένων με χρήση γραφικού διαγράμματος.



Εικόνα 15 Απεικόνιση τυχαίου συστήματος συντεταγμένων στο χώρο βάσει πίνακα περιστροφής R

PART B

Functions R = rotX(th),R = rotY(th) και R = rotZ(th)

Το δεύτερο μέρος της εργασίας επικεντρώνεται στη δημιουργία και απεικόνιση προκαθορισμένων περιστροφών ως προς τους κύριους άξονες του τρισδιάστατου χώρου. Οι συναρτήσεις που υλοποιούνται είναι οι rotx(theta), roty(theta) και rotz(theta) και επιστρέφουν πίνακες περιστροφής 3×3 για γωνία περιστροφής theta γύρω από τους άξονες X, Y και Z αντίστοιχα.

Η μαθηματική μορφή αυτών των πινάκων είναι η εξής:

Για περιστροφή γύρω από τον άξονα X:

$$Rx(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix}$$

Για περιστροφή γύρω από τον άξονα Y:

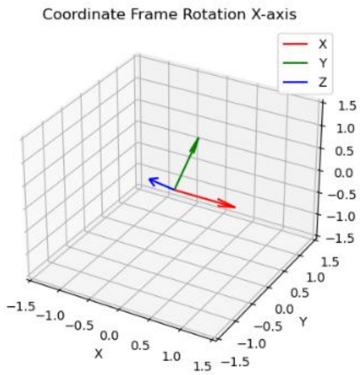
$$Ry(\theta) = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\Theta & 0 & cos\theta \end{bmatrix}$$

Για περιστροφή γύρω από τον άξονα Z:

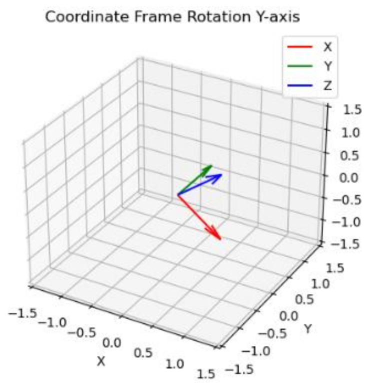
$$Rz(\theta) = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Αφού οριστούν οι συναρτήσεις, δημιουργείται μια γωνία περιστροφής 45° (δηλαδή π/4 ραντ), και υπολογίζονται οι αντίστοιχοι πίνακες. Οι πίνακες εκτυπώνονται στην έξοδο για έλεγχο, και στη συνέχεια χρησιμοποιείται η συνάρτηση plot_Rot() για να απεικονιστούν τα συστήματα συντεταγμένων που προκύπτουν από κάθε μία από αυτές τις περιστροφές.

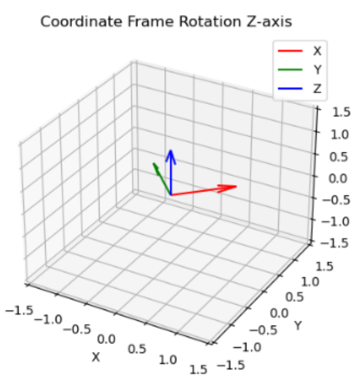
Κάθε γράφημα δείχνει την επίδραση της αντίστοιχης περιστροφής στους άξονες του συστήματος συντεταγμένων. Για παράδειγμα, μετά από περιστροφή γύρω από τον άξονα X, οι άξονες Y και Z αλλάζουν κατεύθυνση, ενώ ο X παραμένει αμετάβλητος.



Εικόνα 16 Σύστημα συντεταγμένων μετά από περιστροφή 45° ως προς τον άξονα X. Οι άξονες Y και Z έχουν αλλάξει κατεύθυνση, ενώ ο X παραμένει αμετάβλητος.



Εικόνα 17 Σύστημα συντεταγμένων μετά από περιστροφή 45° ως προς τον άξονα Y. Οι άξονες X και Z έχουν περιστραφεί, ενώ ο Y παραμένει σταθερός.



Εικόνα 18 Σύστημα συντεταγμένων μετά από περιστροφή 45° ως προς τον άξονα Z. Οι άξονες X και Y έχουν περιστραφεί στο επίπεδο XY, ενώ ο Z παραμένει αμετάβλητος.

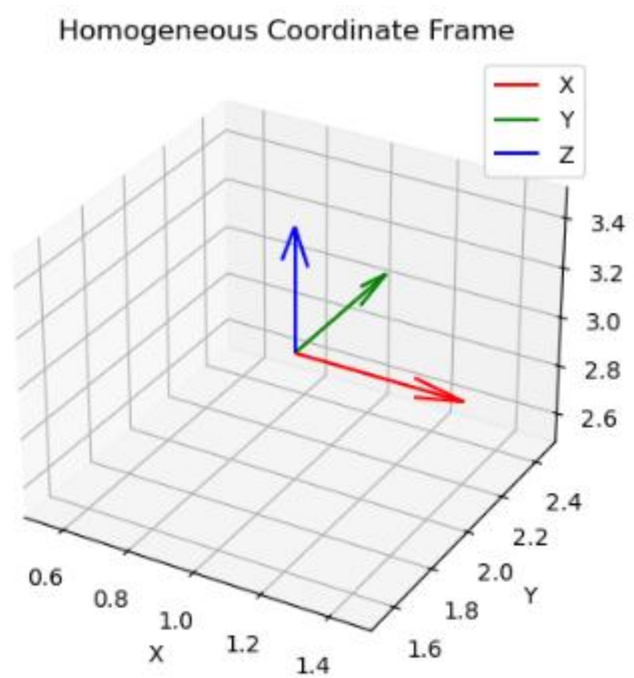
Function plot_hom(G,scale)

Η δεύτερη συνάρτηση plot_hom(G, scale), η οποία είναι υπεύθυνη για την απεικόνιση ενός συστήματος συντεταγμένων που περιγράφεται από έναν ομογενή μετασχηματισμό 4×4.

Ο πίνακας G περιέχει τις πρώτες τρεις στήλες που ορίζουν την περιστροφή (δηλαδή τους κατευθυντήριους άξονες X, Y, Z) και την τέταρτη στήλη (τα πρώτα τρία στοιχεία της) ορίζει τη μετατόπιση (θέση του συστήματος στο χώρο).

Η συνάρτηση εξάγει τον προσανατολισμό και τη θέση από τον πίνακα G, και στη συνέχεια σχεδιάζει τρία διανύσματα από το σημείο origin που αντιστοιχεί στη θέση του συστήματος, με χρώματα κόκκινο (άξονας X), πράσινο (άξονας Y) και μπλε (άξονας Z). Η παράμετρος scale καθορίζει το μήκος των αξόνων στο γράφημα, ώστε να είναι οπτικά κατάλληλοι για το εκάστοτε μέγεθος μετασχηματισμού.

Στο παράδειγμα που ακολουθεί, χρησιμοποιείται ο ταυτοτικός ομογενής πίνακας (np.eye(4)), στον οποίο προστίθεται μετατόπιση στις θέσεις (1, 2, 3) για τους άξονες X, Y και Z αντίστοιχα. Το αποτέλεσμα είναι η απεικόνιση ενός συστήματος συντεταγμένων μετατοπισμένου στον χώρο, αλλά χωρίς αλλαγή προσανατολισμού.



Εικόνα 19 Απεικόνιση συστήματος συντεταγμένων με χρήση ομογενούς μετασχηματισμού. Το σύστημα έχει μετατοπιστεί στο σημείο (1, 2, 3) χωρίς περιστροφή.

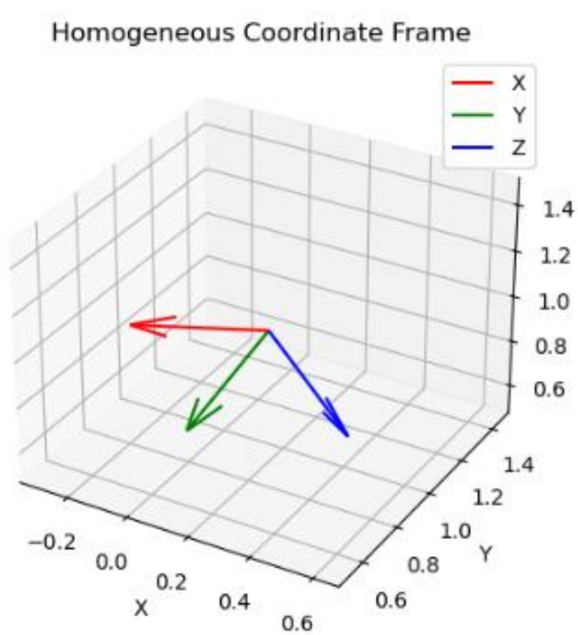
Functions G = homogen(R, p)

Η τρίτη συνάρτηση homegen(R, p) δημιουργεί έναν ομογενή μετασχηματισμό 4×4, συνδυάζοντας μια περιστροφή και μια μετατόπιση στον τρισδιάστατο χώρο. Η συνάρτηση δέχεται ως είσοδο τον πίνακα περιστροφής R, ο οποίος είναι ένας πίνακας 3×3 που καθορίζει τον προσανατολισμό ενός τοπικού συστήματος συντεταγμένων και το διάνυσμα θέσης p, που είναι μια λίστα ή πίνακας με 3 στοιχεία και καθορίζει τη μετατόπιση του συστήματος στον χώρο.

Η υλοποίηση της συνάρτησης κατασκευάζει έναν ταυτοτικό πίνακα 4×4 και στη συνέχεια τοποθετεί τον πίνακα R στα πρώτα 3×3 στοιχεία του πίνακα και τοποθετεί το διάνυσμα p στη στήλη της μετατόπισης, δηλαδή στα πρώτα 3 στοιχεία της τελευταίας στήλης.

Το αποτέλεσμα είναι ένας πλήρης ομογενής μετασχηματισμός, ο οποίος μπορεί να χρησιμοποιηθεί για να περιγράψει τη θέση και τον προσανατολισμό ενός αντικειμένου στον χώρο.

Στο παράδειγμα που ακολουθεί, δημιουργείται ένας τυχαίος πίνακας περιστροφής με χρήση της συνάρτησης generate_rot() και ένα τυχαίο διάνυσμα θέσης. Ο προκύπτων πίνακας μετασχηματισμού G περνάει στη συνάρτηση plot_hom(G, scale) για να απεικονιστεί στο τρισδιάστατο χώρο.



Εικόνα 20 Σύστημα συντεταγμένων που προκύπτει από τη σύνθεση τυχαίου πίνακα περιστροφής και διανύσματος θέσης, όπως κατασκευάζεται με τη συνάρτηση homegen(R, p).

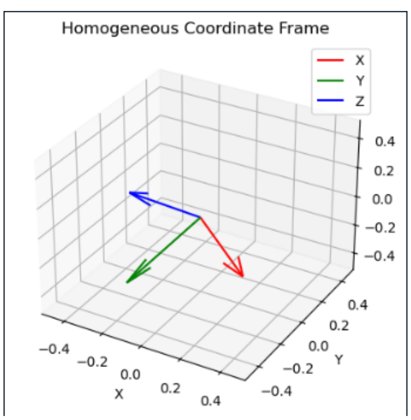
Functions G = gr(R),G = gp(p)

Στο τέταρτο μέρος υλοποιούνται δύο συναρτήσεις: gr(R) και gp(p). Οι συναρτήσεις αυτές κατασκευάζουν μεμονωμένους ομογενείς μετασχηματισμούς για περιστροφή και μετατόπιση αντίστοιχα.

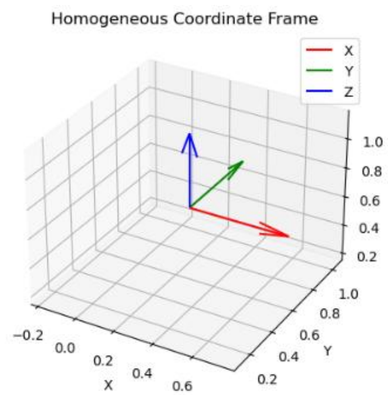
Η συνάρτηση gr(R) δέχεται έναν πίνακα περιστροφής 3×3 και επιστρέφει έναν πίνακα 4×4, όπου το υποπλέγμα των 3×3 περιέχει τον πίνακα R, και τα υπόλοιπα στοιχεία διαμορφώνουν έναν έγκυρο ομογενή μετασχηματισμό. Η μετατόπιση σε αυτήν την περίπτωση παραμένει μηδενική.

Αντίστοιχα, η συνάρτηση gp(p) κατασκευάζει έναν ομογενή μετασχηματισμό που περιλαμβάνει μόνο μετατόπιση. Ο πίνακας ξεκινά ως μοναδιαίος (ταυτότητα), και τα πρώτα τρία στοιχεία της τελευταίας στήλης αντικαθίστανται από τα στοιχεία του διανύσματος p. Ο πίνακας περιστροφής είναι στην προεπιλεγμένη μορφή της ταυτότητας (δηλαδή χωρίς περιστροφή).

Οι δύο συναρτήσεις χρησιμοποιούνται για τη δημιουργία ξεχωριστών μετασχηματισμών, πρώτα παράγεται ένας τυχαίος πίνακας περιστροφής με τη generate_rot() και μετατρέπεται σε ομογενή μορφή με τη gr(R) και έπειτα δημιουργείται ένα τυχαίο διάνυσμα μετατόπισης p, και μέσω της gp(p) κατασκευάζεται ο αντίστοιχος πίνακας μεταφοράς και οι δύο μετασχηματισμοί απεικονίζονται με χρήση της plot_hom(), αποκαλύπτοντας χωριστά την επίδραση της καθαρής περιστροφής και της καθαρής μετατόπισης.



Εικόνα 21 Απεικόνιση ομογενούς μετασχηματισμού που περιέχει μόνο περιστροφή. Το σύστημα συντεταγμένων παραμένει στη θέση (0, 0, 0), αλλάζει μόνο ο προσανατολισμός του.



Εικόνα 22 Απεικόνιση ομογενούς μετασχηματισμού που περιέχει μόνο μετατόπιση. Το σύστημα συντεταγμένων έχει μετακινηθεί στον χώρο, χωρίς να αλλάξει ο προσανατολισμός του.

Functions G = gRX(theta),G = gRY(theta),G = gRZ(theta)

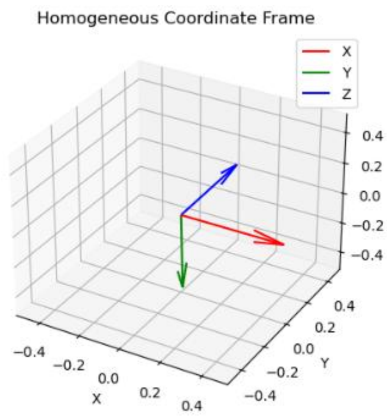
Οι συναρτήσεις gRX(θ), gRY(θ) και gRZ(θ) αποτελούν βοηθητικά εργαλεία για την κατασκευή ομογενών μετασχηματισμών που περιλαμβάνουν καθαρή περιστροφή γύρω από έναν από τους τρεις κύριους άξονες (X, Y, Z) στον τρισδιάστατο χώρο.

Κάθε συνάρτηση δέχεται ως είσοδο μια γωνία περιστροφής σε μοίρες, την μετατρέπει σε ακτίνια με χρήση της np.deg2rad, δημιουργεί τον αντίστοιχο πίνακα περιστροφής 3×3 σύμφωνα με τη γνωστή μορφή των περιστροφών Euler και χρησιμοποιεί την gr(R) για να τοποθετήσει τον πίνακα σε έναν πλήρη πίνακα ομογενούς μετασχηματισμού 4×4.

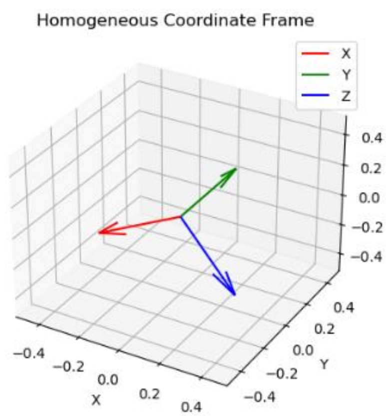
Οι πίνακες που προκύπτουν είναι Gx, Gy και Gz, που αντιστοιχούν σε περιστροφές ως προς τους άξονες X, Y και Z αντίστοιχα.

Για την επίδειξη, παράγονται τυχαίες γωνίες περιστροφής στο διάστημα [−180°, 180°] και εμφανίζονται τα αποτελέσματα με χρήση της plot_hom(), η οποία επιτρέπει την οπτική επιβεβαίωση των μετασχηματισμένων αξόνων.

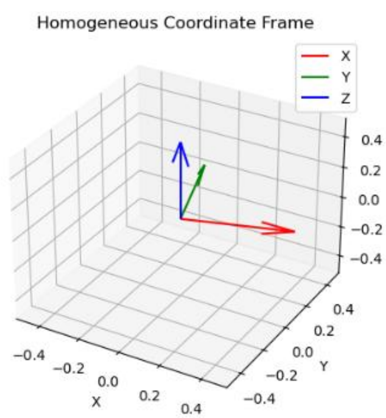
Κάθε απεικόνιση δείχνει τον προσανατολισμό του συστήματος συντεταγμένων μετά από την αντίστοιχη περιστροφή. Οι περιστροφές αυτές, όταν απομονώνονται, είναι εξαιρετικά χρήσιμες για την κατανόηση της επίδρασης καθενός από τους άξονες χωριστά.



Εικόνα 23 Ομογενής μετασχηματισμός με περιστροφή γύρω από τον άξονα X κατά τυχαία γωνία. Παρατηρείται αλλαγή κατεύθυνσης των αξόνων Y και Z, ενώ ο άξονας X παραμένει σταθερός.



Εικόνα 24 Ομογενής μετασχηματισμός με περιστροφή γύρω από τον άξονα Z κατά τυχαία γωνία. Οι άξονες X και Y περιστρέφονται στο επίπεδο XY, με τον Z να παραμένει σταθερός.



Εικόνα 25 Ομογενής μετασχηματισμός με περιστροφή γύρω από τον άξονα Y κατά τυχαία γωνία. Ο άξονας Y διατηρείται, ενώ οι άξονες X και Z περιστρέφονται στο επίπεδο XZ.

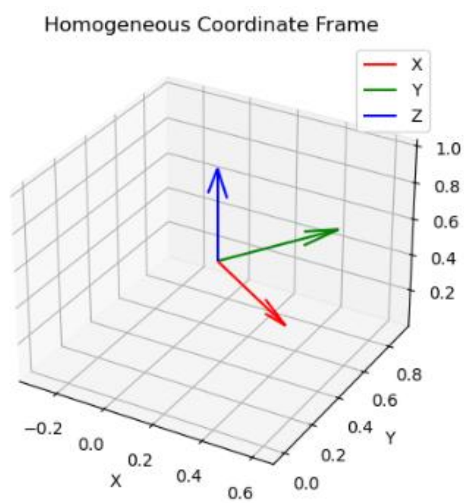
Function vout =rotAndTranVec(G,vin)

Η συνάρτηση rotAndTranVec(G, vin) εφαρμόζει έναν ομογενή μετασχηματισμό σε ένα διάνυσμα τριών διαστάσεων. Συγκεκριμένα, μετασχηματίζει ένα σημείο του χώρου (το vin) με βάση έναν πίνακα 4×4 που συνδυάζει περιστροφή και μετατόπιση.

Η λειτουργία είναι, του διάνυσμα εισόδου vin, που είναι 3 στοιχείων (X, Y, Z), μετατρέπεται σε ομογενή συντεταγμένη 4 στοιχείων με την προσθήκη μιας μονάδας στο τέλος (vin_hom = [x, y, z, 1]), ο πίνακας G πολλαπλασιάζεται με το vin_hom, προκειμένου να εφαρμοστεί ο πλήρης μετασχηματισμός (περιστροφή + μετατόπιση), το αποτέλεσμα vout_hom είναι ένα νέο διάνυσμα 4 στοιχείων, από το οποίο επιστρέφονται μόνο τα πρώτα 3 (οι νέες συντεταγμένες στο χώρο).

Για τη δοκιμή της συνάρτησης δημιουργείται πρώτα ένας πίνακας περιστροφής γύρω από τον άξονα Z μέσω της gRZ(θ) με τυχαία γωνία θ, δημιουργείται επίσης μια τυχαία μετάθεση p και μετατρέπεται σε ομογενή πίνακα με gp(p) και οι δύο πίνακες G_rot και G_trans πολλαπλασιάζονται για να σχηματίσουν τον συνολικό μετασχηματισμό G_total.

Ένα τυχαίο διάνυσμα vin υποβάλλεται στον μετασχηματισμό και προκύπτει το vout. Το αποτέλεσμα εκτυπώνεται μαζί με την τιμή της γωνίας και το διάνυσμα μετάθεσης. Η τελική μορφή του μετασχηματισμού απεικονίζεται με τη plot_hom(G_total).



Εικόνα 26 Απεικόνιση συνδυασμένου ομογενούς μετασχηματισμού: περιστροφή γύρω από τον άξονα Z και μετατόπιση στον χώρο. Το τοπικό σύστημα συντεταγμένων έχει αλλάξει τόσο θέση όσο και προσανατολισμό.

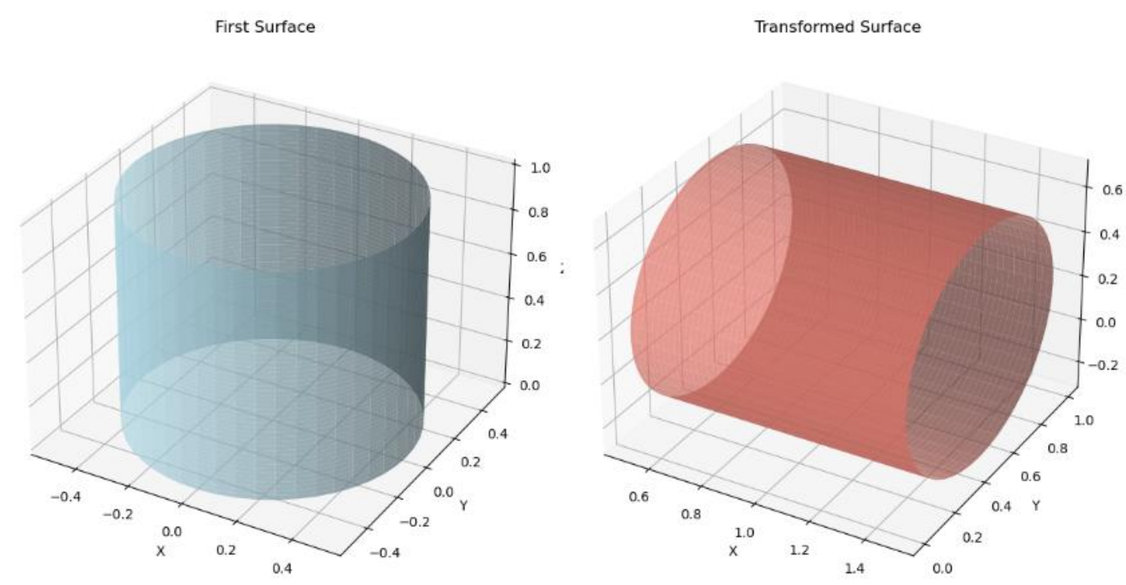
Function [Xout,Yout,Zout] = rotAndTrans_shape(X,Y,Z,G)

Η συνάρτηση rotAndTrans_shape(G, X, Y, Z) εφαρμόζει έναν ομογενή μετασχηματισμό 4×4 σε ένα σύνολο σημείων που περιγράφουν μια επιφάνεια στον τρισδιάστατο χώρο. Οι πίνακες X, Y και Z αναπαριστούν τις συντεταγμένες μιας παραμετρικής επιφάνειας, όπως ενός κυλίνδρου, τα σημεία μετατρέπονται σε μορφή 4×N, ώστε κάθε στήλη να αποτελεί ένα σημείο στην ομογενή του μορφή: $[x,y,z,1]^T$, ο πίνακας G εφαρμόζεται ταυτόχρονα σε όλα τα σημεία με πολλαπλασιασμό κατά μήκος των στηλών και τα μετασχηματισμένα σημεία επανατοποθετούνται στα αντίστοιχα σχήματα πλέγματος και επιστρέφονται ως X_new, Y_new, Z_new.

Η επιφάνεια που χρησιμοποιείται στο παράδειγμα είναι ένας κύλινδρος, ο οποίος παράγεται μέσω της συνάρτησης generate_cylinder(). Η συνάρτηση επιστρέφει πίνακες X, Y, Z με βάση παραμετρική περιγραφή σε κυλινδρικές συντεταγμένες.

Στη συνέχεια, εφαρμόζεται περιστροφή 90° γύρω από τον άξονα Y μέσω της gRY(θ), προστίθεται μετατόπιση με διάνυσμα p = [0.5, 0.5, 0.2] και οι δύο μετασχηματισμοί συνδυάζονται με G = G_trans @ G_rot.

Το αρχικό και το μετασχηματισμένο σχήμα απεικονίζονται δίπλα-δίπλα με χρήση της plot_surfaces, όπου το πρώτο εμφανίζεται σε γαλάζιο και το μετασχηματισμένο σε ροζ.



Εικόνα 27 Σύγκριση αρχικής (αριστερά) και μετασχηματισμένης (δεξιά) επιφάνειας κυλίνδρου. Ο μετασχηματισμός περιλαμβάνει περιστροφή 90° γύρω από τον άξονα Y και μετατόπιση κατά [0.5,0.5,0.2]

Το ρομπότ

Η συνάρτηση g0e(q1, q2, q3, L1=0.5, L2=0.5) υπολογίζει τον ομογενή μετασχηματισμό από τη βάση έως το άκρο ενός ρομποτικού βραχίονα με τρεις κινηματικούς βαθμούς ελευθερίας. Η υλοποίηση βασίζεται στην έννοια της προώθησης κινηματικής (forward kinematics), όπου κάθε άρθρωση και κάθε κρίκος αντιπροσωπεύονται από κατάλληλους μετασχηματισμούς.

Αναλυτικά:

- Το q1 είναι η κάθετη μετατόπιση στον άξονα Z
- Το q2 είναι μια περιστροφή γύρω από τον άξονα Y.
- Το q3 είναι μια περιστροφή γύρω από τον άξονα Z.
- Τα L1 και L2 είναι τα μήκη των δύο συνδέσμων του βραχίονα.

Η συνολική κινηματική αλυσίδα υλοποιείται ως εξής:

1. T1 = gp([0, 0, q1]): κατακόρυφη μετατόπιση κατά q1 στον άξονα Z.
2. T2 = gRY(...): περιστροφή κατά q2 γύρω από τον άξονα Y.
3. T3 = gp([L1, 0, 0]): μετατόπιση κατά μήκος του πρώτου συνδέσμου στον άξονα X.
4. T4 = gRZ(...): περιστροφή κατά q3 γύρω από τον άξονα Z.
5. T5 = gp([L2, 0, 0]): τελική μετατόπιση κατά μήκος του δεύτερου συνδέσμου στον άξονα X.

Ο συνολικός μετασχηματισμός προκύπτει από τη σύνθεση των παραπάνω:

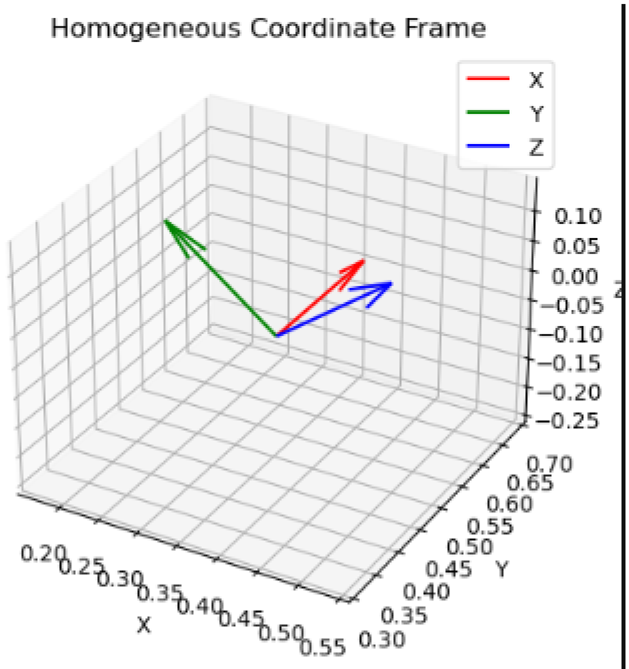
$$G=T1\cdot T2\cdot T3\cdot T4\cdot T5$$

Το αποτέλεσμα είναι ένας πίνακας 4×4 που περιγράφει τη θέση και τον προσανατολισμό του τελικού εκτελεστή (end-effector) στο παγκόσμιο σύστημα συντεταγμένων.

Η συνάρτηση plot_robot(q1, q2, q3) καλεί εσωτερικά τη g0e(q1, q2, q3) για να υπολογίσει τη θέση και τον προσανατολισμό του τελικού σημείου ενός ρομποτικού βραχίονα με τρεις βαθμούς ελευθερίας, και στη συνέχεια απεικονίζει το σύστημα συντεταγμένων που αντιστοιχεί στον εκτελεστή χρησιμοποιώντας τη plot_hom().

Το επιλεγμένο scale (0.2) εξασφαλίζει ότι οι άξονες θα είναι ευδιάκριτοι αλλά όχι δυσανάλογοι στο γράφημα.

Στο παράδειγμα, η μετατόπιση στον άξονα Z (q1) είναι 0.3 μονάδες, η γωνία περιστροφής γύρω από τον Y είναι 45° και η γωνία περιστροφής γύρω από τον Z είναι 90°.



Εικόνα 28 Απεικόνιση της τελικής θέσης και προσανατολισμού του εκτελεστή ρομποτικού βραχίονα για τις τιμές $q_1 = 0.3$, $q_2 = 45°$, $q_3 = 90°$

Χώρος εργασίας

Το συγκεκριμένο τμήμα του κώδικα υπολογίζει και απεικονίζει το γεωμετρικό χώρο εργασίας του ρομποτικού βραχίονα, δηλαδή το σύνολο όλων των πιθανών θέσεων που μπορεί να καταλάβει το άκρο του (end-effector), λαμβάνοντας υπόψη τα όρια των τριών παραμέτρων q1, q2, και q3.

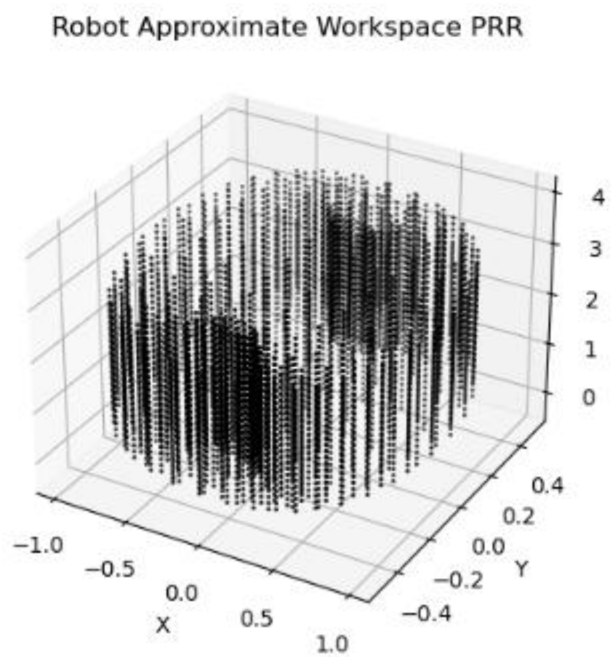
Το q1 μεταβάλλεται από 0.7 έως 3.0 και αντιστοιχεί σε κατακόρυφη μετατόπιση, έπειτα το q2 καλύπτει εύρος από −170° έως +170°, αντιπροσωπεύοντας περιστροφή γύρω από τον άξονα Y και το q3 καλύπτει εύρος από −135° έως +135°, αντιπροσωπεύοντας περιστροφή γύρω από τον άξονα Z.

Για κάθε δυνατό συνδυασμό τιμών των παραμέτρων Υπολογίζεται ο συνολικός ομογενής μετασχηματισμός μέσω της g0e(q1, q2, q3).

1. Εξάγεται το διάνυσμα θέσης του εκτελεστή (το 4ο στοιχείο της τελευταίας στήλης του πίνακα G).
2. Η θέση αποθηκεύεται στη λίστα points.

Αφού υπολογιστούν όλες οι δυνατές θέσεις, αυτές απεικονίζονται ως σύννεφο σημείων (scatter plot) σε τρισδιάστατο χώρο. Κάθε μαύρη κουκκίδα στην τελική εικόνα αντιστοιχεί σε μια διαφορετική θέση του εκτελεστή για συγκεκριμένο συνδυασμό γωνιών και μετατόπισης.

Ο τελικός τίτλος του γραφήματος "Robot Approximate Workspace PRR" περιγράφει επακριβώς το περιεχόμενο: πρόκειται για μια **προσέγγιση του χώρου εργασίας** ενός ρομποτικού βραχίονα τύπου PRR.



Εικόνα 29 Γραφική αναπαράσταση του προσεγγιστικού χώρου εργασίας του ρομποτικού βραχίονα PRR. Κάθε σημείο αντιστοιχεί σε μια διαφορετική θέση του εκτελεστή για επιτρεπτές τιμές των παραμέτρων q_1, q_2 και q_3

Function [q1,q2,q3,d]=ikine(p)

Η συνάρτηση ikine(p_desired, all_q, all_p) υλοποιεί έναν απλό αλγόριθμο για την επίλυση του προβλήματος της αντίστροφης κινηματικής, δηλαδή την εύρεση των τιμών των παραμέτρων (q_1, q_2, q_3) που οδηγούν σε μια επιθυμητή θέση του εκτελεστή (end-effector).

Λαμβάνει ως είσοδο την επιθυμητή θέση p_desired καθώς και δύο πίνακες τον all_q όπου είναι όλοι οι συνδυασμοί των παραμέτρων που έχουν δοκιμαστεί προηγουμένως και τον all_p που τις αντίστοιχες θέσεις του εκτελεστή για κάθε τέτοιο συνδυασμό. Στην συνέχεια υπολογίζει την ευκλείδεια απόσταση από κάθε θέση all_p προς το p_desired, επιλέγει τη θέση που έχει τη μικρότερη απόσταση (δηλαδή είναι η πλησιέστερη δυνατή προσέγγιση) και επιστρέφει τις αντίστοιχες παραμέτρους (q_1, q_2, q_3) και την απόσταση d ως δείκτη σφάλματος.

Στο παράδειγμα που ακολουθεί, το σημείο p = [1.5, 0.2, 1.1] επιλέγεται ως στόχος. Η ikine() επιστρέφει τη βέλτιστη λύση και το αντίστοιχο σφάλμα. Η τιμή επαληθεύεται μέσω της συνάρτησης g0e() για τις προκύπτουσες γωνίες, και η τελική θέση εκτυπώνεται.

Η μέθοδος αυτή είναι απλή αλλά αποτελεσματική όταν υπάρχει ήδη υπολογισμένο ένα σύνολο σημείων από forward kinematics, όπως έγινε στο προηγούμενο βήμα (workspace PRR). Παρόλο που δεν παρέχει ακριβή αναλυτική λύση, είναι πολύτιμη για την προσεγγιστική εύρεση λύσεων σε προβλήματα ελέγχου ή προγραμματισμού ρομποτικής τροχιάς.

```
Solution: 0.9421052631578947 -0.15616103833633455 0.12401023632591279 error: 0.5341454475065656
Calculated position: [0.98403864 0.06184632 1.0970352 ]
```

Στην παραπάνω φωτογραφία βλέπουμε την συνάρτηση να επιλύει προσεγγιστικά την αντίστροφη κινηματική βρίσκοντας το κοντινότερο σημείο σε ένα προϋπολογισμένο σύνολο θέσεων. Στο παράδειγμα, η επιθυμητή θέση βρέθηκε εκτός του χώρου εργασίας, με σφάλμα περίπου 0.53.

Functions [u,u_dot] = get_trajectory(p0,pf,t,tf)

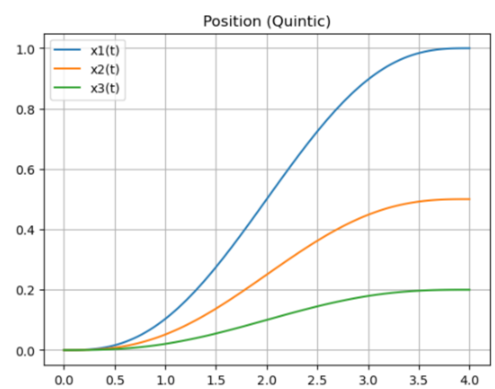
Η συνάρτηση get_trajectory(p0, pf, t, tf) υπολογίζει την κίνηση σημείου στον χώρο από αρχική θέση p0 προς τελική θέση pf μέσα σε χρόνο tf, χρησιμοποιώντας πεντασωνύμους πολυωνύμους. Η επιλογή τέτοιου βαθμού πολυωνύμου επιτρέπει τον ορισμό εξομαλυνμένων τροχιών με μηδενικές αρχικές και τελικές ταχύτητες και επιταχύνσεις.

Για κάθε διάσταση (X, Y, Z) ορίζεται ένα σύστημα 6 εξισώσεων με 6 αγνώστους (τους συντελεστές του πολυωνύμου 5ου βαθμού), η συνάρτηση επιβάλλει αρχικές και τελικές συνθήκες θέσης, ταχύτητας και επιτάχυνσης και η λύση του συστήματος δίνει τους συντελεστές των πολυωνύμων για κάθε διάσταση.

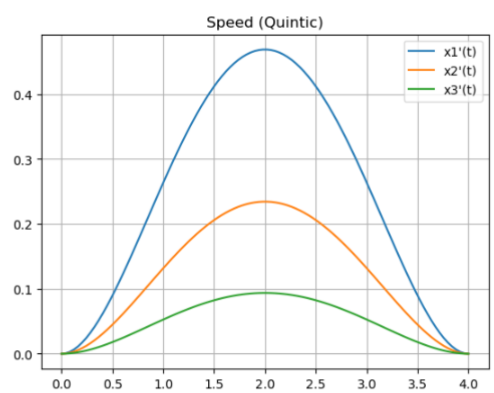
Μετά τον υπολογισμό των συντελεστών, οι θέσεις και οι ταχύτητες σε οποιαδήποτε χρονική στιγμή t υπολογίζονται με πολυωνυμική αξιολόγηση.

Στο παράδειγμα το σημείο μετακινείται από [0, 0, 0] προς [1, 0.5, 0.2] σε 4 δευτερόλεπτα, οι τροχιές υπολογίζονται σε 100 χρονικές στιγμές, και αποθηκεύονται οι αντίστοιχες θέσεις και ταχύτητες, δημιουργούνται δύο γραφήματα. Το πρώτο απεικονίζει τη θέση σε κάθε άξονα σε συνάρτηση με τον χρόνο και το δεύτερο δείχνει την ταχύτητα, επίσης ως προς τον χρόνο.

Τα διαγράμματα παρουσιάζουν εξομαλυσμένες καμπύλες, χωρίς ασυνέχειες ή απότομες μεταβολές, κάτι που εξασφαλίζει ομαλή και ασφαλή κίνηση.



Εικόνα 30 Χρονική εξέλιξη της θέσης στους άξονες X_1, X_2 και X_3 μέσω πεντασωνυμικής τροχιάς (quintic). Η τροχιά εξασφαλίζει ομαλή εκκίνηση και ακινητοποίηση.



Εικόνα 31 Χρονική εξέλιξη της ταχύτητας στους άξονες X_1, X_2 και X_3. Οι καμπύλες είναι λείες και συμμετρικές, με μηδενικές ταχύτητες στα άκρα της τροχιάς.

Script trajectory_joint_demo

Στο συγκεκριμένο τμήμα του κώδικα, σχεδιάζεται **μια ομαλή και χρονικά** ελεγχόμενη τροχιά για έναν ρομποτικό βραχίονα PRR, χρησιμοποιώντας τις γωνιακές παραμέτρους q_1, q_2, q_3. Η τροχιά ορίζεται από αρχικές τιμές:

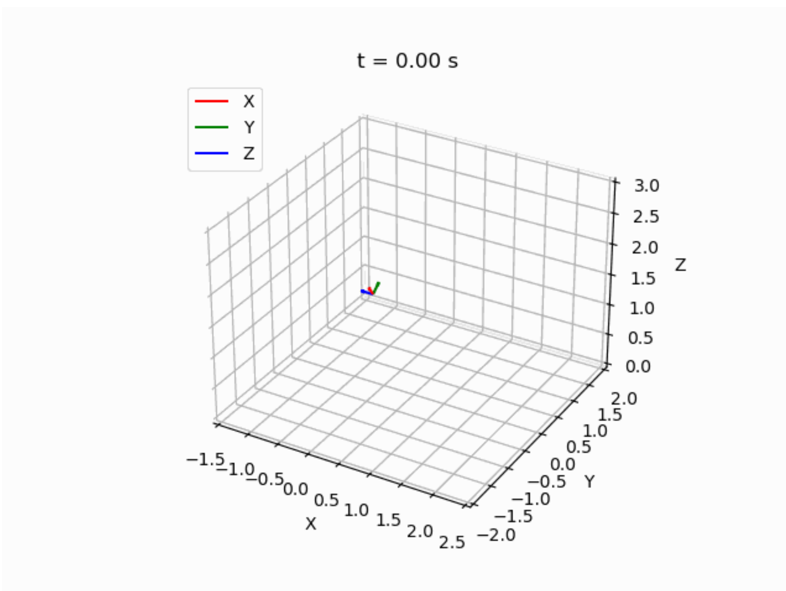
- $q_0 = 0.7, -90°, -35°$

και τελικές τιμές:

- $qf = 2.0, +70°, +145°$

με χρονική διάρκεια 6 δευτερολέπτων.

Η συνάρτηση get_trajectory χρησιμοποιείται για να δημιουργήσει μια πεντασωνυμική τροχιά για κάθε παράμετρο, εξασφαλίζοντας ομαλή έναρξη, μέγιστη ενδιάμεση ταχύτητα και ομαλή ακινητοποίηση.



Εικόνα 32 Animation της τροχιάς του συστήματος συντεταγμένων του τελικού εκτελεστικού σημείου του ρομπότ PRR κατά την κίνηση από αρχική σε τελική στάση. Οι άξονες X (κόκκινο), Y (πράσινο) και Z (μπλε) απεικονίζουν τον προσανατολισμό του εργαλείου σε κάθε

ο animation παρουσιάζει τη χρονική εξέλιξη της στάσης του εκτελεστικού σημείου ρομπότ PRR σε 3D χώρο. Οι άξονες X, Y και Z αποδίδονται χρωματικά και συνοδεύονται από υπόμνημα. Η χρήση της συνάρτησης plot_hom με υποστήριξη για animation καθιστά σαφή τη δυναμική αλλαγή προσανατολισμού, διευκολύνοντας την κατανόηση της τροχιάς.

Για την οπτικοποίηση της κίνησης του αρθρωτού ρομπότ PRR χρησιμοποιήθηκε η βιβλιοθήκη matplotlib και η συνάρτηση FuncAnimation. Οι αρχικές και τελικές τιμές των παραμέτρων των αρθρώσεων δόθηκαν με τη μορφή πινάκων q0 και qf, και χρησιμοποιήθηκε πεντασωνυμική παρεμβολή μέσω της get_trajectory για τον υπολογισμό των ενδιάμεσων θέσεων. Το σύστημα σχεδιάστηκε σε 3D για κάθε χρονική στιγμή και αποθηκεύτηκε ως αρχείο .gif.

Η βασική ιδέα ήταν να παράγεται σε κάθε frame το ομογενές σύστημα συντεταγμένων του τελικού εκτελεστικού σημείου, το οποίο υπολογίζεται με τη g0e, και στη συνέχεια να σχεδιάζεται το σύστημα αξόνων του μέσω της plot_hom. Για τη δημιουργία της τροχιάς, το animation διατρέχει 100 χρονικά δείγματα της διαδρομής, εμφανίζοντας τις αντίστοιχες θέσεις του ρομπότ.

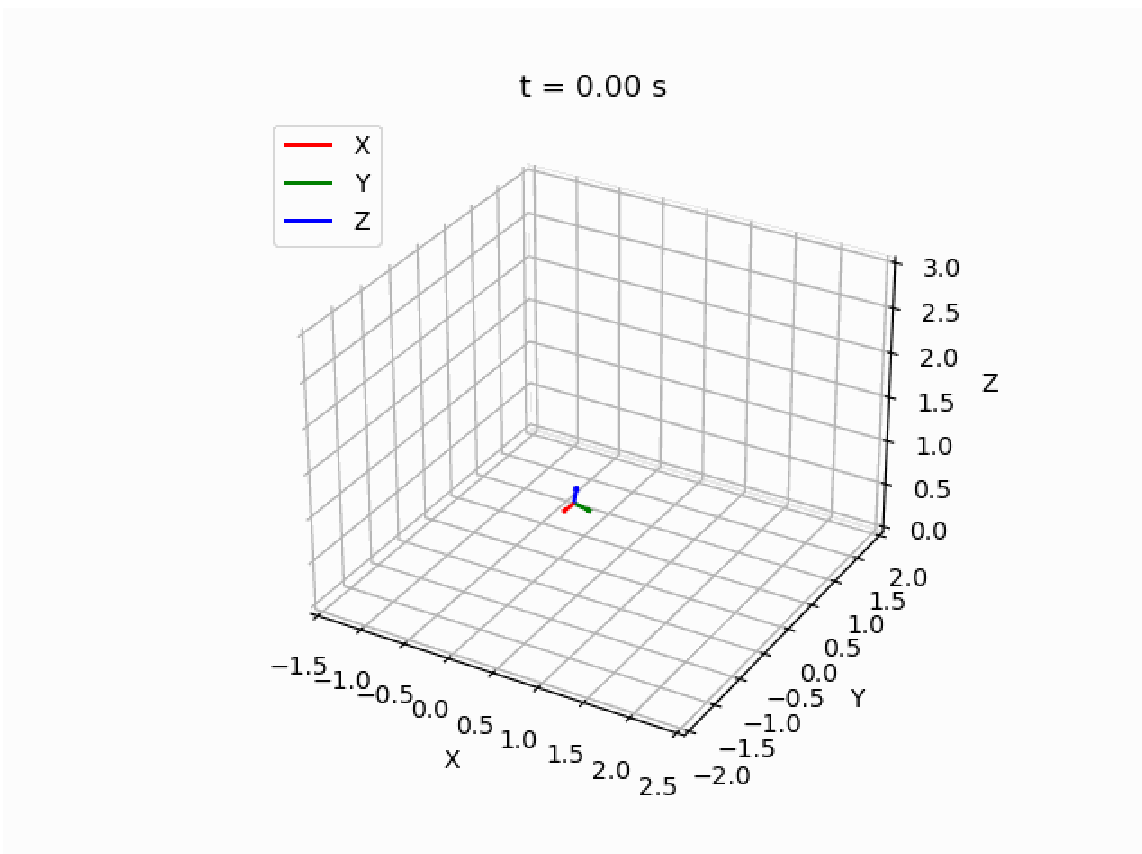
Επίσης να σημειωθεί ότι ξαναγράφτηκε η plot_hom για να είναι κατάλληλη για animation:

Έγινε στην καινούργια plot_hom προσθήκη ετικετών (label=...) στους άξονες και προσθήκη λεζάντας:

- ax.quiver(*origin, *x_axis, color='r', label='X')
- ax.quiver(*origin, *y_axis, color='g', label='Y')
- ax.quiver(*origin, *z_axis, color='b', label='Z')
- ax.legend(loc='upper left')

Script trajectory_task_demo

Ο κώδικας δημιουργεί ένα animation που αναπαριστά την κίνηση ενός ρομποτικού βραχίονα στον τρισδιάστατο χώρο, ακολουθώντας μια τροχιά η οποία έχει οριστεί στον task space. Αρχικά, υπολογίζεται η επιθυμητή πορεία του εργαλείου (end-effector) από μια αρχική θέση p0 σε μια τελική pf μέσω πλειστικής (quintic) πολυωνυμικής παρεμβολής. Στη συνέχεια, για κάθε χρονική στιγμή, εφαρμόζεται αλγόριθμος ανάστροφης κινηματικής ώστε να προκύψουν οι αντίστοιχες αρθρώσεις (q1, q2, q3) που τοποθετούν τον βραχίονα στην επιθυμητή θέση. Τέλος, η συνάρτηση FuncAnimation δημιουργεί και αποθηκεύει το κινούμενο γράφημα της διαδρομής, προσφέροντας εποπτική κατανόηση της συμπεριφοράς του ρομπότ καθώς εκτελεί μια εργασία μεταφοράς στο χώρο.



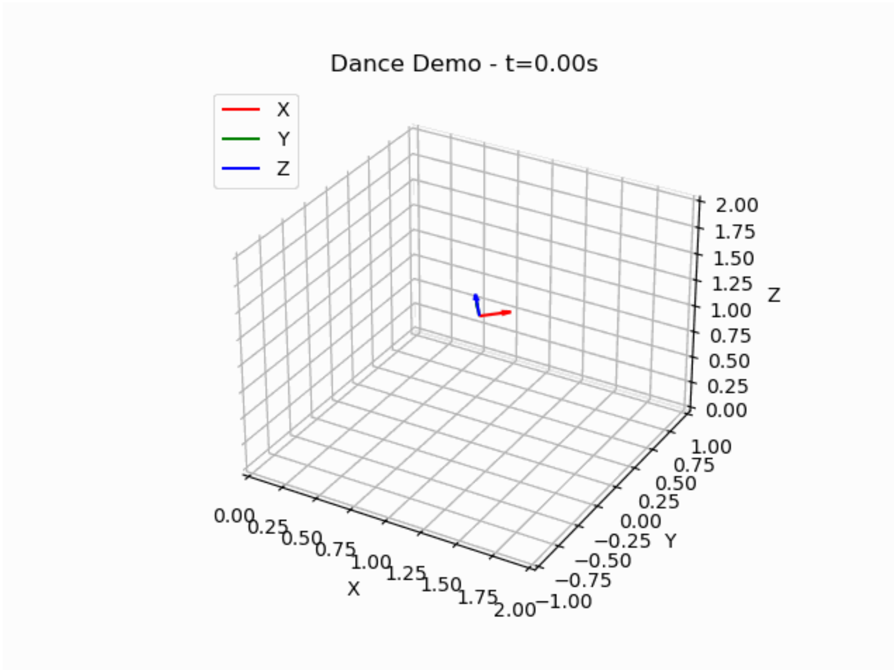
Εικόνα 33 Animation της εκτέλεσης ευθύγραμμης τροχιάς του τελικού εκτελεστικού σημείου του ρομπότ στον τρισδιάστατο χώρο. Η τροχιά σχεδιάζεται στον χώρο εργασίας και αναπαρίσταται δυναμικά μέσω αντίστροφης κινηματικής και οπτικοποίησης των αξόνων του εργαλείου.

Το animation ξεκινά από το σημείο p0 = [0.5, -0.5, 1.0] και καταλήγει στο pf = [1.0, 0.8, 1.8], αναδεικνύοντας την ικανότητα του ρομπότ να εκτελέσει μεταφορά εργαλείου με ακριβή έλεγχο στάσης και θέσης.

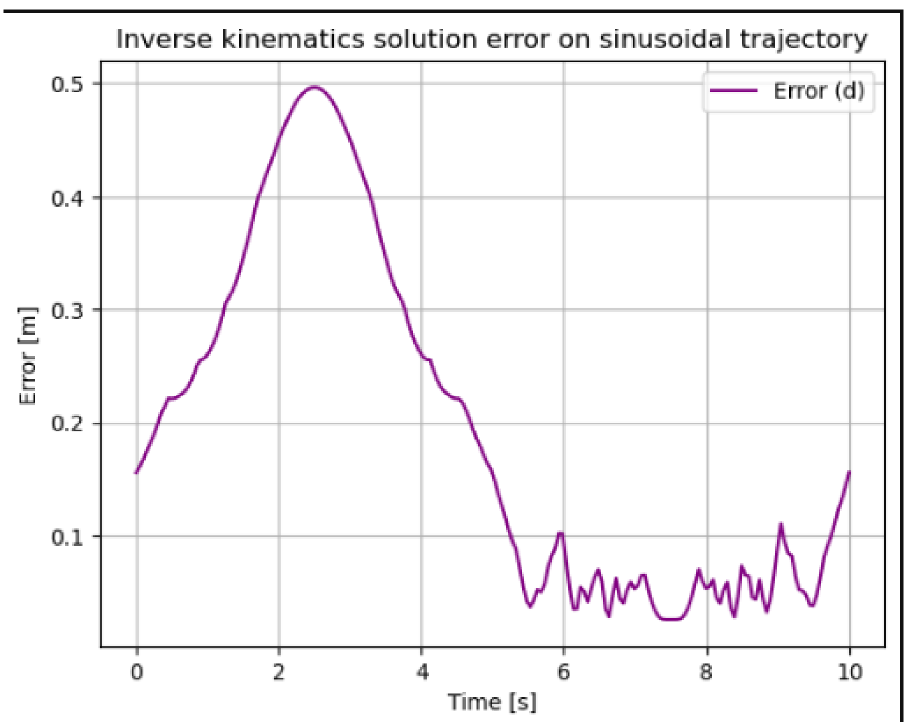
Script dance_demo

Ο κώδικας υλοποιεί μία περιοδική τροχιά τύπου “χορευτικής” κίνησης του τελικού εκτελεστή του ρομποτικού βραχίονα στον 3D χώρο. Η τροχιά ορίζεται παραμετρικά με ημιτονοειδείς συναρτήσεις ως προς τον χρόνο, ώστε να επιτευχθεί μία κίνηση που παρουσιάζει κυκλική μεταβολή στη διεύθυνση X, συχνότερη παλινδρόμηση στον άξονα Y, και κυματοειδή μεταβολή του ύψους στον Z. Η ακολουθία των σημείων της τροχιάς μετατρέπεται μέσω αντίστροφης κινηματικής σε τιμές αρθρώσεων (q1, q2, q3), ενώ σε κάθε χρονική στιγμή εμφανίζεται το αποτέλεσμα με χρήση της συνάρτησης plot_hom.

Η κίνηση καταγράφεται σε μορφή κινούμενης εικόνας (dance_demo.gif) επιτρέποντας έτσι την καθαρή απεικόνιση του τρόπου με τον οποίο το ρομπότ ακολουθεί την τροχιά. Επιπλέον, σχεδιάζεται και το σφάλμα της αντίστροφης κινηματικής σε σχέση με το χρόνο, επιβεβαιώνοντας ότι το ρομπότ ακολουθεί την επιθυμητή τροχιά με ικανοποιητική ακρίβεια.



Εικόνα 34 Ένα στιγμιότυπο της αρχικής κατάστασης της τροχιάς "Dance Demo", η οποία ακολουθεί μια περιοδική, πολύπλοκη διαδρομή στον χώρο. Οι άξονες X (κόκκινος), Y (πράσινος) και Z (μπλε) του εργαλείου δείχνουν την προσανατολισμένη θέση του στο τρισδιάστατο σύστη



Εικόνα 35 Χρονική εξέλιξη του σφάλματος θέσης κατά την παρακολούθηση της περιοδικής τροχιάς από το εργαλείο του ρομπότ. Το σφάλμα παραμένει εντός αποδεκτών ορίων, με μέγιστο γύρω στα 0.5 m και αισθητή μείωση μετά το 3ο δευτερόλεπτο, υποδεικνύοντας αποτελεσματική π