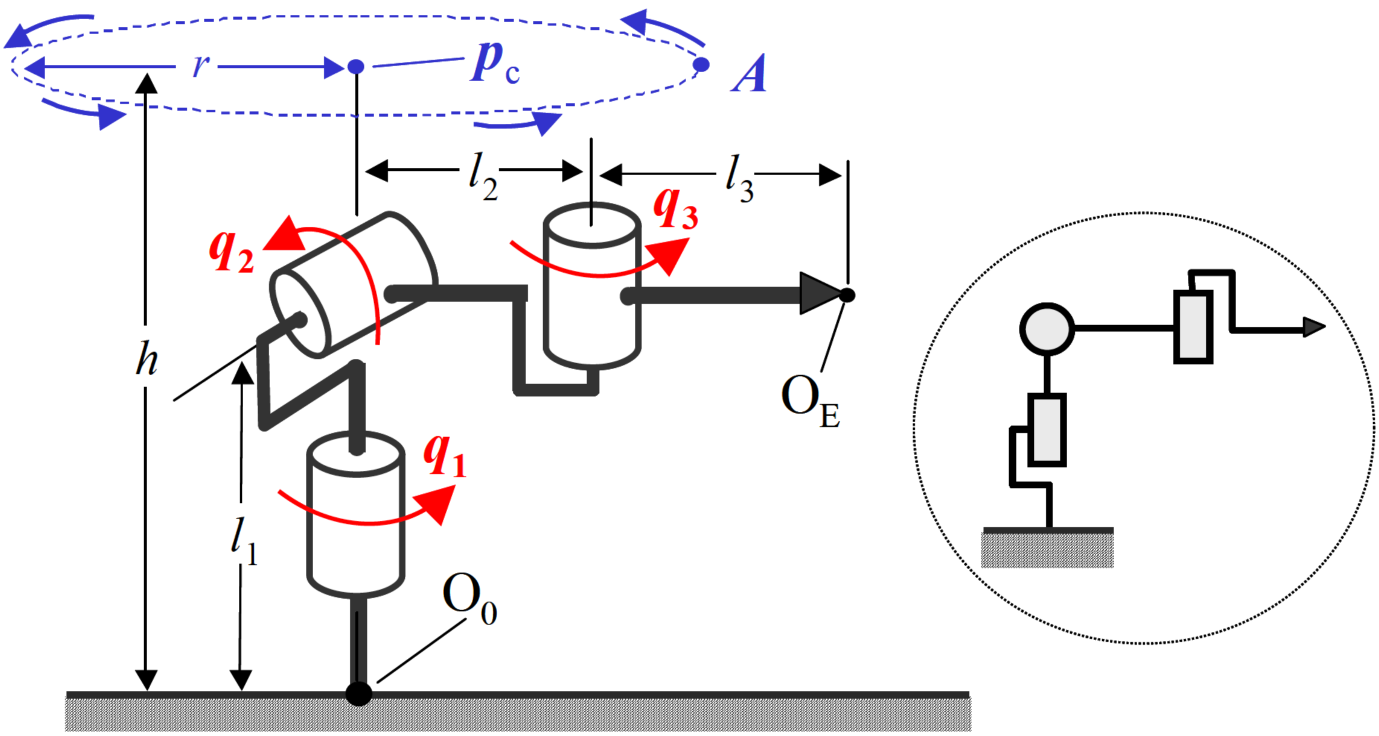
Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Μάθημα :
Ρομποτική Ι : Ανάλυση, Έλεγχος, Εργαστήριο

Εξαμηνιαία Εργασία : Robotic Manipulator with 3 rotational DOF

Ον/μο : Βαβουλιώτης Γεώργιος
Α.Μ. : 03112083
Εξάμηνο : 7ο

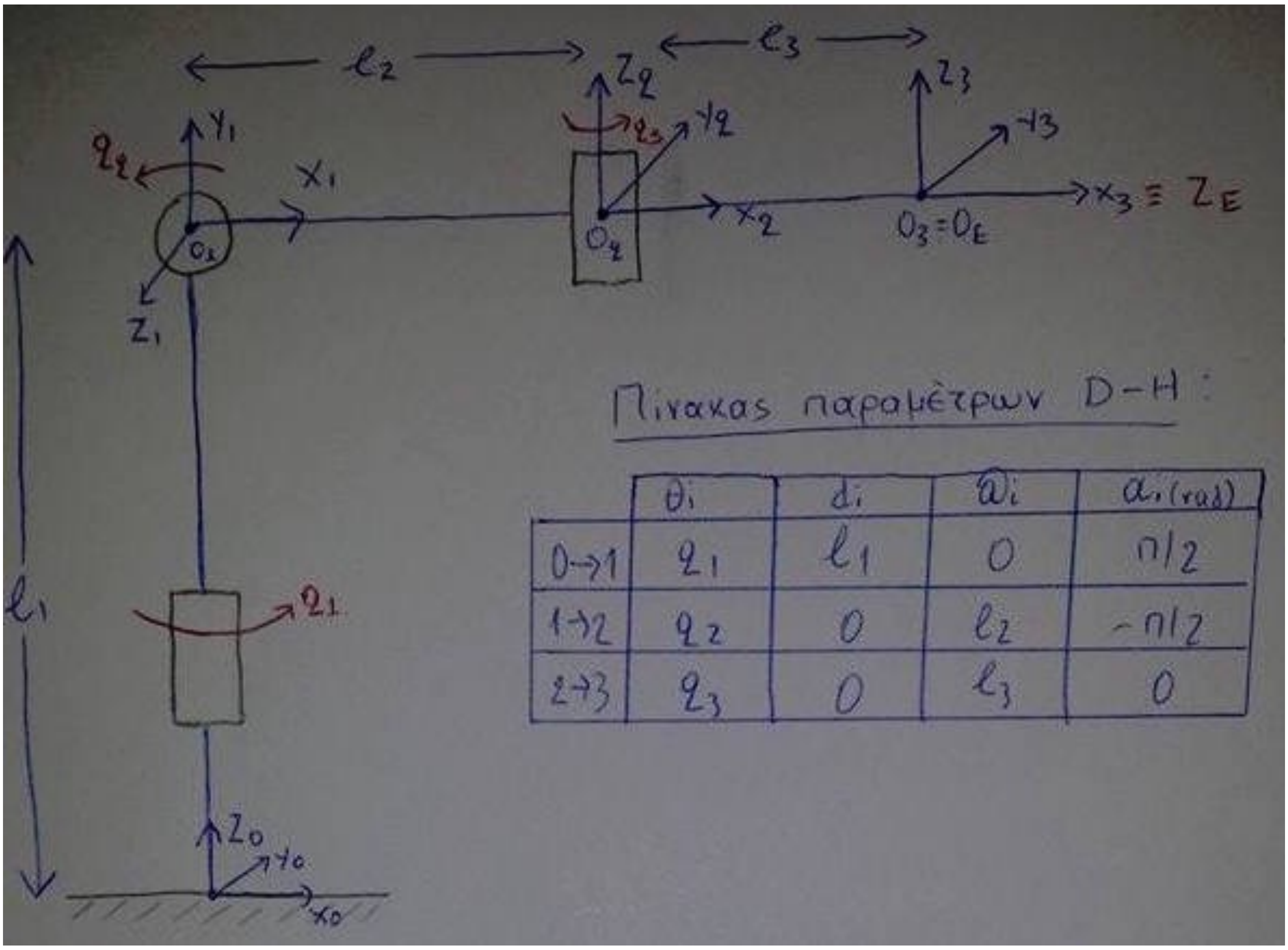


Ακαδημαικό έτος : 2015-2016

<u>Θεωρητική Ανάλυση</u>

1. Να υπολογιστεί ο πίνακας παραμέτρων Denavit-Hartenberg του ρομποτικού βραχίονα.

Στην παρακάτω εικόνα σας παραθέτω το σχήμα του ρομποτικού βραχίονα με σχεδιασμένους τους άξονες που επέλεξα για να βρω τις παραμέτρους D-H :



Ο πίνακας παραμέτρων D-H φαίνεται και στην παραπάνω φωτογραφία, ωστόσο σας τον παραθέτω και παρακάτω, επειδή μπορεί να μην είναι καλή η ανάλυση της εικόνας:

	θi	di	ai	αi
0->1	q1	l1	0	π/2
1->2	q2	0	l2	-π/2
2->3	q3	0	l3	0

Θα πρέπει να τονιστεί οτι για να εξάγουμε τα αποτελέσματα αυτά χρησιμοποιήσαμε και ενα βοηθητικό σύστημα, όπως μπορεί να παρατηρήσει κανείς απο την παραπάνω εικόνα άρα ουσιαστικά υπάρχει και μια περιστροφή ως προς τον άξονα y κατλα γωνία -π/2, η οποία προφανώς δεν αποτυπώνεται στον πίνακα παραμέτρων D-H, αλλά θα πρέπει να ληφθεί υπόψην στη συνέχεια.

2. Να γραφεί η κινηματική εξίσωση (ευθύ γεωμετρικό μοντέλο) του ρομπότ.

Απο τις διαφάνειες του μαθήματος πήρα τον εξής τύπο για να υπολογίσω τις μήτρες Denavit-Hartenberg :

$$A_i^{i-1} = A_{\Sigma_i}^{i-1} \cdot A_i^{\Sigma_i} = \left(\begin{array}{ccc|c} \cos\boldsymbol{\theta}_i & -\sin\boldsymbol{\theta}_i \cos\boldsymbol{\alpha}_i & \sin\boldsymbol{\theta}_i \sin\boldsymbol{\alpha}_i & \mathbf{a}_i \cos\boldsymbol{\theta}_i \\ \sin\boldsymbol{\theta}_i & \cos\boldsymbol{\theta}_i \cos\boldsymbol{\alpha}_i & -\cos\boldsymbol{\theta}_i \sin\boldsymbol{\alpha}_i & \mathbf{a}_i \sin\boldsymbol{\theta}_i \\ 0 & \sin\boldsymbol{\alpha}_i & \cos\boldsymbol{\alpha}_i & \mathbf{d}_i \\ \hline 0 & 0 & 0 & 1 \end{array}\right)$$

Με βάση την πραπάνω σχέση πήρα τα εξής αποτελέσματα, τα οποία έβαλα χειροκίνητα και στο Matlab(γι'αυτο και σας τα παρουσιάζω όπως τα έβαλα στο Matlab και όχι σε χειρόγραφη μορφή) :

A01 :

```
>> pretty(A01)
/ cos(q1), 0,  sin(q1),  0 \
|                          |
| sin(q1), 0, -cos(q1),  0 |
|                          |
|    0,    1,     0,    l1 |
|                          |
\    0,    0,     0,     1 /
```

A12:

```
>> pretty(A12)
/ cos(q2),  0, -sin(q2), l2 cos(q2) \
|                                   |
| sin(q2),  0,  cos(q2), l2 sin(q2) |
|                                   |
|    0,    -1,     0,         0     |
|                                   |
\    0,     0,     0,         1     /
```

A23:

```
>> pretty(A23)
/ cos(q3), -sin(q3), 0, l3 cos(q3) \
|                                  |
| sin(q3),  cos(q3), 0, l3 sin(q3) |
|                                  |
|    0,        0,    1,      0     |
|                                  |
\    0,        0,    0,      1     /
```

A3E:

```
>> pretty(A3E)
/ 0, 0, -1, 0 \
|             |
| 0, 1,  0, 0 |
|             |
| 1, 0,  0, 0 |
|             |
\ 0, 0,  0, 1 /
```

A0E:

```
>> pretty(A0E)
[[-cos(q1) sin(q2), - cos(q3) sin(q1) - cos(q1) cos(q2) sin(q3), sin(q1) sin(q3) - cos(q1) cos(q2) cos(q3),

  l2 cos(q1) cos(q2) - l3 sin(q1) sin(q3) + l3 cos(q1) cos(q2) cos(q3)],

  [-sin(q1) sin(q2), cos(q1) cos(q3) - cos(q2) sin(q1) sin(q3), - cos(q1) sin(q3) - cos(q2) cos(q3) sin(q1),

  l2 cos(q2) sin(q1) + l3 cos(q1) sin(q3) + l3 cos(q2) cos(q3) sin(q1)],

  [cos(q2), -sin(q2) sin(q3), -cos(q3) sin(q2), l1 + l2 sin(q2) + l3 cos(q3) sin(q2)],

  [0, 0, 0, 1]]
```

<u>Παρατηρήσεις</u> :

- Καθένας απο τους παραπάνω πίνακες είναι ενα screenshoot απο το Matlab και πάνω απο κάθε screenshoot υπάρχει μια εντολή pretty(), η οποία είναι μια συνάρτηση του Matlab η οποία μου δίνει την δυνατότητα να δω το αποτέμεσμα σε μια καλοσχηματισμένη μορφή που να χωράει στο command line.
- Ο πίνακας A3E ουσιαστικά είναι ο πίνακας περιστροφής ως προς τον άξονα y κατά γωνία -π/2 και υπολογίζεται απο τον τύπο:

$$\mathrm{Rot}(y,\boldsymbol{\theta}_y) = \begin{pmatrix} \cos\boldsymbol{\theta}_y & 0 & \sin\boldsymbol{\theta}_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\boldsymbol{\theta}_y & 0 & \cos\boldsymbol{\theta}_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

3. Να υπολογιστεί η Ιακωβιανή μήτρα για τυχαία διάταξη, και να γραφεί το ευθύ διαφορικό κινηματικό μοντέλο του ρομπότ.

Για να υπολογίσουμε την Ιακωβιανή μήτρα χρησιμοποιήσαμε τους παρακάτω τύπους απο τις διαφάνειες του μαθήματος :

$$\begin{bmatrix} \boldsymbol{J}_{L_i} \\ \boldsymbol{J}_{A_i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_{i-1} \times \boldsymbol{r}_{i-1,E} \\ \boldsymbol{b}_{i-1} \end{bmatrix} \quad \text{: για στροφική άρθρωση}$$

όπου $\boldsymbol{b}_{i-1}$: ο άξονας της άρθρωσης i

$\boldsymbol{r}_{i-1,E}$: διάνυσμα $O_{i-1} \rightarrow O_E$

$$\begin{bmatrix} \boldsymbol{J}_{L_i} \\ \boldsymbol{J}_{A_i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_{i-1} \\ \boldsymbol{0} \end{bmatrix} \quad \text{: για πρισματική άρθρωση}$$

$$\boldsymbol{b}_{i-1} = \mathbf{R}^0_{i-1}(q_1,...,q_{i-1}) \cdot \underline{\boldsymbol{b}}$$

όπου: $\underline{\boldsymbol{b}} = [0, 0, 1]^T$

(στη μεθοδολογία Denavit-Hartenberg)

$$\boldsymbol{r}_{i-1,E} = \mathbf{A}^0_n(q_1,...,q_n) \cdot \underline{\boldsymbol{r}} - \mathbf{A}^0_{i-1}(q_1,...,q_{i-1}) \cdot \underline{\boldsymbol{r}}$$

όπου: $\underline{\boldsymbol{r}} = [0, 0, 0, 1]^T$

Ωστόσο επειδή στην άσκηση αυτή έχουμε μόνο στροφικές αρθρώσεις χρησιμοποιήσαμε μόνο τον τύπο για τις στροφικές αρθρώσεις.
Επειδή οι πράξεις ήταν πολλές, προτίμησα να κάνω χρήση του Matlab για να κάνω τον υπολογισμό της Ιακωβιανής. Ωστόσο η Ιακωβιανή ήταν πολύ μεγάλη ώστε να μπορώ να την κάνω screenshot για να σας την παρουσιάσω. Για να το καταφέρω χρησιμοποίησα την συνάρτηση pretty() του Matlab και το αποτέλεσμα είναι το εξής :

```
>> pretty(Jacobian)
/                                                                                                         2    \
| - #4 - #2 - #1,           -cos(q1) #3,              - cos(q2) (#2 + #1) - l3 cos(q3) sin(q1) sin(q2)         |
|                                                                                                              |
|                                                                                     2                        |
|       #5,                  -sin(q1) #3,               l3 cos(q1) cos(q3) sin(q2)  - cos(q2) (#7 - #6)         |
|                                                                                                              |
|        0,      sin(q1) (#4 + #2 + #1) + cos(q1) #5, - cos(q1) sin(q2) (#2 + #1) - sin(q1) sin(q2) (#7 - #6)  |
|                                                                                                              |
|        0,                    sin(q1),                                        -cos(q1) sin(q2)                 |
|                                                                                                              |
|        0,                   -cos(q1),                                        -sin(q1) sin(q2)                 |
|                                                                                                              |
\        1,                       0,                                              cos(q2)                      /

where

   #1 == l3 cos(q2) cos(q3) sin(q1)

   #2 == l3 cos(q1) sin(q3)

   #3 == l2 sin(q2) + l3 cos(q3) sin(q2)

   #4 == l2 cos(q2) sin(q1)

   #5 == l2 cos(q1) cos(q2) - #7 + #6

   #6 == l3 cos(q1) cos(q2) cos(q3)

   #7 == l3 sin(q1) sin(q3)
```

<u>**Ευθύ Διαφορικό Κινηματικό Μοντέλο**</u>

Αφού υπολόγισα παραπάνω την Ιακωβιανή είναι πλέον πολύ εύκολο να διατυπώσω το ευθύ διαφορικό κινηματικό μοντέλο του ρομπότ. Θα χρησιμοποιήσω την παρακάτω σχέση η οποία υπάρχει στις διαφάνειες του μαθήματος :

$$\boxed{\dot{p} = J \cdot \dot{q}}$$

όπου q είναι το διάνυσμα των ταχυτήτων των αρθρώσεων.

Με την βοήθεια του Matlab μπορώ να υπολογίσω την παραπάνω σχέση. Το αποτέλεσμα που παίρνω είναι το εξής (κι εδώ έκανα χρήση της συνάρτησης pretty):

```
>> pretty(p1_derivative)
                                          2
- vvz (l3 cos(q3) sin(q1) sin(q2)  + cos(q2) (l3 cos(q1) sin(q3) + #1)) - vvx (l2 cos(q2) sin(q1) + l3 cos(q1) sin(q3) + #1)

   - vvy cos(q1) (l2 sin(q2) + l3 cos(q3) sin(q2))

where

   #1 == l3 cos(q2) cos(q3) sin(q1)
```

```
>> pretty(p2_derivative)
                                                                                                                     2
vvx (l2 cos(q1) cos(q2) - l3 sin(q1) sin(q3) + #1) - vvz (cos(q2) (l3 sin(q1) sin(q3) - #1) - l3 cos(q1) cos(q3) sin(q2) )

   - vvy sin(q1) (l2 sin(q2) + l3 cos(q3) sin(q2))

where

   #1 == l3 cos(q1) cos(q2) cos(q3)
```

```
>> pretty(p3_derivative)
vvy (sin(q1) (l2 cos(q2) sin(q1) + #4 + #1) + cos(q1) (l2 cos(q1) cos(q2) - #3 + #2)) - vvz (cos(q1) sin(q2) (#4 + #1) + sin(q1) sin(q2) (#3 - #2))

where

   #1 == l3 cos(q2) cos(q3) sin(q1)

   #2 == l3 cos(q1) cos(q2) cos(q3)

   #3 == l3 sin(q1) sin(q3)

   #4 == l3 cos(q1) sin(q3)
```

όπου :

```
vv =

 vvx
 vvy
 vvz
```

4. Να μελετηθεί το αντίστροφο γεωμετρικό, καθώς και το αντίστροφο διαφορικό κινηματικό μοντέλο του ρομπότ, και να προσδιορισθούν πιθανές ιδιόμορφες διατάξεις του συστήματος (singular configurations).

Αντίστροφο Κινηματικό Μοντέλο

Αρχικά προσπάθησα να επιλύσω το αντίστροφο γεωμετρικό μοντέλο με την βοήθεια του Matlab και με την βοήθεια τις γεωμετρίας αλλά δεν μπόρεσα να βγάλω αποτέλεσμα. Η μόνη λύση ήταν πλέον να λύσω αναλυτικά τις εξισώσεις ώστε να καταφέρω να εξάγω τα τις σχέσεις που μου δίνουν τα q1,q2,q3 συναρτήσει των pex,pey,pez. Συγκεκριμένα πήρα τις εξισώσεις που φαίνονται παρακάτω :

$$P_{ex} = \ell_2 \cos(q_1)\cos q_2 - \ell_3 \sin(q_1)\sin(q_3) + \ell_3 \cos(q_1)\cos(q_2)\cos(q_3) \quad (1)$$

$$P_{ey} = \ell_2 \sin(q_1)\cos q_2 + \ell_3 \cos(q_1)\sin(q_3) + \ell_3 \sin(q_1)\cos(q_2)\cos(q_3) \quad (2)$$

$$P_{ez} = \ell_1 + \ell_3 \sin(q_2)\cos(q_3) + \ell_2 \sin(q_2) \quad (3)$$

Στο εξής για λιγότερες πράξεις θα συμβολίζω το $\cos(q_i) = C_i$ και το $\sin q_i = S_i$ για $i=1,2,3$.

Βήμα 1: Πολλαπλασιάζω την (2) × s1 και την (1) × C_1 και παίρνω:

$$P_{ex} \cdot C_1 = \ell_2 C_1^2 C_2 - \ell_3 S_1 S_3 C_1 + \ell_3 C_1^2 C_2 C_3 \quad (4)$$

$$P_{ey} S_1 = \ell_2 S_1^2 C_2 + \ell_3 C_1 S_3 S_1 + \ell_3 S_1^2 C_2 C_3 \quad (5)$$

Προσθέτω κατά μέλη τις (4),(5):

$$P_{ex} C_1 + P_{ey} S_1 = \ell_2 C_2 + \ell_3 C_2 C_3 \quad (6)$$

Υψώνω στο τετράγωνο την σχέση (6):

$$(P_{ex} C_1 + P_{ey} S_1)^2 = (\ell_2 C_2 + \ell_3 C_2 C_3)^2 \Rightarrow (P_{ex} C_1)^2 + (P_{ey} S_1)^2 + 2 P_{ex} P_{ey} S_1 C_1 = (\ell_2 C_2 + \ell_3 C_2 C_3)^2 \quad (7)$$

Βήμα 2: Πολλαπλασιάζω την ② $\times c_1$ και την ① $\times s_1$ και παίρνω:

$$P_{ex} s_1 = l_2 c_1 c_2 s_1 - l_3 s_1^2 s_3 + l_3 c_1 c_2 c_3 s_1 \quad (8)$$

$$P_{ey} c_1 = l_2 c_1 c_2 s_1 + l_3 c_1^2 s_3 + l_3 c_1 c_2 c_3 s_1 \quad (9)$$

Αφαιρώ κατά μέλη τις ⑧, ⑨:

$$-P_{ex} s_1 + P_{ey} c_1 = l_3 s_3 \quad (10)$$

Υψώνω στο τετράγωνο την σχέση ⑩:

$$(P_{ex} s_1)^2 + (P_{ey} c_1)^2 - 2 P_{ex} P_{ey} s_1 c_1 = (l_3 s_3)^2 \quad (11)$$

Βήμα 3: Από τις σχέσεις ⑦ και ⑪ παίρνω:

$$P_{ex}^2 + P_{ey}^2 = (l_2 c_2 + l_3 c_2 c_3)^2 + (l_3 s_3)^2 \Rightarrow$$

$$P_{ex}^2 + P_{ey}^2 = (l_2 c_2)^2 + (l_3 c_2 c_3)^2 + 2 l_2 l_3 c_2^2 c_3 + (l_3 s_3)^2 \quad (12)$$

Βήμα 4: Από τη σχέση ③ παίρνω το εξής:

$$P_{ez} - l_1 = l_2 s_2 + l_3 s_2 c_3 \Rightarrow$$

$$(P_{ez} - l_1)^2 = (l_2 s_2 + l_3 s_2 c_3)^2 \Rightarrow (P_{ez} - l_1)^2 = l_2^2 s_2^2 + l_3^2 s_2^2 c_3^2 + 2 l_2 l_3 s_2^2 c_3 \quad (13)$$

Προσθέτω κατά μέλη τις σχέσεις ⑫, ⑬:

$$P_{ex}^2 + P_{ey}^2 + (P_{ez} - l_1)^2 = l_2^2 + l_3^2 + 2 l_2 l_3 c_3 \Rightarrow$$

$$c_3 = \frac{P_{ex}^2 + P_{ey}^2 + (P_{ez} - l_1)^2 - l_2^2 - l_3^2}{2 l_2 l_3} \Rightarrow$$

$$\boxed{q_3 = \begin{cases} \arccos\left(\dfrac{P_{ex}^2 + P_{ey}^2 + (P_{ez} - l_1)^2 - l_2^2 - l_3^2}{2 l_2 l_3}\right) \\[2ex] -\arccos\left(\dfrac{P_{ex}^2 + P_{ey}^2 + (P_{ez} - l_1)^2 - l_2^2 - l_3^2}{2 l_2 l_3}\right) \end{cases}}$$

Βήμα 5 : Απο την σχέση ③ (με γνωστό πλέον το c_3) βρίσκω το q_2:

$$P_{ez} = l_1 + l_2 s_2 + l_3 s_2 c_3 \Rightarrow s_2 = \frac{P_{ez}-l_1}{l_2+l_3c_3} \Rightarrow$$

$$\boxed{q_2 = \begin{cases} \arcsin\left(\dfrac{P_{ez}-l_1}{l_2+l_3c_3}\right) \\ -\arcsin\left(\dfrac{P_{ez}-l_1}{l_2+l_3c_3}\right) \end{cases} \quad \text{όπου } c_3 = \frac{P_{ex}^2+P_{ey}^2+(P_{ez}-l_1)^2-l_2^2-l_3^2}{2l_2l_3}}$$

Βήμα 6 : Διαιρώ κατά μέλη τις σχέσεις ⑥ και ⑩:

$$\frac{P_{ex}c_1+P_{ey}s_1}{P_{ey}c_1-P_{ex}s_1} = \frac{l_2c_2+l_3c_2c_3}{l_3s_3} \xRightarrow{\text{διαιρώ με } c_1}$$

$$\frac{P_{ex}+P_{ey}\tan(q_1)}{P_{ey}-P_{ex}\tan(q_1)} = \frac{l_2c_2+l_3c_2c_3}{l_3s_3} \Rightarrow$$

$$l_3s_3P_{ex} + l_3s_3P_{ey}\tan(q_1) = l_2c_2P_{ey} + l_3c_2c_3P_{ey} - l_2c_2P_{ex}\tan(q_1) - P_{ex}\tan(q_1)\,l_3c_2c_3$$

$$\Rightarrow$$

$$\tan(q_1) = \frac{P_{ey} - \dfrac{l_3s_3}{l_2c_2+l_3c_2c_3}P_{ex}}{P_{ex} + \dfrac{l_3s_3}{l_2c_2+l_3c_2c_3}\cdot P_{ey}} \Rightarrow$$

$$\boxed{q_1 = \begin{cases} \arctan\left(\dfrac{P_{ey}-X\cdot P_{ex}}{P_{ex}+X P_{ey}}\right) \\ \pi + \arctan\left(\dfrac{P_{ey}-X P_{ex}}{P_{ex}+X P_{ey}}\right) \end{cases} \qquad \text{όπου } X = \frac{l_3s_3}{l_2c_2+l_3c_2c_3}}$$

Αντίστροφο Διαφορικό Κινηματικό Μοντέλο

Για να έχει νόημα να μελετήσουμε το Αντίστροφο Διαφορικό Κινηματικό Μοντέλο θα πρέπει να μην βρισκόμαστε με ιδιόμορφη κατάσταση, δηλαδή θα πρέπει να ισχύει απο μαθηματικής άποψης οτι det(Jacobian) != 0, δηλαδή η Ιακωβιανή θα πρέπει να είναι αντιστρέψιμη. Τότε καλόυμαι να λύσω την εξίσωση :

$$\dot{q} = J^{-1}(q) \cdot \dot{p}$$

όπου q είναι ταχύτητες αρθρώσεων για να επιτύχουμε επιθυμητή ταχύτητα p τελικού στοιχείου δράσης.

Κάνοντας χρήση του Matlab έχω οτι ο υπολογισμός της αντίστροφης Ιακωβιανής μήτρας ανάγεται σε μια εντολή, άρα με χρήση της συνάρτησης inv() που υπάρχει στο Matlab υπολογίζω την αντίστροφη Ιακωβιανή μήτρα και εφαρμόζω την παραπάνω εξίσωση. Το αποτέλεσμα φαίνεται παρακάτω (κι εδώ έκανα χρήση της συνάρτησης pretty):

```
>> pretty(q1_derivative)
lvy (cos(q1) sin(q3) + cos(q2) cos(q3) sin(q1))   lvx (sin(q1) sin(q3) - cos(q1) cos(q2) cos(q3))   lvz cos(q3) sin(q2
----------------------------------------------- - ----------------------------------------------- + ------------------
                      #1                                                #1                                 #1

where

                                        2                            2
   #1 == l2 cos(q2) sin(q3) cos(q1)  + l2 cos(q2) sin(q3) sin(q1)
```

```
>> pretty(q2_derivative)
               2                     2
lvz (l2 cos(q2)  - l3 cos(q3) sin(q2) )   lvx sin(q2) (l2 cos(q1) cos(q2) - l3 sin(q1) sin(q3) + l3 cos(q1) cos(q2) cos(q3))
--------------------------------------- - ----------------------------------------------------------------------------------
                  #1                                                          #1

     lvy sin(q2) (l2 cos(q2) sin(q1) + l3 cos(q1) sin(q3) + l3 cos(q2) cos(q3) sin(q1))
   - ----------------------------------------------------------------------------------
                                             #1

where

           2        2        3      2        2                2      2        3        2      2               2        2                               2        3
   #1 == l2  cos(q1)  cos(q2)  + l2  cos(q1)  cos(q2) sin(q2)  + l2  cos(q2)  sin(q1)  + l2  cos(q2) sin(q1)  sin(q2)  + l3 cos(q3) l2 cos(q1)  cos(q2)

                                2                 2                      3        2                                2        2
      + l3 cos(q3) l2 cos(q1)  cos(q2) sin(q2)  + l3 cos(q3) l2 cos(q2)  sin(q1)  + l3 cos(q3) l2 cos(q2) sin(q1)  sin(q2)
```

```
>> pretty(q3_derivative)
  lvz (l2 sin(q2) + l3 cos(q3) sin(q2))   lvx (l2 cos(q1) cos(q2) - l3 sin(q1) sin(q3) + l3 cos(q1) cos(q2) cos(q3))
- ------------------------------------- - ----------------------------------------------------------------------------
                   #1                                                       #1

     lvy (l2 cos(q2) sin(q1) + l3 cos(q1) sin(q3) + l3 cos(q2) cos(q3) sin(q1))
   - ----------------------------------------------------------------------------
                                        #1

where

                                   2        2                                  2        2                                  2        2                                  2        2
   #1 == l2 l3 sin(q3) cos(q1)  cos(q2)  + l2 l3 sin(q3) cos(q1)  sin(q2)  + l2 l3 sin(q3) cos(q2)  sin(q1)  + l2 l3 sin(q3) sin(q1)  sin(q2)
```

όπου : `lv =`

```
 lvx
 lvy
 lvz
```

Για πληρότητα σας παρουσιάζω και την αντίστροφη μήτρα της Ιακωβιανής όπως αυτή παρουσιάζεται απο το Matlab :

```
>> pretty(inv(Jacobian_1_3))
/   sin(q1) sin(q3) - cos(q1) cos(q2) cos(q3)  cos(q1) sin(q3) + cos(q2) cos(q3) sin(q1)          cos(q3) sin(q2)         \
| - ----------------------------------------, ----------------------------------------,          ---------------         |
|                      #3                                        #3                                     #3                |
|                                                                                                                         |
|                                                                                                 2                   2  |
|                  sin(q2) #5                                 sin(q2) #4                   l2 cos(q2)  - l3 cos(q3) sin(q2) |
|                - ----------,                              - ----------,                  -------------------------------- |
|                      #1                                         #1                                     #1                |
|                                                                                                                         |
|                      #5                                         #4                          l2 sin(q2) + l3 cos(q3) sin(q2) |
|                    - --,                                      - --,                      - ------------------------------ |
\                     #2                                         #2                                     #2                /

where

           2        2        3     2        2                2     2        3        2     2                2        2                        2        3
   #1 == l2  cos(q1)  cos(q2)  + l2  cos(q1)  cos(q2) sin(q2)  + l2  cos(q2)  sin(q1)  + l2  cos(q2) sin(q1)  sin(q2)  + l3 cos(q3) l2 cos(q1)  cos(q2)

                             2                2                        3        2                                2        2
      + l3 cos(q3) l2 cos(q1)  cos(q2) sin(q2)  + l3 cos(q3) l2 cos(q2)  sin(q1)  + l3 cos(q3) l2 cos(q2) sin(q1)  sin(q2)

                              2        2                        2        2                        2        2                        2        2
   #2 == l2 l3 sin(q3) cos(q1)  cos(q2)  + l2 l3 sin(q3) cos(q1)  sin(q2)  + l2 l3 sin(q3) cos(q2)  sin(q1)  + l2 l3 sin(q3) sin(q1)  sin(q2)

                                   2                             2
   #3 == l2 cos(q2) sin(q3) cos(q1)  + l2 cos(q2) sin(q3) sin(q1)

   #4 == l2 cos(q2) sin(q1) + l3 cos(q1) sin(q3) + l3 cos(q2) cos(q3) sin(q1)

   #5 == l2 cos(q1) cos(q2) - l3 sin(q1) sin(q3) + l3 cos(q1) cos(q2) cos(q3)
```

Προσδιορισμός Ιδιόμορφων Διατάξεων(Singular Configurations)

Για να προσδιορίσω τις ιδιόμορφες διατάξεις του συστήματος μου, αρκεί να λύσω την εξίσωση det(Jacobian) = 0 για καθένα απο τα q1,q2,q3. Απο το προηγούμενο ερώτημα έχω υπολογίσει την Ιακωβιανή και πλέον με την βοηθεία του Matlab υπολογίζω την παραπάνω εξίσωση. Για να κάνω το Matlab να λύσει την εξίσωση det(Jacobian) = 0 για καθεμία απο τις μεταβλητές q1,q2,q3 χρησιμοποιώ την έτοιμη συνάρτηση solve(), η οποία σαν πρώτο όρισμα παίρνει την εξίσωση που καλείται να λύσει και σαν δεύτερο όρισμα την μεταβλητή ως προς την οποία καλείται να λύσει την εξίσωση. Αρα γράφοντας τις εντολές

```
solve( det(Jacobian) == 0, q1);
solve( det(Jacobian) == 0, q2);
solve( det(Jacobian) == 0, q3);
```

παίρνω τα εξής αποτελέσματα απο το Matlab :

```
singular_q1 =

Empty sym: 0-by-1


singular_q2 =

pi/2


singular_q3 =

0
```

Το παραπάνω σημαίνει οτι η εξίσωση det(Jacobian) = 0 για την μεταβλητή q1 δεν δίνει λύση(δηλαδή δεν έχουμε κάποια ιδιόμορφη κατάσταση) ενώ για τις q2 και q3 δίνει σαν λύση τα π/2 και 0 αντίστοιχα.
Για πληρότητα σας παραθέτω και την ορίζουσα τις Ιακωβιανής όπως αυτή υπολογίστηκε απο το Matlab για να γίνει προφανές πως οι ιδιόμορφες διατάξεις είναι ορθά υπολογισμένες:

```
det_Jacobian =

-l2*l3*cos(q2)*sin(q3)*(l2 + l3*cos(q3))
```

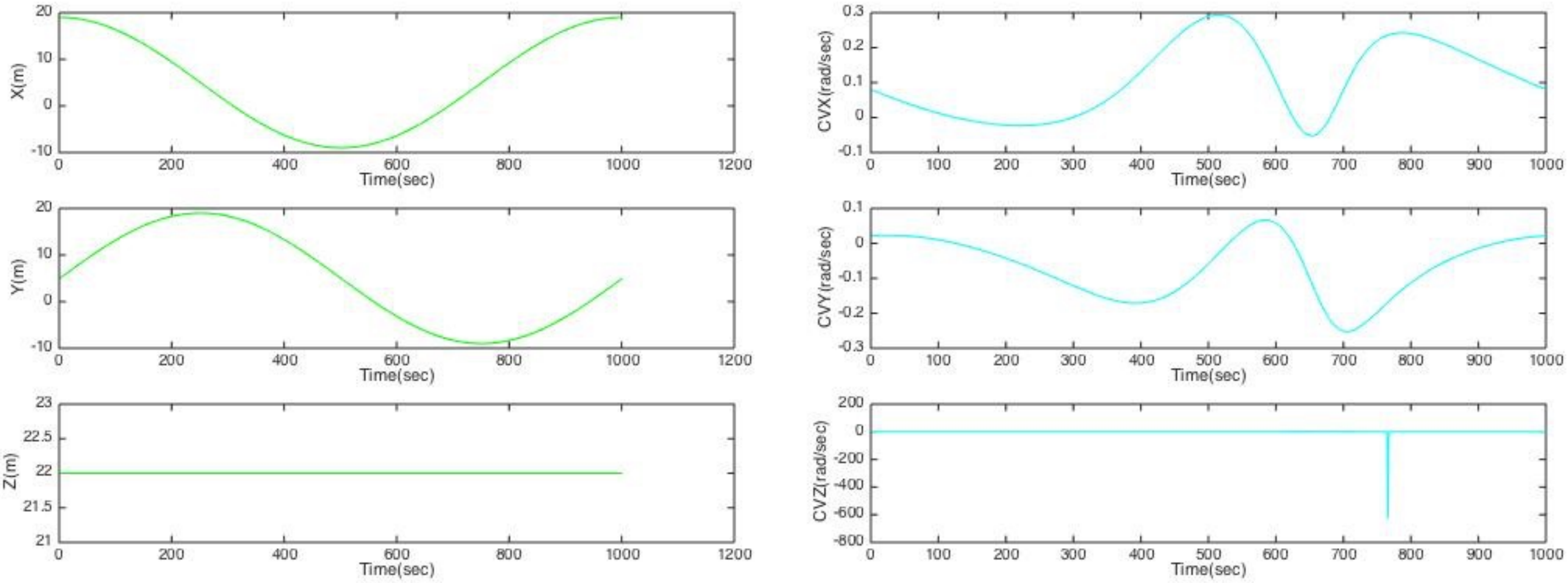
Γενική Παρατήρηση :
Όλα τα παραπάνω ερωτήματα επιλύθηκαν με την βοήθεια του Matlab όπως είπα αρκετές φορές παραπάνω. Το script που παρέχει τις λύσεις για τα παραπάνω ερωτήματα που χρησιμοποίησαν Matlab είναι το **chapter_A.m** και υπάρχει στο zip αρχείο που σας έστειλα.

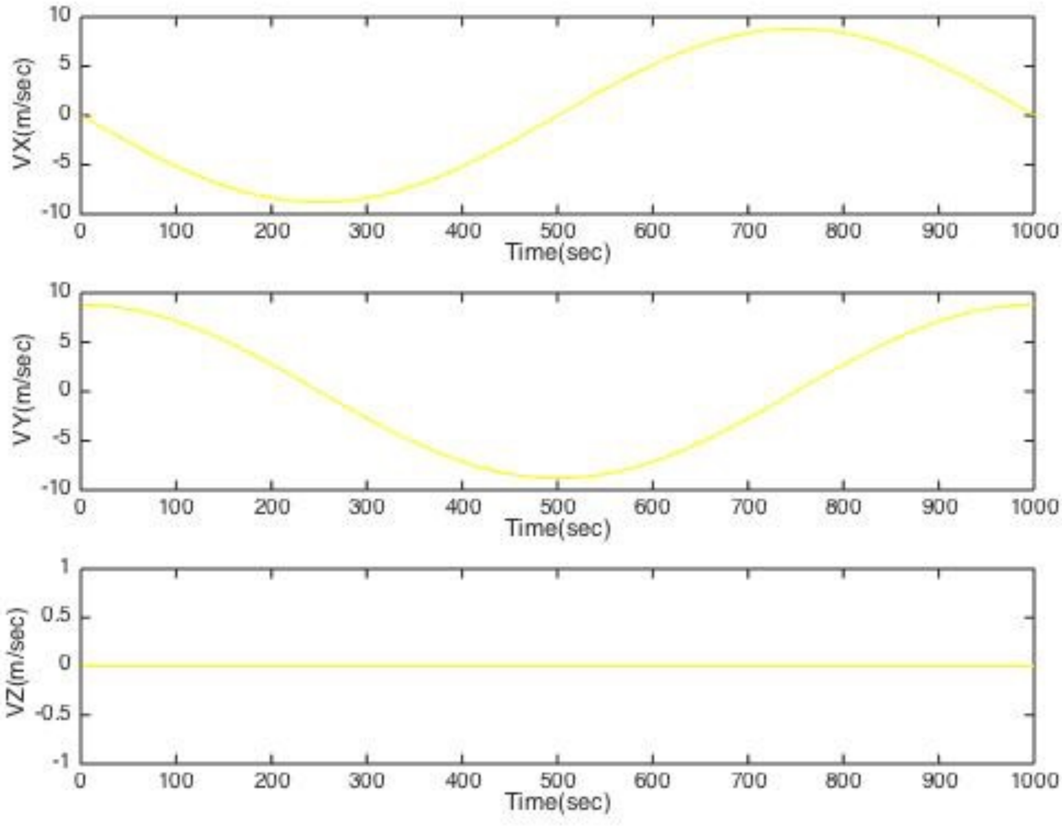
<u>Κινηματική Προσομοίωση</u>

5. Θεωρούμε ότι τη χρονική στιγμή t=0 το τελικό στοιχείο δράσης του ρομπότ βρίσκεται ήδησε δεδομένη αρχική θέση Α πάνω στον κυκλικό δρόμο, και ότι η επιθυμητή τροχιά τουτελικού στοιχείου δράσης πρέπει να διαγραφεί συνολικά εντός 10 secs. Επιθυμητή,επίσης, είναι η ομαλότητα της τροχιάς (χρονική συνέχεια και ως προς την ταχύτητα).Να εκτελεστεί κινηματική προσομοίωση του ρομποτικού χειριστή και να δοθούν οιγραφικές παραστάσεις στο χρόνο (plots) των ακολούθων μεγεθών, που επιτυγχάνουν τηνεκτέλεση της επιθυμητής ρομποτικής εργασίας:
(α) Το επιθυμητό προφίλ κίνησης του τελικού εργαλείου δράσης, δηλαδή: (1) η επιθυ-μητή θέση του άκρου (pEx, pEy, pEz) του ρομπότ σε κάθε χρονική στιγμή t, και (2) η γραμμική και γωνιακή ταχύτητα του εργαλείου δράσης.
(β) Οι γωνίες στροφής και οι γωνιακές ταχύτητες των αρθρώσεων, σε κάθε χρονική στιγμή t, κατά την εκτέλεση της εργασίας.
(γ) Ένα, τουλάχιστον, διάγραμμα κίνησης που θα εικονίζει μια χρονική ακολουθία ενδιάμεσων διατάξεων της ρομποτικής κινηματικής αλυσίδας κατά την εκτέλεση της εργασίας (από το animation της κίνησης).

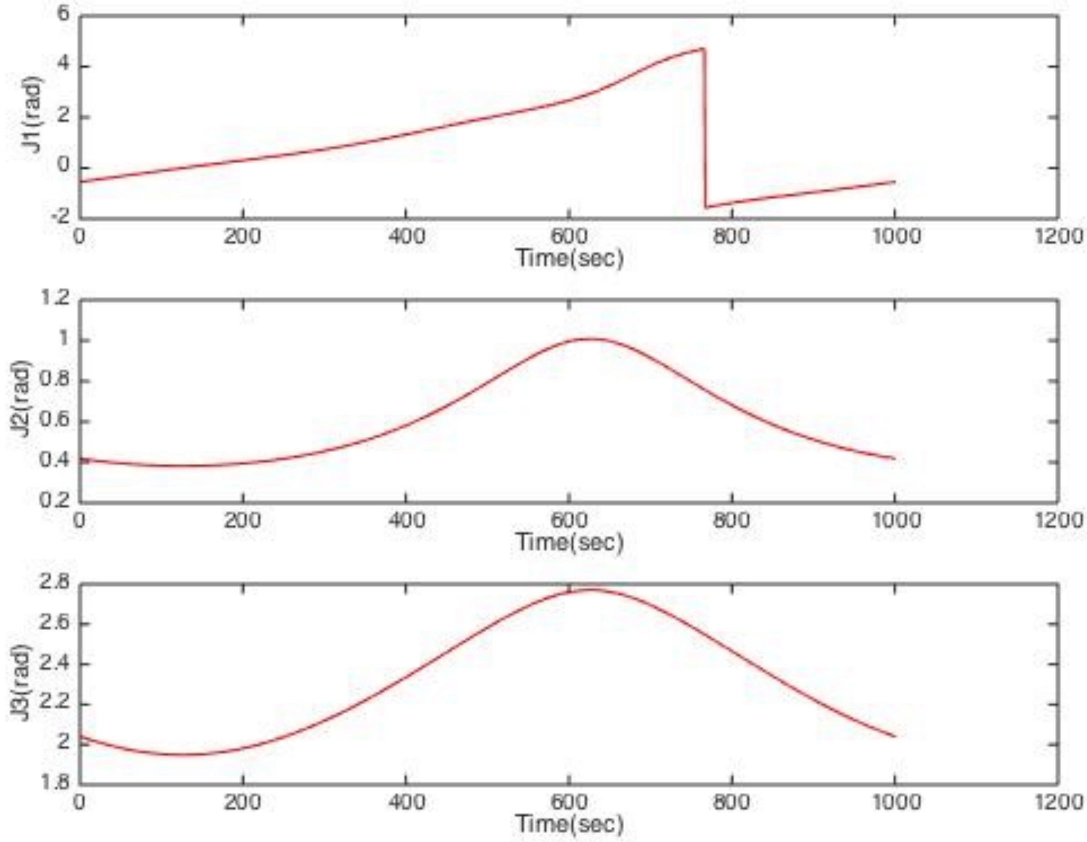
Θα πρέπει να κάνω τις εξής παρατηρήσεις :
- Αρχικά, εφόσον είχα ελευθερία να ορίσω όλα τα γεωμετρικά στοιχεία του προβλήματος πρέπει να σας τα παρουσιάσω : Τα μήκη των συνδέσμων ορίστηκαν 16, 22 και 16 αντίστοιχα, η ακτίνα που διαγράφει το ρομπότ μας είναι 14 και το κέντρο του είναι στην θέση (5,5). Η διάρκεια της κίνησης του ρομπότ ορίστηκε να είναι 10 sec, αφού κάτι τέτοιο ζητούσε η άσκηση.
- Για να καταφέρω να κάνω τις συνδέσεις του ρομπότ, και να μπορέσω να έχω οπτική επαφή με αυτό χρησιμοποίησα ενα toolbox το οποίο βρήκα στο internet, το οποίο περιείχε κάποιες συναρτήσεις που μου ήταν πολύ χρήσιμες. Το toolbox αυτό σας δίνετε στο zip αρχείο που σας υποβάλλω και έχει όνομα *my_robotic_toolbox.*
- Μια τελευταία παρατήρηση είναι οτι αν τρέξει κανείς τον κώδικα της προσομοίωσης (είναι το αρχείο Robot_Run.m που σας δίνω στο zip αρχείο) και χρονομετρήσει την κίνηση του ρομπότ μέχρι αυτό να τελειώσει τον κύκλο αντιάμβάνεται σχετικά εύκολα πως ο χρόνος που απαιτείται είναι μεγαλύτερος απο 10 sec. Κάτι τέτοιο είναι απόλυτα λογικό αφού απο το ενα καρέ στο άλλο μεσολαβεί κάποιος χρόνος υπολογισμού και εμφάνισης του επόμενου καρέ. Το γεγονός αυτό κάνει την προσομοίωση να είναι πιο αργή απο την πραγματικότητα.
- Τέλος, τα δοσμένα script είναι οι εξής :
  1. **Robot_Run.m** : περιέχει όλα όσα ανέφερα παραπάνω καθώς και άλλες λεπτομέριες της προσομοίωσης.
  2. **LocDet.m** : κάνει κάποιους απαραίτητους υπολογισμούς, τους οποίους χρειάζεται η προσομοίωση.
  3. **plot_maker.m** : κάνει όλες τις γραφικές που ζητούνται στα ερωτήματα (α) και (β).

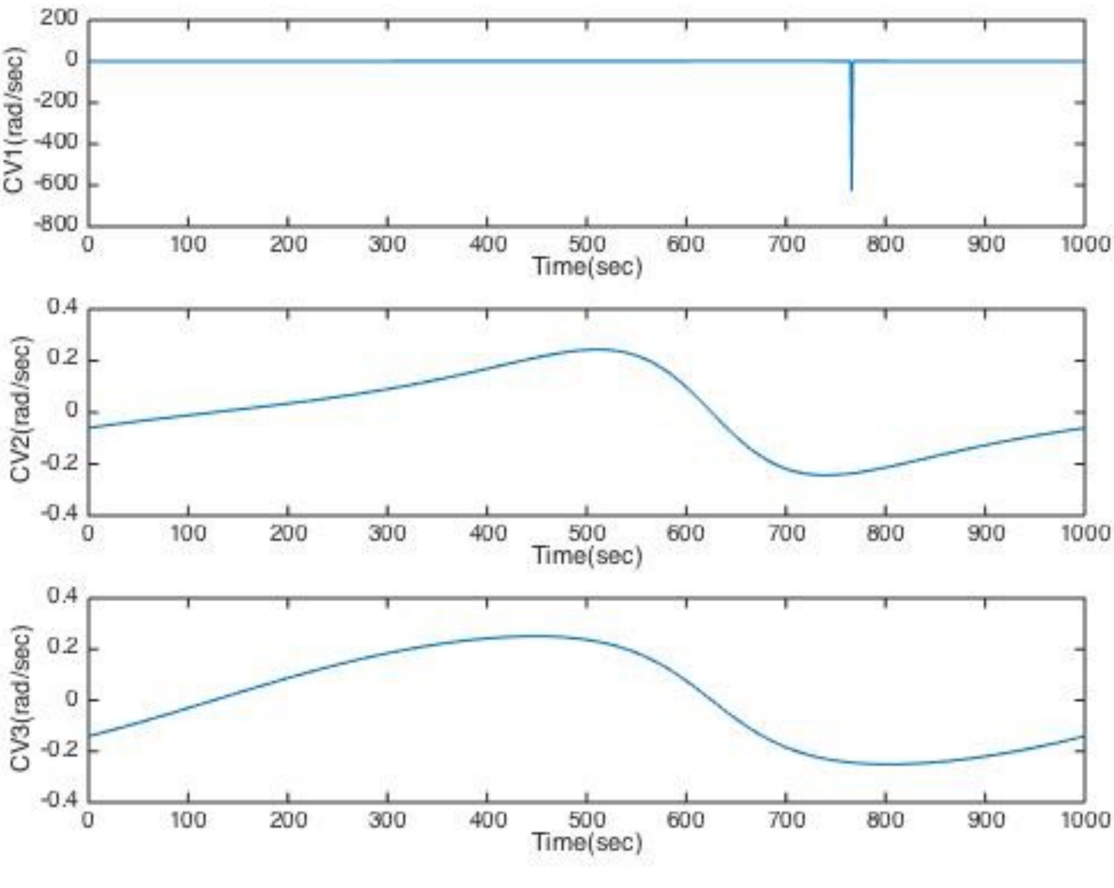
(α) Όλα τα ζητούμενα γραφήματα για το ερώτημα παρατίθενται παρακάτω:



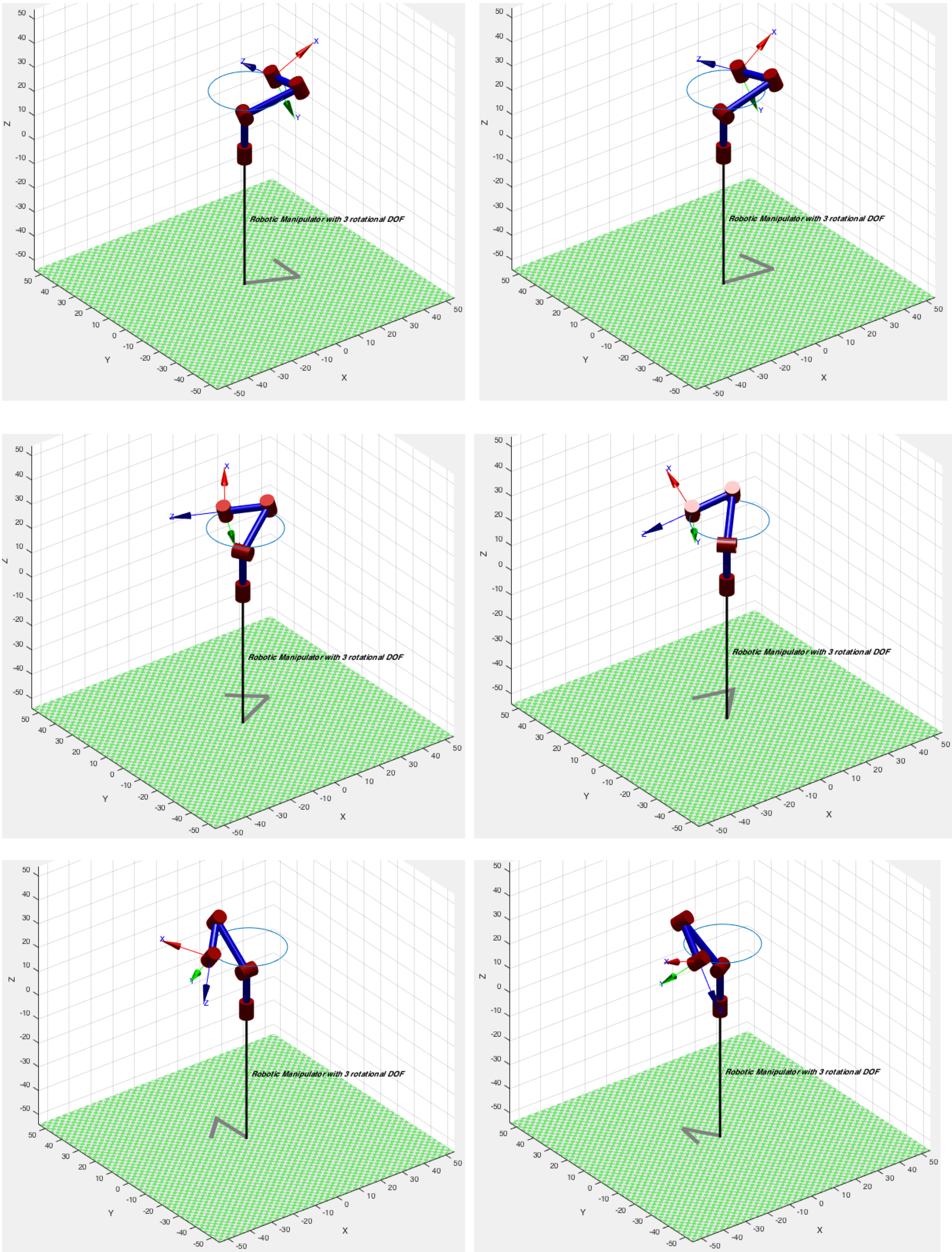


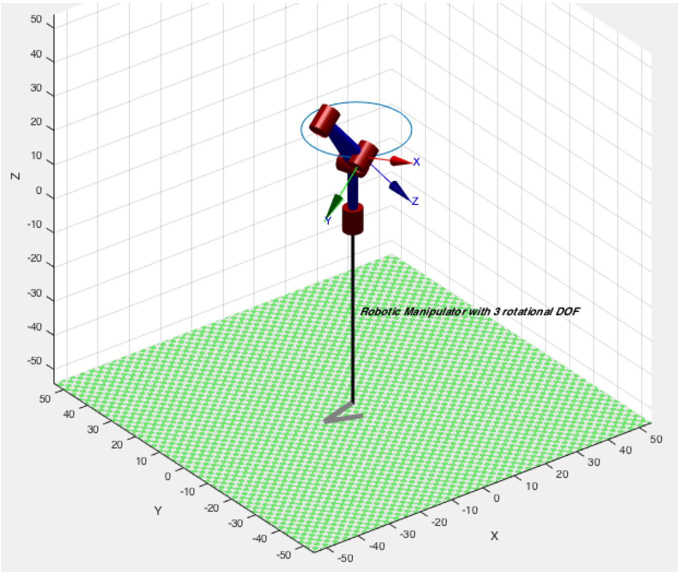
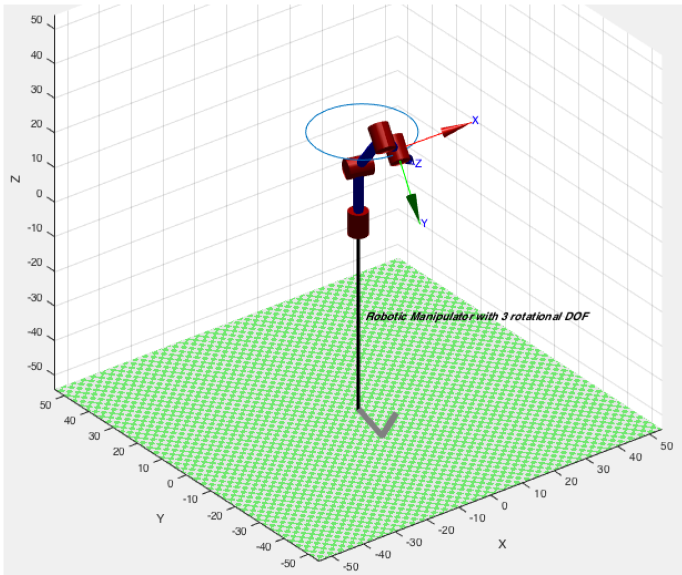
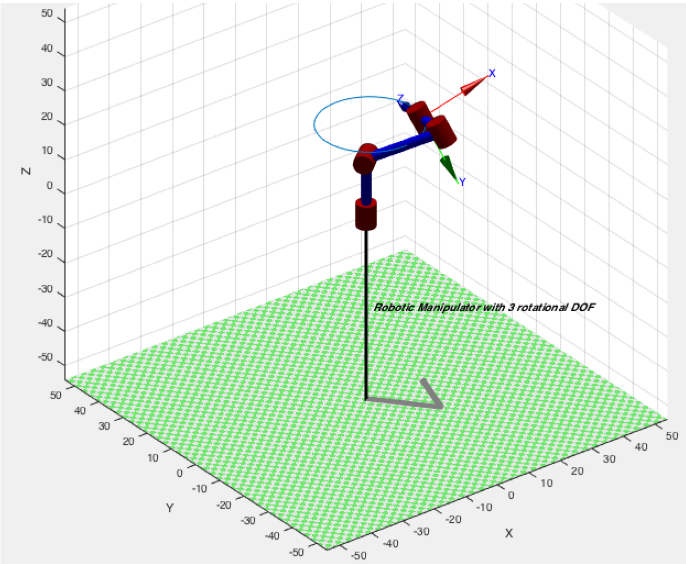
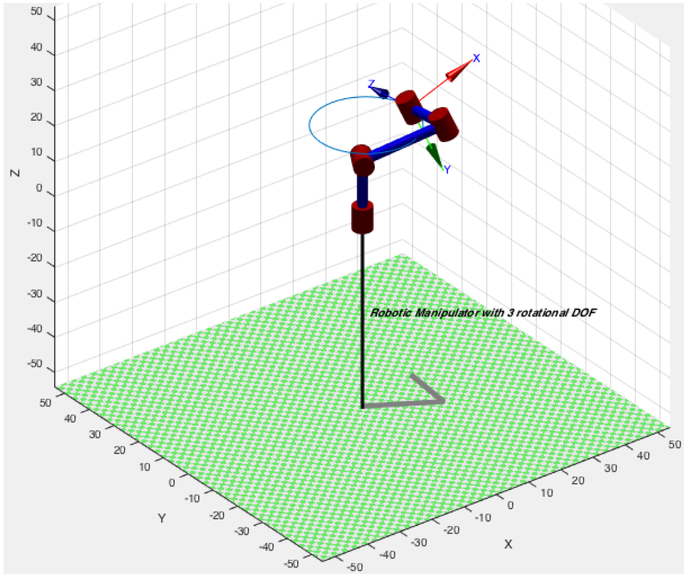
(β) Όλα τα ζητούμενα γραφήματα για το ερώτημα αυτό παρατίθενται παρακάτω:





(γ) Μια χρονική ακολουθία του animation της άσκησης φαίνεται παρακάτω :




Robotic Manipulator with 3 rotational DOF
X
Y
Z

Robotic Manipulator with 3 rotational DOF
X
Y
Z

Robotic Manipulator with 3 rotational DOF
X
Y
Z

Robotic Manipulator with 3 rotational DOF
X
Y
Z