

Αναφορά άσκησης

Ονοματεπώνυμο	Κωνσταντίνος Δρακάκης
ΑΜ	TH20049
Email	th20049@edu.hmu.gr
Τίτλος άσκησης	08-α Άσκηση (Τοποθέτηση αντικειμένου σε τετραγωνική υποδοχή) + Παλιό θέμα 1
Ημερομηνία διεξαγωγής εκφώνησης	7/4/2025

Περιεχόμενα

Εκφώνηση άσκησης 7.....	3
Λύση άσκησης	5
Αριθμητική επίλυση μέσω python	6
Αριθμητική επίλυση μέσω MATLAB	7
Άσκηση 2	8

Εκφώνηση άσκησης 7

Χρήση βασικών συναρτήσεων για τον υπολογισμό και την κατανόηση των Ομογενών Μετασχηματισμών PYTHON version σελίδα 9 - 11

Άσκηση 7η / 08-α

Ένα ρομπότ 6 βαθμών ελευθερίας πρέπει να σηκώσει ένα αντικείμενο με τετραγωνική διατομή από μια συγκεκριμένη θέση και να το τοποθετήσει μέσα στο άνοιγμα μίας τετραγωνικής υποδοχής που βρίσκεται στην ίδια επιφάνεια σε σταθερή θέση. Οι διαστάσεις του αντικειμένου και της υποδοχής και τα πλαίσιά τους $\{C\}$ και $\{F\}$ καθώς και η αρχική κατάσταση του χώρου εργασίας φαίνονται στο σχήμα 1. Σαν αδρανειακό σύστημα αναφοράς θεωρείται το πλαίσιο της βάσης του βραχίονα $\{R\}$. Επίσης το πλαίσιο $\{O\}$ βρίσκεται στο κέντρο του τετραγωνικού ανοίγματος της υποδοχής στην επάνω επιφάνεια της. Στο σχήμα 2 φαίνεται λεπτομέρεια από το σύστημα συντεταγμένων $\{E\}$ του άκρου της αρπάγης σε σχέση με το πλαίσιο $\{H\} \equiv \{6\}$ του καρπού του ρομπότ.

α) Βρείτε τους πίνακες ομογενούς μετασχηματισμού. g_{6E} , g_{RC} , g_{FO} , g_{RF}

β) Για να εκτελεστεί η παραπάνω εργασία πρέπει ο βραχίονας να περάσει από 4 βασικές θέσεις που παρουσιάζονται στο σχήμα 3:

Η θέση (1) όταν η αρπάγη είναι πλησίον του αντικειμένου.

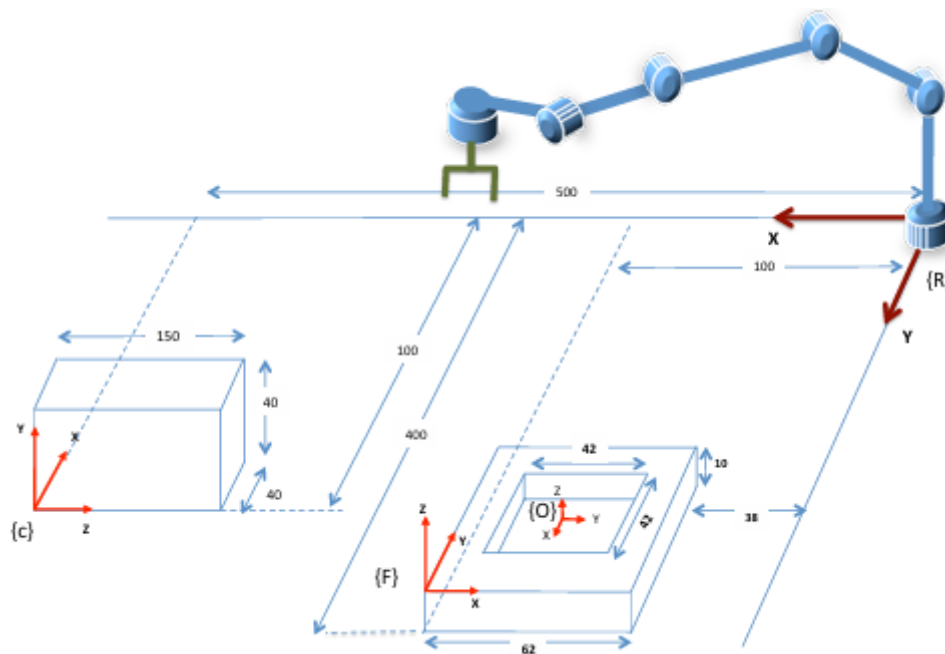
Η θέση (2) όταν η αρπάγη έχει πιάσει το αντικείμενο.

Η θέση (3) όταν το αντικείμενο είναι πλησίον του πλαισίου υποδοχής.

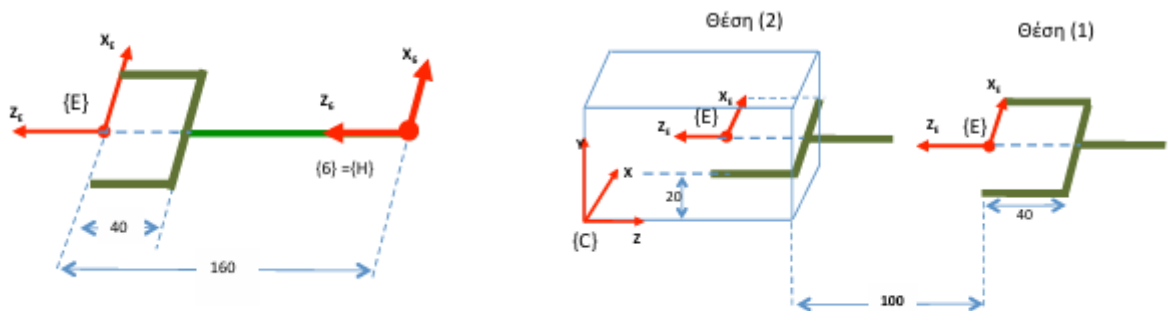
Η τελική θέση (4) με το αντικείμενο τοποθετημένο στο άνοιγμα της υποδοχής.

Για να εκτελεστεί η επιθυμητή εργασία συναρμολόγησης θα δοθούν εντολές κίνησης στον βραχίονα με την μορφή των μετασχηματισμών $g_{RH}(i)$, για τις θέσεις $i=1,2,3,4$. Για κάθε μια από τις θέσεις αυτές διατυπώστε μία εξίσωση ομογενών μετασχηματισμών που να περιέχει σαν μοναδικό άγνωστο τον ομογενή μετασχηματισμό $g_{RH}(i)$ και υπολογίστε τον.

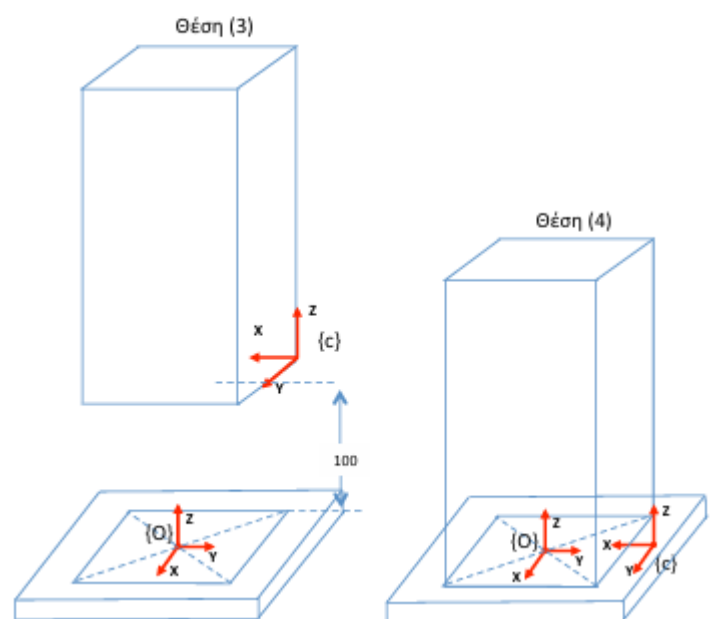
Σχήματα άσκησης



Σχήμα 1



Σχήμα 2



Σχήμα 3

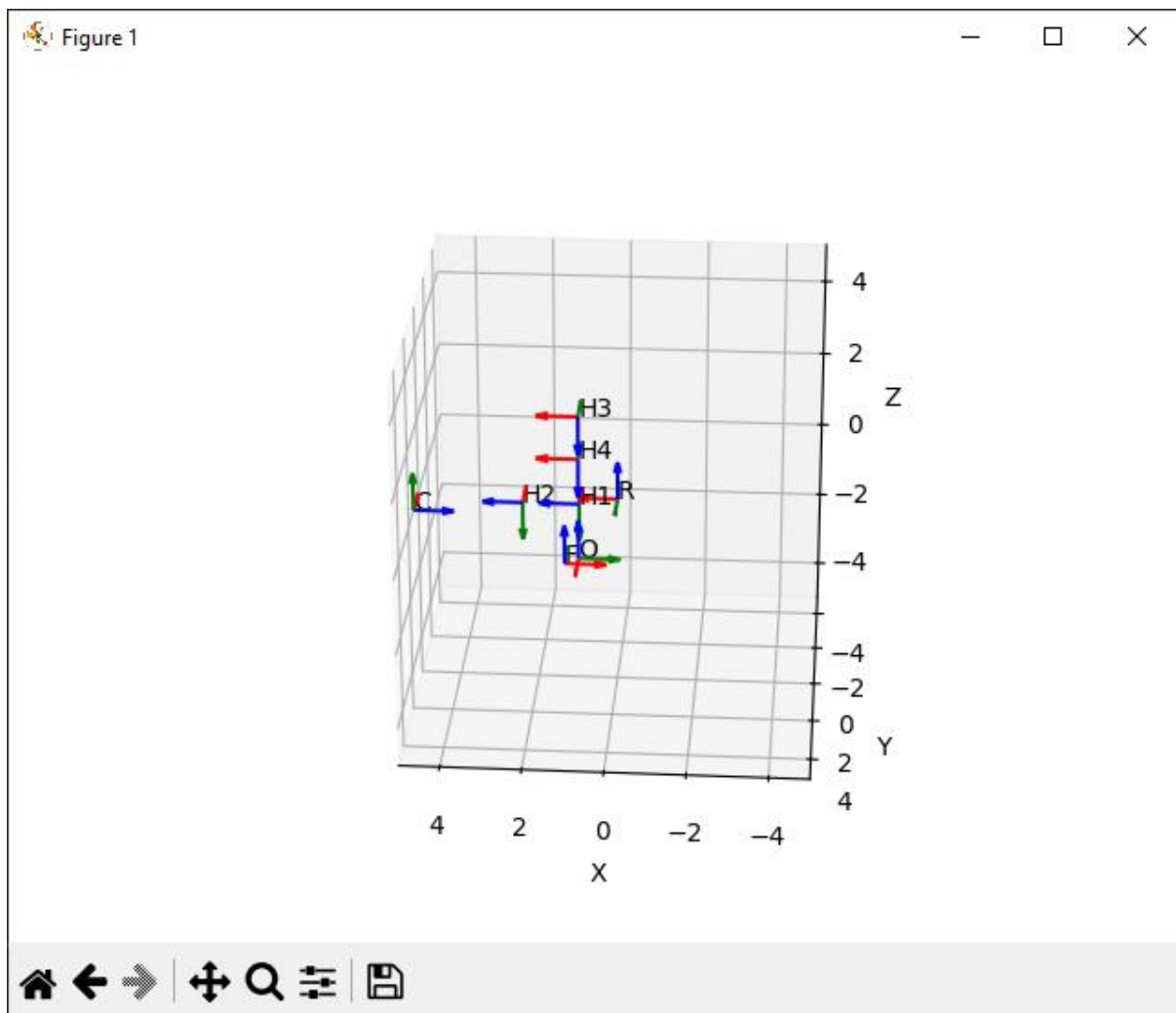
Λύση άσκησης

Χειρόγραφη έκδοση: Αρχείο με τίτλο *handwritten_solution*

Ψηφιακή έκδοση : (Αρχείο *.pdf* που είναι ήδη στο *eclass*):

Αριθμητική επίλυση μέσω python

Αρχείο με τίτλο *task1.py*



Πλαίσια στο χώρο

Στο αρχείο που υπάρχουν οι κλασεις `RobotUtils()` και `RobotViz()` δεν υπάρχει νόημα το να είναι `aspect equal` και μετα να βρίσκουμε το `min` και `max` του `x y z` γιατί πολύ απλά αλλάζει η κλίμακα προβολής και επίσης δεν υπάρχει συνάρτηση να κάνουμε `scale` μόνο τα `vectors` των `frame` και όχι την μετατόπιση τους αρά καταφεύγουμε στο `scale down (/100)` για να χωράει στο `figure` επόμενος πρέπει στην `show_plots(self)` να συμπληρώσουμε αυτές τις γραμμές κωδικα

```
all_points = np.array([max_range[0],max_range[1],max_range[2]])
global_min = -np.max(all_points)
global_max = np.max(all_points)

ax.set_xlim(global_min, global_max)
ax.set_ylim(global_min, global_max)
ax.set_zlim(global_min, global_max)
```

Αριθμητική επίλυση μέσω MATLAB

```
rc1 = [[0 0 -1 500];[-1 0 0 100];[0 1 0 0];[0 0 0 1]];
ce1 = [[1 0 0 20];[0 -1 0 20];[0 0 -1 250];[0 0 0 1]];

het = [[1 0 0 0];[0 1 0 0];[0 0 1 -160];[0 0 0 1]];

rc2 = rc1;
ce2 = [[1 0 0 20];[0 -1 0 20];[0 0 -1 110];[0 0 0 1]];

rc1 = [[0 0 -1 500];[-1 0 0 100];[0 1 0 0];[0 0 0 1]];
ce3 = ce2;

ce4 = ce2;

rh3 = [[1 0 0 49];[0 1 0 349];[0 0 1 110];[0 0 0 1]];
rh4 = [[1 0 0 49];[0 1 0 349];[0 0 1 0];[0 0 0 1]];

a1 = rc1*ce1*het;
a2 = rc2*ce2*het;
a3 = rh3*ce3*het;
a4 = rh4*ce4*het;
```

Άσκηση 2

Ρομποτική Ι - ΕΛΜΕΠΑ Παλιό θέμα

Αρχείο με τίτλο task2.py

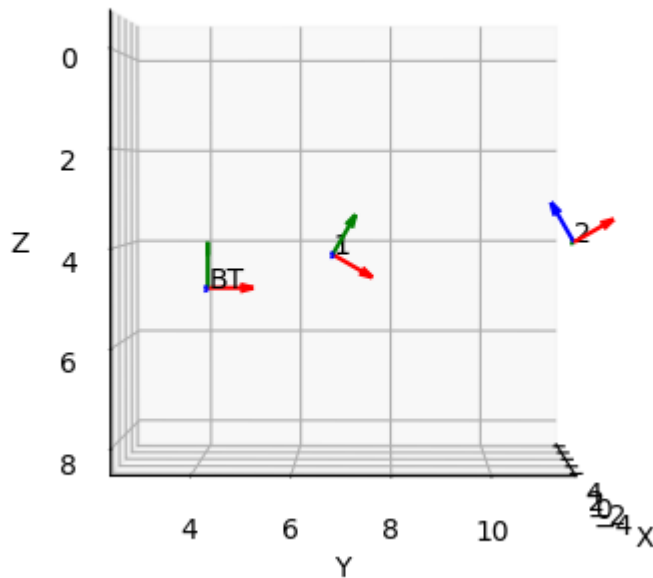


Figure zoomed in and rotated to show

Ξανά με την ίδια διόρθωση στο show_plots(self)

```
all_points = np.array([max_range[0],max_range[1],max_range[2]])
global_min = -np.max(all_points)
global_max = np.max(all_points)

ax.set_xlim(global_min, global_max)
ax.set_ylim(global_min, global_max)
ax.set_zlim(global_min, global_max)
```



```

import numpy as np
from csrl_robotics import RobotUtils, RobotViz

utils = RobotUtils()
viz = RobotViz()

theta1 = 30
theta2 = 30

t1x = 2
t1y = 2

t2x = 8
t2y = 1

theta_rad1 = np.radians(theta1)
theta_rad2 = np.radians(theta2)

gBT = np.array([[0, 0, -1, 3],
                [1, 0, 0, 4],
                [0, -1, 0, 5],
                [0, 0, 0, 1]])

tr1 = utils.gtransl(t1x, t1y, 0)
tr2 = utils.gtransl(t2x, t2y, 0)
gT1 = gBT @ tr1 @ utils.grotz(-theta_rad1)
gT2 = gBT @ tr2 @ utils.grotz(theta_rad2) @ utils.grotx(-np.pi/2)

viz.add_frame(gBT, "BT")
viz.add_frame(gT1, "1")
viz.add_frame(gT2, "2")

viz.show_plots()

```

οπού 3,4, 5 για σωστά αριθμητικά αποτελέσματα 300 400 500 στο gBT