



ΔΙΕΘΝΕΣ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΤΗΣ ΕΛΛΑΔΟΣ

Σχολή Μηχανικών

Τμήμα : “Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων”

---

**“ΜΕΛΕΤΗ ΚΑΙ ΠΡΟΣΟΜΟΙΩΣΗ ΡΟΜΠΟΤΙΚΟΥ  
ΒΡΑΧΙΩΝΑ 5 ΒΑΘΜΩΝ ΕΛΕΥΘΕΡΙΑΣ”**

=====

**“STUDY AND SIMULATION OF A ROBOTIC ARM OF 5  
DEEGRES OF FREDOM”**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΤΟΥ**

**ΣΩΤΗΡΙΟΥ ΔΟΚΑ ΚΑΣ 514035**

**ΚΩΔΙΚΟΣ ΠΤΥΧΙΑΚΗΣ 18176Μ**

Επιβλέπων: κος Χρήστος Κ. Μανάβης, Καθηγητής Εφαρμογών

**ΘΕΣΣΑΛΟΝΙΚΗ, ΙΟΥΝΙΟΣ 2020**



## ΠΕΡΙΛΗΨΗ

Ο έλεγχος ενός ρομποτικού βραχίονα ήταν μια τεχνολογική πρόκληση από τα πρώτα χρόνια της τεχνολογικής εξέλιξης και μάλιστα της εξέλιξης των ρομπότ. Το πρόβλημα της κινηματικής ορίζεται ως μετασχηματισμός από τον καρτεσιανό χώρο στον κοινό χώρο και αντίστροφα. Η εργασία αυτή στοχεύει στη μοντελοποίηση του ορθού κινηματικού προβλήματος του ρομποτικού βραχίονα 5 βαθμών ελευθερίας. Θα γίνει μία προσέγγιση της προς τα εμπρός μήτρας με τη χρήση του αλγορίθμου Denavit - Hartenberg. Μια αναλυτική λύση για την εμπρόσθια κινηματική της ρομποτικής 5 βαθμών ελευθερίας που παρουσιάζεται, για να αναλύσει την κίνηση του βραχίονα από ένα σημείο σε άλλο σημείο στο χώρο. Στην παρούσα πτυχιακή εργασία με τη βοήθεια του αλγορίθμου Denavit – Hartenberg θα προσδιορίσουμε τον πίνακα μετασχηματισμού ενός ρομποτικού βραχίονα 5 βαθμών ελευθερίας. Στη συνέχεια θα μελετήσουμε μεγέθη όπως είναι η θέση του βραχίονα, η ταχύτητά του αλλά και η αδράνεια του.

## **ABSTRACT**

Controlling a robotic arm was a technological challenge from the early years of technological development and even the development of robots. The problem of kinematics is defined as a transformation from Cartesian space to common space and vice versa. This paper aims to model the forward kinematics of a 5 DOF Robotic Arm for simple place application. A general algorithm of Denavit - Hartenberg representation of forward matrix is obtained. An analytical solution for the forward kinematics of 5 DOF robotic arm presented, to analyze the movement of arm from one point in space to another point. In this thesis, using the Denavit - Hartenberg algorithm, we will identify the transformation panel of a 5 degree robotic arm. Then we will study sizes such as the arm's position, its speed and its inertia.

## ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ .....	1
Αντικείμενο εργασίας .....	1
Δομή εργασίας .....	1
ΚΕΦΑΛΑΙΟ 1	
ΡΟΜΠΟΤΙΚΟΙ ΒΡΑΧΙΟΝΕΣ .....	3
1.1 Ρομποτική .....	3
1.1.1 Ιστορική αναδρομή .....	3
1.1.2 Τομείς εφαρμογής .....	8
1.2 Χαρακτηριστικά των ρομπότ .....	9
1.3 Δομή των ρομπότ .....	11
1.4 Τύποι ρομποτικών βραχιόνων και χώρος εργασίας .....	12
1.3.1 Ρομπότ καρτεσιανών συντεταγμένων .....	13
1.3.2 Ρομπότ κυλινδρικών συντεταγμένων .....	14
1.3.3 Ρομπότ σφαιρικών συντεταγμένων .....	14
1.3.4 Αρθρωτό ρομπότ .....	14
1.3.5 Βαθμοί ελευθερίας .....	15
ΚΕΦΑΛΑΙΟ 2	
ΚΙΝΗΜΑΤΙΚΗ ΘΕΩΡΙΑ .....	16
2.1 Εισαγωγή στην κινηματική θεωρία .....	16
2.2 Ευθύ κινηματικό πρόβλημα .....	16
2.1.1 Αλγόριθμος Denavit – Hartenberg .....	17
2.3 Είδη αρθρώσεων .....	19
2.2.1 Σύνθετες αρθρώσεις .....	20
ΚΕΦΑΛΑΙΟ 3	
ΤΟ ΕΥΘΥ ΚΙΝΗΜΑΤΙΚΟ ΠΡΟΒΛΗΜΑ ΕΝΟΣ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ 5	
ΒΑΘΜΩΝ ΕΛΕΥΘΕΡΙΑΣ (5 DOF) .....	23

*Περιεχόμενα*

ΚΕΦΑΛΑΙΟ 4

ΡΟΜΠΟΤΙΚΟΣ ΒΡΑΧΙΟΝΑΣ 5 ΒΑΘΜΩΝ ΕΛΕΥΘΕΡΙΑΣ (5 DOF)  
ΠΡΟΣΟΜΟΙΩΣΗ GUI MATLAB ..... 39

ΚΕΦΑΛΑΙΟ 5

ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΣΟΜΟΙΩΣΗΣ ..... 49

ΣΥΜΠΕΡΑΣΜΑΤΑ ..... 54

ΠΑΡΑΡΤΗΜΑ Α

Ο κώδικας σε Matlab για την ενεργοποίηση του πλήκτρου Forward ..55

ΠΑΡΑΡΤΗΜΑ Β

Ο κώδικας σε Matlab για τη δημιουργία του GUI .....56

ΠΑΡΑΡΤΗΜΑ Γ

Ο κώδικας σε C για την προσομοίωση του ρομποτικού βραχίονα των 5 βαθμών  
ελευθερίας .....63

**ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ**

Πίνακας 2.1: Παράμετροι Denavit – Hardenberg ..... 19

Πίνακας 3.1: Παράμετροι DH για το ρομποτικό βραχίονα 5 – DOF του σχήματος 3.1.  
..... 23

**ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ**

Εικόνα 1.1: Το πρώτο βιομηχανικό ρομπότ από τη Unimation το 1961	.... 5
Εικόνα 1.2: Ο βραχίονας του Στάνφορντ	..... 6
Εικόνα 1.3: Ο Scheinman και ο βραχίονας PUMA	..... 7
Εικόνα 1.4: Βραχίονας KUKA Robotics του 2015	..... 8
Εικόνα 1.5: Τυπική μορφή βιομηχανικού ρομπότ (Cincinnati Milacron)	.... 10
Εικόνα 1.6: Η βασική δομή ενός βιομηχανικού ρομποτικού βραχίονα	.... 13
Εικόνα 1.7: Οι τέσσερις βασικοί δομικοί τύποι βιομηχανικού ρομπότ.	.... 15
Εικόνα 1.8: Οι 6 βαθμοί κινητικότητας και οι 6 βαθμοί ελευθερίας	.... 15
Σχήμα 2.1: Συντεταγμένες σε γενικού τύπου βραχίονα(wikipedia)	....17
Σχήμα 2.2: Είδη αρθρώσεων	..... 20
Σχήμα 2.3: Πρισματική άρθρωση	..... 21
Σχήμα 3.1: Η παράσταση των αξόνων των επιμέρους τμημάτων ενός ρομποτικού βραχίονα 5 βαθμών ελευθερίας (5 – DOF).....	23
Εικόνα 4.1: Η εντολή guide	..... 39
Εικόνα 4.2: Η επιλογή του GUI	..... 40
Εικόνα 4.3: Το παράθυρο δημιουργίας του GUI	..... 40
Εικόνα 4.4: Δημιουργία των πεδίων $\theta_1$	..... 41
Εικόνα 4.5: Δημιουργία των 5 πεδίων $\theta_i$	..... 41
Εικόνα 4.6: Η επιλογή για τη δημιουργία του πλήκτρου	..... 42
Εικόνα 4.7: Η δημιουργία του πλήκτρου	..... 42
Εικόνα 4.8: Το παράθυρο με το οποίο θα οριστούν οι λειτουργίες των κάθε πεδίων και πλήκτρων του GUI	..... 43
Εικόνα 4.9: Το πεδίο «String» καθαρίζεται	..... 44
Εικόνα 4.10: Το πεδίο «Tag» ορίζεται ως Theta_1	..... 44
Εικόνα 4.11: Το πεδίο «String» ορίζεται ως Forward	..... 45
Εικόνα 4.12: Το πεδίο «Tag» ορίζεται ως btn_Forward	..... 45

Εικόνα 4.13: Ορισμός των «String» και «Tag» των τριών πεδίων που ορίζουν τις συντεταγμένες του άρπαγα. ....	46
Εικόνα 4.14: Ορισμός ονομάτων στα πεδία που έχουν δημιουργηθεί.....	47
Εικόνα 4.15: Η μορφή του παραθύρου του GUI μετά τον ορισμό όλων των πεδίων.....	47
Εικόνα 4.16: Η δημιουργία των αξόνων του τρισδιάστατου γραφήματος. ....	48
Εικόνα 4.17: Η μορφή του GUI και με τους άξονες. ....	48
Εικόνα 4.18: Η ενεργοποίηση του πλήκτρου Forward μέσω Matlab κώδικα. ....	49
Εικόνα 4.19: Η έναρξη της προσομοίωσης. ....	49
Εικόνα 4.20: Η τελική εικόνα του GUI. ....	50
Εικόνα 5.1: Η τελική εικόνα του GUI. ....	51
Εικόνα 5.2: Ορισμός των τιμών των γωνιών $\theta_i$ ....	52
Εικόνα 5.3: Ενεργοποίηση της προσομοίωσης από το πλήκτρο Forward ....	52
Εικόνα 5.4: Ο ρομποτικός βραχίονας των 5 βαθμών ελευθερίας σε 3 διαστάσεις και οι συντεταγμένες θέσης του άρπαγα. ....	53



## ΕΙΣΑΓΩΓΗ

### Αντικείμενο εργασίας

Στην παρούσα πτυχιακή εργασία θα γίνει μία ιστορική αναδρομή στην εξέλιξη των ρομπότ και των ρομποτικών βραχιόνων, η οποία ξεκινάει από το 1961 όπου κατασκευάστηκε το πρώτο βιομηχανικό ρομπότ από τη Unimation μέχρι και τους πιο σύγχρονους ρομποτικούς βραχίονες, όπως για παράδειγμα είναι ο βραχίονας της KUKA Robotics. Στη συνέχεια θα γίνει μία αναλυτική επεξήγηση των εννοιών και των ορισμών που αφορούν τα ρομποτικά συστήματα. Το επόμενο στάδιο της μελέτης των ρομποτικών βραχιόνων είναι η μελέτη και ανάλυση των αλγορίθμων στους οποίους στηρίζονται οι μαθηματικές εξισώσεις, οι οποίες αποδεικνύουν την ορθότητα των κινήσεών τους.

Θα ασχοληθούμε επίσης με την μελέτη και την προσομοίωση ενός ρομποτικού συστήματος 5 βαθμών ελευθερίας (5 DOF). Θα αναφερθούμε στις συντεταγμένες της θέσης του άρπαγα, στις ταχύτητες των αρθρώσεων αλλά και στις αδράνειες που αναπτύσσονται σε αυτές. Η επιβεβαίωση της ορθότητας της παραπάνω μελέτης θα πραγματοποιηθεί με την χρήση της γλώσσας προγραμματισμού C, αλλά και με τη δημιουργία ενός GUI σε περιβάλλον Matlab, στην οποία προσομοιώνεται ο ρομποτικός βραχίονας των 5 βαθμών ελευθερίας που μελετήσαμε. Με το GUI παρουσιάζεται ο ρομποτικός βραχίονας σε ένα τρισδιάστατο σχήμα.

Στόχοι της εργασίας είναι:

- Η ενασχόληση μας με το τομέα της ρομποτικής
- Η ευκολότερη κατανόηση των βασικών εννοιών της ρομποτικής και
- Η επιβεβαίωση της ορθότητας της κινηματικής (ευθύ κινηματικό πρόβλημα) ενός ρομποτικού βραχίονα 5 βαθμών ελευθερίας.

Στην επόμενη υποενότητα θα παρουσιαστεί αναλυτικότερα η δομή της εργασίας ανά κεφάλαιο.

### 1.2 Δομή εργασίας

Το **κεφάλαιο 1** αποτελεί μια εισαγωγή στην εργασία όπου παρουσιάζεται το αντικείμενο μελέτης αλλά και η δομή της.

Στο **κεφάλαιο 2** παρουσιάζεται η απαραίτητη βιβλιογραφική έρευνα για την ιστορία και τη λειτουργία των ρομποτικών συστημάτων, καθώς και την ορολογία που χρησιμοποιείται για τη μελέτη τους.

Το **κεφάλαιο 3** αποτελεί μία εκτενή παρουσίαση της κινηματικής θεωρίας, του ευθέως κινηματικού προβλήματος αλλά και του αλγορίθμου Denavit–Hartenberg στον οποίο στηρίζεται η μαθηματική μελέτη της κίνησης ενός ρομποτικού βραχίονα.

Στο **κεφάλαιο 4** αναλύεται ένα παράδειγμα ενός ρομποτικού βραχίονα με 5 DOF (βαθμούς ελευθερίας). Στο κεφάλαιο αυτό παρουσιάζεται η αναλυτική μελέτη του ευθέως κινηματικού προβλήματος ενός ρομποτικού βραχίονα 5 βαθμών ελευθερίας κάνοντας χρήση του αλγορίθμου Denavit–Hartenberg. Υπολογίζονται οι συντεταγμένες της θέσης του άρπαγα, η περιστροφικές και μεταφορικές ταχύτητες των αρθρώσεων καθώς και η αδράνεια τους.

Στο **κεφάλαιο 5** παρουσιάζονται τα στάδια κατασκευής ενός GUI με τη χρήση της Matlab, με το οποίο γίνεται η προσομοίωση του ρομποτικού βραχίονα που αναπτύχθηκε στο κεφάλαιο 3.

Στο **κεφάλαιο 6** παρουσιάζονται τα αποτελέσματα τόσο της προσομοίωσης μέσω της Matlab, όσο και τα αποτελέσματα της προσομοίωσης μέσω της γλώσσας προγραμματισμού C.

Στο τέλος της παρούσας εργασίας υπάρχουν δύο **παραρτήματα** στα οποία παρουσιάζονται τμήματα των κωδικών που αναπτύχθηκαν για την προσομοίωση του ρομποτικού βραχίονα που αναπτύχθηκε στο κεφάλαιο 3. Ο ένας κώδικας αφορά το GUI της Matlab, ενώ ο άλλος είναι σε γλώσσα προγραμματισμού C.

## ΚΕΦΑΛΑΙΟ 1

### ΡΟΜΠΟΤΙΚΟΙ ΒΡΑΧΙΟΝΕΣ

#### 1.1 ΡΟΜΠΟΤΙΚΗ

Η Ρομποτική είναι ένας κλάδος της επιστήμης και της τεχνολογίας, που ασχολείται:

- με τη σύλληψη,
- τη σχεδίαση,
- την κατασκευή,
- τη θεωρία και
- τις εφαρμογές

των ρομπότ.

Τα ρομπότ είναι μηχανές, οι οποίες μπορούν να εκτελέσουν εργασίες που μπορούν να κάνουν οι άνθρωποι αλλά δεν έχουν ούτε τη μορφή ή ούτε τη συμπεριφορά του ανθρώπου. Η κατασκευή και η χρήση των ρομπότ, έχει ως σκοπό την αντικατάσταση του ανθρώπου στην εκτέλεση εργασιών, τόσο στο φυσικό επίπεδο, όσο και στο επίπεδο λήψης απόφασης.

Η εξέλιξη των ρομπότ έχει περάσει από πολλά στάδια. Πιο συγκεκριμένα, τα ρομπότ της πρώτης γενιάς δεν είχαν ούτε την ικανότητα υπολογισμού ούτε την ικανότητα της αίσθησης. Σε αντίθεση τα ρομπότ της δεύτερης γενιάς διέθεταν περιορισμένη υπολογιστική ισχύ, γλώσσες προγραμματισμού υψηλού επιπέδου και αισθητήρες ανατροφοδότησης. Στη συνέχεια τα ρομπότ της τρίτης γενιάς άρχισαν να διαθέτουν νοημοσύνη, δηλαδή έχουν πλέον την ικανότητα να παίρνουν αποφάσεις κατά τη διάρκεια εκτέλεσης της εργασίας τους. Οι ικανότητες αυτές ενεργοποιούνται μέσω τεχνικών της τεχνητής νοημοσύνης σε συνδυασμό με εξελιγμένες μορφές αισθητήρων αφής, δύναμης, απόστασης, όρασης, κ.ο.κ.. Τα πιο εξελιγμένα ρομπότ, τα βιομηχανικά ρομπότ, είναι εξελιγμένα συστήματα αυτοματισμού, τα οποία κάνουν χρήση ηλεκτρονικού υπολογιστή προκειμένου να επιτευχθεί ο έλεγχος τους. Σήμερα, οι υπολογιστές είναι το βασικότερο τμήμα του βιομηχανικού αυτοματισμού, οι οποίοι κατευθύνουν γραμμές παραγωγής και ελέγχουν συστήματα κατασκευής, όπως εργαλειομηχανές, συγκολλητές, κοπτικές διατάξεις Laser κ.α.. Τα πιο τεχνολογικά εξελιγμένα ρομπότ εκτελούν ποικίλες εργασίες στα βιομηχανικά συστήματα και γενικά μπορούν να συμμετέχουν στον πλήρη αυτοματισμό των εργοστασίων.

Η ευελιξία είναι το βασικότερο πλεονέκτημα των ρομπότ. Προσαρμόζονται σε διάφορα προϊόντα στην ίδια γραμμή παραγωγής, ανάλογα με τις απαιτήσεις των αλλαγών της αγοράς και επαναπρογραμματίζονται, ώστε να μπορούν να ικανοποιούν μικρές ή μεγάλες μεταβολές του παραγόμενου προϊόντος. Έτσι, προσφέρουν στη βιομηχανία μαζικής παραγωγής έναν τρόπο να αντιμετωπίζει τις μεταβολές της απαιτούμενης ποσότητας ή του τύπου του προς παραγωγή προϊόντος. Για να ορίσουμε το τι είναι ρομπότ θα βασιστούμε στον ορισμό που δίνει το Robot Institute της Αμερικής. Οπότε, ως ρομπότ μπορούμε να ορίσουμε ένα σχεδιασμένο μηχανισμό, ώστε μέσω προγραμματιζόμενων κινήσεων να μπορεί να μεταφέρει υλικά, τεμάχια, εργαλεία ή ειδικευμένες συσκευές με σκοπό την εκτέλεση ποικιλίας εργασιών. Ένας τέτοιος μηχανισμός περιλαμβάνει, συνήθως, τις ακόλουθες συνιστώσες:

- ένα μηχανολογικό υποσύστημα,
- ένα υποσύστημα αίσθησης και
- ένα σύστημα ελέγχου.

Αναλυτικότερα, το μηχανολογικό υποσύστημα επιτρέπει στο ρομπότ να εκτελεί την εργασία, που του έχει ανατεθεί. Το υποσύστημα αίσθησης βοηθάει το ρομπότ να συλλέγει πληροφορίες για την κατάσταση στην οποία βρίσκονται τόσο το ίδιο το ρομπότ, όσο και το περιβάλλον του. Εκτός των άλλων, δέχεται εξωτερικές εντολές, τις οποίες και επεξεργάζεται, τις μετατρέπει σε ηλεκτρική ισχύ, η οποία θα προσφερθεί στους κινητήρες του ρομπότ, καθώς, επίσης, παράγει σήματα εξόδου, που πληροφορούν για την κατάσταση του συστήματος. Το υποσύστημα αίσθησης περιλαμβάνει όργανα, όπως είναι όργανα μέτρησης, αισθητήρες, ηλεκτρονικά στοιχεία κ.λπ.. Επίσης συνήθως περιλαμβάνονται και μηχανισμοί, που επιτρέπουν στο ρομπότ να κινείται, όπως αρθρώσεις, σύστημα μετάδοσης κίνησης, επενεργητές – κινητήρες, οδηγούς, κ.λπ..

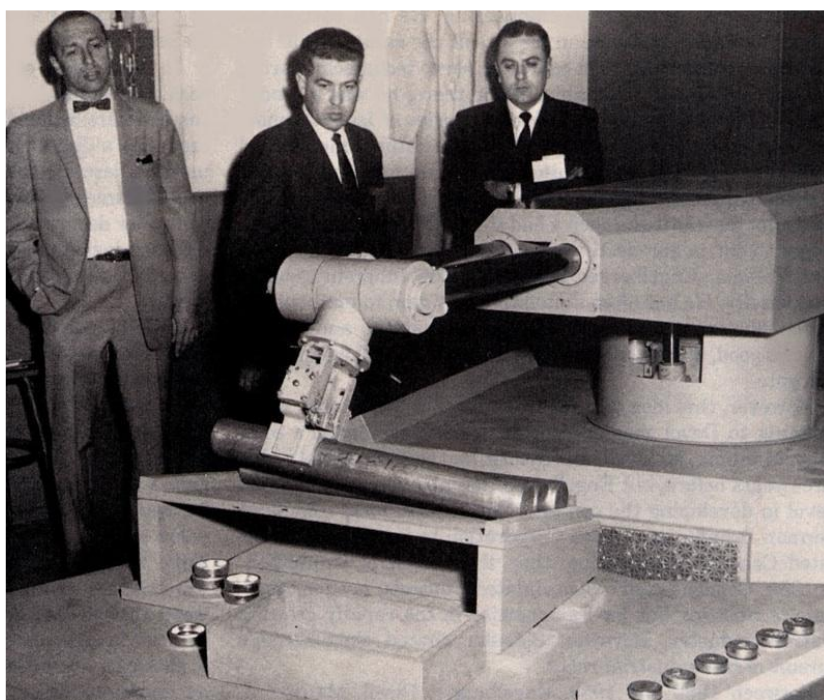
### 1.1.1 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

Τα βιομηχανικά ρομπότ αναπτύχθηκαν ταυτόχρονα με τον υπολογιστικό αριθμητικό έλεγχο (CNC). Το πρώτο ρομπότ κατασκευάστηκε το 1961, αλλά τα ρομπότ άρχισαν να παίζουν πρωτεύοντα ρόλο στη βιομηχανική παραγωγή γύρω στα τέλη της δεκαετίας του 1970.

Η λέξη «ρομπότ» χρησιμοποιήθηκε για πρώτη φορά πριν από έναν αιώνα. Ο Τσέχος δραματογράφος Karel Capek, χρησιμοποίησε για πρώτη φορά το 1921 τον όρο ρομπότ στο έργο του R.U.R.( "Rossum's Universal Robots"). Ο όρος αυτός επινοήθηκε από την τσέχικη λέξη «robot», η οποία σημαίνει καταναγκαστική εργασία. Στο έργο του περιγράφεται η κατασκευή έξυπνων συσκευών, οι οποίες χρησιμοποιούνται ως υπηρέτες του ανθρώπινου δημιουργού τους.

Κατά τη δεκαετία του 1940, ο Ρώσος Isaac Asimov, συγγραφέας επιστημονικής φαντασίας, αναφέρθηκε στο ρομπότ ως μια μηχανή με ανθρώπινη εμφάνιση χωρίς συναισθήματα. Η συμπεριφορά του καθοριζόταν από έναν εγκέφαλο, τον οποίο προγραμματίζαν με έναν ιδιαίτερο τρόπο άνθρωποι, ώστε να ακολουθεί βασικές ηθικές αρχές. Για πρώτη φορά, επίσης, χρησιμοποιήθηκε από τον Asimov ο όρος της «ρομποτικής» για να περιγράψει τον τομέα της επιστήμης, που ασχολείται με τα ρομπότ.

Ο όρος «βιομηχανικό ρομπότ» καθιερώθηκε το 1954 από τον G.C. Devol (ΗΠΑ), ο οποίος περιέγραψε πώς μπορεί να υλοποιηθεί ένα ελεγχόμενο μηχανικό χέρι, το οποίο μπορεί να βρει εφαρμογή στη βιομηχανία και να εκτελεί διάφορες εργασίες. Το 1961 η εταιρία Unimation κατασκεύασε και έθεσε σε λειτουργία το πρώτο βιομηχανικό ρομπότ. Έκτοτε, τέθηκαν σε λειτουργία χιλιάδες ρομπότ στην Αμερική, στην Ιαπωνία και στην Ευρώπη.



Εικόνα 1.1: Το πρώτο βιομηχανικό ρομπότ από τη Unimation το 1961

Στο Πανεπιστήμιο του Στάνφορντ, το 1969 ο Victor Scheinman δημιούργησε το "βραχίονα του Στάνφορντ", όπως φαίνεται στην εικόνα 1.1.



*Εικόνα 1.2: Ο βραχίονας του Στάνφορντ*

Οι δυνατότητες των ρομπότ άρχισαν να επεκτείνονται, οπότε η χρήση τους γίνεται και σε πιο εξελιγμένες εφαρμογές, όπως η συναρμολόγηση και συγκόλληση. Έτσι, ένας ηλεκτρικός βραχίονας με έξι αρθρώσεις, είχε τη δυνατότητα να κινείται με ακρίβεια σε πολλά σημεία στο χώρο. Ο Scheinman σχεδίασε κι ένα δεύτερο βραχίονα για το εργαστήριο Τεχνητής Νοημοσύνης του MIT. Στη συνέχεια εξέλιξε τα σχέδιά του, τα οποία πούλησε στην ίδια εταιρία, ενώ συνέχισε να τα εξελίσει με την υποστήριξη της General Motors. Το αποτέλεσμα αυτών των συνεργασιών ήταν η καθολικά προγραμματιζόμενη μηχανή για συναρμολόγηση (PUMA), όπως φαίνεται στο εικόνα 1.2.



*Εικόνα 1.3: Ο Scheinman και ο βραχίονας PUMA*

Η ρομποτική βιομηχανία, στην Ευρώπη, αναπτύχθηκε πολύ γρήγορα. Σε αυτό συνέβαλαν οι εταιρίες ABB Robotics και KUKA Robotics, όπου διέθεσαν ρομπότ στην αγορά το 1972. Από αυτές η ABB Robotics ήταν από τις πρώτες εταιρίες στον κόσμο, που διέθεσαν στο εμπόριο εξ ολοκλήρου ηλεκτρικά ρομπότ, τα οποία ελέγχονταν από μικροεπεξεργαστή. Η KUKA Robotics, το 1973, δημιούργησε το πρώτο ρομπότ, γνωστό ως FAMULUS, το ποίο όπως και το PUMA ήταν ένα από τα πρώτα αρθρωτά ρομπότ που δούλευαν με έξι ηλεκτρομηχανικούς άξονες.

Αξίζει να σημειωθεί ότι οι ABB και η KUKA Robotics πρωταγωνιστούν ακόμα στο χώρο της ρομποτικής, μαζί με αρκετές ακόμη εταιρίες, όπως η Automatrix, η Adept Technology, η Stäubli-Unimation, η Swedish-Swiss και άλλες, κυρίως Ιαπωνικές, όπως η FANUC LTD.

Στην εικόνα 1.4 φαίνεται ένας βραχίονας της KUKA Robotics, τελευταίας τεχνολογίας, ο οποίος χρησιμοποιείται αρκετά από τις αυτοκινητοβιομηχανίες.



*Εικόνα 1.4: Βραχίονας KUKA Robotics του 2015*

Η πρώτη γενιά των σύγχρονων ρομπότ είχε πολλές εμφανείς διαφορές από τα ανθρωπόμορφα μηχανήματα, καθώς οι περισσότεροι κατασκευαστές δεν ήθελαν να μιμηθούν το ανθρώπινο γένος. Το πιο δημοφιλές ρομπότ της δεκαετίας του 1960, το Unimate, μπορούσε να κινήσει μόνο το ένα του χέρι προς διάφορες κατευθύνσεις και να ανοιγοκλείνει την παλάμη του. Σήμερα, έχουν κατασκευαστεί περισσότερα από δύο εκατομμύρια ρομπότ Roomba, τα οποία κινούνται και εκτελούν εργασίες, που άλλοτε εκτελούνταν από τους ανθρώπους, αλλά μοιάζουν περισσότερο με γρήγορες χελώνες παρά με υπηρέτες.

### **1.1.2 ΤΟΜΕΙΣ ΕΦΑΡΜΟΓΗΣ**

Τα τελευταία χρόνια σημειώνεται μία ραγδαία ανάπτυξη στις επιστήμες και στην τεχνολογία. Κατά τη διάρκεια του 20ου αιώνα εμφανίστηκε ενδιαφέρον για την ανάπτυξη της ρομποτικής και έτσι το ρομπότ αποτέλεσε αντικείμενο μελέτης. Η Ρομποτική βρήκε εφαρμογή σε αρκετά πεδία. Κατά κύριο λόγο εφαρμόστηκε περισσότερο στους τομείς όπως η βιομηχανία και ο στρατός. Βέβαια εφαρμόστηκε και εφαρμόζεται και σε άλλους τομείς.

Τα βασικά χαρακτηριστικά των βιομηχανικών ρομπότ είναι τα παρακάτω:

- Υποκατάστατο του ανθρώπινου χεριού
- Ευελιξία

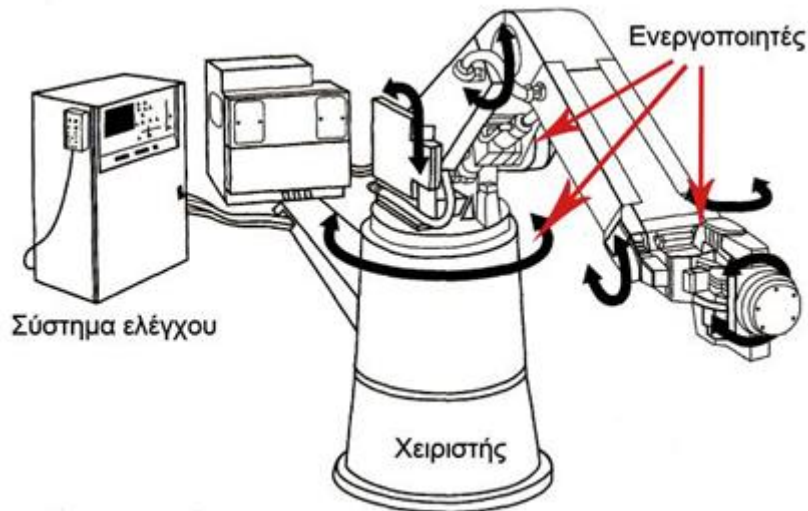
- Ευφυΐα και ειδικούς αισθητήρες
- Ταχύτητα και ακρίβεια
- Μεγαλύτερη δύναμη από τον άνθρωπο
- Γρήγορη ανάπτυξη – ολοκλήρωση
- Οικονομικές λύσεις για μικρή παραγωγή
- Σταθερή ποιότητα προϊόντος και παραγωγικότητα

Ακόμα αξίζει να αναφερθεί και η εφαρμογή της ρομποτικής τεχνολογίας στον τομέα της ιατρικής. Τις τελευταίες δεκαετίες έχει αναπτυχθεί η Ρομποτική Ιατρική (Medical Robotics). Η έρευνα που γίνεται στον τομέα αυτό, χρησιμοποιώντας τη ρομποτική, συγκεντρώνεται σε συγκεκριμένα πεδία, από τα οποία τα πιο κύρια είναι αυτά που ακολουθούν:

- Ρομποτική Χειρουργική (Robotic Surgery)
- Ρομποτική Αποκατάστασης (Rehabilitation Robotics)
- Προσομοιωτές Χειρουργικών Επεμβάσεων Εικονικής Πραγματικότητας με Οπτική Ανάδραση - Ανατροφοδότηση Δύναμης ή χωρίς (Training Simulators with Haptic Display)
- Διαγνωστική Ρομποτική Τεχνολογία (Diagnostics Robotic Technology)
- Μικρορομποτική (Micro -Robotics)

## 1.2 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΩΝ ΡΟΜΠΟΤ

Στην εικόνα που ακολουθεί παρουσιάζεται η τυπική μορφή ενός ρομπότ-βραχίονα (εικόνα 1.5). Ένα βιομηχανικό ρομπότ μιμείται το ανθρώπινο χέρι. Επειδή τα βιομηχανικά ρομπότ εκτελούν διάφορους χειρισμούς, ονομάζονται και μηχανικοί χειριστές ή απλώς χειριστές. Έτσι, λοιπόν ένα ρομπότ έχει αρθρώσεις ανάλογες αυτών του ανθρώπινου χεριού.



Εικόνα 1.5: Τυπική μορφή βιομηχανικού ρομπότ (Cincinnati Milacron)

Ένα σύγχρονο ρομπότ περιέχει ένα ρομπότ έναν υπολογιστή, ο οποίος περιέχει δύο προγράμματα. Ένα πρόγραμμα ελέγχου και ένα πρόγραμμα εργασίας.

Το πρόγραμμα ελέγχου ορίζεται από τον κατασκευαστή. Με αυτό το πρόγραμμα ρυθμίζει την κίνηση κάθε μιας άρθρωσης του ρομπότ. Το πρόγραμμα εργασίας ορίζεται από το χρήστη. Έτσι, καθορίζονται οι κινήσεις, που χρειάζονται για να εκτελεστεί κάθε φορά η επιθυμητή εργασία. Η παραγωγή ενός προγράμματος εργασίας μπορεί να γίνει είτε περνώντας το ρομπότ από τις θέσεις που απαιτεί η συγκεκριμένη εργασία, είτε χρησιμοποιώντας κατάλληλες γλώσσες προγραμματισμού. Όταν γίνεται η χρήση μιας γλώσσας προγραμματισμού, το ρομπότ περιέχει έναν κατάλληλο επεξεργαστή, ο οποίος μπορεί να μεταφράζει το πρόγραμμα εργασίας και να παρέχει τα δεδομένα, που απαιτούνται από το πρόγραμμα ελέγχου για να καθοδηγήσει το ρομπότ στις επιθυμητές κινήσεις. Το πρόγραμμα ελέγχου χρησιμοποιεί το πρόγραμμα εργασίας σαν είσοδο και συνεπώς, για κάθε εργασία ο χρήστης πρέπει να γράψει ένα διαφορετικό πρόγραμμα εργασίας.

Τα πλεονεκτήματα των βιομηχανικών ρομπότ συνοψίζονται ως εξής:

- Ευελιξία
- Υψηλή παραγωγικότητα
- Καλύτερη ποιότητα προϊόντος
- Βελτιωμένη ποιότητα ζωής

### 1.3 ΔΟΜΗ ΤΩΝ ΡΟΜΠΟΤ

Το βιομηχανικό ρομπότ μπορεί να οριστεί ως ένας προγραμματιζόμενος μηχανικός χειριστής, που είναι ικανός να κινείται προς διάφορες διευθύνσεις, ο οποίος στο άκρο του είναι εφοδιασμένος με μια διάταξη εργασίας, που ονομάζεται τελικό σημείο δράσης (ή εργαλείο). Επίσης, είναι ικανός να εκτελεί διάφορες βιομηχανικές εργασίες, που συνήθως γίνονται από τον άνθρωπο. Ένα σύγχρονο ρομπότ αποτελείται από τουλάχιστον τρία βασικά δομικά μέρη:

- Το χειριστή (βραχίονα, χέρι), που αποτελεί το κινούμενο μηχανικό τμήμα,
- Τα στοιχεία δράσης (κινητήρες κ.λ.π.), τα οποία ενεργοποιούν τις αρθρώσεις του βραχίονα και
- Τον υπολογιστή, ο οποίος αποθηκεύει και εκτελεί τα προγράμματα εργασίας αλλά και ελέγχει τις κινήσεις του ρομπότ (εικόνα 1.6).

Ο βραχίονας του ρομπότ αποτελείται από δύο βασικά μέρη:

- το κύριο σώμα και
- τον καρπό, που στο τέλος του έχει το εργαλείο που αποτελεί το τελικό στοιχείο δράσης.

Το εργαλείο μπορεί να είναι είτε μια κεφαλή συγκόλλησης είτε ένα πιστόλι χρωματίσματος. Επίσης μπορεί να είναι ένα μηχανικό εργαλείο ή μια αρπάγη, που ανοιγοκλείνει ανάλογα με την εφαρμογή, στην οποία πρόκειται να χρησιμοποιηθεί το ρομπότ. Επειδή όλα τα παραπάνω εργαλεία στερεώνονται στο άκρο του ρομποτικού βραχίονα, λέγονται και "τελικά στοιχεία δράσης".

Ο κορμός ή αλλιώς το κύριο σώμα του ρομπότ αποτελείται από μια διαδοχή συνδέσμων ή μελών (links), που συνδέονται με αρθρώσεις (joints), οι οποίες ελέγχουν τις κινήσεις των συνδέσμων. Από τις αρθρώσεις αυτές, το σύνολο αυτών που ελέγχουν την κίνηση του εργαλείου, ονομάζεται "καρπός". Κάθε μια από τις αρθρώσεις του βραχίονα και του καρπού παρέχει έναν βαθμό ελευθερίας στην κίνηση του τελικού στοιχείου δράσης. Έτσι, ένα ρομπότ με  $n$  βαθμούς ελευθερίας περιέχει  $n$  αρθρώσεις ή  $n$  άξονες κίνησης γενικά. Οι παράγοντες που ρυθμίζουν την κίνηση του τελικού στοιχείου δράσης είναι ο έλεγχος της θέσης και της ταχύτητας των αξόνων κίνησης του ρομπότ. Στη ρομποτική ένας άξονας κίνησης ισοδυναμεί με έναν βαθμό ελευθερίας, ως προς τον οποίο μπορεί να κινηθεί το ρομπότ. Για να μπορέσει ένα ρομπότ να προσεγγίσει ένα οποιαδήποτε σημείο (μέσα στο χώρο εργασίας του) με

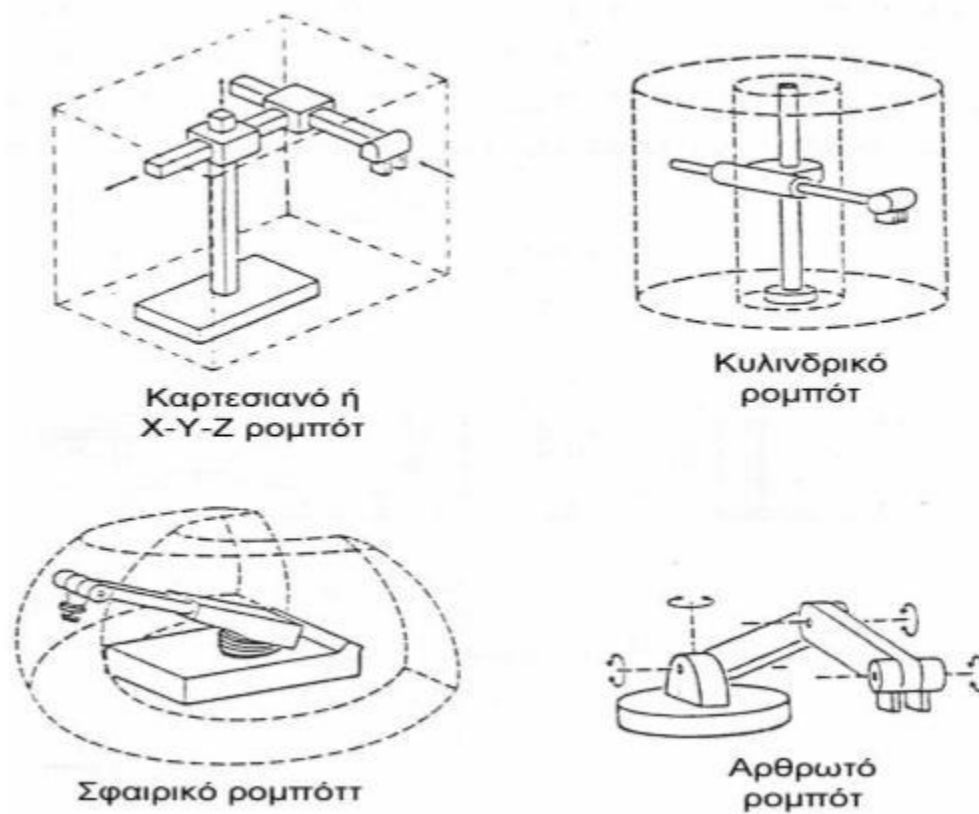
έναν επιθυμητό προσανατολισμό του εργαλείου, χρειάζεται να έχει έξι άξονες (βαθμούς ελευθερίας) κίνησης. Έστω και μόνο ένας διαφορετικός προσανατολισμός του εργαλείου μπορεί να αλλάξει εντελώς τη θέση του βραχίονα του ρομπότ.

#### 1.4 ΤΥΠΟΙ ΡΟΜΠΟΤΙΚΩΝ ΒΡΑΧΙΟΝΩΝ ΚΑΙ ΧΩΡΟΣ ΕΡΓΑΣΙΑΣ

Το μηχανικό τμήμα των ρομπότ αποτελείται από μια αλληλουχία μηχανικών συνδέσμων (links) και αρθρώσεων (joints), οι οποίες ενώνουν τους συνδέσμους ανά δύο. Ένα τυπικό ρομπότ αποτελείται από το κυρίως τμήμα (δηλαδή τον βραχίονα) και από τον καρπό. Τόσο ο βραχίονας (arm) όσο και ο καρπός (wrist) έχουν τρεις βαθμούς ελευθερίας, ο καθένας. Υπάρχουν όμως ρομπότ στα οποία ο καρπός έχει λιγότερους βαθμούς ελευθερίας.

Τα ρομπότ ανάλογα με το σύστημα συντεταγμένων του βραχίονα ταξινομούνται σε (εικόνα 1.7):

- **Καρτεσιανά:** Τρεις γραμμικοί άξονες
- **Κυλινδρικά:** Δυο γραμμικοί και ένας στροφικός άξονας
- **Σφαιρικά:** Ένας γραμμικός και δυο στροφικοί άξονες
- **Αρθρωτά:** Τρεις στροφικοί άξονες



Εικόνα 1.6: Οι τέσσερις βασικοί δομικοί τύποι βιομηχανικού ρομπότ.

Οι γραμμικές αρθρώσεις μπορούν να είναι αρθρώσεις ολίσθησης (συμβολικά S: sliding), ή πρισματικές (συμβολικά P: prismatic). Μια, στροφική άρθρωση συμβολίζεται με R (revolute joint). Έτσι ο τύπος ενός ρομπότ με βάση τα συστήματα συντεταγμένων των αρθρώσεων του συμβολίζεται με την αλληλουχία των συμβόλων S, P και R αρχίζοντας από τη βάση και προχωρώντας προς τον καρπό. Ένα σφαιρικό ρομπότ μπορεί να είναι της μορφής RRP, ενώ ένα αρθρωτό ρομπότ συμβολίζεται με RRR .

### 1.3.1 ΡΟΜΠΟΤ ΚΑΡΤΕΣΙΑΝΩΝ ΣΥΝΤΕΤΑΓΜΕΝΩΝ

Το κυρίως σώμα ενός ρομπότ καρτεσιανών συντεταγμένων αποτελείται από τρεις γραμμικούς άξονες όπως φαίνεται στην εικόνα 1.7. Η δομή μπορεί να είναι όμοια με των εργαλειομηχανών (βάση, τραπέζι εργασίας, κ.ά.) αλλά τότε ο λόγος μεταξύ του χώρου εργασίας του ρομπότ και του χώρου που καταλαμβάνει είναι μικρότερος. Γενικά, τα χαρακτηριστικά (ηλεκτρονικό υλικό, hardware, πρόγραμμα έλεγχου κλπ)

ενός καρτεσιανού ρομπότ είναι όμοια με εκείνα των εργαλειομηχανών αριθμητικού ελέγχου (CNC) .

### **1.3.2 ΡΟΜΠΟΤ ΚΥΛΙΝΔΡΙΚΩΝ ΣΥΝΤΕΤΑΓΜΕΝΩΝ**

Το κυρίως σώμα ενός ρομπότ κυλινδρικών συντεταγμένων αποτελείται από ένα οριζόντιο βραχίονα στερεωμένο σε μια κατακόρυφη κολώνα. Η κολώνα είναι με τη σειρά της στερεωμένη πάνω σε μια περιστρεφόμενη βάση (εικόνα 1.7). Ο οριζόντιος βραχίονας κινείται προς τα εμπρός και προς τα πίσω κατά τη διεύθυνση του διαμήκους άξονα του και επίσης ανεβοκατεβαίνει στη κολώνα. Κολώνα και βραχίονας στρέφονται σαν ένα σώμα πάνω στη βάση γύρω από τον κατακόρυφο άξονα.

### **1.3.3 ΡΟΜΠΟΤ ΣΦΑΙΡΙΚΩΝ ΣΥΝΤΕΤΑΓΜΕΝΩΝ**

Τα ρομπότ των σφαιρικών συντεταγμένων αποτελούνται από μια στρεφόμενη βάση, ένα ανυψούμενο στέλεχος και ένα τηλεσκοπικό βραχίονα που κινείται προς τα μέσα και προς τα έξω όπως φαίνεται στην εικόνα 1.7. Το μέγεθος της περιστροφής μετράται από ψηφιακούς κωδικοποιητές που είναι στερεωμένοι στους άξονες. Το βασικό μειονέκτημα των σφαιρικών ρομπότ είναι και πάλι η μικρή διακριτική ικανότητα θέσης των δυο στροφικών αξόνων που μεταβάλλεται με το μήκος του βραχίονα. Τα σφαιρικά ρομπότ, εκτός από το πλεονέκτημα της αυξημένης ταχύτητας κίνησης των στροφικών αξόνων, έχουν και το πλεονέκτημα της αυξημένης ευελιξίας σε σχέση τόσο με τα καρτεσιανά όσο και με τα κυλινδρικά ρομπότ.

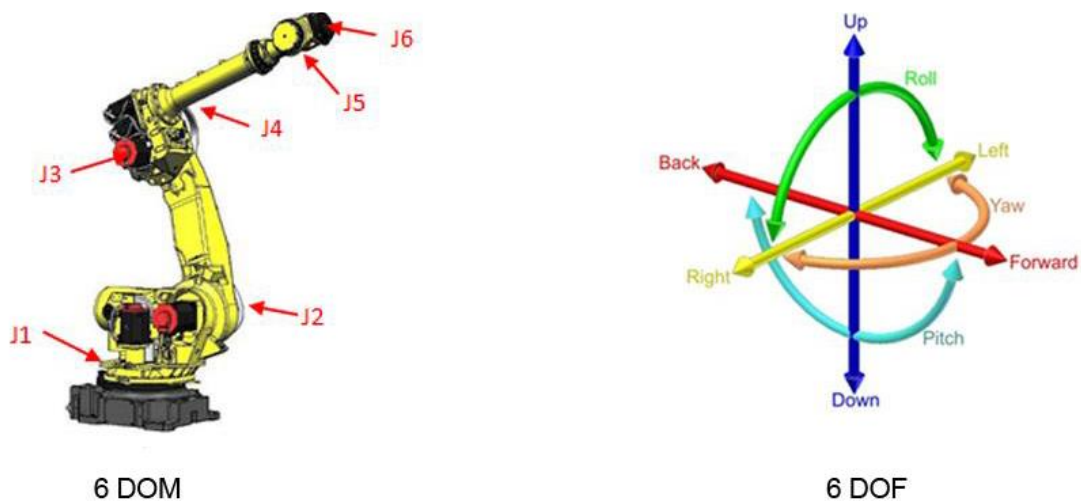
### **1.3.4 ΑΡΘΡΩΤΟ ΡΟΜΠΟΤ**

Τα αρθρωτά ρομπότ αποτελούνται από τρία σταθερά μέλη (συνδέσμους) που ενώνονται με στροφικές αρθρώσεις και είναι τοποθετημένα πάνω σε μια στρεφόμενη βάση (εικόνα 1.7). Η κινηματική διάταξη μοιάζει με εκείνη του ανθρώπινου χεριού. Το εργαλείο (αρπάγη) είναι ανάλογο της παλάμης και προσαρμόζεται στον πήχη (Forearm) μισώ του καρπού. Ο «αγκώνας» συνδέει τον πήχη με τον άνω βραχίονα και ο «ώμος» συνδέει τον άνω βραχίονα με τη βάση. Να σημειωθεί ότι σε κάποια ρομπότ στην άρθρωση του ώμου διατίθεται και μια περιστροφική κίνηση σε οριζόντιο επίπεδο.

### 1.3.5 ΒΑΘΜΟΙ ΕΛΕΥΘΕΡΙΑΣ

Ο αριθμός των βαθμών ελευθερίας ενός ρομποτικού βραχίονα είναι ο αριθμός των ανεξάρτητων μεταβλητών θέσης. Ο αριθμός αυτός πρέπει να είναι γνωστός και μονοσήμαντα ορισμένος για κάθε διάταξη, έτσι ώστε να είναι εφικτός ο προσδιορισμός της θέσης των τμημάτων που τη συνιστούν. Πρόκειται για μια γενική αρχή, που εφαρμόζεται σε όλους τους μηχανισμούς.

Κρίνεται σκόπιμο να επισημάνουμε τη διαφορά που υπάρχει ανάμεσα στους βαθμούς κινητικότητας ενός βραχίονα και τους βαθμούς ελευθερίας που απαιτούνται για την εκτέλεση ενός έργου. Για έναν βραχίονα το πλήθος των βαθμών κινητικότητας είναι σταθερό και ίσο με το πλήθος των αρθρώσεων του. Από την άλλη πλευρά οι βαθμοί ελευθερίας είναι άμεσα συνδεδεμένοι με το συγκεκριμένο έργο που καλείται να φέρει εις πέρας ο βραχίονα. Για τη γενική περίπτωση που θέλουμε να τοποθετήσουμε και να προσανατολίσουμε ένα αντικείμενο στο χώρο απαιτούνται έξι βαθμοί ελευθερίας (τρεις για να τοποθετήσουμε ένα σημείο του αντικείμενου στο χώρο και τρεις για να προσανατολίσουμε το αντικείμενο ως προς ένα σύστημα συντεταγμένων αναφοράς) είναι προφανές ότι ένας ρομποτικός βραχίονας με 6 βαθμούς κινητικότητας μπορεί να ανταπεξέλθει σ' αυτό το έργο, όπως επίσης και σε οποιοδήποτε άλλο έργο που απαιτεί μέχρι έξι βαθμούς ελευθερίας (εικόνα 1.8).



Εικόνα 1.7: Οι 6 βαθμοί κινητικότητας και οι 6 βαθμοί ελευθερίας

## ΚΕΦΑΛΑΙΟ 2

### ΚΙΝΗΜΑΤΙΚΗ ΘΕΩΡΙΑ

#### 2.1 ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΚΙΝΗΜΑΤΙΚΗ ΘΕΩΡΙΑ

Κινηματική είναι η επιστήμη που μελετά την κίνηση χωρίς να λαμβάνει υπόψη τις δυνάμεις που την προκαλούν. Στην κινηματική μελετάμε τη θέση, την ταχύτητα, την επιτάχυνση και όλες τις παραγώγους των μεταβλητών που περιγράφουν τη θέση.

Η κινηματική ανάλυση ενός βραχίονα αφορά την επίλυση δύο προβλημάτων:

- του ευθέος και του αντίστροφου κινηματικού προβλήματος. Το ευθύ κινηματικό πρόβλημα συνίσταται την εύρεση της θέσης και του προσανατολισμού του άκρου του βραχίονα ως προς την βάση του, όταν γνωρίζουμε την θέση κάθε άρθρωσης,
- του αντίστροφου κινηματικού προβλήματος, που συνίσταται στην εύρεση της θέσης της κάθε άρθρωσης, η οποία τοποθετεί το άκρο του βραχίονα σε μια συγκεκριμένη θέση και προσανατολισμό σε σχέση με την βάση του. Ο βαθμός δυσκολίας επίλυσης των προβλημάτων αυτών εξαρτάται από την γεωμετρία του βραχίονα.

Για τη μετακίνηση του άκρου του βραχίονα από την αρχική του θέση στην επιθυμητή είναι απαραίτητη η γνώση των γωνιών των αρθρώσεων που αντιστοιχούν στην επιθυμητή θέση. Οι γωνίες αυτές υπολογίζονται επιλύοντας το αντίστροφο κινηματικό πρόβλημα. Η μετάβαση μεταξύ δύο διαδοχικών καταστάσεων του βραχίονα γίνεται μέσω ευθύγραμμης τροχιάς με παραβολική μείξη στο επίπεδο των αρθρώσεων ώστε να επιτυγχάνεται τόσο η ομαλή κίνηση της κατασκευής όσο και η ταυτόχρονη έναρξη και λήξη της κίνησης όλων των κινητήρων.

#### 2.2 ΕΥΘΥ ΚΙΝΗΜΑΤΙΚΟ ΠΡΟΒΛΗΜΑ

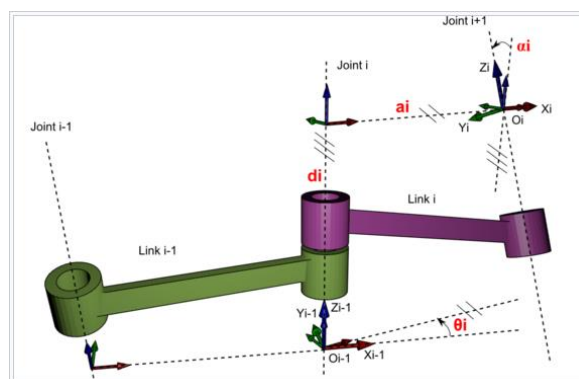
Ένας βραχίονας αποτελείται από σειριακούς συνδέσμους οι οποίοι τοποθετούνται σε κάθε άρθρωση από τη βάση μέχρι το εργαλείο τελικής δράσης. Για να εφαρμοστεί το ευθύ κινηματικό σε ένα ρομπότ θα πρέπει να χρησιμοποιήσουμε ένα κατάλληλο κινηματικό μοντέλο. Η μεθοδολογία Denavit - Hartenberg που χρησιμοποιεί τέσσερις παραμέτρους είναι η πιο κοινή μέθοδος για την περιγραφή του κινηματικού προβλήματος ενός ρομπότ. Ένα σύστημα συντεταγμένων επισυνάπτεται σε κάθε

άρθρωση και καθορίζει τις DH παραμέτρους. Βάσει της μεθόδου αυτής επιλέγονται με συγκεκριμένο τρόπο τα συστήματα συντεταγμένων που είναι προσαρμοσμένα σε κάθε σύνδεσμο του ρομποτικού βραχίονα εκτελώντας τα βήματα του αλγορίθμου. Έτσι προσδιορίζεται η θέση και ο προσανατολισμός του άκρου εργασίας ως προς το ακίνητο σύστημα συντεταγμένων.

### 2.2.1 ΑΛΓΟΡΙΘΜΟΣ DENAVIT - HARTENBERG

Ο αλγόριθμος των D-H, είναι μια διαδικασία σύμφωνα με την οποία τοποθετούνται ορθοκανονικά και δεξιόστροφα Συστήματα Συντεταγμένων στους συνδέσμους του βραχίονα. Η συστηματική αυτή μέθοδος μας βοηθάει στο να ορίσουμε την σχετική θέση και τον προσανατολισμό μεταξύ δύο διαδοχικών συνδέσμων. Βασικό στοιχείο της μεθόδου αποτελεί ο σωστός ορισμός των πλαισίων κάθε άρθρωσης. Προκειμένου να γίνει αυτό, πρέπει να τηρηθούν οι παρακάτω τέσσερις κανόνες(Denavit – Hartenberg).

1. Επιλέγουμε τον άξονα  $Z_i$  κατά μήκος του άξονα της άρθρωσης  $i+1$
1. Τοποθετούμε το κέντρο  $O_i$ , στην τομή του άξονα  $Z_i$ , με την κοινή κάθετο των αξόνων  $Z_{i-1}$  και  $Z_i$ .
2. Επιλέγουμε τον άξονα  $x_0$  κατά μήκος της κοινής κάθετου των αξόνων  $Z_{i-1}$  και  $Z_i$  με τη φορά από την άρθρωση  $i$  στην άρθρωση  $i+1$ .
3. Επιλέγουμε τον άξονα  $y_i$ , ώστε να έχουμε δεξιόστροφο σύστημα συντεταγμένων για το πλαίσιο  $i$ , έχοντας ορίσει τα πλαίσια αναφοράς κάθε άρθρωσης, όπως φαίνεται στο σχήμα 2.1 το μόνο που μας απομένει είναι να προσδιορίσουμε τις τέσσερις παραμέτρους της μεθόδου που είναι οι  $d_i$ ,  $\theta_i$ ,  $a_i$ ,  $\alpha_i$ .



Σχήμα 2.1: Συντεταγμένες σε γενικού τύπου βραχίονα (wikipedia)

Οι παράμετροι αυτοί ορίζονται ως εξής:

- $a_i$  = η απόσταση μεταξύ των  $Z_i$  και  $Z_{i+1}$  μετρούμενη κατά μήκος του  $X_i$
- $\alpha_i$  = η γωνία μεταξύ των  $Z_i$  και  $Z_{i+1}$  μετρούμενη ως προς τον  $X_i$
- $d_i$  = η απόσταση μεταξύ των  $X_{i-1}$  και  $X_i$  μετρούμενη κατά μήκος του  $Z_i$
- $\theta_i$  = η γωνία μεταξύ των  $X_{i-1}$  και  $X_i$  μετρούμενη ως προς  $Z_i$

Εφόσον έχουμε ορίσει τα πλαίσια των αρθρώσεων και τις παραμέτρους της μεθόδου D-H, ακολουθούμε τα παρακάτω βήματα, προκειμένου να ορίσουμε τη θέση και τον προσανατολισμό του πλαισίου ως προς το πλαίσιο  $i-1$ .

**Βήμα 1:** Περιστροφή του πλαισίου  $i-1$  γύρω από τον άξονα  $Z_{i+1}$  κατά γωνία  $\theta_i$

**Βήμα 2:** Μετατόπιση  $d_i$  του πλαισίου  $i-1$  κατά μήκος του άξονα  $Z_{i-1}$

**Βήμα 3:** Μετατόπιση  $a_i$  (μήκος της κοινής καθέτου) κατά τον άξονα  $x_0$

**Βήμα 4:** Περιστροφή γύρω από τον άξονα  $X_i$  κατά γωνία  $\alpha_i$

Από το βήμα 1 και βήμα 2, παίρνουμε την μήτρα μετασχηματισμού:

$$T_{\Sigma_i^{i-1}} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ενώ εφαρμόζοντας το βήμα 3 και βήμα 4 έχουμε την μήτρα μετασχηματισμού:

$$T_i^{\Sigma_i} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Οπότε συνολικά η μήτρα μετασχηματισμού που μας δίνει τη σχετική θέση και τον προσανατολισμό του πλαισίου  $i$  ως προς το πλαίσιο  $i-1$ , είναι η ακόλουθη:

$$T_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Πίνακας 2.1: Παράμετροι Denavit – Hardenberg

<b>i</b>	<b><math>a_{i-1}</math></b>	<b><math>\alpha_{i-1}</math></b>	<b><math>d_i</math></b>	<b><math>\theta_i</math></b>
<b>1</b>	0	0	0	$\theta_1$
<b>2</b>	0	$L_1$	0	$\theta_2$
<b>3</b>	0	$L_2$	0	$\theta_3$

Στον παραπάνω πίνακα παραμέτρων κάθε σειρά περιέχει όλες τις απαραίτητες παραμέτρους για τη δημιουργία των ομογενών μετασχηματισμών μεταξύ των πλαισίων των γειτονικών συνδέσμων.

### 2.3 ΕΙΔΗ ΑΡΘΡΩΣΕΩΝ

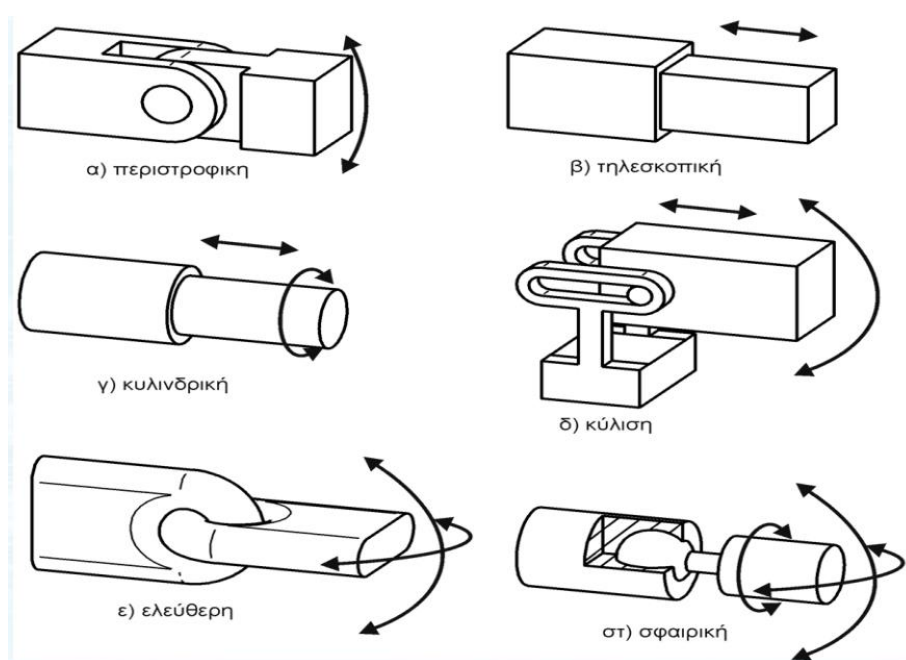
Η περιστροφική άρθρωση είναι άρθρωση που επιτρέπει σχετική στροφή μεταξύ δύο γειτονικών συνδέσμων. Δίνει ένα βαθμό ελευθερίας αφήνοντας το σώμα να περιστραφεί σε ένα επίπεδο και αποκόπτει κάθε άλλη δυνατότητα κίνησης. Στην περιστροφική άρθρωση μεταβάλλεται η γωνία  $\theta_i$ , ενώ οι  $\alpha_i, d_i, a_i$  παραμένουν σταθερές.

Η πρισματική (ή τηλεσκοπική) άρθρωση είναι άρθρωση που επιτρέπει σχετική μετατόπιση (σε ευθεία γραμμή) μεταξύ δύο γειτονικών συνδέσμων. Δίνει και αυτή ένα βαθμό ελευθερίας αφήνοντας το σώμα να μετατοπίζεται στη διεύθυνση ενός από τους άξονες, και αποκόπτει κάθε άλλη δυνατότητα κίνησης. Στην πρισματική άρθρωση μεταβάλλεται η  $d_i$ , ενώ οι  $\alpha_i, \theta_i, a_i$  παραμένουν σταθερές.

Σύνθετες αρθρώσεις είναι αυτές που αναλύονται γεωμετρικά σε υπέρθεση δύο ή περισσότερων από τις βασικές αρθρώσεις (στροφική και πρισματική).

### 2.2.1 ΣΥΝΘΕΤΕΣ ΑΡΘΡΩΣΕΙΣ

Η κυλινδρική άρθρωση δίνει δύο βαθμούς ελευθερίας αφήνοντας μία μεταφορική κίνηση στη διεύθυνση ενός άξονα και μία περιστροφική γύρω από τον άξονα αυτό. Η άρθρωση της κύλισης δίνει και αυτή δύο βαθμούς ελευθερίας, δηλαδή μία μεταφορική και μία περιστροφική κίνηση, άλλα σε αυτή την περίπτωση ο άξονας της περιστροφικής κίνησης είναι κάθετος στη διεύθυνση του άξονα που πραγματοποιείται η μεταφορική κίνηση. Η ελεύθερη άρθρωση δίνει δύο βαθμούς ελευθερίας αφήνοντας δύο περιστροφικές κινήσεις και εμποδίζοντας όλες τις υπόλοιπες. Η σφαιρική άρθρωση δίνει τρεις βαθμούς ελευθερίας αφήνοντας και τις τρεις περιστροφικές κινήσεις ελεύθερες και εμποδίζοντας όλες τις μεταφορικές.

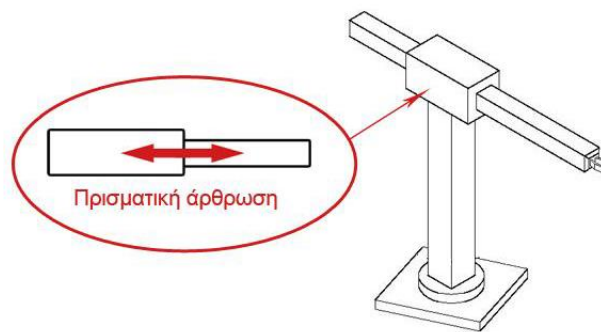


Σχήμα 2.2: Είδη αρθρώσεων.

**Περιστροφική άρθρωση:** Η περιστροφική άρθρωση (revolute joint), η οποία συμβολίζεται με R, επιτρέπει τη σχετική περιστροφική κίνηση δύο διαδοχικών συνδέσμων. Η άρθρωση αυτή δίνει ένα βαθμό ελευθερίας στο βραχίονα, επιτρέποντας στο σώμα του να περιστραφεί σε ένα επίπεδο, ενώ ταυτόχρονα αποκόπτει κάθε άλλη δυνατότητα κίνησης. Η μεταβλητή που ορίζει την κίνησή της

είναι η γωνία  $\theta$  ή  $q$ , η οποία θα αναλυθεί παρακάτω, στην παράγραφο της κινηματικής των βραχιόνων.

**Πρισματική άρθρωση:** Η πρισματική (ή γραμμική ή τηλεσκοπική) άρθρωση (prismatic joint), που με συμβολισμό P, επιτρέπει τη σχετική μετατόπιση (σε ευθεία γραμμή) μεταξύ δύο διαδοχικών συνδέσμων. Η άρθρωση αυτή δίνει ένα βαθμό ελευθερίας στο βραχίονα επιτρέποντας στο σώμα του να μετατοπίζεται στη διεύθυνση του άξονα στον οποίο είναι προσανατολισμένη, ενώ αποκόπτει κάθε άλλη δυνατότητα κίνησης. Η μεταβλητή που ορίζει την κίνησή της είναι η μετατόπιση  $d$ , η οποία θα αναλυθεί παρακάτω, στην παράγραφο της κινηματικής των βραχιόνων. Στην εικόνα 10 φαίνεται μια πρισματική άρθρωση.



Σχήμα2.3: Πρισματική άρθρωση

**Σύνθετες αρθρώσεις:** Σύνθετες αρθρώσεις είναι αυτές, που μπορούν να αναλυθούν γεωμετρικά σε υπέρθεση δύο ή περισσότερων από τις βασικές αρθρώσεις (περιστροφική και πρισματική). Σύνθετες αρθρώσεις μπορούν να θεωρηθούν οι παρακάτω:

- Η κυλινδρική άρθρωση, που δίνει δύο βαθμούς ελευθερίας στο βραχίονα και επιτρέπει στο σώμα του να εκτελεί μία μεταφορική κίνηση στη διεύθυνση ενός άξονα και μία περιστροφική γύρω από τον άξονα αυτό.
- Η άρθρωση της κύλισης, που δίνει και αυτή δύο βαθμούς ελευθερίας, δηλαδή μία μεταφορική και μία περιστροφική κίνηση, άλλα σε αυτή την περίπτωση ο άξονας της περιστροφικής κίνησης είναι κάθετος στη διεύθυνση του άξονα που πραγματοποιείται η μεταφορική κίνηση.

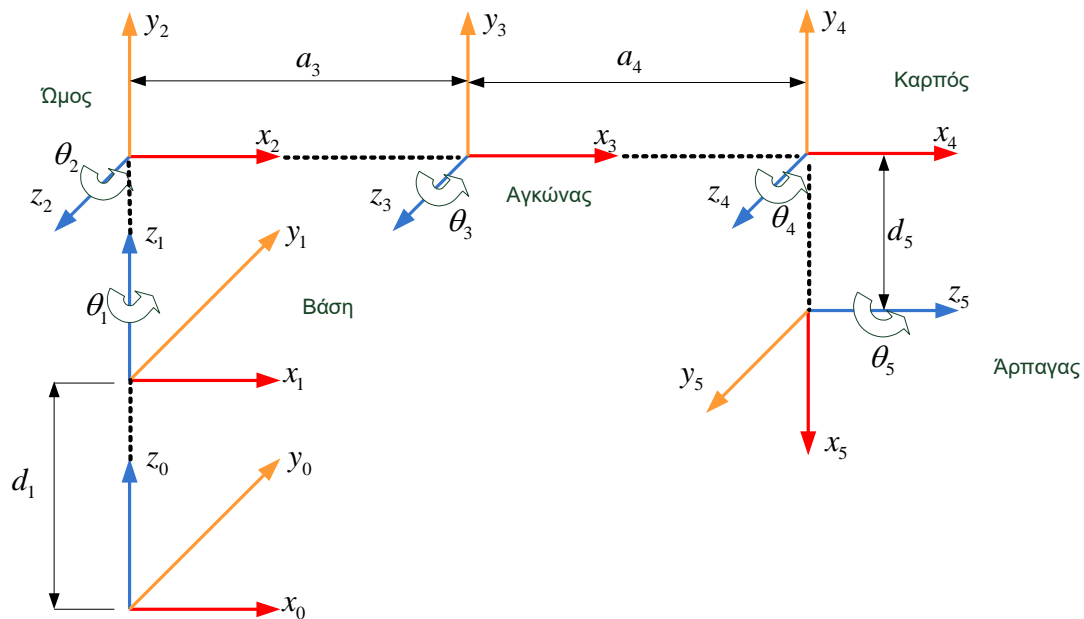
## Κεφάλαιο 2

- Η ελεύθερη άρθρωση. Η άρθρωση αυτή δίνει δύο βαθμούς ελευθερίας επιτρέποντας δύο περιστροφικές κινήσεις και εμποδίζοντας όλες τις υπόλοιπες.
- Η σφαιρική άρθρωση, η οποία δίνει τρεις βαθμούς ελευθερίας αφήνοντας και τις τρεις περιστροφικές κινήσεις ελεύθερες και εμποδίζοντας όλες τις μεταφορικές.

## ΚΕΦΑΛΑΙΟ 3

### ΤΟ ΕΥΘΥ ΚΙΝΗΜΑΤΙΚΟ ΠΡΟΒΛΗΜΑ ΕΝΟΣ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ 5 ΒΑΘΜΩΝ ΕΛΕΥΘΕΡΙΑΣ (5 DOF)

Θεωρούμε τον παρακάτω πίνακα (πίνακας 3.1) παραμέτρων Denavit – Hartenberg (DH) για ένα ρομποτικό βραχίονα 5 βαθμών ελευθερίας (5 – DOF), που παρουσιάζεται στο σχήμα που ακολουθεί, σχήμα 3.1.



Σχήμα 3.1: Η παράσταση των αξόνων των επιμέρους τμημάτων ενός ρομποτικού βραχίονα 5 βαθμών ελευθερίας (5 – DOF) [22].

Πίνακας 3.1: Παράμετροι DH για το ρομποτικό βραχίονα 5 – DOF του σχήματος 3.1. [22]

		$a_i$ (deg)	$a_i$ (mm)	$d_i$ (mm)	$\theta_i$ (deg)
1	Βάση	0	0	86	50
2	Ωμος	90	0	0	30
3	Αγκώνας	0	96	0	45
4	Καρπός	0	96	0	25
5	Άρπαγας	90	0	59.5	0

Να σημειωθεί ότι για τη γωνία  $\theta_i$  (η γωνία μεταξύ των  $X_{i-1}$  και  $X_i$  μετρούμενη ως προς  $Z_i$ ) έχουν οριστεί οι παρακάτω περιορισμοί [5]:

$$-360^\circ \leq \theta_1 \leq 360^\circ$$

$$30^\circ \leq \theta_2 \leq 120^\circ$$

$$30^\circ \leq \theta_3 \leq 210^\circ$$

$$-45^\circ \leq \theta_4 \leq 45^\circ$$

$$-45^\circ \leq \theta_5 \leq 45^\circ$$

Σύμφωνα με τις τιμές του πίνακα 3.1 και από τη συνολική μήτρα μετασχηματισμού έχουμε:

$$T_1^0 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

όπου:  $c_1 = \cos\theta_1$  και  $s_1 = \sin\theta_1$

$$T_2^1 = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

όπου:  $c_2 = \cos\theta_2$  και  $s_2 = \sin\theta_2$

$$T_3^2 = \begin{bmatrix} c_3 & -s_3 & 0 & a_3 c_3 \\ s_3 & c_3 & 0 & a_3 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

όπου:  $c_3 = \cos\theta_3$  και  $s_3 = \sin\theta_3$

$$T_4^3 = \begin{bmatrix} c_4 & -s_4 & 0 & a_4 c_4 \\ s_4 & c_4 & 0 & a_4 s_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

όπου:  $c_4 = \cos\theta_4$  και  $s_4 = \sin\theta_4$

$$T_5^4 = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

όπου:  $c_5 = \cos\theta_5$  και  $s_5 = \sin\theta_5$

$$T_5^0 = \begin{bmatrix} c_{12}c_{345} & s_{12} & c_{12}s_{345} & s_{12}d_5 + c_{12}a_4c_{34} + c_{12}a_3c_3 \\ s_{12}c_{345} & -c_{12} & s_{12}s_{345} & -c_{12}d_5 + s_{12}a_4c_{34} + s_{12}a_3c_3 \\ s_{345} & 0 & -c_{345} & a_4s_{34} + a_3s_3 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

όπου:  $c_{12} = \cos(\theta_1 + \theta_2)$ ,  $s_{12} = \sin(\theta_1 + \theta_2)$ ,  $c_{34} = \cos(\theta_3 + \theta_4)$ ,

$s_{34} = \sin(\theta_3 + \theta_4)$ ,  $c_{345} = \cos(\theta_3 + \theta_4 + \theta_5)$ ,  $s_{345} = \sin(\theta_3 + \theta_4 + \theta_5)$

Ο πίνακας  $T_e$  που αντιστοιχεί στη μήτρα μετασχηματισμού των τελεστών:

**θέσης:**  $p_x, p_y, p_z$

**προσανατολισμού στον άξονα x:**  $n_x, n_y, n_z$

**προσανατολισμού στον άξονα y:**  $o_x, o_y, o_z$

**προσανατολισμού στον άξονα z:**  $a_x, a_y, a_z$

έχει τη μορφή:

$$T_e = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Χρησιμοποιώντας τις γωνίες  $\theta_i$  (η γωνία μεταξύ των  $X_{i-1}$  και  $X_i$  μετρούμενη ως προς  $Z_i$ ) του πίνακα 1, τότε προκύπτουν τα παρακάτω αποτελέσματα:

$$T_1^0 = \begin{bmatrix} 0,645 & -0,766 & 0 & 0 \\ 0,766 & 0,645 & 0 & 0 \\ 0 & 0 & 1 & 86 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^1 = \begin{bmatrix} 0,866 & 0 & 0,5 & 0 \\ 0,5 & 0 & -0,866 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^2 = \begin{bmatrix} 0,707 & -0,707 & 0 & 67,882 \\ 0,707 & 0,707 & 0 & 67,882 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^3 = \begin{bmatrix} 0,906 & -0,432 & 0 & 87,006 \\ 0,423 & 0,906 & 0 & 40,751 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_5^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 59,5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Άρα  $T_5^0 = T_1^0 \cdot T_2^1 \cdot T_3^2 \cdot T_4^3 \cdot T_5^4 =$

$$= \begin{bmatrix} 0,0594 & 0,9848 & 0,1632 & 76,0852 \\ 0,3368 & -0,1736 & 0,9254 & 88,854 \\ 0,9397 & 0 & -0,342 & 244,0927 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Από την ισότητα των δύο τελευταίων πινάκων προκύπτουν οι παρακάτω τελεστές:

- Προσανατολισμού στον άξονα x

$$n_x = 0,0594$$

$$n_y = 0,3368$$

$$n_z = 0,9397$$

- Προσανατολισμού στον άξονα y

$$o_x = 0,9848$$

$$o_y = -0,1736$$

$$o_z = 0$$

- Προσανατολισμού στον άξονα z

$$a_x = 0,1632$$

$$a_y = 0,9254$$

$$a_z = -0,342$$

- θέσης

$$p_x = 76,0852$$

$$p_y = 88,854$$

$$p_z = 244,0927$$

Για τον υπολογισμό της ταχύτητας του ρομποτικού βραχίονα πρέπει να προσδιοριστεί η Ιακωβιανή ορίζουσα, η οποία δίνεται από τον τύπο που ακολουθεί:

$$J = \begin{bmatrix} z_0 x(o_5 - o_0) & z_1 x(o_5 - o_1) & z_2 x(o_5 - o_2) & z_3 x(o_5 - o_3) & z_4 x(o_5 - o_4) \\ z_0 & z_1 & z_2 & z_3 & z_4 \end{bmatrix}$$

Η ταχύτητα μίας άρθρωσης μπορεί να είναι:

- γωνιακή:  $J_v$
- μεταφορική:  $J_w$

Η γωνιακή ταχύτητα δίνεται από τη σχέση:

$$J_{vi} = \begin{cases} z_{i-1} x(o_n - o_{i-1}), & \text{για περιστροφική άρθρωση } i \\ z_{i-1} & , \quad \text{για πρισματική άρθρωση } i \end{cases}$$

Η μεταφορική ταχύτητα δίνεται από τη σχέση:

$$J_{wi} = \begin{cases} z_{i-1}, & \text{για περιστροφική άρθρωση } i \\ 0, & \text{για πρισματική άρθρωση } i \end{cases}$$

Για να υπολογιστούν οι παραπάνω ταχύτητες πρέπει πρώτα να οριστούν τα διανύσματα  $z_{i-1}$ ,  $o_n$  και  $o_{i-1}$ , τα οποία προκύπτουν ως εξής:

Θα χρησιμοποιήσουμε τους παρακάτω πίνακες μετασχηματισμού:

$$T_1^0 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

όπου:  $c_1 = \cos\theta_1$  και  $s_1 = \sin\theta_1$

$$T_2^1 = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

όπου:  $c_2 = \cos\theta_2$  και  $s_2 = \sin\theta_2$

$$T_3^2 = \begin{bmatrix} c_3 & -s_3 & 0 & a_3 c_3 \\ s_3 & c_3 & 0 & a_3 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

όπου:  $c_3 = \cos\theta_3$  και  $s_3 = \sin\theta_3$

$$T_4^3 = \begin{bmatrix} c_4 & -s_4 & 0 & a_4 c_4 \\ s_4 & c_4 & 0 & a_4 s_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

όπου:  $c_4 = \cos\theta_4$  και  $s_4 = \sin\theta_4$

$$T_5^4 = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Το διάνυσμα  $o_n = o_5$  ισούται με τα στοιχεία

$$o_5 = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

του πίνακα:

$$T_5^0 = \begin{bmatrix} c_{12}c_{345} & s_{12} & c_{12}s_{345} & s_{12}d_5 + c_{12}a_4c_{34} + c_{12}a_3c_3 \\ s_{12}c_{345} & -c_{12} & s_{12}s_{345} & -c_{12}d_5 + s_{12}a_4c_{34} + s_{12}a_3c_3 \\ s_{345} & 0 & -c_{345} & a_4s_{34} + a_3s_3 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Άρα έχουμε:

$$o_5 = \begin{bmatrix} s_{12}d_5 + c_{12}a_4c_{34} + c_{12}a_3c_3 \\ -c_{12}d_5 + s_{12}a_4c_{34} + s_{12}a_3c_3 \\ a_4s_{34} + a_3s_3 + d_1 \end{bmatrix}$$

Το διάνυσμα  $o_4$  ισούται με τα στοιχεία

$$o_4 = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

ενώ το διάνυσμα  $z_4$  ισούται με τα στοιχεία

$$z = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$$

του πίνακα:

### Κεφάλαιο 3

$$T_4^0 = T_1^0 \cdot T_2^1 \cdot T_3^2 \cdot T_4^3 = \begin{bmatrix} c_{12}c_{34} & -c_{12}s_{34} & s_{12} & c_{12}a_4c_{34} \\ s_{12}c_{34} & -s_{12}s_{34} & -c_{12} & s_{12}a_4c_{34} \\ s_{34} & -c_{34} & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Άρα έχουμε:

$$o_4 = \begin{bmatrix} c_{12}a_4c_{34} \\ s_{12}a_4c_{34} \\ d_1 \end{bmatrix}$$

και

$$z_4 = \begin{bmatrix} s_{12} \\ -c_{12} \\ 1 \end{bmatrix}$$

Το διάνυσμα  $o_3$  ισούται με τα στοιχεία

$$o_3 = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

ενώ το διάνυσμα  $z_3$  ισούται με τα στοιχεία

$$z_3 = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$$

του πίνακα:

$$T_3^0 = T_1^0 \cdot T_2^1 \cdot T_3^2 = \begin{bmatrix} c_{12}c_3 & -c_{12}s_3 & s_{12} & c_{12}a_3c_3 \\ s_{12}c_3 & -s_{12}s_3 & -c_{12} & s_{12}a_3c_3 \\ s_3 & c_3 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Άρα έχουμε:

$$o_3 = \begin{bmatrix} c_{12}a_3c_3 \\ s_{12}a_3c_3 \\ d_1 \end{bmatrix}$$

και

$$z_3 = \begin{bmatrix} s_{12} \\ -c_{12} \\ 1 \end{bmatrix}$$

Το διάνυσμα  $o_2$  ισούται με τα στοιχεία

$$o_2 = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

ενώ το διάνυσμα  $z_2$  ισούται με τα στοιχεία

$$z_2 = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$$

του πίνακα:

$$T_2^0 = T_1^0 \cdot T_2^1 = \begin{bmatrix} c_{12} & 0 & s_{12} & 0 \\ s_{12} & 0 & -c_{12} & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Άρα έχουμε:

$$o_2 = \begin{bmatrix} 0 \\ 0 \\ d_1 \end{bmatrix}$$

και

$$z_2 = \begin{bmatrix} s_{12} \\ -c_{12} \\ 0 \end{bmatrix}$$

Το διάνυσμα  $o_1$  ισούται με τα στοιχεία

$$o_1 = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

ενώ το διάνυσμα  $z_1$  ισούται με τα στοιχεία

$$z_1 = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$$

του πίνακα:

$$T_1^0 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Άρα έχουμε:

$$o_1 = \begin{bmatrix} 0 \\ 0 \\ d_1 \end{bmatrix}$$

### Κεφάλαιο 3

και

$$z_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Τα διανύσματα  $o_0$  και  $z_0$  αρχικοποιούνται:

$$o_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

και

$$z_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Ο υπολογισμός του εξωτερικού γινομένου δύο διανυσματικών πινάκων γίνεται με τη χρήση της παρακάτω ισότητας:

$$z \ x \ o = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \ x \ \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} b \cdot z - c \cdot y \\ c \cdot x - a \cdot z \\ a \cdot y - b \cdot x \end{bmatrix}$$

Άρα έχουμε:

$$z_0 x(o_5 - o_0) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \ x \ \left( \begin{bmatrix} s_{12}d_5 + c_{12}a_4c_{34} + c_{12}a_3c_3 \\ -c_{12}d_5 + s_{12}a_4c_{34} + s_{12}a_3c_3 \\ a_4s_{34} + a_3s_3 + d_1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \Rightarrow$$

$$\Rightarrow z_0 x(o_5 - o_0) = \begin{bmatrix} c_{12}d_5 - s_{12}a_4c_{34} - s_{12}a_3c_3 \\ s_{12}d_5 + c_{12}a_4c_{34} + c_{12}a_3c_3 \\ 0 \end{bmatrix}$$

$$z_1 x(o_5 - o_1) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \ x \ \left( \begin{bmatrix} s_{12}d_5 + c_{12}a_4c_{34} + c_{12}a_3c_3 \\ -c_{12}d_5 + s_{12}a_4c_{34} + s_{12}a_3c_3 \\ a_4s_{34} + a_3s_3 + d_1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ d_1 \end{bmatrix} \right) \Rightarrow$$

$$\Rightarrow z_1 x(o_5 - o_1) = \begin{bmatrix} c_{12}d_5 - s_{12}a_4c_{34} - s_{12}a_3c_3 \\ s_{12}d_5 + c_{12}a_4c_{34} + c_{12}a_3c_3 \\ a_4s_{34} + a_3s_3 \end{bmatrix}$$

$$z_2 x(o_5 - o_2) = \begin{bmatrix} s_{12} \\ -c_{12} \\ 0 \end{bmatrix} \ x \ \left( \begin{bmatrix} s_{12}d_5 + c_{12}a_4c_{34} + c_{12}a_3c_3 \\ -c_{12}d_5 + s_{12}a_4c_{34} + s_{12}a_3c_3 \\ a_4s_{34} + a_3s_3 + d_1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ d_1 \end{bmatrix} \right) \Rightarrow$$

$$\Rightarrow z_2 x(o_5 - o_2) = \begin{bmatrix} a_4 c_{34} + a_3 c_3 \\ s_{12}(a_4 s_{34} + a_3 s_3) \\ -c_{12}(a_4 s_{34} + a_3 s_3) \end{bmatrix}$$

$$z_3 x(o_5 - o_3) = \begin{bmatrix} s_{12} \\ -c_{12} \\ 1 \end{bmatrix} x \left( \begin{bmatrix} s_{12}d_5 + c_{12}a_4c_{34} + c_{12}a_3c_3 \\ -c_{12}d_5 + s_{12}a_4c_{34} + s_{12}a_3c_3 \\ a_4s_{34} + a_3s_3 + d_1 \end{bmatrix} - \begin{bmatrix} c_{12}a_3c_3 \\ s_{12}a_3c_3 \\ d_1 \end{bmatrix} \right) \Rightarrow$$

$$\Rightarrow z_3 x(o_5 - o_3) = \begin{bmatrix} -c_{12}(a_4s_{34} + a_3s_3) + c_{12}d_5 - s_{12}a_4c_{34} \\ s_{12}d_5 + c_{12}a_4c_{34} - s_{12}(a_4s_{34} + a_3s_3) \\ a_4c_{34} \end{bmatrix}$$

$$z_4 x(o_5 - o_4) = \begin{bmatrix} s_{12} \\ -c_{12} \\ 1 \end{bmatrix} x \left( \begin{bmatrix} s_{12}d_5 + c_{12}a_4c_{34} + c_{12}a_3c_3 \\ -c_{12}d_5 + s_{12}a_4c_{34} + s_{12}a_3c_3 \\ a_4s_{34} + a_3s_3 + d_1 \end{bmatrix} - \begin{bmatrix} c_{12}a_4c_{34} \\ s_{12}a_4c_{34} \\ d_1 \end{bmatrix} \right) \Rightarrow$$

$$\Rightarrow z_4 x(o_5 - o_4) = \begin{bmatrix} -c_{12}(a_4s_{34} + a_3s_3) + c_{12}d_5 - s_{12}a_4c_{34} \\ s_{12}d_5 + s_{12}a_3c_3 - s_{12}(a_4s_{34} + a_3s_3) \\ a_3c_{34} \end{bmatrix}$$

Άρα η Ιακωβιανή ορίζουσα ισούται με:

$$J = \begin{bmatrix} c_{12}d_5 - s_{12}a_4c_{34} - s_{12}a_3c_3 & c_{12}d_5 - s_{12}a_4c_{34} - s_{12}a_3c_3 & a_4c_{34} + a_3c_3 & -c_{12}(a_4s_{34} + a_3s_3) + c_{12}d_5 - s_{12}a_4c_{34} - c_{12}(a_4s_{34} + a_3s_3) + c_{12}d_5 - s_{12}a_3c_3 \\ s_{12}d_5 + c_{12}a_4c_{34} + c_{12}a_3c_3 & s_{12}d_5 + c_{12}a_4c_{34} + c_{12}a_3c_3 & s_{12}(a_4s_{34} + a_3s_3) & s_{12}d_5 + c_{12}a_4c_{34} - s_{12}(a_4s_{34} + a_3s_3) & s_{12}d_5 + s_{12}a_3c_3 - s_{12}(a_4s_{34} + a_3s_3) \\ 0 & a_4s_{34} + a_3s_3 & -c_{12}(a_4s_{34} + a_3s_3) & a_4c_{34} & a_3c_{34} \\ 0 & 0 & s_{12} & s_{12} & s_{12} \\ 0 & 0 & -c_{12} & -c_{12} & -c_{12} \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Από τον παραπάνω Ιακωβιανό πίνακα μπορούμε να ορίσουμε την κινητική ενέργεια κάθε κόμβου του ρομποτικού βραχίονα. Επομένως έχουμε:

**Κόμβος 1:**

$$\dot{x}_1 = (c_{12}d_5 - s_{12}a_4c_{34} - s_{12}a_3c_3) \cdot \dot{\theta}_1$$

$$\dot{y}_1 = (s_{12}d_5 + c_{12}a_4c_{34} + c_{12}a_3c_3) \cdot \dot{\theta}_1$$

$$\dot{z}_1 = 0 \cdot \dot{\theta}_1$$

$$\omega_{x1} = 0 \cdot \dot{\theta}_1$$

$$\omega_{y1} = 0 \cdot \dot{\theta}_1$$

$$\omega_{z1} = \dot{\theta}_1$$

Άρα μπορούμε να ορίσουμε την κινητική ενέργεια του πρώτου κόμβου του ρομποτικού

Λόγω μεταφορικής κίνησης:

$$\begin{aligned} T_1 &= \frac{1}{2} \cdot m \cdot (\dot{x}_1^2 + \dot{y}_1^2 + \dot{z}_1^2) \cdot \dot{\theta}_1^2 \\ &= \frac{1}{2} \cdot m \cdot (d_5^2 + a_4^2 \cdot c_{34}^2 + a_3^2 \cdot c_3^2 + 2a_4c_{34}a_3c_3) \cdot \dot{\theta}_1^2 \end{aligned}$$

Λόγω περιστροφικής κίνησης:

$$T_1 = \frac{1}{2} \cdot I \cdot (\dot{\omega}_{x1}^2 + \dot{\omega}_{y1}^2 + \dot{\omega}_{z1}^2) \cdot \dot{\theta}_1^2 = \frac{1}{2} \cdot I \cdot \dot{\theta}_1^2$$

ενώ η δυναμική ενέργεια (θεωρούμε ότι κάθε στέλεχος έχει μάζα ίση με m) δίνεται:

$$U_1 = m \cdot g \cdot p_z = m \cdot g \cdot (d_1)$$

**Κόμβος 2:**

$$\dot{x}_2 = (c_{12}d_5 - s_{12}a_4c_{34} - s_{12}a_3c_3) \cdot \dot{\theta}_2$$

$$\dot{y}_2 = (s_{12}d_5 + c_{12}a_4c_{34} + c_{12}a_3c_3) \cdot \dot{\theta}_2$$

$$\dot{z}_2 = (a_4s_{34} + a_3s_3) \cdot \dot{\theta}_2$$

$$\omega_{x2} = 0 \cdot \dot{\theta}_2$$

$$\omega_{y2} = 0 \cdot \dot{\theta}_2$$

$$\omega_{z2} = \dot{\theta}_2$$

Άρα μπορούμε να ορίσουμε την κινητική ενέργεια του πρώτου κόμβου του ρομποτικού

Λόγω μεταφορικής κίνησης:

$$T_2 = \frac{1}{2} \cdot m \cdot (\dot{x}_2^2 + \dot{y}_2^2 + \dot{z}_2^2) \cdot \dot{\theta}_2^2 = \frac{1}{2} \cdot m \cdot (2a_4^2 \cdot c_{34}^2 + 2a_3^2 \cdot c_3^2 + 4a_4c_{34}a_3c_3) \cdot \dot{\theta}_2^2$$

Λόγω περιστροφικής κίνησης:

$$T_2 = \frac{1}{2} \cdot I \cdot (\dot{\omega}_{x2}^2 + \dot{\omega}_{y2}^2 + \dot{\omega}_{z2}^2) \cdot \dot{\theta}_2^2 = \frac{1}{2} \cdot I \cdot \dot{\theta}_2^2$$

ενώ η δυναμική ενέργεια δίνεται:

$$U_3 = 2m \cdot g \cdot p_z = 2m \cdot g \cdot (d_1)$$

**Κόμβος 3:**

$$\dot{x}_3 = (a_4 c_{34} + a_3 c_3) \cdot \dot{\theta}_3$$

$$\dot{y}_3 = (s_{12}(a_4 s_{34} + a_3 s_3)) \cdot \dot{\theta}_3$$

$$\dot{z}_3 = (-c_{12}(a_4 s_{34} + a_3 s_3)) \cdot \dot{\theta}_3$$

$$\omega_{x3} = s_{12} \cdot \dot{\theta}_3$$

$$\omega_{y3} = -c_{12} \cdot \dot{\theta}_3$$

$$\omega_{z3} = 0 \cdot \dot{\theta}_3$$

Άρα μπορούμε να ορίσουμε την κινητική ενέργεια του πρώτου κόμβου του ρομποτικού

Λόγω περιστροφικής κίνησης:

$$T_3 = \frac{1}{2} \cdot m \cdot (\dot{x}_3^2 + \dot{y}_3^2 + \dot{z}_3^2) \cdot \dot{\theta}_3^2 = \frac{1}{2} \cdot m \cdot (a_4^2 + a_3^2 + 2a_4 a_3 c_4) \cdot \dot{\theta}_3^2$$

Λόγω περιστροφικής κίνησης:

$$T_3 = \frac{1}{2} \cdot I \cdot (\dot{\omega}_{x3}^2 + \dot{\omega}_{y3}^2 + \dot{\omega}_{z3}^2) \cdot \dot{\theta}_3^2 = \frac{1}{2} \cdot I \cdot \dot{\theta}_3^2$$

ενώ η δυναμική ενέργεια δίνεται:

$$U_3 = 3m \cdot g \cdot p_z = 3m \cdot g \cdot (d_1)$$

**Κόμβος 4:**

$$\dot{x}_4 = (-c_{12}(a_4 s_{34} + a_3 s_3) + c_{12} d_5 - s_{12} a_4 c_{34}) \cdot \dot{\theta}_4$$

$$\dot{y}_4 = (s_{12} d_5 + c_{12} a_4 c_{34} - s_{12}(a_4 s_{34} + a_3 s_3)) \cdot \dot{\theta}_4$$

$$\dot{z}_4 = (a_4 c_{34}) \cdot \dot{\theta}_4$$

$$\omega_{x4} = s_{12} \cdot \dot{\theta}_4$$

$$\omega_{y4} = -c_{12} \cdot \dot{\theta}_4$$

$$\omega_{z4} = \dot{\theta}_4$$

Άρα μπορούμε να ορίσουμε την κινητική ενέργεια του πρώτου κόμβου του ρομποτικού

Λόγω μεταφορικής κίνησης:

$$T_4 = \frac{1}{2} \cdot m \cdot (\dot{x}_4^2 + \dot{y}_4^2 + \dot{z}_4^2) \cdot \dot{\theta}_4^2$$

$$T_4 = \frac{1}{2} \cdot m \cdot (2a_4^2 \cdot c_{34}^2 + 2a_3^2 \cdot c_3^2 + 4a_4c_{34}a_3s_3s_{34} - 2d_5(a_4s_{34} + a_3s_3)) \cdot \dot{\theta}_4^2$$

Λόγω περιστροφικής κίνησης:

$$T_4 = \frac{1}{2} \cdot I \cdot (\dot{\omega}_{x4}^2 + \dot{\omega}_{y4}^2 + \dot{\omega}_{z4}^2) \cdot \dot{\theta}_4^2 = \frac{1}{2} \cdot (I + 1) \cdot \dot{\theta}_4^2$$

ενώ η δυναμική ενέργεια δίνεται:

$$U_4 = 4m \cdot g \cdot p_z = 4m \cdot g \cdot (d_1)$$

**Κόμβος 5:**  $\dot{x}_5 = (-c_{12}(a_4s_{34} + a_3s_3) + c_{12}d_5 - s_{12}a_4c_{34}) \cdot \dot{\theta}_5$

$$\dot{y}_5 = (s_{12}d_5 + c_{12}a_4c_{34} - s_{12}(a_4s_{34} + a_3s_3)) \cdot \dot{\theta}_5$$

$$\dot{z}_5 = (a_3c_{34}) \cdot \dot{\theta}_5$$

$$\omega_{x5} = s_{12} \cdot \dot{\theta}_5$$

$$\omega_{y5} = -c_{12} \cdot \dot{\theta}_5$$

$$\omega_{z5} = \dot{\theta}_5$$

Άρα μπορούμε να ορίσουμε την κινητική ενέργεια του πρώτου κόμβου του ρομποτικού

Λόγω μεταφορικής κίνησης:

$$T_5 = \frac{1}{2} \cdot m \cdot (\dot{x}_5^2 + \dot{y}_5^2 + \dot{z}_5^2) \cdot \dot{\theta}_5^2$$

$$T_5 = \frac{1}{2} \cdot m \cdot ((a_4^2 + a_3^2) \cdot c_{34}^2 + 2a_3^2 \cdot c_3^2 + 4a_4c_{34}a_3s_3s_{34} - 2d_5(a_4s_{34} + a_3s_3)) \cdot \dot{\theta}_5^2$$

Λόγω περιστροφικής κίνησης:

$$T_5 = \frac{1}{2} \cdot I \cdot (\dot{\omega}_{x5}^2 + \dot{\omega}_{y5}^2 + \dot{\omega}_{z5}^2) \cdot \dot{\theta}_5^2 = \frac{1}{2} \cdot (I + 1) \cdot \dot{\theta}_5^2$$

ενώ η δυναμική ενέργεια δίνεται:

$$U_5 = 5m \cdot g \cdot p_z = 5m \cdot g \cdot (a_4s_{34} + a_3s_3 + d_1)$$

### Κεφάλαιο 3

Η συνάρτηση Lagrange ισούται με το άθροισμα της κινητικής και της δυναμικής ενέργειας. Δηλαδή:

$$L = T - U$$

Η εξίσωση Lagrange προκύπτει από τη σχέση:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_i} \right) - \frac{\partial L}{\partial \theta_i} = 0$$

Για κάθε βαθμό ελευθερίας προκύπτει και μία εξίσωση Lagrange, δηλαδή:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} = 0 \Rightarrow m \cdot (d_5^2 + a_4^2 \cdot c_{34}^2 + a_3^2 \cdot c_3^2 + 2a_4 c_{34} a_3 c_3) \cdot \ddot{\theta}_1 + I \cdot \ddot{\theta}_1 = 0$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2} = 0 \Rightarrow m \cdot (2a_4^2 \cdot c_{34}^2 + 2a_3^2 \cdot c_3^2 + 4a_4 c_{34} a_3 c_3) \cdot \ddot{\theta}_2 + I \cdot \ddot{\theta}_2 = 0$$

Μετά από πράξεις οι εξισώσεις Lagrange οδηγούν στις δυναμικές εξισώσεις τύπου Euler-Lagrange του ρομπότ, που δίνονται από:

$$B(\theta_i) \ddot{\theta}_i + C(\theta_i, \dot{\theta}_i) \dot{\theta}_i + G(\theta_i) = 0$$

Όπου

$B(\theta_i)$ : μητρώο αδράνειας.

- Τα διαγώνια στοιχεία του  $b_{ii}(\theta_i)$  αναπαριστούν τη ροπή αδράνειας του άξονα της άρθρωσης  $i$ , στην εκάστοτε θέση του βραχίονα, όταν οι υπόλοιπες αρθρώσεις είναι ακίνητες
- Τα μηδιαγώνια στοιχεία του  $b_{ij}(\theta_i)$  περιγράφουν το αποτέλεσμα της επιτάχυνσης της άρθρωσης  $j$  στην άρθρωση  $i$

$C(\theta_i, \dot{\theta}_i)$ : μητρώο, το οποίο περιέχει τις φυγόκεντρες ροπές και τις ροπές Coriolis που αναπτύσσονται στο βραχίονα. Προκύπτει από το μητρώο της κινητικής ενέργειας μετά από παραγωγίσεις.

$G(\theta_i)$ : διάνυσμα βαρυτικών ορών, εξαρτώνται μόνο από τις μετατοπίσεις των αρθρώσεων του βραχίονα και προέρχονται από την ολική δυναμική ενέργεια. Αναπαριστούν τη ροπή που αναπτύσσεται στον άξονα μίας άρθρωσης στην εκάστοτε θέση του βραχίονα, λόγω βαρύτητας.

Άρα

$$B(\theta) = \begin{bmatrix} b_{11} & 0 & 0 & 0 & 0 \\ 0 & b_{22} & 0 & 0 & 0 \\ 0 & 0 & b_{33} & 0 & 0 \\ 0 & 0 & 0 & b_{44} & 0 \\ 0 & 0 & 0 & 0 & b_{55} \end{bmatrix}$$

$$b_{11} = m \cdot (d_5^2 + a_4^2 \cdot c_{34}^2 + a_3^2 \cdot c_3^2 + 2a_4c_{34}a_3c_3) + I$$

$$b_{22} = m \cdot (2a_4^2 \cdot c_{34}^2 + 2a_3^2 \cdot c_3^2 + 4a_4c_{34}a_3c_3) + I$$

$$b_{33} = m \cdot (a_4^2 + a_3^2 + 2a_4a_3c_4) + I$$

$$b_{44} = m \cdot (2a_4^2 \cdot c_{34}^2 + 2a_3^2 \cdot c_3^2 + 4a_4c_{34}a_3s_3s_{34} - 2d_5(a_4s_{34} + a_3s_3)) + (I + 1)$$

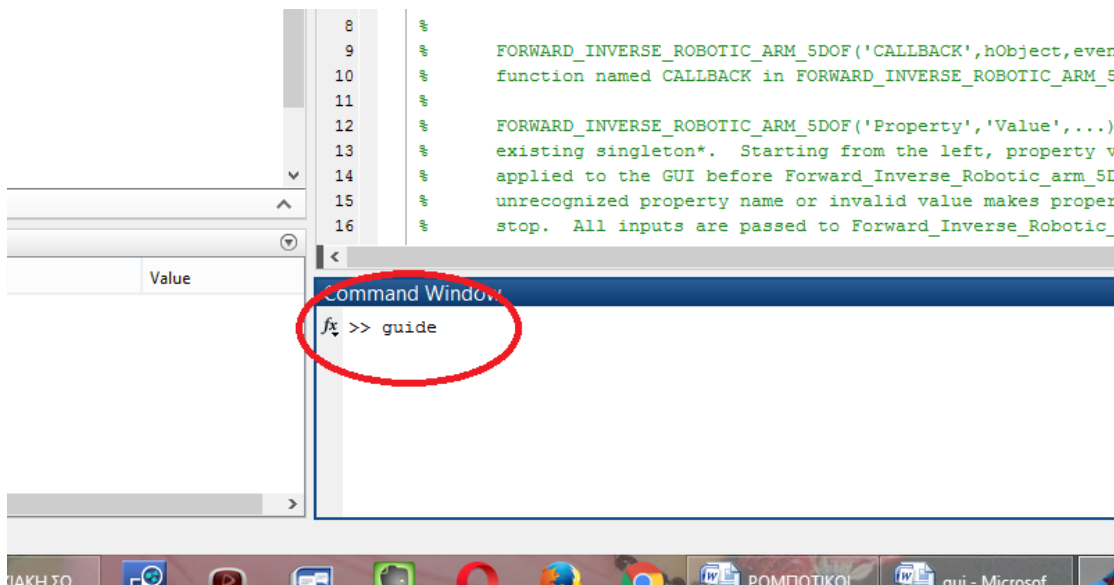
$$b_{55} = m \cdot ((a_4^2 + a_3^2) \cdot c_{34}^2 + 2a_3^2 \cdot c_3^2 + 4a_4c_{34}a_3s_3s_{34} - 2d_5(a_4s_{34} + a_3s_3)) + (I + 1)$$

## ΚΕΦΑΛΑΙΟ 4

### ΡΟΜΠΟΤΙΚΟΣ ΒΡΑΧΙΟΝΑΣ 5 ΒΑΘΜΩΝ ΕΛΕΥΘΕΡΙΑΣ (5 DOF) ΠΡΟΣΟΜΟΙΩΣΗ GUIMATLAB

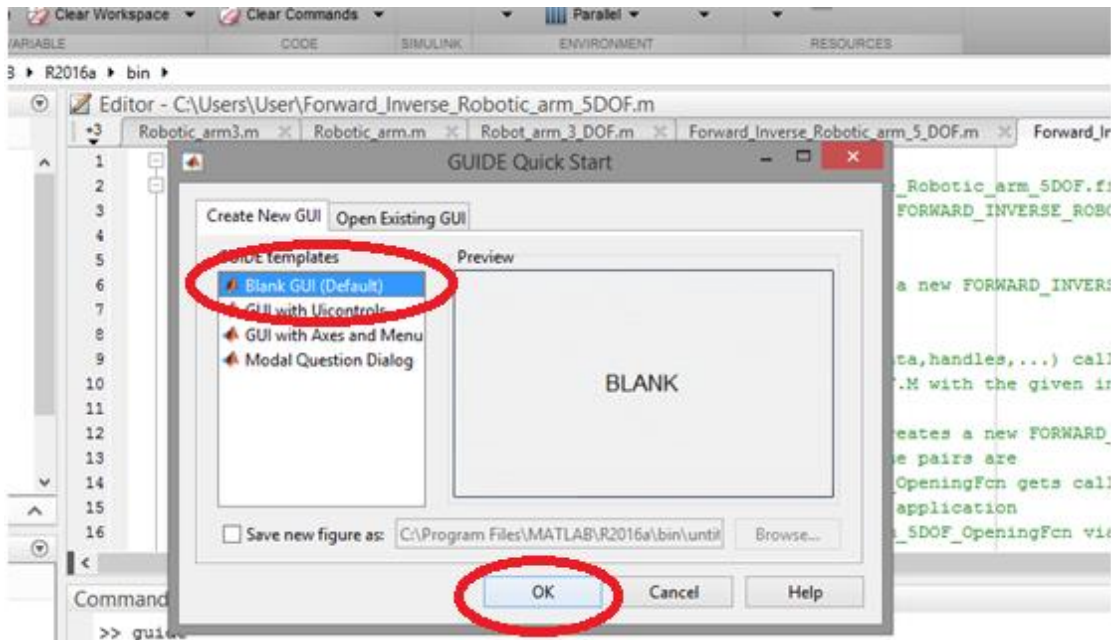
Για την προσομοίωση του ρομποτικού βραχίονα δημιουργήσαμε ένα GUI σε περιβάλλον Matlab. Τα βήματα της προσομοίωσης παρουσιάζονται αναλυτικότερα παρακάτω.

Αρχικά εισερχόμαστε στο περιβάλλον της Matlab και πληκτρολογούμε την εντολή `guide`, όπως φαίνεται στην εικόνα 5.1.

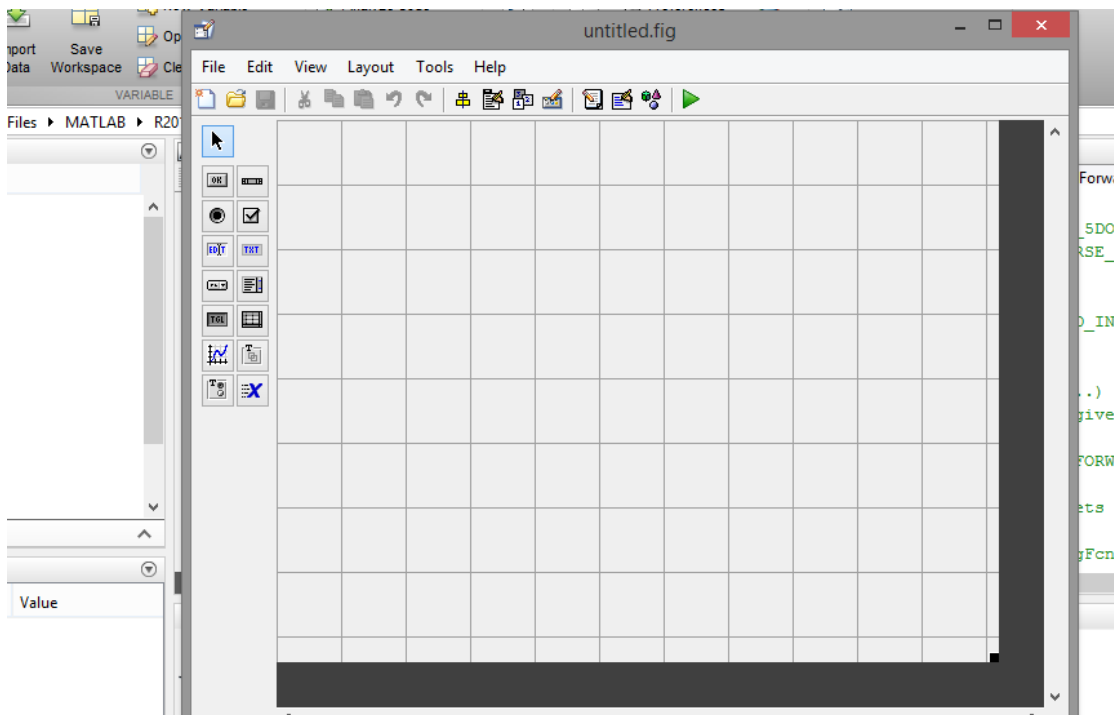


Εικόνα 5.1: Η εντολή `guide`

Με την εντολή αυτή ανοίγει το περιβάλλον στο οποίο θα δημιουργηθεί το GUI. Στην εικόνα 5.2 φαίνεται το πρώτο στάδιο της δημιουργίας του GUI, στο οποίο επιλέγουμε `BlankGUI(Default)`, και μετά `OK`. Έτσι, ανοίγει το παράθυρο της εικόνας 5.3, όπου εκεί θα δημιουργηθεί η προσομοίωση.

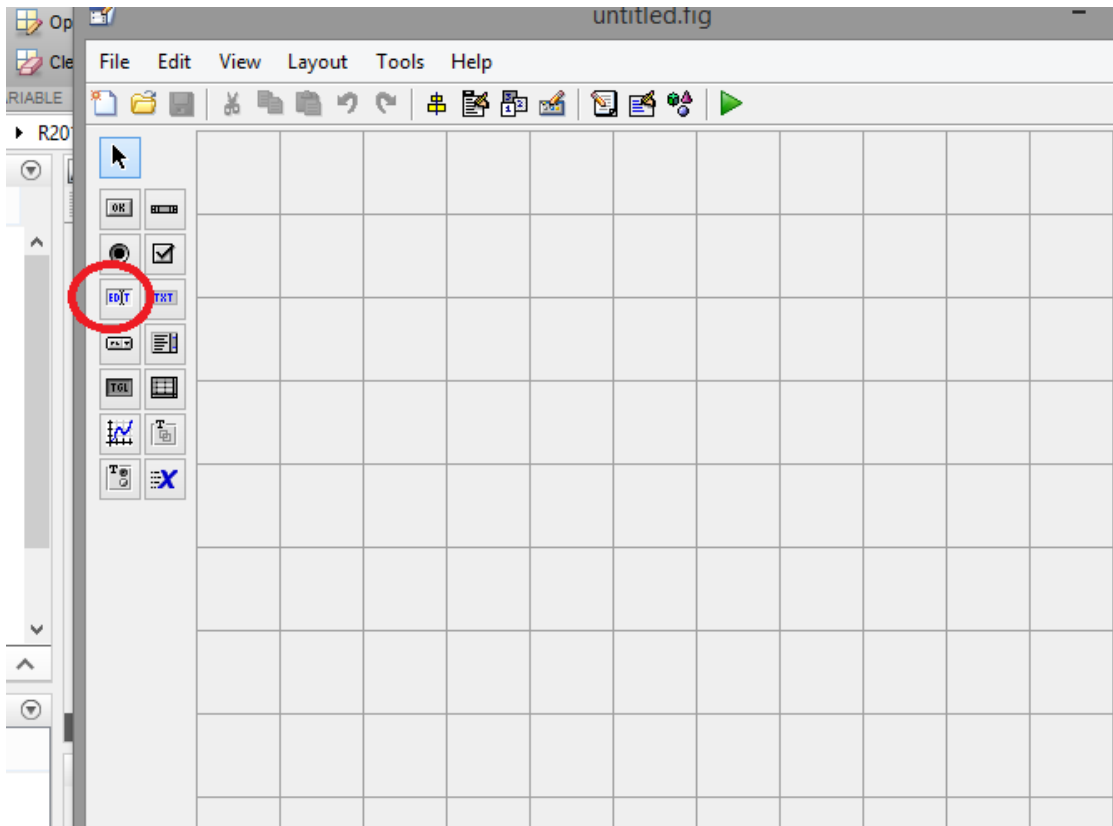


Εικόνα 4.2: Η επιλογή του GUI

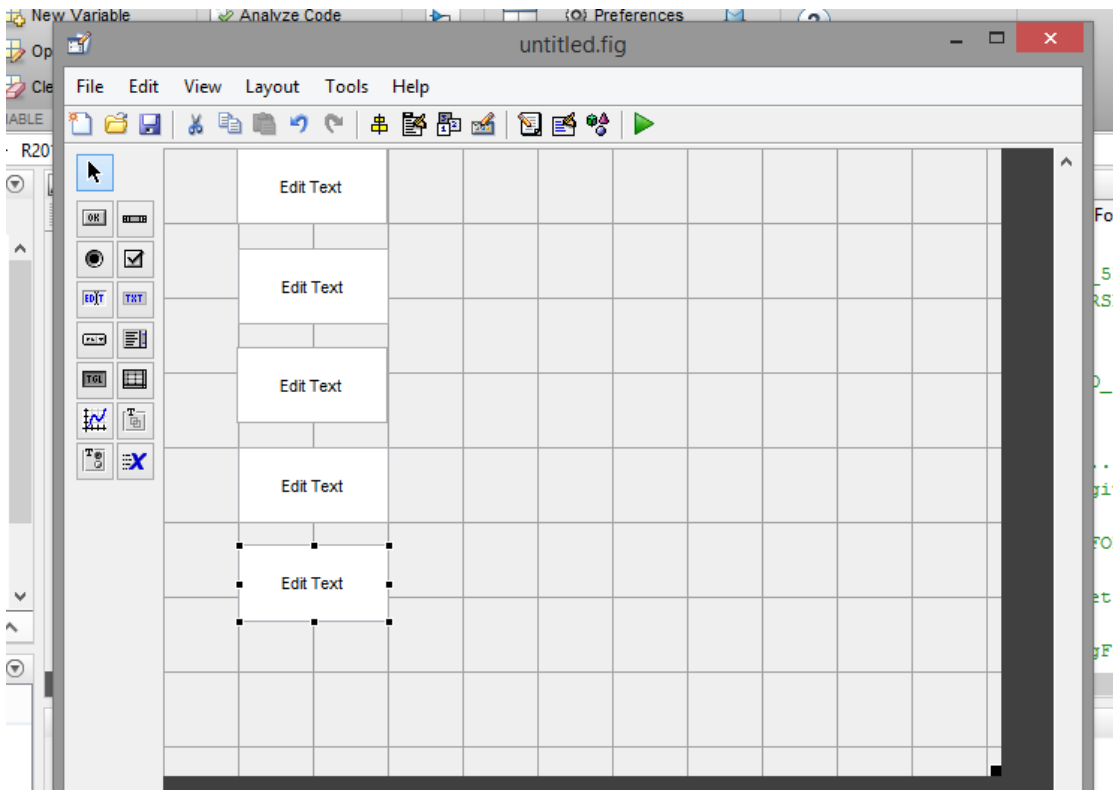


Εικόνα 4.3: Το παράθυρο δημιουργίας του GUI

Για τη δημιουργία του GUI θα πρέπει πρώτα να σχηματίσουμε τα πεδία όπου θα πληκτρολογούμε τις γωνίες  $\theta_i$ . Αυτό γίνεται επιλέγοντας το πλήκτρο που φαίνεται στην εικόνα 4.3. Χρειαζόμαστε πέντε τέτοιες γωνίες αφού ο ρομποτικός βραχίονας είναι 5 βαθμών ελευθερίας. Έτσι προκύπτει το παράθυρο της εικόνας 4.4.



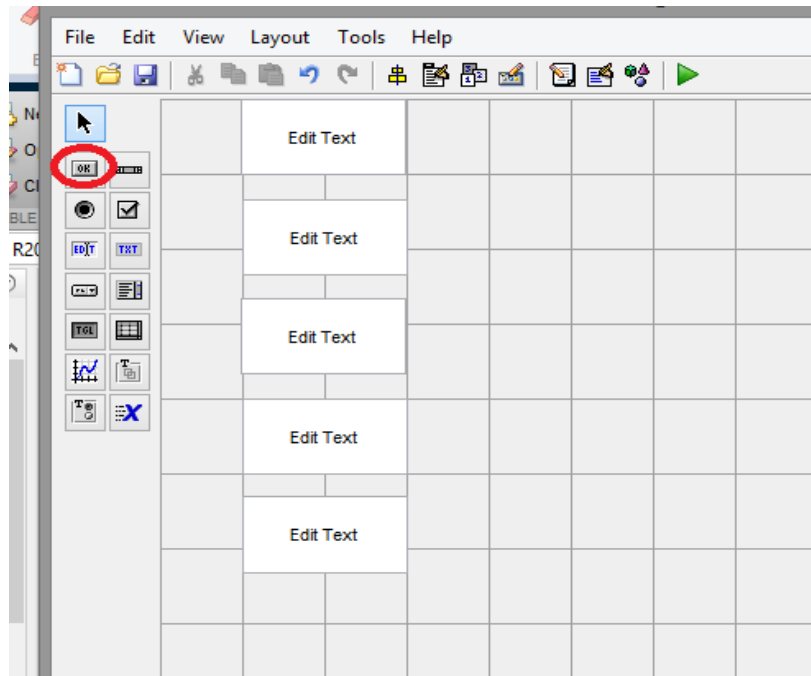
Εικόνα 4.4: Δημιουργία των πεδίων  $\theta_i$



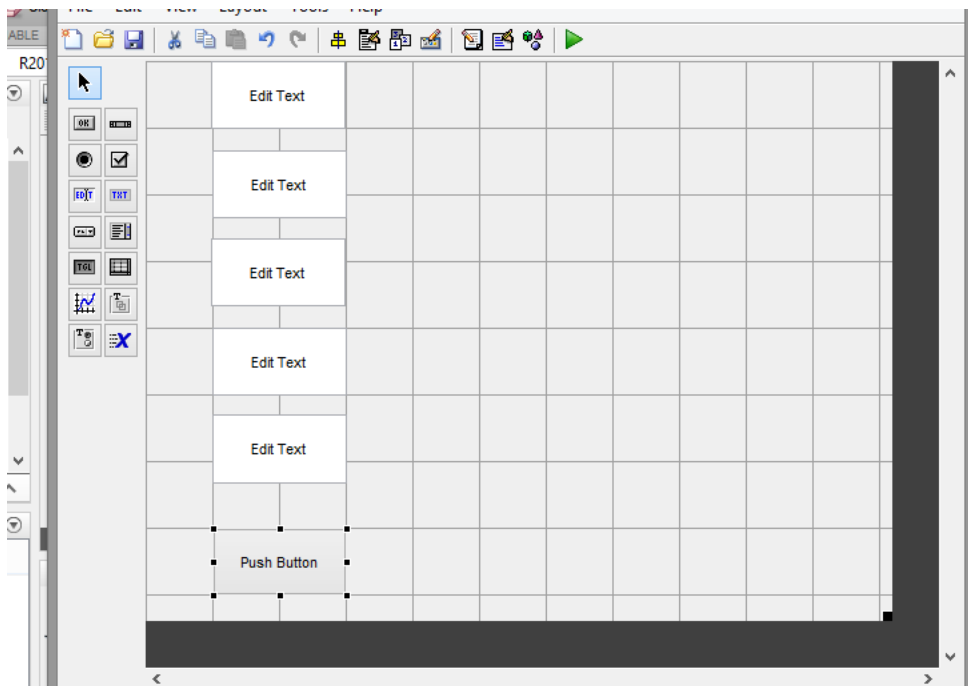
Εικόνα 4.5: Δημιουργία των 5 πεδίων  $\theta_i$

## Κεφάλαιο 4

Το επόμενο βήμα είναι να οριστεί και το πλήκτρο το οποίο με το πάτημά του θα ενεργοποιεί την προσομοίωση. Έτσι, λοιπόν η επιλογή που φαίνεται στην εικόνα 4.6 μας δίνει τη δυνατότητα να δημιουργήσουμε ένα «πλήκτρο». Στην εικόνα 4.7 φαίνεται η δημιουργία αυτού του πλήκτρου.

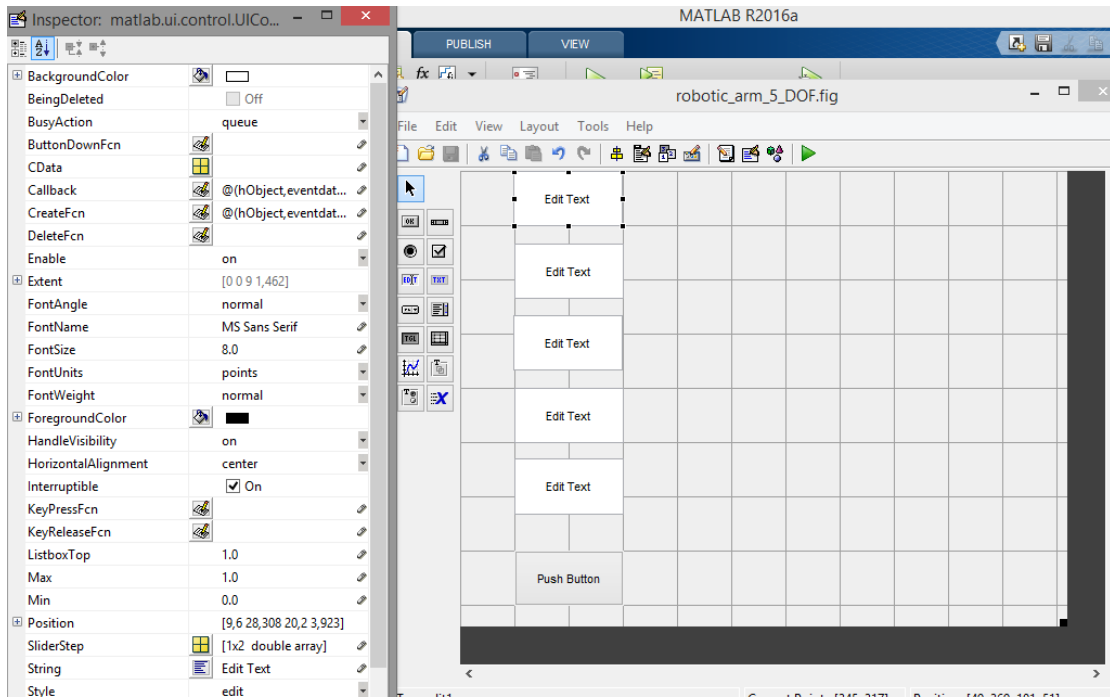


Εικόνα 4.6: Η επιλογή για τη δημιουργία του πλήκτρου



Εικόνα 4.7: Η δημιουργία του πλήκτρου

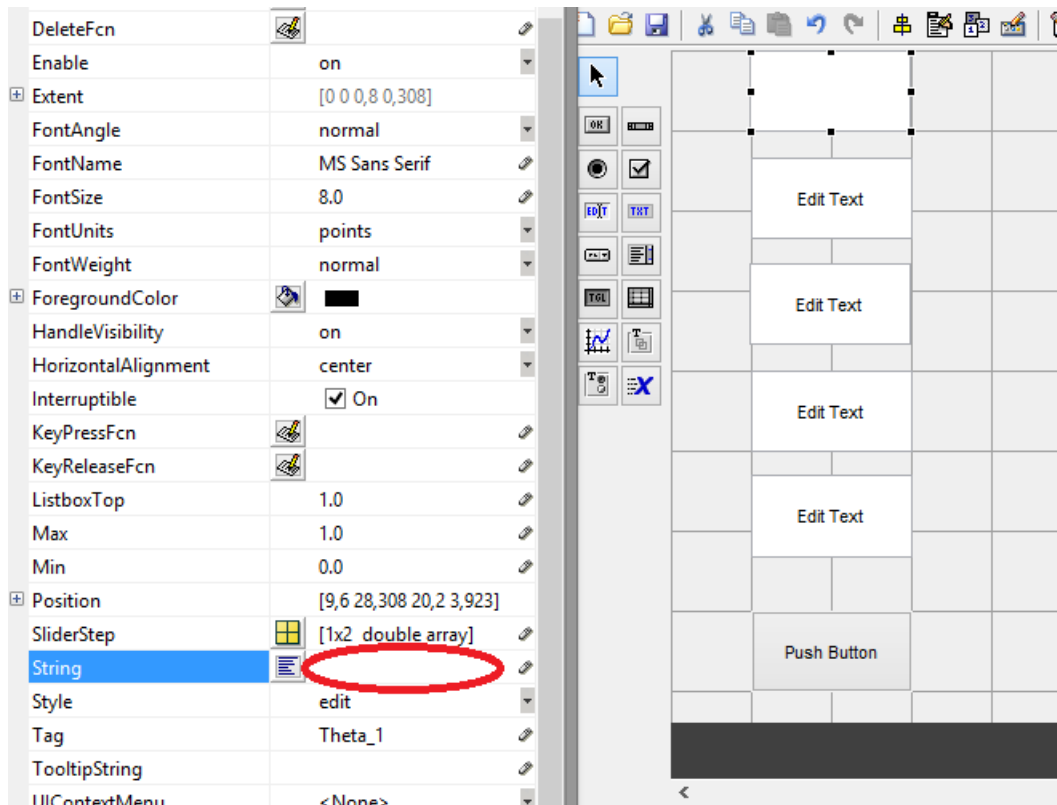
Στη συνέχεια θα πρέπει να ενεργοποιηθούν τόσο τα 5 πεδία ώστε να αναγνωρίζεται η τιμή που θα πληκτρολογείται σε αυτά, όσο και το πλήκτρο το οποίο με το πάτημά του θα ενεργοποιείται η προσομοίωση. Αρχικά με διπλό κλικ πάνω σε ένα από τα πεδία ανοίγει ένα νέο παράθυρο, όπως φαίνεται στην εικόνα 4.8.



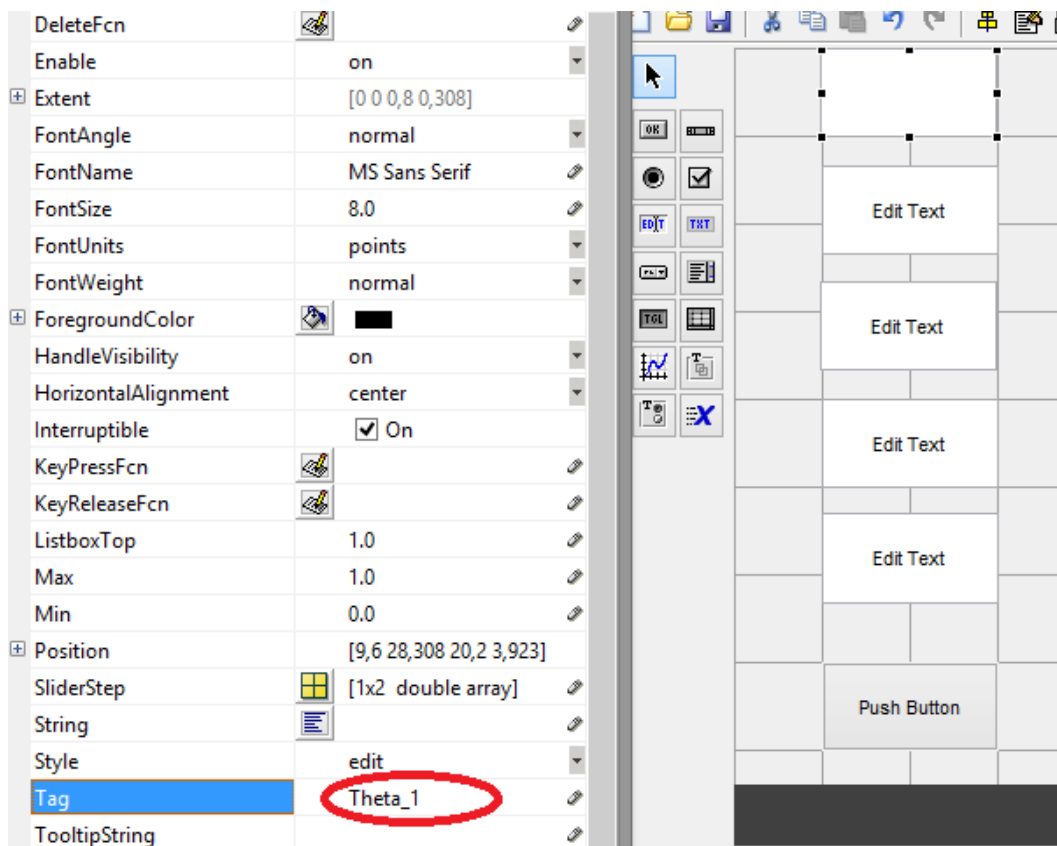
Εικόνα 4.8: Το παράθυρο με το οποίο θα οριστούν οι λειτουργίες των κάθε πεδίων και πλήκτρων του GUI

Στην εικόνα που ακολουθεί (εικόνα 4.9) παρατηρούμε ότι το πεδίο «String» θα πρέπει να είναι κενό, ώστε το αντίστοιχο πεδίο του GUI να μπορεί να δεχτεί την όποια τιμή πληκτρολογήσουμε ώστε να ορίσουμε την τιμή της γωνίας  $\theta_i$ .

Επίσης στο πεδίο «Tag» ορίζεται το όνομα «Theta\_1», το οποίο θα είναι και η μεταβλητή που θα αντιστοιχεί στη γωνία  $\theta_1$ . Αυτή η ενέργεια φαίνεται στην εικόνα 4.10. Αυτό επαναλαμβάνεται και για τα 5 πεδία του GUI ώστε το κάθε ένα να ισοδυναμεί με μία από τις 5 γωνίες  $\theta_i$ .

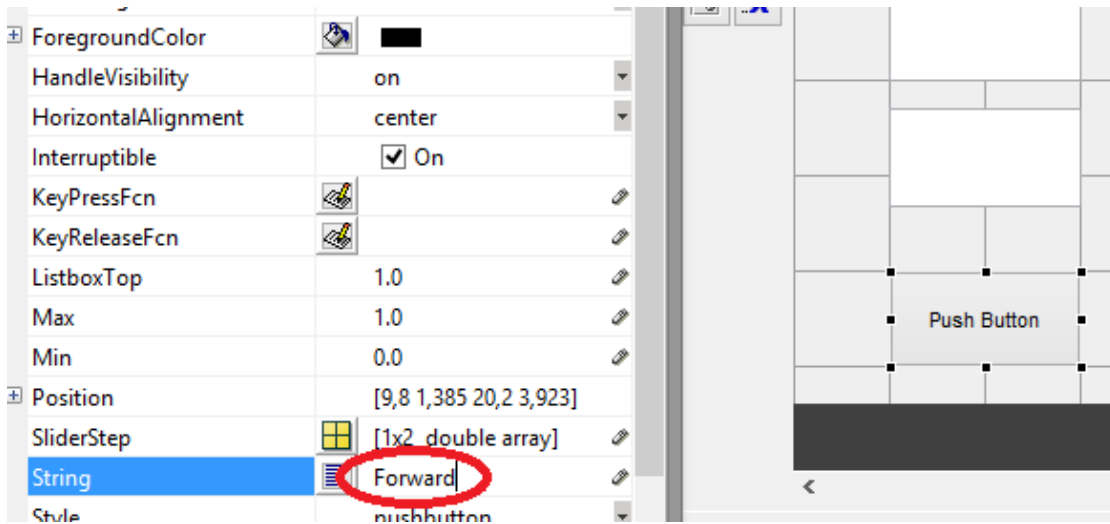


Εικόνα 4.9: Το πεδίο «String» καθαρίζεται

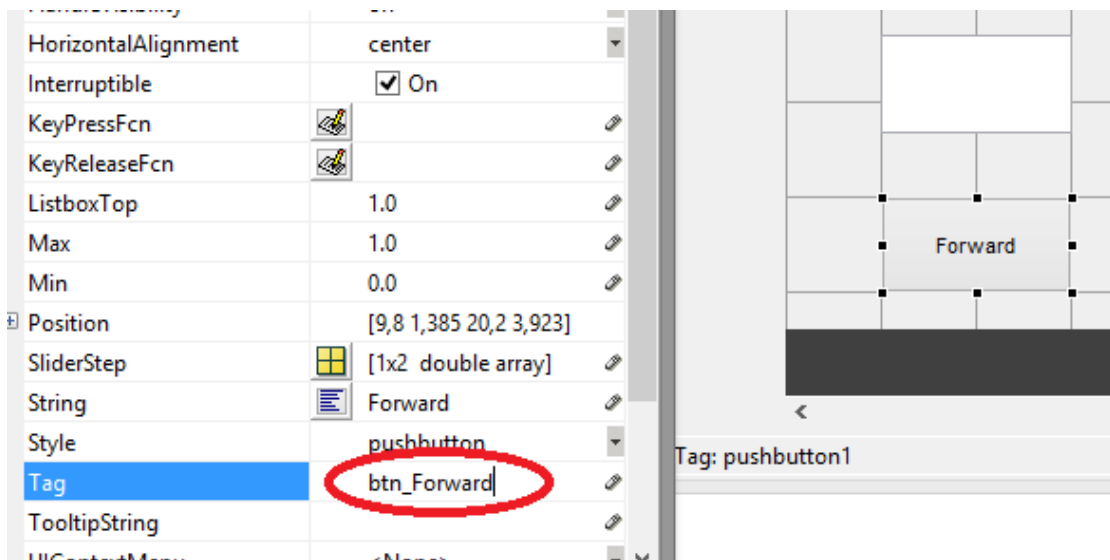


Εικόνα 4.10: Το πεδίο «Tag» ορίζεται ως Theta\_1

Επίσης, με παρόμοιο τρόπο θα πρέπει να ορισθεί και το πεδίο που θα χρησιμοποιηθεί ως πλήκτρο. Έτσι, λοιπόν το πεδίο «String» ορίζεται ως Forward, οπότε το πλήκτρο από PushButton μετονομάζεται σε Forward (εικόνα 4.11). Επίσης, το πεδίο «Tag» ορίζεται ως btn\_Forward. Με τις ενέργειες αυτές το πλήκτρο Forward είναι πλέον ενεργό. Όταν ολοκληρωθεί το GUI και οριστούν οι τιμές των 5 γωνιών  $\theta_i$  τότε με το πάτημα του πλήκτρου Forward θα εμφανιστεί η προσομοίωση του ρομποτικού βραχίονα των 5 βαθμών ελευθερίας.



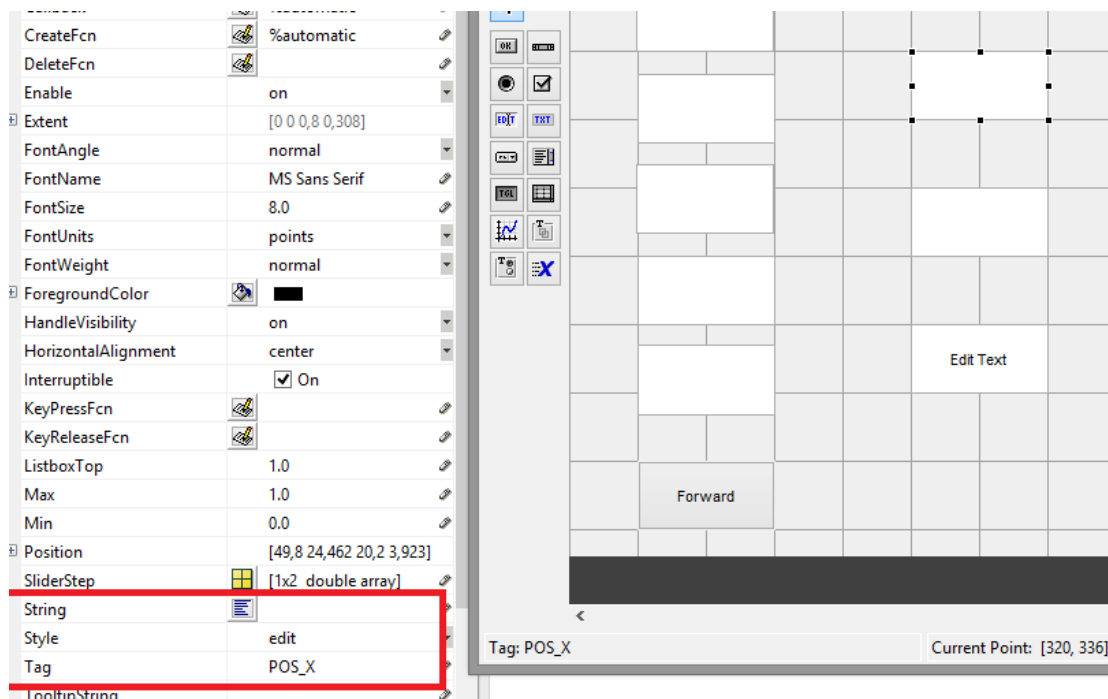
Εικόνα 4.11: Το πεδίο «String» ορίζεται ως Forward



Εικόνα 4.12: Το πεδίο «Tag» ορίζεται ως btn\_Forward

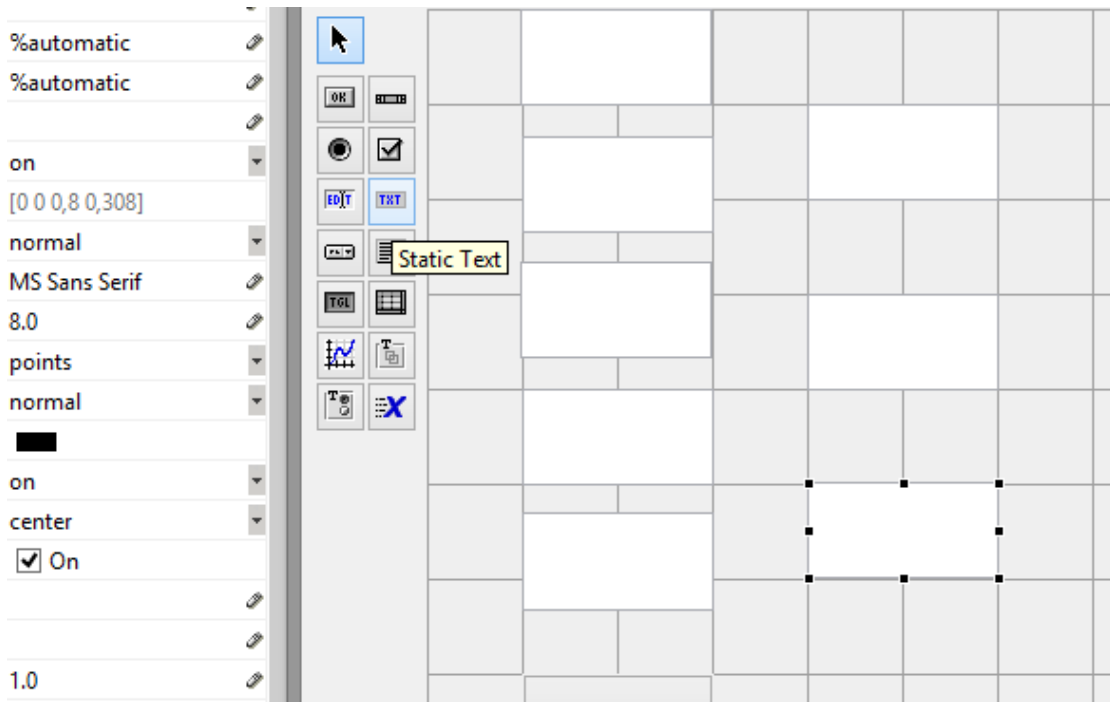
Στη συνέχεια πρέπει να οριστούν και τα τρία πεδία στα οποία θα εμφανιστούν τα αποτελέσματα της προσομοίωσης που είναι οι καρτεσιανές συντεταγμένες της θέσης

του άρπαγα. Στην εικόνα 4.13 παρουσιάζεται ο ορισμός των πεδίων «String» και «Tag».

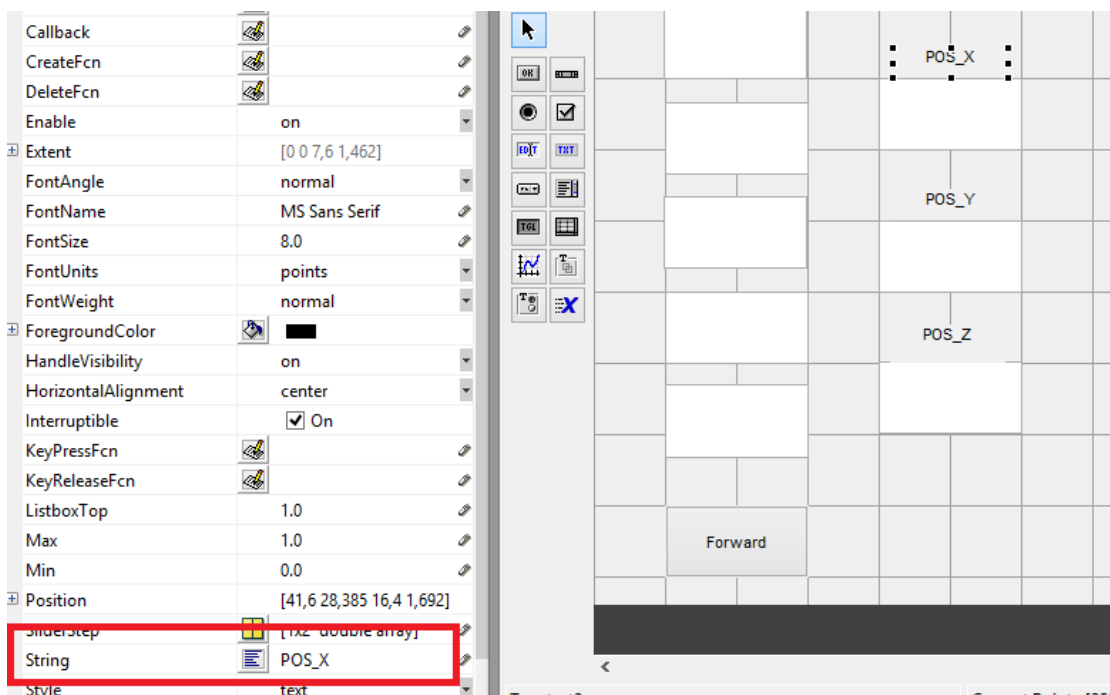


Εικόνα 4.13: Ορισμός των «String» και «Tag»των τριών πεδίων που ορίζουν τις συντεταγμένες του άρπαγα.

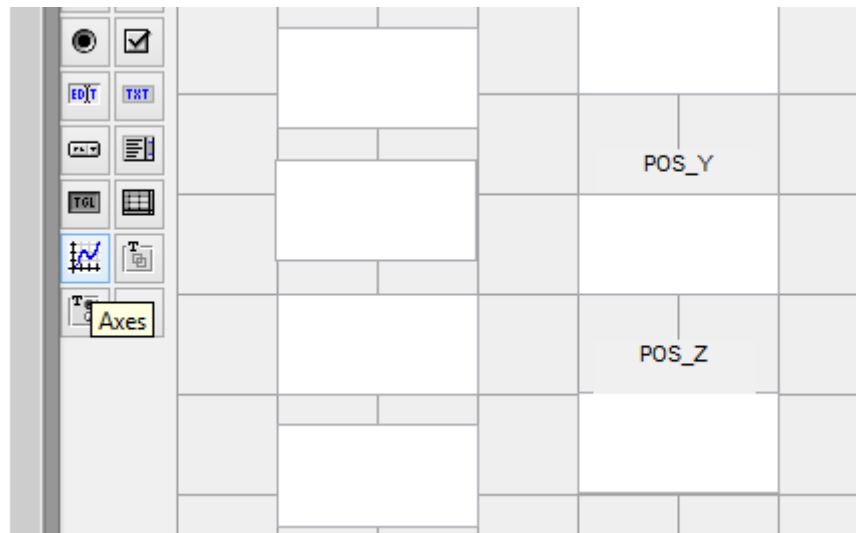
Για να δώσουμε ετικέτες στα πεδία που έχουμε δημιουργήσει στο παράθυρο του GUI επιλέγουμε το εικονίδιο txt, όπως φαίνεται στην εικόνα 4.13. Με την επιλογή αυτή μπορούμε να ορίσουμε ότι τα τρία δεξιά πεδία αντιστοιχούν στις συντεταγμένες της θέσης του άρπαγα. Γι' αυτό και τα ονομάζουμε POS\_X, POS\_Y και POS\_Z αντίστοιχα. Έτσι, το παράθυρο του GUI στο οποίο εργαζόμαστε έχει πάρει τη μορφή που παρουσιάζεται στην εικόνα 4.14. Επιλέγοντας την επιλογή axes, τότε μπορούμε να εμφανίζουμε στο παράθυρο ένα γράφημα τριών διαστάσεων στο οποίο θα εμφανιστεί ο ρομποτικός βραχίονας των 5 βαθμών ελευθερίας (εικόνα 4.16 και εικόνα 4.17).



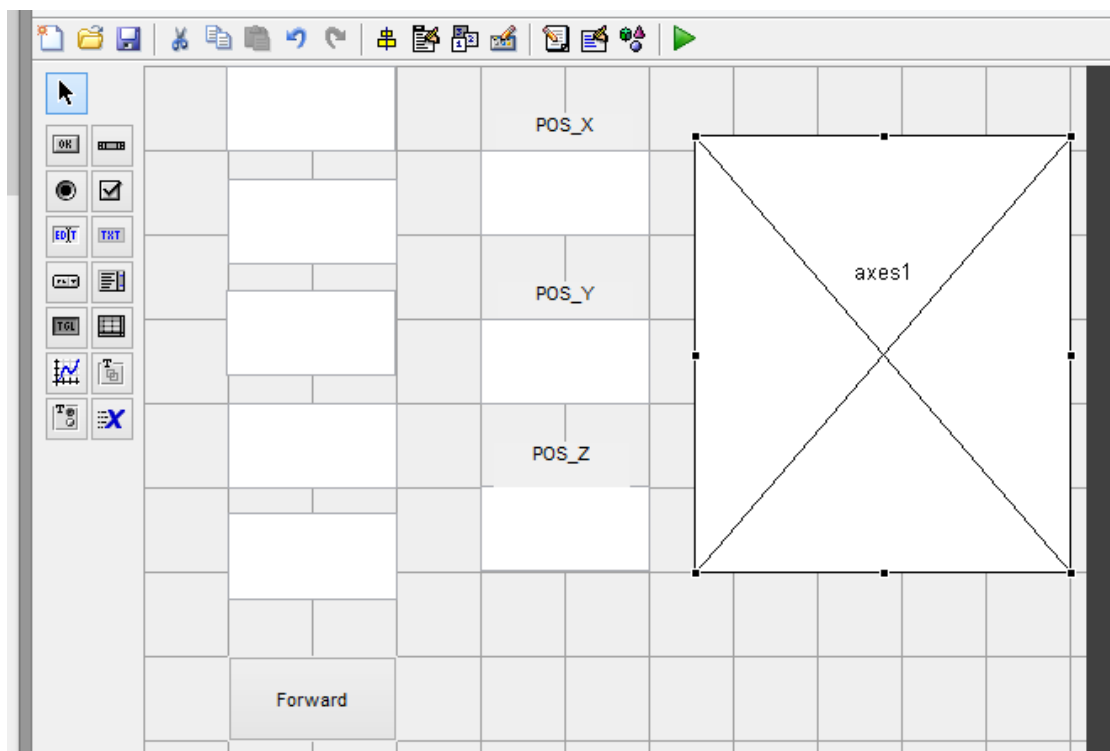
Εικόνα 4.14: Ορισμός ονομάτων στα πεδία που έχουν δημιουργηθεί



Εικόνα 4.15: Η μορφή του παραθύρου του GUI μετά τον ορισμό όλων των πεδίων.



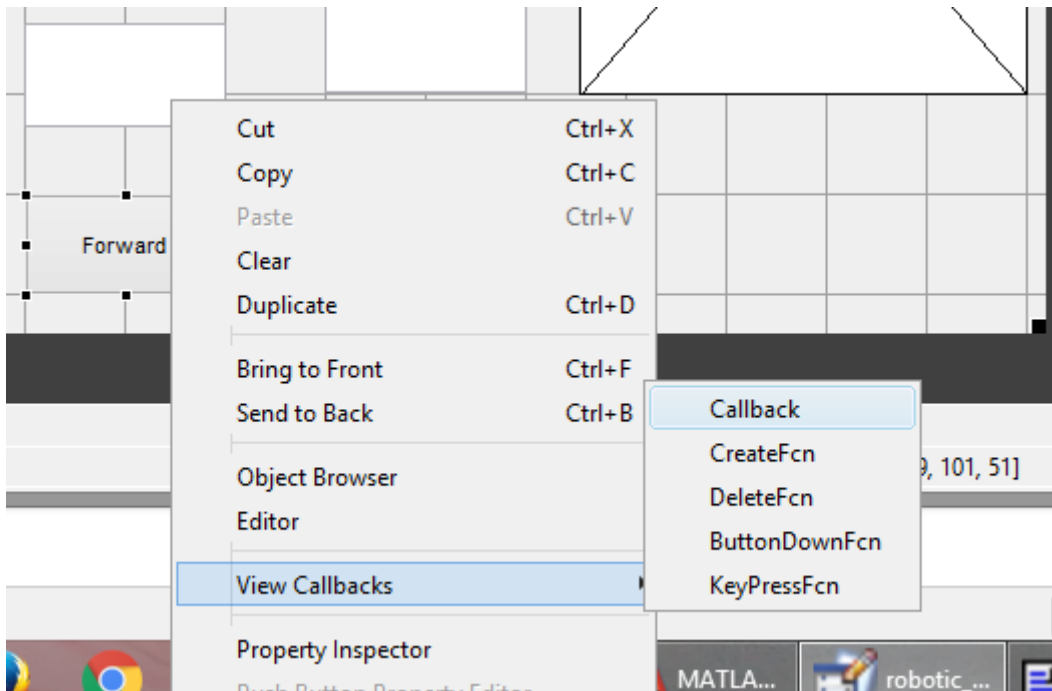
Εικόνα 4.16: Η δημιουργία των αξόνων του τρισδιάστατου γραφήματος.



Εικόνα 4.17: Η μορφή του GUI και με τους άξονες.

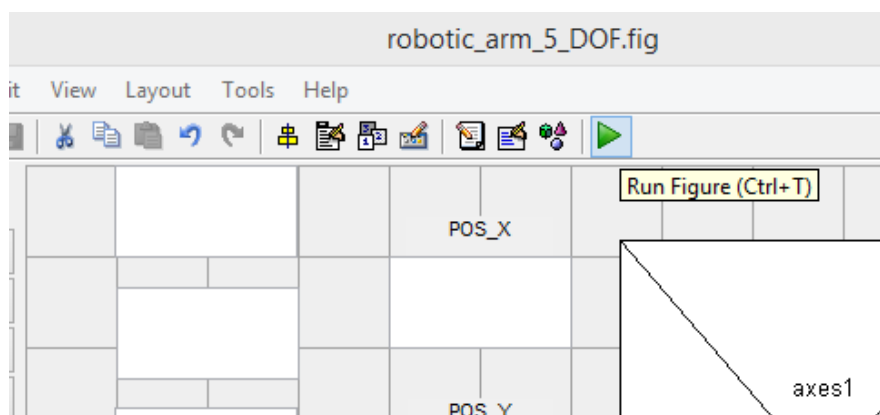
Στη συνέχεια θα πρέπει να αποθηκεύσουμε το GUI που έχουμε δημιουργήσει. Στη συνέχεια θα πρέπει να ορίσουμε την πραγματική λειτουργία του κουμπιού Forward, το οποίο με το που θα πατηθεί θα πρέπει να ενεργοποιήσει την προσομοίωση του ευθέως κινηματικού προβλήματος του ρομποτικού βραχίονα. Επομένως, με δεξί κλικ πάνω στο κουμπί Forward επιλέγουμε ViewCallbacks και μετά επιλέγουμε Callbacks (εικόνα 4.18). Το GUI μας συνδέει με το παράθυρο της Matlab στο οποίο θα πρέπει

να γράψουμε τον κώδικα με τον οποίο το GUI θα μας προσομοιώσει το ρομποτικό βραχίονα που έχουμε σχεδιάσει στο κεφάλαιο 3.



Εικόνα 4.18: Η ενεργοποίηση του πλήκτρου Forward μέσω Matlabκώδικα.

Ο κώδικας σε Matlab για την ενεργοποίηση του πλήκτρου Forward παρουσιάζεται στο Παράρτημα Α, ενώ για ολόκληρο του GUI στο Παράρτημα Β. Στην εικόνα 4.19 επιλέγοντας το Run αρχίζει η προσομοίωση του GUI. Πατώντας το πλήκτρο αυτό εμφανίζεται η τελική μορφή του GUI στο οποίο μπορούμε να ορίσουμε πλέον τις τιμές των γωνιών  $\theta_i$ . Με το πάτημα του πλήκτρου Forward θα ξεκινήσει η προσομοίωση τα αποτελέσματα της οποίας θα παρουσιαστούν στο επόμενο κεφάλαιο. Στην εικόνα 4.20 παρουσιάζεται το GUI της προσομοίωσης.



Εικόνα 4.19: Η έναρξη της προσομοίωσης.

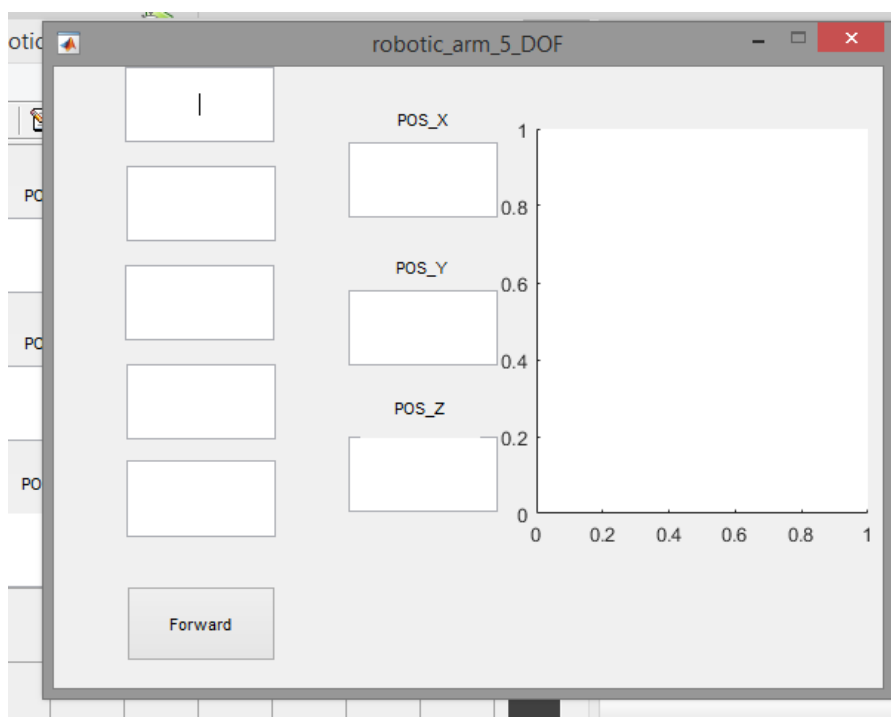


Εικόνα 4.20: Η τελική εικόνα του GUI.

## ΚΕΦΑΛΑΙΟ 5

### ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΣΟΜΟΙΩΣΗΣ

Στην ενότητα αυτή θα παρουσιαστούν τα αποτελέσματα που προέκυψαν τόσο από την προσομοίωση του GUI που παρουσιάστηκε στο προηγούμενο κεφάλαιο, όσο και από την προσομοίωση του ρομποτικού βραχίονα με το πρόγραμμα της C. Να σημειωθεί ότι το πρόγραμμα σε γλώσσα C για την προσομοίωση παρουσιάζεται στο Παράρτημα Γ.



Εικόνα 5.1: Η τελική εικόνα του GUI.

Στην εικόνα 5.1 παρουσιάζεται η τελική μορφή του GUI. Αντίστοιχα, στην εικόνα 5.2 βλέπουμε ότι το GUI μας δίνει τη δυνατότητα να ορίσουμε τις γωνίες  $\theta_i$ . Ορίζουμε λοιπόν, ως γωνίες  $\theta_i$ , όπως και στη θεωρητική μελέτη του ρομποτικού βραχίονα, που αναλύθηκε στο κεφάλαιο 4, τις εξής:

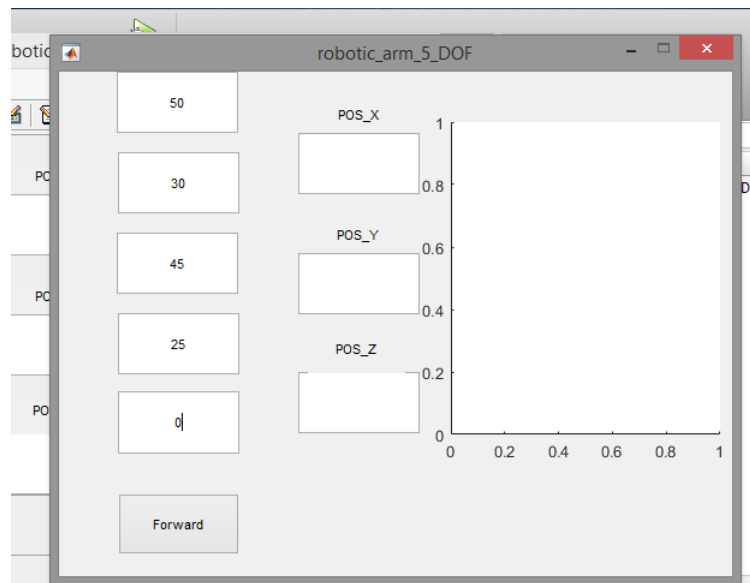
$$\theta_1 = 50^\circ$$

$$\theta_2 = 30^\circ$$

$$\theta_3 = 45^\circ$$

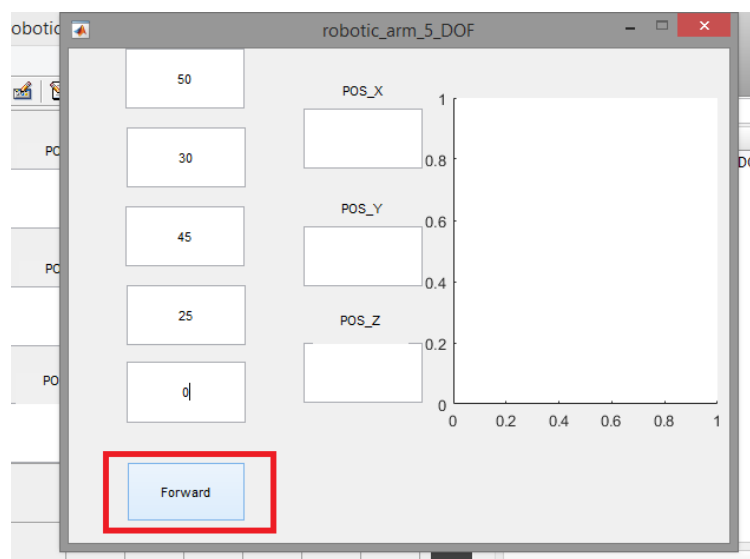
$$\theta_4 = 25^\circ$$

$$\theta_5 = 0^\circ$$

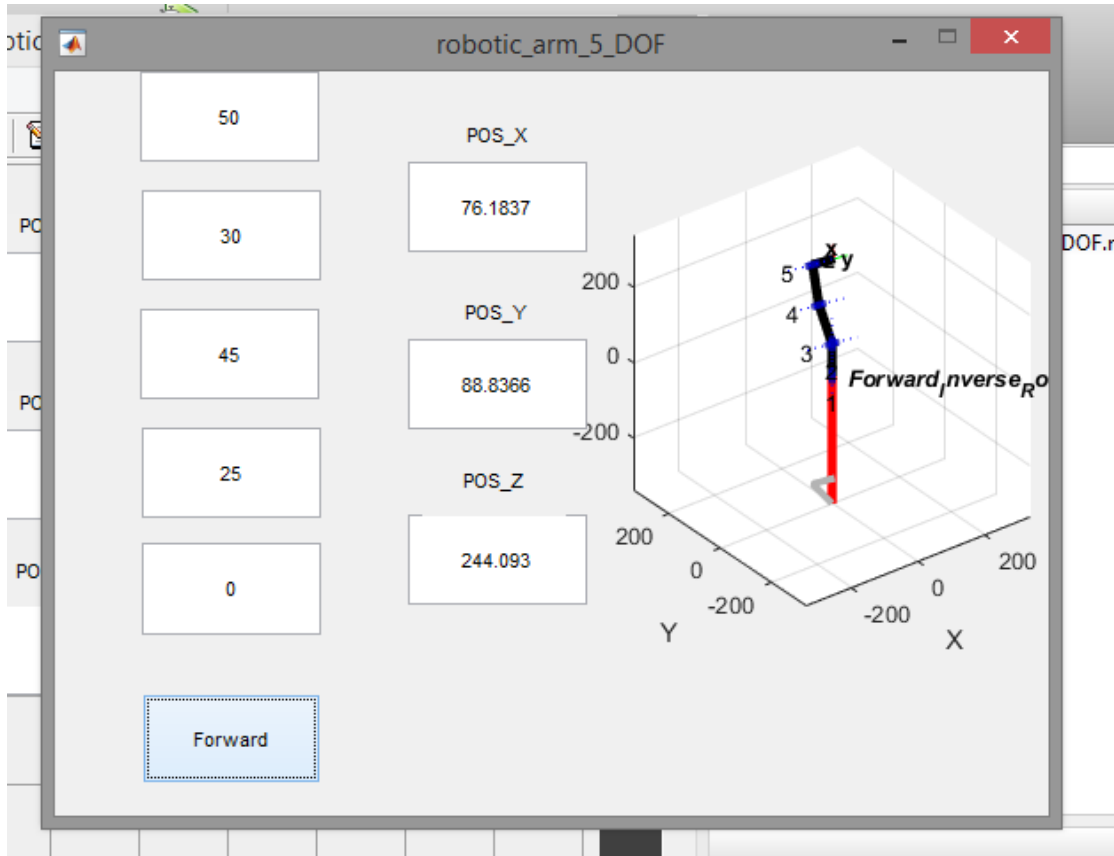


Εικόνα 5.2: Ορισμός των τιμών των γωνιών  $\theta_i$

Πατώντας το πλήκτρο Forward (εικόνα 5.3) αρχίζει η ενεργοποίηση της προσομοίωσης και μετά από μερικά δευτερόλεπτα θα εμφανιστεί ο ρομποτικός βραχίονας των 5 βαθμών ελευθερίας σε τρισδιάστατη μορφή, καθώς και στα πεδία POS\_X, POS\_Y και POS\_Z οι συντεταγμένες θέσης του άρπαγα του βραχίονα. Αυτά τα αποτελέσματα παρουσιάζονται στην εικόνα 5.4 που ακολουθεί.



Εικόνα 5.3: Ενεργοποίηση της προσομοίωσης από το πλήκτρο Forward



Εικόνα 5.4: Ο ρομποτικός βραχίονας των 5 βαθμών ελευθερίας σε 3 διαστάσεις και οι συντεταγμένες θέσης του άρπαγα.

Από τα αποτελέσματα της παραπάνω προσομοίωσης (εικόνα 5.4), παρατηρούμε ότι η θέση του άρπαγα έχει συντεταγμένες:

$$POS\_X = 76.1837$$

$$POS\_Y = 88.8366$$

$$POS\_Z = 244.093$$

Οι παραπάνω τιμές επαληθεύουν τις αντίστοιχες τιμές της θεωρητικής μελέτης του ρομποτικού βραχίονα, μελέτη που έγινε λεπτομερώς στο κεφάλαιο 3.

## ΣΥΜΠΕΡΑΣΜΑΤΑ

Στο προηγούμενο κεφάλαιο παρουσιάστηκαν τα αποτελέσματα της προσομοίωσης του ρομποτικού βραχίονα των 5 βαθμών ελευθερίας που αναλύθηκε στο κεφάλαιο 3. Από τη θεωρητική μελέτη υπολογίσαμε τις εξισώσεις από τις οποίες προκύπτουν οι συντεταγμένες θέσεις του άρπαγα, οι ταχύτητες περιστροφής και μεταφοράς σε κάθε μία από τις 5 αρθρώσεων του καθώς και οι αδράνειες σε κάθε άρθρωση.

Από τη σύγκριση των θεωρητικών και πειραματικών αποτελεσμάτων παρατηρήσαμε ότι δεν υπάρχει μεγάλη απόκλιση. Βέβαια αυτή η μικρή απόκλιση οφείλεται στις στρογγυλοποιήσεις που έγιναν στη θεωρητική προσέγγιση του ρομποτικού βραχίονα.

Ως μία μελλοντική επέκταση τις παρούσας εργασίας είναι η υλοποίηση του ρομποτικού βραχίονα με χρήση της πλατφόρμας Arduino. Βέβαια, ως μία επέκταση της παρούσας εργασίας θα ήταν και η μελέτη του αντίστροφου κινηματικού προβλήματος του ρομποτικού βραχίονα των 5 βαθμών ελευθερίας.

## ΠΑΡΑΡΤΗΜΑ Α

### Ο κώδικας σε Matlab για την ενεργοποίηση του πλήκτρου

#### Forward

```
Th_1 = str2double(handles.Theta_1.String)*pi/180;
Th_2 = str2double(handles.Theta_1.String)*pi/180;
Th_3 = str2double(handles.Theta_2.String)*pi/180;
Th_4 = str2double(handles.Theta_3.String)*pi/180;
Th_5 = str2double(handles.Theta_4.String)*pi/180;

d_1 = 86;
a_3 = 96;
a_4 = 96;
L_5 = 59.6;

L(1) = Link([0 d_1 0 0]);
L(2) = Link([0 0 0 pi/2]);
L(3) = Link([0 0 a_3 0]);
L(4) = Link([0 0 a_4 0]);
L(5) = Link([0 L_5 0 pi/2]);

Robot = SerialLink(L);
Robot.name = 'Forward_Inverse_Robotic_arm_5DOF';
Robot.plot ([Th_1 Th_2 Th_3 Th_4 Th_5]);

T = Robot.fkine([Th_1 Th_2 Th_3 Th_4 Th_5]);

POS_X = T(1,4);
POS_Y = T(2,4);
POS_Z = T(3,4);

set(handles.POS_X, 'string', POS_X);
set(handles.POS_Y, 'string', POS_Y);
set(handles.POS_Z, 'string', POS_Z);
```

## ΠΑΡΑΡΤΗΜΑ Β

### Ο κώδικας σε Matlab για τη δημιουργία του GUI

```

function varargout = robotic_arm_5_DOF(varargin)
% ROBOTIC_ARM_5_DOF MATLAB code for robotic_arm_5_DOF.fig
%   ROBOTIC_ARM_5_DOF, by itself, creates a new ROBOTIC_ARM_5_DOF
or raises the existing
%   singleton*.
%
%   H = ROBOTIC_ARM_5_DOF returns the handle to a new
ROBOTIC_ARM_5_DOF or the handle to
%   the existing singleton*.
%
%   ROBOTIC_ARM_5_DOF('CALLBACK', hObject,eventData,handles,...) calls
the local
%   function named CALLBACK in ROBOTIC_ARM_5_DOF.M with the given
input arguments.
%
%   ROBOTIC_ARM_5_DOF('Property','Value',...) creates a new
ROBOTIC_ARM_5_DOF or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before robotic_arm_5_DOF_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to robotic_arm_5_DOF_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help robotic_arm_5_DOF

% Last Modified by GUIDE v1.5 24-Nov-2018 09:21:57

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
'gui_Singleton', gui_Singleton, ...
'gui_OpeningFcn', @robotic_arm_5_DOF_OpeningFcn, ...
'gui_OutputFcn', @robotic_arm_5_DOF_OutputFcn, ...
'gui_LayoutFcn', [] , ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});

```

```
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before robotic_arm_5_DOF is made visible.
function robotic_arm_5_DOF_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to robotic_arm_5_DOF (see VARARGIN)

% Choose default command line output for robotic_arm_5_DOF
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes robotic_arm_5_DOF wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = robotic_arm_5_DOF_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function Theta_1_Callback(hObject, eventdata, handles)
% hObject    handle to Theta_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Theta_1 as text
%        str2double(get(hObject,'String')) returns contents of Theta_1 as a double

% --- Executes during object creation, after setting all properties.
function Theta_1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Theta_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```

% Hint: edit controls usually have a white background on Windows.
%   See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function Theta_2_Callback(hObject, eventdata, handles)
% hObject   handle to Theta_2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Theta_2 as text
%       str2double(get(hObject,'String')) returns contents of Theta_2 as a double

```

```

% --- Executes during object creation, after setting all properties.
function Theta_2_CreateFcn(hObject, eventdata, handles)
% hObject   handle to Theta_2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
%   See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function Theta_3_Callback(hObject, eventdata, handles)
% hObject   handle to Theta_3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Theta_3 as text
%       str2double(get(hObject,'String')) returns contents of Theta_3 as a double

```

```

% --- Executes during object creation, after setting all properties.
function Theta_3_CreateFcn(hObject, eventdata, handles)
% hObject   handle to Theta_3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.

```

```
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function Theta_4_Callback(hObject, eventdata, handles)
% hObject handle to Theta_4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Theta_4 as text
% str2double(get(hObject,'String')) returns contents of Theta_4 as a double
```

```
% --- Executes during object creation, after setting all properties.
function Theta_4_CreateFcn(hObject, eventdata, handles)
% hObject handle to Theta_4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function Theta_5_Callback(hObject, eventdata, handles)
% hObject handle to Theta_5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Theta_5 as text
% str2double(get(hObject,'String')) returns contents of Theta_5 as a double
```

```
% --- Executes during object creation, after setting all properties.
function Theta_5_CreateFcn(hObject, eventdata, handles)
% hObject handle to Theta_5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in btn_Forward.
function btn_Forward_Callback(hObject, eventdata, handles)
% hObject    handle to btn_Forward (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

Th_1 = str2double(handles.Theta_1.String)*pi/180;
Th_2 = str2double(handles.Theta_1.String)*pi/180;
Th_3 = str2double(handles.Theta_2.String)*pi/180;
Th_4 = str2double(handles.Theta_3.String)*pi/180;
Th_5 = str2double(handles.Theta_4.String)*pi/180;

d_1 = 86;
a_3 = 96;
a_4 = 96;
L_5 = 59.6;

L(1) = Link([0 d_1 0 0]);
L(2) = Link([0 0 0 pi/2]);
L(3) = Link([0 0 a_3 0]);
L(4) = Link([0 0 a_4 0]);
L(5) = Link([0 L_5 0 pi/2]);

Robot = SerialLink(L);
Robot.name = 'Forward_Inverse_Robotic_arm_5DOF';
Robot.plot ([Th_1 Th_2 Th_3 Th_4 Th_5]);

T = Robot.fkine([Th_1 Th_2 Th_3 Th_4 Th_5]);

POS_X = T(1,4);
POS_Y = T(2,4);
POS_Z = T(3,4);

set(handles.POS_X, 'string', POS_X);
set(handles.POS_Y, 'string', POS_Y);
set(handles.POS_Z, 'string', POS_Z);

function POS_X_Callback(hObject, eventdata, handles)
% hObject    handle to POS_X (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of POS_X as text
%       str2double(get(hObject,'String')) returns contents of POS_X as a double

```

```
% --- Executes during object creation, after setting all properties.
function POS_X_CreateFcn(hObject, eventdata, handles)
% hObject handle to POS_X (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function POS_Y_Callback(hObject, eventdata, handles)
% hObject handle to POS_Y (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of POS_Y as text
% str2double(get(hObject,'String')) returns contents of POS_Y as a double
```

```
% --- Executes during object creation, after setting all properties.
function POS_Y_CreateFcn(hObject, eventdata, handles)
% hObject handle to POS_Y (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function POS_Z_Callback(hObject, eventdata, handles)
% hObject handle to POS_Z (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of POS_Z as text
% str2double(get(hObject,'String')) returns contents of POS_Z as a double
```

```
% --- Executes during object creation, after setting all properties.
function POS_Z_CreateFcn(hObject, eventdata, handles)
% hObject    handle to POS_Z (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```

## ΠΑΡΑΡΤΗΜΑ Γ

### Ο κώδικας σε C για την προσομοίωση του ρομποτικού βραχίονα των 5 βαθμών ελευθερίας

```
//FORWARD ROBOTIC ARM 5DOF

#include <stdlib.h>

#include <stdio.h>

#include <math.h>

#define PI 2.14159265

#define m 1

#define I 10

//void function1(void);

unsigned long double function(int i, int j, float theta, int aplha, float a, float d);

int main(void)

{

double val;

val = PI/180.0;

int i, j, k, z;

int alpha[5];

float a[5];

float d[5], theta[5], Px, Py, Pz;

unsigned long float T0[4][4], T1[4][4], T2[4][4], T3[4][4], T4[4][4];

float T_total0[4][4], T_total1[4][4], T_total2[4][4], T_total3[4][4];

float Z0[3][1], Z1[3][1], Z2[3][1], Z3[3][1], Z4[3][1];

float O0[3][1], O1[3][1], O2[3][1], O3[3][1], O4[3][1], O5[3][1];
```

```

float O5_0[3][1], O5_1[3][1], O5_2[3][1], O5_3[3][1], O5_4[3][1];
float Jv0[3][1], Jv1[3][1], Jv2[3][1], Jv3[3][1], Jv4[3][1];
float Jacobian[5][6];
float Inertia[5][5];

FILE *infile_alpha;
FILE *infile_a;
FILE *infile_d;
FILE *infile_theta;

/*---- DHMIOYRGIA TWN TESSARWN PINAKWN ----*/

/*---- PINAKAS ME ONOMA alpha = H GWNIA METAKSY TWN Zi KAI Zi+1
METROYMENH WS PROS TON AKSONA Xi ----*/

/*---- PINAKAS ME ONOMA a = H APOSTASH METAKSY TWN Zi KAI Zi+1
METROYMENH KATA MHKOS TOY AKSONA Xi ----*/

/*---- PINAKAS ME ONOMA d = H APOSTASH METAKSY TWN Xi-1 KAI Xi
METROYMENH KATA MHKOS TOY AKSONA Zi ----*/

/*---- PINAKAS ME ONOMA theta = H GWNIA METAKSY TWN Xi-1 KAI Xi
METROYMENH WS PROS TON AKSONA Zi ----*/

infile_alpha = fopen("alpha.txt","r");
infile_a = fopen("a.txt","r");
infile_d = fopen("d.txt","r");
infile_theta = fopen("theta.txt","r");

/*-----*/
/*-----anagnwsi pinakwn-----*/
/*-----*/

for (i=0;i<5;i++)
{
fscanf(infile_alpha,"%d\n",&alpha[i]);

```

## Παράρτημα Γ

```
fscanf(infile_a,"%f\n",&a[i]);
fscanf(infile_d,"%f\n",&d[i]);
fscanf(infile_theta,"%f\n",&theta[i]);

//printf("alpha[%1d]=%d\n",i,alpha[i]);
//printf("a[%1d]=%d\n",i,a[i]);
//printf("d[%1d]=%0.2f\n",i,d[i]);
//printf("theta[%1d]=%0.2f\n",i,theta[i]);
}
/*-----*/
/*-----end anagnwsi pinakwn-----*/
/*-----*/

/*---- GIA KATHE BATHMO ELETHERIAS (k) YPOLOGIZOYME THN
ANTISTOIXH MHTRA METASXHMATISMOY T ----*/

/*---- PERIORISMOI GIA TH GWNIA theta = H GWNIA METAKSY TWN Xi-1
KAI Xi METROYMENH WS PROS TON AKSONA Zi ----*/

/*---- -360 <= THETA0 <= 360 ----*/
/*---- 30 <= THETA1 <= 120 ----*/
/*---- 30 <= THETA2 <= 210 ----*/
/*---- -45 <= THETA3 <= 45 ----*/
/*---- -45 <= THETA4 <= 45 ----*/

for (k=0;k<5;k++)
{
for (i=0;i<4;i++)
{
for (j=0;j<4;j++)
{
if (k == 0)
{
```

```

if (theta[k] >= -360 && theta[k] <= 360)
{
    T0[i][j] = function(i, j, theta[k], alpha[k], a[k], d[k]);
    printf("T0[%d][%d]=%0.3f\n",i,j,T0[i][j]);
}
else
{
    printf("ERROR!\n");
    printf("theta0 must be between the degrees: -360 <= theta0 <= 360\n");
    exit(1);
}
}

else if (k == 1)
{
    if (theta[k] >= 30 && theta[k] <= 120)
        {
            T1[i][j] = function(i, j, theta[k], alpha[k], a[k], d[k]);
            printf("T1[%d][%d]=%0.3f\n",i,j,T1[i][j]);
        }

    else
    {
        printf("ERROR!\n");
        printf("theta0 must be between the degrees: 30 <= theta2 <= 120\n");
        exit(1);
    }
}

else if (k == 2)
{
    if (theta[k] >= 30 && theta[k] <= 210)

```

```
{
T2[i][j] = function(i, j, theta[k], alpha[k], a[k], d[k]);
printf("T2[%d][%d]=%0.3f\n",i,j,T2[i][j]);
}
else
{
printf("ERROR!\n");
printf("theta0 must be between the degrees: 30 <= theta3 <= 210\n");
exit(1);
}
}
else if (k == 3)
{
if (theta[k] >= -45 && theta[k] <= 45)
{
T3[i][j] = function(i, j, theta[k], alpha[k], a[k], d[k]);
printf("T3[%d][%d]=%0.3f\n",i,j,T3[i][j]);
}
else
{
printf("ERROR!\n");
printf("theta0 must be between the degrees: -45 <= theta4 <= 45\n");
exit(1);
}
}
else if (k == 4)
{
if (theta[k] >= -45 && theta[k] <= 45)
{
```



```

        T_total1[i][j] = 0;
        printf("T_total1[%d][%d]=%0.3f\n",i,j,T_total1[i][j]);
    }
else if (k == 2)
    {
        T_total2[i][j] = 0;
        printf("T_total2[%d][%d]=%0.3f\n",i,j,T_total2[i][j]);
    }
else if (k == 3)
    {
        T_total3[i][j] = 0;
        printf("T_total3[%d][%d]=%0.3f\n",i,j,T_total3[i][j]);
    }
}
}

/*---- ΥΠΟΛΟΓΙΣΜΟΣ ΤΗΣ ΤΕΛΙΚΗΣ ΜΗΤΡΑΣ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ Τ, ΠΟΥ
ΠΡΟΚΥΠΤΕΙ ΑΠΟ ΤΟ ----*/

/*---- ΓΙΝΟΜΕΝΟ ΤΩΝ ΜΗΤΡΩΝ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ ΓΙΑ ΚΑΘΕ
ΒΑΘΜΟ ΕΛΕΥΘΕΡΙΑΣ (k) ----*/

/*---- Η ΤΕΛΙΚΗ ΜΗΤΡΑ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ ΘΑ ΕΧΕΙ ΤΗΝ ΠΑΡΑΚΑΤΩ
ΜΟΡΦΗ ----*/

/*----  ___          ___ ----*/
/*---- |          | ----*/
/*---- | nx          ox          ax          Px | ----*/
/*---- |          | ----*/
/*---- | ny          oy          ay          Py | ----*/
/*---- |          | ----*/
/*---- | nz          oz          az          Pz| ----*/
/*---- |          | ----*/
/*---- | 0          0          0          1| ----*/

```

```

/*---- |                               | ----*/
/*---- --                               -----*/

for (k=0;k<4;k++)
{
    for (i=0;i<4;i++)
    {
        for (j=0;j<4;j++)
        {
            for (z=0;z<4;z++)
            {
                if (k == 0)
                {
                    T_total0[i][j] += T0[i][z]*T1[z][j];
                }
                else if (k == 1)
                {
                    T_total1[i][j] += T_total0[i][z]*T2[z][j];
                }
                else if (k == 2)
                {
                    T_total2[i][j] += T_total1[i][z]*T3[z][j];
                }
                else if (k == 3)
                {
                    T_total3[i][j] += T_total2[i][z]*T4[z][j];
                }
            }
        }
    }
}

```

```
    }

/*---- EMFANISH STHN OTHONH TWN MHTRWN METASXHMATISMOY --*/

    for (k=0;k<4;k++)
    {
        for (i=0;i<4;i++)
        {
            for (j=0;j<4;j++)
            {
                if (k == 0)
                {
printf("T_total0[%d][%d]=%0.3f\n",i,j,T_total0[i][j]);
                }
                else if (k == 1)
                {

printf("T_total1[%d][%d]=%0.3f\n",i,j,T_total1[i][j]);
                }
                else if (k == 2)
                {
printf("T_total2[%d][%d]=%0.3f\n",i,j,T_total2[i][j]);
                }
                else if (k == 3)
                {
printf("T_total3[%d][%d]=%0.3f\n",i,j,T_total3[i][j]);
                }
            }
        }
    }

printf("\n");
```

```

/*---- Η ΤΕΚΙΚΗ ΤΗΣ ΤΟΥ ΑΚΡΟΥ ΤΟΥ ΒΡΑΧΙΟΝΑ Px, Py ΚΑΙ Pz ----*/
    printf("THE FINAL END - EFFECTOR POSITION IS:\n\n");
        Px = T_total3[0][3];
        Py = T_total3[1][3];
        Pz = T_total3[2][3];
        printf("Px=%0.3f\n",Px);
        printf("Py=%0.3f\n",Py);
        printf("Pz=%0.3f\n\n",Pz);

/*-----*/

/*----- ΥΠΟΛΟΓΙΣΜΟΣ ΠΙΝΑΚΩΝ Z0, Z1, Z2, Z3, Z4 ΚΑΙ Z5 -----*/

/*-----*/

/*----- Ο ΠΙΝΑΚΑΣ Z0 ΑΡΧΙΚΟΠΟΙΕΙΤΑΙ ΣΤΗΝ ΤΙΜΗ ΜΗΔΕΝ ΚΑΙ ΤΑ ΤΡΙΑ
STOIXEIA ΤΟΥ -----*/
    for (i=0;i<3;i++)
        {
            for (j=0;j<1;j++)
                {
                    if (i<2)
                        {
                            Z0[i][j] = 0;
                        }
                    else if(i==2)
                        {
                            Z0[i][j] = 1;
                        }
                    printf("Z0[%d][%d]=%0.3f\n",i,j,Z0[i][j]);
                }
        }
}

```

```
/*----- Ο ΠΙΝΑΚΑΣ Z1 ΙΣΟΥΤΑΙ ΜΕ ΤΑ ΤΡΙΑ ΠΡΩΤΑ ΣΤΟΙΧΕΙΑ ΤΗΣ  
ΤΡΙΤΗΣ ΣΤΗΛΗΣ ΤΟΥ ΠΙΝΑΚΑ Τ0 -----*/
```

```
    for (i=0;i<3;i++)  
    {  
        for (j=0;j<1;j++)  
        {  
            Z1[i][j] = T0[i][j+2];  
            printf("Z1[%d][%d]=%0.3f\n",i,j,Z1[i][j]);  
        }  
    }
```

```
/*----- Ο ΠΙΝΑΚΑΣ Z2 ΙΣΟΥΤΑΙ ΜΕ ΤΑ ΤΡΙΑ ΠΡΩΤΑ ΣΤΟΙΧΕΙΑ ΤΗΣ  
ΤΡΙΤΗΣ ΣΤΗΛΗΣ ΤΟΥ ΠΙΝΑΚΑ T_total0 -----*/
```

```
    for (i=0;i<3;i++)  
    {  
        for (j=0;j<1;j++)  
        {  
            Z2[i][j] = T_total0[i][j+2];  
            printf("Z2[%d][%d]=%0.3f\n",i,j,Z2[i][j]);  
        }  
    }
```

```
/*----- Ο ΠΙΝΑΚΑΣ Z3 ΙΣΟΥΤΑΙ ΜΕ ΤΑ ΤΡΙΑ ΠΡΩΤΑ ΣΤΟΙΧΕΙΑ ΤΗΣ  
ΤΡΙΤΗΣ ΣΤΗΛΗΣ ΤΟΥ ΠΙΝΑΚΑ T_total1 -----*/
```

```
    for (i=0;i<3;i++)  
    {  
        for (j=0;j<1;j++)  
        {  
            Z3[i][j] = T_total1[i][j+2];  
            printf("Z3[%d][%d]=%0.3f\n",i,j,Z3[i][j]);  
        }  
    }
```

```

/*----- O PINAKAS Z4 ISOYTAI ME TA TRIA PRWTA STOIXEIA THS
TRITHS STHLHS TOY PINAKA T_total2 -----*/

    for (i=0;i<3;i++)
        {
            for (j=0;j<1;j++)
                {
                    Z4[i][j] = T_total2[i][j+2];
                    printf("Z4[%d][%d]=%0.3f\n",i,j,Z4[i][j]);
                }
        }

/*-----*/

/*----- YPOLOGISMOS PINAKWN O0, O1, O2, O3, O4 KAI O5 -----*/

/*-----*/

/*----- O PINAKAS O0 ARXIKOPOIEITAI STHN TIMH MHDEN KAI TA TRIA
STOIXEIA TOY -----*/

    for (i=0;i<3;i++)
        {
            for (j=0;j<1;j++)
                {
                    O0[i][j] = 0;
                    printf("O0[%d][%d]=%0.3f\n",i,j,O0[i][j]);
                }
        }

/*----- O PINAKAS O1 ISOYTAI ME TA TRIA PRWTA STOIXEIA THS
TETARTH STHLHS TOY PINAKA T0 -----*/

    for (i=0;i<3;i++)
        {
            for (j=0;j<1;j++)
                {
                    O1[i][j] = T0[i][j+3];

```

```
                printf("O1[%d][%d]=%0.3f\n",i,j,O1[i][j]);
            }
        }

/*----- O PINAKAS O2 ISOYTAI ME TA TRIA PRWTA STOIXEIA THS
TETARTHS STHLHS TOY PINAKA T_total0 -----*/

        for (i=0;i<3;i++)
        {
            for (j=0;j<1;j++)
            {
                O2[i][j] = T_total0[i][j+3];
                printf("O2[%d][%d]=%0.3f\n",i,j,O2[i][j]);
            }
        }

/*----- O PINAKAS O3 ISOYTAI ME TA TRIA PRWTA STOIXEIA THS
TETARTHS STHLHS TOY PINAKA T_total1 -----*/

        for (i=0;i<3;i++)
        {
            for (j=0;j<1;j++)
            {
                O3[i][j] = T_total1[i][j+3];
                printf("O3[%d][%d]=%0.3f\n",i,j,O3[i][j]);
            }
        }

/*----- O PINAKAS O4 ISOYTAI ME TA TRIA PRWTA STOIXEIA THS
TETARTHS STHLHS TOY PINAKA T_total2 -----*/

        for (i=0;i<3;i++)
        {
            for (j=0;j<1;j++)
            {
                O4[i][j] = T_total2[i][j+3];
                printf("O4[%d][%d]=%0.3f\n",i,j,O4[i][j]);
```

```

    }
}

/*----- Ο ΠΙΝΑΚΑΣ Ο5 ΙΣΟΥΤΑΙ ΜΕ ΤΑ ΤΡΙΑ ΠΡΩΤΑ ΣΤΟΙΧΕΙΑ ΤΗΣ
ΤΕΤΑΡΤΗΣ ΣΤΗΛΗΣ ΤΟΥ ΠΙΝΑΚΑ T_total3 -----*/

    for (i=0;i<3;i++)
    {
        for (j=0;j<1;j++)
        {
            O5[i][j] = T_total3[i][j+3];
            printf("O5[%d][%d]=%0.3f\n",i,j,O5[i][j]);
        }
    }

/*-----*/

/*----- ΥΠΟΛΟΓΙΣΜΟΣ ΤΗΣ JACOBIAN MATRIX -----*/

/*----- ΥΠΟΛΟΓΙΣΜΟΣ ΤΩΝ ΔΙΑΝΥΣΜΑΤΩΝ (On - Oi) -----*/

/*----- (O5 - O0) -----*/

    for (i=0;i<3;i++)
    {
        for (j=0;j<1;j++)
        {
            O5_0[i][j] = O5[i][j] - O0[i][j];
            printf("O5_0[%d][%d]=%0.3f\n",i,j,O5_0[i][j]);
        }
    }

/*----- (O5 - O1) -----*/

    for (i=0;i<3;i++)
    {
        for (j=0;j<1;j++)

```

## Παράρτημα Γ

```

        {
            O5_1[i][j] = O5[i][j] - O1[i][j];
            printf("O5_1[%d][%d]=%0.3f\n",i,j,O5_1[i][j]);
        }
    }
/*----- (O5 - O2) -----*/
    for (i=0;i<3;i++)
    {
        for (j=0;j<1;j++)
        {
            O5_2[i][j] = O5[i][j] - O2[i][j];
            printf("O5_2[%d][%d]=%0.3f\n",i,j,O5_2[i][j]);
        }
    }
/*----- (O5 - O3) -----*/
    for (i=0;i<3;i++)
    {
        for (j=0;j<1;j++)
        {
            O5_3[i][j] = O5[i][j] - O3[i][j];
            printf("O5_3[%d][%d]=%0.3f\n",i,j,O5_3[i][j]);
        }
    }
/*----- (O5 - O4) -----*/
    for (i=0;i<3;i++)
    {
        for (j=0;j<1;j++)
        {
            O5_4[i][j] = O5[i][j] - O4[i][j];

```

```

        printf("O5_4[%d][%d]=%0.3f\n",i,j,O5_4[i][j]);
    }
}
/*--- YPOLOGISMOS TWN EKSWTERIKWN DIANYSMATWN Zix(On - Oi) ---*/
/*----- Z0x(O5 - O0) -----*/
    for (i=0;i<3;i++)
    {
        for (j=0;j<1;j++)
        {
            if(i == 0)
                {
                    Jv0[i][j] = Z0[i+1][j]*O5_0[i+2][j] - Z0[i+2][j]*O5_0[i+1][j];
                    printf("Jv0[%d][%d]=%0.3f\n",i,j,Jv0[i][j]);
                }
            else if(i == 1)
            {
                Jv0[i][j] = Z0[i+1][j]*O5_0[i-1][j] - Z0[i-1][j]*O5_0[i+1][j];
                printf("Jv0[%d][%d]=%0.3f\n",i,j,Jv0[i][j]);
            }
            else if(i == 2)
            {
                Jv0[i][j] = Z0[i-2][j]*O5_0[i-1][j] - Z0[i-1][j]*O5_0[i-2][j];
                printf("Jv0[%d][%d]=%0.3f\n",i,j,Jv0[i][j]);
            }
        }
    }
/*----- Z1x(O5 - O1) -----*/

```

```

for (i=0;i<3;i++)
    {
        for (j=0;j<1;j++)
            {
                if(i == 0)
                    {

Jv1[i][j] = Z1[i+1][j]*O5_1[i+2][j] - Z1[i+2][j]*O5_1[i+1][j];
printf("Jv1[%d][%d]=%0.3f\n",i,j,Jv1[i][j]);

                    }

                if(i == 1)
                    {
Jv1[i][j] = Z1[i+1][j]*O5_1[i-1][j] - Z1[i-1][j]*O5_1[i+1][j];
printf("Jv1[%d][%d]=%0.3f\n",i,j,Jv1[i][j]);

                    }

                if(i == 2)
                    {

Jv1[i][j] = Z1[i-2][j]*O5_1[i-1][j] - Z1[i-1][j]*O5_1[i-2][j];
printf("Jv1[%d][%d]=%0.3f\n",i,j,Jv1[i][j]);

                    }

            }
        }
}

/*----- Z2x(O5 - O2) -----*/
for (i=0;i<3;i++)
    {
        for (j=0;j<1;j++)
            {
                if(i == 0)
                    {

```

```

Jv2[i][j] = Z2[i+1][j]*O5_2[i+2][j] - Z2[i+2][j]*O5_2[i+1][j];
printf("Jv2[%d][%d]=%0.3f\n",i,j,Jv2[i][j]);
    }
    if(i == 1)
    {
Jv2[i][j] = Z2[i+1][j]*O5_2[i-1][j] - Z2[i-1][j]*O5_2[i+1][j];
printf("Jv2[%d][%d]=%0.3f\n",i,j,Jv2[i][j]);
    }
    if(i == 2)
    {
Jv2[i][j] = Z2[i-2][j]*O5_2[i-1][j] - Z2[i-1][j]*O5_2[i-2][j];
printf("Jv2[%d][%d]=%0.3f\n",i,j,Jv2[i][j]);
    }
    }
}
/*----- Z3x(O5 - O3) -----*/
for (i=0;i<3;i++)
    {
        for (j=0;j<1;j++)
        {
            if(i == 0)
            {
Jv3[i][j] = Z3[i+1][j]*O5_3[i+2][j] - Z3[i+2][j]*O5_3[i+1][j];
printf("Jv3[%d][%d]=%0.3f\n",i,j,Jv3[i][j]);
            }
            if(i == 1)
            {
Jv3[i][j] = Z3[i+1][j]*O5_3[i-1][j] - Z3[i-1][j]*O5_3[i+1][j];
printf("Jv3[%d][%d]=%0.3f\n",i,j,Jv3[i][j]);
            }
        }
    }

```

```

        }
        if(i == 2)
        {
            Jv3[i][j] = Z3[i-2][j]*O5_3[i-1][j] - Z3[i-1][j]*O5_3[i-2][j];
            printf("Jv3[%d][%d]=%0.3f\n",i,j,Jv3[i][j]);
        }
    }
}
/*----- Z4x(O5 - O4) -----*/
for (i=0;i<3;i++)
{
    for (j=0;j<1;j++)
    {
        if(i == 0)
        {
            Jv4[i][j] = Z4[i+1][j]*O5_4[i+2][j] - Z4[i+2][j]*O5_4[i+1][j];
            printf("Jv4[%d][%d]=%0.3f\n",i,j,Jv4[i][j]);
        }
        if(i == 1)
        {
            Jv4[i][j] = Z4[i+1][j]*O5_4[i-1][j] - Z4[i-1][j]*O5_4[i+1][j];
            printf("Jv4[%d][%d]=%0.3f\n",i,j,Jv4[i][j]);
        }
        if(i == 2)
        {
            Jv4[i][j] = Z4[i-2][j]*O5_4[i-1][j] - Z4[i-1][j]*O5_4[i-2][j];
            printf("Jv4[%d][%d]=%0.3f\n",i,j,Jv4[i][j]);
        }
    }
}

```

```
    }  
for (i=0;i<6;i++)  
    {  
        for (j=0;j<5;j++)  
            {  
                if(i == 0 && j == 0)  
                    {  
                        Jacobian[i][j] = Jv0[0][0];  
                    }  
                else if(i == 0 && j == 1)  
                    {  
                        Jacobian[i][j] = Jv1[0][0];  
                    }  
                else if(i == 0 && j == 2)  
                    {  
                        Jacobian[i][j] = Jv2[0][0];  
                    }  
                else if(i == 0 && j == 3)  
                    {  
                        Jacobian[i][j] = Jv3[0][0];  
                    }  
                else if(i == 0 && j == 4)  
                    {  
                        Jacobian[i][j] = Jv4[0][0];  
                    }  
                else if(i == 1 && j == 0)  
                    {  
                        Jacobian[i][j] = Jv0[1][0];  
                    }  
            }  
    }  
}
```

```
else if(i == 1 && j == 1)
    {
        Jacobian[i][j] = Jv1[1][0];
    }
else if(i == 1 && j == 2)
    {
        Jacobian[i][j] = Jv2[1][0];
    }
else if(i == 1 && j == 3)
    {
        Jacobian[i][j] = Jv3[1][0];
    }
else if(i == 1 && j == 4)
    {
        Jacobian[i][j] = Jv4[1][0];
    }
else if(i == 2 && j == 0)
    {
        Jacobian[i][j] = Jv0[2][0];
    }
else if(i == 2 && j == 1)
    {
        Jacobian[i][j] = Jv1[2][0];
    }
else if(i == 2 && j == 2)
    {
        Jacobian[i][j] = Jv2[2][0];
    }
else if(i == 2 && j == 3)
```

```
        {  
            Jacobian[i][j] = Jv3[2][0];  
        }  
else if(i == 2 && j == 4)  
        {  
            Jacobian[i][j] = Jv4[2][0];  
        }  
else if(i == 3 && j == 0)  
        {  
            Jacobian[i][j] = Z0[0][0];  
        }  
else if(i == 3 && j == 1)  
        {  
            Jacobian[i][j] = Z1[0][0];  
        }  
else if(i == 3 && j == 2)  
        {  
            Jacobian[i][j] = Z2[0][0];  
        }  
else if(i == 3 && j == 3)  
        {  
            Jacobian[i][j] = Z3[0][0];  
        }  
else if(i == 3 && j == 4)  
        {  
            Jacobian[i][j] = Z4[0][0];  
        }  
else if(i == 4 && j == 0)  
        {
```

```
        Jacobian[i][j] = Z0[1][0];
    }
else if(i == 4 && j == 1)
    {
        Jacobian[i][j] = Z1[1][0];
    }
else if(i == 4 && j == 2)
    {
        Jacobian[i][j] = Z2[1][0];
    }
else if(i == 4 && j == 3)
    {
        Jacobian[i][j] = Z3[1][0];
    }
else if(i == 4 && j == 4)
    {
        Jacobian[i][j] = Z4[1][0];
    }
else if(i == 5 && j == 0)
    {
        Jacobian[i][j] = Z0[2][0];
    }
else if(i == 5 && j == 1)
    {
        Jacobian[i][j] = Z1[2][0];
    }
else if(i == 5 && j == 2)
    {
        Jacobian[i][j] = Z2[2][0];
    }
```

```

        }
    else if(i == 5 && j == 3)
        {
            Jacobian[i][j] = Z3[2][0];
        }
    else if(i == 5 && j == 4)
        {
            Jacobian[i][j] = Z4[2][0];
        }
    }
}

/*----- EMFANISH STHN O8ONH TOY JACOBIAN MATRIX -----*/
    for (i=0;i<6;i++)
        {
            for (j=0;j<5;j++)
                {
                    printf("Jacobian[%d][%d]=%0.3f\n",i,j,Jacobian[i][j]);
                }
        }

/*----- YPOLOGISMOS TOY MATRIX THS INERTIA -----*/

    for (i=0;i<5;i++)
        {
            for (j=0;j<5;j++)
                {
                    if (i!=j)
                        {
                            Inertia[i][j] = 0;
                        }
                }
        }

```

```

else if (i==0 && j==0)
{
Inertia[i][j] = m*(d[4]*d[4]
+a[3]*a[3]*cos((theta[3]+theta[2])*val)*cos((theta[3]+theta[2])*val) +
a[2]*a[2]*cos(theta[2]*val)*cos(theta[2]*val) +
2*a[3]*a[2]*cos((theta[3]+theta[2])*val)*cos(theta[2]*val)) + I;
}
else if (i==1 && j==1)
{
Inertia[i][j] =
m*(2*a[3]*a[3]*cos((theta[3]+theta[2])*val)*cos((theta[3]+theta[2])*val) +
2*a[2]*a[2]*cos(theta[2]*val)*cos(theta[2]*val) +
4*a[3]*a[2]*cos((theta[3]+theta[2])*val)*cos(theta[2]*val)) + I;
}
else if (i==2 && j==2)
{
Inertia[i][j] = m*(2*a[3]*a[3] + a[2]*a[2] + 2*a[3]*a[2]*cos(theta[3]*val)) + I;
}
else if (i==3 && j==3)
{
Inertia[i][j] =
m*(2*a[3]*a[3]*cos((theta[3]+theta[2])*val)*cos((theta[3]+theta[2])*val) +
2*a[2]*a[2]*cos(theta[2]*val)*cos(theta[2]*val) +
4*a[3]*a[2]*cos((theta[3]+theta[2])*val)*sin((theta[3]+theta[2])*val)*sin(theta[2]*va
l) - 2*d[4]*(a[3]*sin((theta[3]+theta[2])*val) + a[2]*sin(theta[2]*val))) + (I+1);
}
else if (i==4 && j==4)
{
Inertia[i][j] = m*(2*(a[3]*a[3] +
a[2]*a[2])*cos((theta[3]+theta[2])*val)*cos((theta[3]+theta[2])*val) +
2*a[2]*a[2]*cos(theta[2]*val)*cos(theta[2]*val) +
4*a[3]*a[2]*cos((theta[3]+theta[2])*val)*sin((theta[3]+theta[2])*val)*sin(theta[2]*va
l) - 2*d[4]*(a[3]*sin((theta[3]+theta[2])*val) + a[2]*sin(theta[2]*val))) + (I+1);
}
printf("Inertia[%d][%d]=%0.3f\n",i,j,Inertia[i][j]);

```

```

        }
    }
    return(0);
}

/*---- DHMIOYRGIA SYNARTHSHS ME THN OPOIOA DHMIOYRGEITAI H
MHTRA METASXHMATISMOY T ----*/
unsigned long double function(int i, int j, float theta, int alpha, float a, float d)
{
    float F;
    double val;
    val = PI/180.0;
    if (i == 0 && j == 0)
    {
        F = cos(theta*val);
    }
    else if (i == 0 && j == 1)
    {
        F = (-1)*sin(theta*val)*cos(alpha*val);
    }
    else if (i == 0 && j == 2)
    {
        F = sin(theta*val)*sin(alpha*val);
    }
    else if (i == 0 && j == 3)
    {
        F = a*cos(theta*val);
    }
    else if (i == 1 && j == 0)
    {

```

```
        F = sin(theta*val);
    }
else if (i == 1 && j == 1)
    {
        F = cos(theta*val)*cos(alpha*val);
    }
else if (i == 1 && j == 2)
    {
        F = (-1)*cos(theta*val)*sin(alpha*val);
    }
else if (i == 1 && j == 3)
    {
        F = a*sin(theta*val);
    }
else if (i == 2 && j == 0)
    {
        F = 0;
    }
else if (i == 2 && j == 1)
    {
        F = sin(alpha*val);
    }
else if (i == 2 && j == 2)
    {
        F = cos(alpha*val);
    }
else if (i == 2 && j == 3)
    {
        F = d;
```

```
    }  
else if (i == 3 && j == 0)  
    {  
        F = 0;  
    }  
else if (i == 3 && j == 1)  
    {  
        F = 0;  
    }  
else if (i == 3 && j == 2)  
    {  
        F = 0;  
    }  
else if (i == 3 && j == 3)  
    {  
        F = 1;  
    }  
return(F);  
}
```

## ΑΝΑΦΟΡΕΣ

- [1] Aristidou, A and Lasenby, J Inverse kinematics: a review of existing techniques and introduction of a new fast iterative solver. Technical Report. Cambridge University Engineering Department(2009).
- [2] Balestrino, G. De Maria, and L. Sciavicco, Robust control of robotic manipulators, in Proceedings of the 9th IFAC World Congress, Vol. 5, pp. 2435–2440 (1984).
- [3] Barrientos, A., Peñin, L.F., Balaguer, C., and Aracil, R., *Fundamentals of Robotics*, McGrawHill, 1997.
- [4] Bruno Siciliano, Oussama Khatib, Springer Handbook of Robotics. Springer-Verlag Berlin Heidelberg (2008).
- [5] C. A. Mondragon Sanchez, Dynamics and Motion of a Six Degree of Freedom Robot Manipulator, University of Saskatchewan, December 2012.
- [6] C. Pedreira, J.Martinez, R.Q.Quiroga, Neural prostheses: linking brain signals to prosthetic devices. In: Proceedings on the ICROSSICE.
- [7] C. W. Wampler, Manipulator inverse kinematic solutions based on vector formulations and damped least squares methods, IEEE Transactions on Systems, Man, and Cybernetics, pp. 93–101 (1986).
- [8] D. E. Whitney, Resolved motion rate control of manipulators and human prostheses, IEEE Transactions on Man-Machine Systems, pp. 47–53 (1969).
- [9] D'Souza, S. Vijayakumar, S. Schaal, Learning inverse kinematics, in Proc. IEEE IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, pp. 298–303 (2001).
- [10] E. Scheme, K. Englehart, Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use. J. Rehab. Res. Develop., vol. 48, no. 6, pp. 643-659 (2011).
- [11] I.S. Barmanov, D.A. Bizyanova, O.D. Riman, Energy calculation of manipulator electromechanical actuators. Samara university, Dynamics and Vibroacoustics of Machines, Samara, Russia, June (2016).
- [12] J. Zhao and N. I. Badler, Inverse kinematics positioning using nonlinear programming for highly articulated figures, ACM Transactions on Graphics, pp. 313–336 (1994).
- [13] J.R. Wolpaw, E.W. Wolpaw, Brain-Computer Interfaces: Something New Under the Sun., Oxford University Press. (2012).

- [14] JOHN GRAIN, ΕΙΣΑΓΩΓΗ ΣΤΗ ΡΟΜΠΟΤΙΚΗ, Μηχανική και Αυτόματος έλεγχος, Εκδόσεις Τζιόλα, 2008
- [15] Klafter, R.D., Chmielewski, T.A., Negin, M., *Robotic Engineering –An integral Approach*, Prentice Hall, 1989.
- [16] L.-C. T. Wang, C. C. Chen, A combined optimization method for solving the inverse kinematics problem of mechanical manipulators, *IEEE Transactions on Robotics and Automation*, pp. 489–499 (1991).
- [17] M. Atzori, H. Müller, Control capabilities of myoelectric robotic prostheses by hand amputees: a scientific research and market overview. Information Systems Institute, University of Applied Sciences Western Switzerland, Sierre, Switzerland(2015).
- [18] McKerrow, P.J., *Introduction to Robotics*, Addison-Wesley, 1991.
- [19] Niku, S.B., *Introduction to Robotics: Analysis, Systems, Applications*, Prentice Hall, 2001.
- [20] R. Grzeszczuk and D. Terzopoulos, Automated learning of muscleactuated locomotion through control abstraction, in *Proc. ACM SIGGRAPH'95*, ACM Press, pp. 63–70, New York (1995).
- [21] S.L. Zenkevich, A.S. Ushenko, *Basics of Robotic Manipulators Control*, MGTU of N. E. Bauman publishing house, 2004 (in Russian).
- [22] V. Deshpande, P.M. George, Kinematic modeling and analysis of 5 DOF robotic arm, *International Journal of Robotics, Research and Development (IJRRD)*, Vol . 4, Issue 2, Apr 2014, 17-24.
- [23] ΔΟΥΛΓΕΡΗ Ζωή, ΡΟΜΠΟΤΙΚΗ, κινηματική δυναμική και έλεγχος αρθρωτών βραχιόνων, Εκδόσεις Κριτική, 2007
- [24] Ε.ΚΑΝΑΡΑΧΟΣ Ανδρέας-Καθηγητής Ε.Μ.Π, *Μηχανισμοί και Ρομποτικά Συστήματα*, Εκδόσεις Παπασωτηρίου, 1996
- [25] *Ρομποτική- Αυτόματη Θεωρία-Πράξη γύρω από τα ρομπότ*, Εκδόσεις Σύγχρονη Έκδοση, 2005.