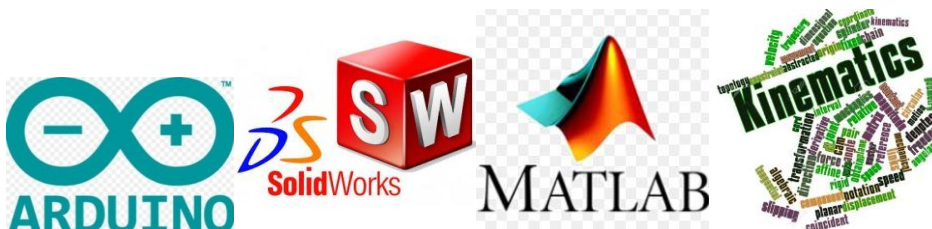


ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗ ΡΟΜΠΟΤΙΚΗ

**ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ
ΚΑΘΟΔΗΓΟΥΜΕΝΟΥ ΑΠΟ PC ΚΑΙ ARDUINO ΚΑΙ ΥΛΟΠΟΙΗΣΗ
ΑΛΓΟΡΙΘΜΩΝ ΜΕΣΩ ΛΟΓΙΣΜΙΚΟΥ ΜΑΤΛΑΒ ΓΙΑ ΚΙΝΗΣΗ ΤΟΥ
ΒΡΑΧΙΟΝΑ ΣΤΟΝ ΧΩΡΟ ΜΕ ΑΠΟΦΥΓΗ ΕΜΠΟΔΙΩΝ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ ΤΟΥ
ΛΟΥΚΑ ΖΗΣΗ ΤΟΥ ΚΩΝ/ΝΟΥ
ΑΕΜ:12

ΕΠΙΒΛΕΠΩΝ: Καθηγητής Δρ. ΚΑΖΑΡΛΗΣ ΣΠΥΡΙΔΩΝ
ΣΕΡΡΕΣ 2019



*Στον αγαπημένο μου Γιο που την παρούσα στιγμή βρίσκεται μέσα
στον αμνιακό σάκο και στην αγαπημένη μου Σύζυγο Στέλλα που μου
συμπαραστάθηκε σε όλες τις απαιτητικές και δύσκολες συνθήκες κατά
τη διάρκεια των σπουδών μου*

Στην οικογένειά μου

Ευχαριστίες

Με την ολοκλήρωση της παρούσας Διπλωματικής εργασίας , η οποία σηματοδοτεί και το τέλος των μεταπτυχιακών σπουδών μου στο πρόγραμμα μεταπτυχιακών σπουδών στη Ρομποτική του τμήματος Μηχανικών Πληροφορικής του ΤΕΙ Κεντρικής Μακεδονίας , θα ήθελα να ευχαριστήσω ολόψυχα όλους όσους στήριξαν εμένα και την προσπάθειά μου, με διαφορετικό τρόπο ο καθένας , αλλά με κοινό σκοπό.

Θα ήθελα να ευχαριστήσω τη σύζυγο μου Στυλιανή η οποία με στήριξε ηθικά και ψυχολογικά σε όλα τα στάδια των σπουδών μου , που μέσα από την κατανόηση την οποία έδειξε για την όλη προσπάθειά μου συνετέλεσε στην καλύτερη απόδοσή μου.

Έχω την ανάγκη να εκφράσω ένα μεγάλο ευχαριστώ στον συνοδοιπόρο μου αρχικά σε όλο αυτό το ταξίδι Διδάκτωρ Σπυρίδων Καζαρή κι έπειτα επιβλέπων Καθηγητή της παρούσας Διπλωματικής εργασίας , που με σωστή μεθοδολογία και στήριξη σε όλους τους τομείς συνετέλεσε στην ολοκλήρωση της παρούσας διπλωματικής εργασίας με τον καλύτερο τρόπο , οι συμβουλές του και οι καθοδήγησή του σε όλες τις φάσεις ήταν πολύτιμες.

Όπως επίσης ένα μεγάλο ευχαριστώ και στο τμήμα μηχανικών πληροφορικής του ΤΕΙ Κεντρικής Μακεδονίας και προτίστω τον Πρόεδρο του Μεταπτυχιακού Διδάκτωρ Ιωάννη Καλόμοιρο , για την πρωτοβουλία να δημιουργηθεί ένα τόσο ενδιαφέρον πρόγραμμα , όπως επίσης που έγινα δεκτός στο πρόγραμμα και μου δόθηκε τη δυνατότητα να παρακολουθήσω , να αναλύσω και κατανοήσω πολλές πτυχές αυτής της απίστευτης επιστήμης η οποία ονομάζεται Ρομποτική, είμαι σίγουρος πως αυτό ήταν μόνο η αρχή και τα καλύτερα έρχονται.

Δε θα μπορούσα να παραλείψω να ευχαρισήσω τον Διδάκτωρ Φασουλά Ιωάννη Διευθυντή του εργαστηρίου Ρομποτικής του Τμήματος Μηχανολόγων Μηχανικών Κρήτης, για την ανεκτίμητη βοήθεια, καθοδήγηση και στήριξή του κάθε φορά που τη χρειάστηκα.

Επίσης δε θα μπορούσα να παραλείψω να ευχαριστήσω τη Διοίκηση και όλους τους συναδέλφους μου στην εταιρία παραγωγής Μονωτικών υλικών Fibran Αναστασιάδης στην οποία εργάζομαι, οι οποίοι με στήριξαν σε όλους τους τομείς κατά τη διάρκεια των σπουδών μου.

Λουκάς Κων. Ζήσης

Σέπρες Μαΐος 2019

Περιεχόμενα

Ευχαριστίες.....σελ. 4	σελ. 4
Περιεχόμενα.....σελ. 5-8	σελ. 5-8
Περίληψη-Ελληνικάσελ. 8	σελ. 8
Περίληψη-Αγγλικάσελ. 8	σελ. 8
Εισαγωγή.....σελ. 9	σελ. 9
Ιστορικές αναφορές.....σελ.10	σελ.10

Κεφάλαιο 1.....σελ. 11	σελ. 11
------------------------	---------

Εισαγωγή στην επιστήμη της Ρομποτικής

1.1 Εισαγωγικές έννοιες.....σελ 11-12	σελ 11-12
1.2 Εφαρμογές της Ρομποτικής.....σελ 13-15	σελ 13-15
1.3 Αναφορά στους διάφορους τύπους ρομπότ.....σελ.16	σελ.16
1.3.1 Ρομπότ Σταθερής Βάσης.....σελ.16-17	σελ.16-17
1.3.2 Κινούμενα Ρομπότ.....σελ.18-21	σελ.18-21
1.4 Χρήση των ρομπότ και η επιρροή στον άνθρωπο.....σελ.22	σελ.22
1.4.1 Πλεονεκτήματα.....σελ. 22-23	σελ. 22-23
1.4.2 Μειονεκτήματα.....σελ. 23-24	σελ. 23-24
1.4.3 Σύνοψη.....σελ.24	σελ.24

Κεφάλαιο 2.....σελ. 25	σελ. 25
------------------------	---------

Ρομποτικοί Βραχίονες

2.1.1 Ορισμός.....σελ. 25-26	σελ. 25-26
2.2 Ταξινόμηση των Ρομποτικών Βραχιόνων.....σελ. 27	σελ. 27
2.2.1 Βραχίονες Καρτεσιανής Γεωμετρίας.....σελ. 27	σελ. 27
2.2.2 Βραχίονες τύπου Gantry.....σελ. 28	σελ. 28
2.2.3 Κυλινδρικοί βραχίονεςσελ.28-29	σελ.28-29
2.2.4 Σφαιρικοί βραχίονες.....σελ.29	σελ.29
2.2.5 SCARA.....σελ.30	σελ.30
2.2.6 Ανθρωπομορφικοί Βραχίονες.....σελ. 30-31	σελ. 30-31
2.3 Είδη αρθρώσεων Ρομποτικού βραχίωνα.....σελ.31-34	σελ.31-34
2.4 Σύστημα ελέγχου των Βραχιόνων.....σελ. 35	σελ. 35
2.4.1. Υλικό τμήμα-Hardware.....σελ. 35	σελ. 35
2.4.2 Λογισμικό τμήμα -Software.....σελ.35	σελ.35
2.5 Βαθμοί Ελευθερίας.....σελ.36	σελ.36
2.6 Ενεργοποιητές.....σελ.37	σελ.37

2.6.1 Κινητήρες AC.....	σελ.38-40
2.6.2 Κινητήρες DC.....	σελ.40-42
2.6.3 Βηματικοί κινητήρες.....	σελ.42-43
2.6.4 Servo κινητήρες.....	σελ. 44-47
2.6.5 Πνευματικά έμβολα Ενεργοποιητές.....	σελ.47-50
2.6.6 Υδραυλικά έμβολα Ενεργοποιητές.....	σελ.50-52
2.7 Χώρος εργασίας.....	σελ.53-54
2.8 Διαχωρισμός με βάση το τελικό εργαλείο δράσης.....	σελ.55-57
2.9 Κατασκευαστές Ρομποτικών Βραχιόνων-Στατιστικά.....	σελ.58-60

Κεφάλαιο 3.....σελ.61

Υλοποίηση Ρομποτικού Βραχίονα

3.1 Συναρμολόγηση Μηχανικού Μέρους-Τροφοδοσία.....	σελ. 61-70
3.2 Κατάλογος υλικών που απαιτήθηκαν για την υλοποίηση.....	σελ. 71

Κεφάλαιο 4.....σελ.72

Εισαγωγή στην κινηματική θεωρία-Κινηματική ανάλυση

4.1 Εισαγωγή.....	σελ.72-73
4.1.1Χαρακτηριστικά μεγέθη Ρομποτικού Βραχίονα	σελ.74-77
4.2 Ευθύ κινηματικό – επίλυση-παραδείγματα.....	σελ. 77-90
4.3.1 Ανάστροφο κινηματικό επίλυση.....	σελ.91-99
4.3.2 Εποπτικά παραδείγματα ανάστροφου κινηματικού	σελ.99-116

Κεφάλαιο 5.....σελ.117

Πλακέτα Arduino UnoR3

5.1 Ηλεκτρονικά μέρη και Λογισμικό.....	σελ.117-119
5.2 Arduino.....	σελ.119-124
5.3 Ιστορικό.....	σελ.125
5.4 Εκδόσεις.....	σελ.126
5.5 Ανάπτυξη	σελ.126
5.6 Περιβάλλον Arduino IDE.....	σελ.127-130
5.7 Έλεγχος servo-κινητήρων με Arduino.....	σελ.130-135

Κεφάλαιο 6.....σελ.136

Περιβάλλον Matlab και σειριακή επικοινωνία με Arduino

6.1 Εισαγωγή στο Matlab.....	σελ.136-137
------------------------------	-------------

6.2.1 Πρωτόκολλο επικοινωνίας Arduino-Matlab.....σελ.138-142	σελ.138-142
6.2.2 Εγκατάσταση rvctools by Peter Corke για προσομείωση.....σελ.142-143	σελ.142-143
6.3 Κίνηση και χειρισμός του βραχίονα με διάφορες επιλογές.....σελ.144	σελ.144
6.3.1 Κίνηση δίνοντας inputs γωνίες.....σελ.144-150	σελ.144-150
6.3.2 Κίνηση δίνοντας inputs επιθυμητά σημεία τοποθέτησης του άκρου Px,Py,Pz.....σελ.151-158	σελ.151-158
6.3.3 Κίνηση του βραχίονα για σχηματισμό ευθείας γραμμής.....σελ.159-163	σελ.159-163

Κεφάλαιο 7.....σελ.164

Εισαγωγή Μηχανικής ευφυΐας στο σύστημα

7.1 Εισαγωγή για την Μέθοδο PSO(Particle Swarm Optimization).σελ.....164-168	σελ.....164-168
7.2 Επίλυση προβλημάτων ρομποτικής με τη μέθοδο της Βελτιστοποίησης Σμήνους Σωματιδίων-PSO(Particle Swarm Optimization).....σελ.169	σελ.169
7.2.1 Κατασκευή της συνάρτησης δημιουργίας του μοντέλου του χώρουσελ.169-170	σελ.169-170
7.2.2 Κατασκευή της συνάρτησης για ανάπτυξη και έλεγχο της εκάστοτε λύσης.....σελ.171-172	σελ.171-172
7.2.3 Κατασκευή της συνάρτησης ποιότητας για τις λύσεις του προβλήματος..σελ172-173	σελ172-173
7.2.4 Κατασκευή της συνάρτησης που θα σχεδιάζει τις λύσεις του προβλήματος...σελ.173	σελ.173
7.2.5 Κατασκευή του κυρίως προγράμματος για Βέλτιστη σχεδίαση Διαδρομής με χρήση του αλγορίθμου PSO.....σελ.174	σελ.174
7.3 Κίνηση στο χώρο εύρεση βέλτιστης διαδρομής και αποφυγή εμποδίων με τη χρήση του PSO.....σελ.175	σελ.175
7.3.1 Επίλυση σχεδιασμού διαδρομής με 3 σημεία καμπήςσελ.176-179	σελ.176-179
7.3.2 Επίλυση σχεδιασμού διαδρομής με 4 σημεία καμπήςσελ.179-182	σελ.179-182
7.3.3 Επίλυση σχεδιασμού διαδρομής με 5 σημεία καμπής.....σελ.182-184	σελ.182-184
7.3.4 Επίλυση σχεδιασμού διαδρομής με 6 σημεία καμπής.....σελ.185-189	σελ.185-189
7.3.5 Τοποθέτηση πραγματικών σημείων στο χώρο.....σελ.190	σελ.190

Κεφάλαιο 8

Λογότυπα κατασκευαστών Βιομηχανικών Βραχιόνων στον κόσμο.....σελ.191	σελ.191
--	---------

Κεφάλαιο 9.....σελ.192

Συμπεράσματα - Μελλοντικές Επεκτάσεις/Βελτιώσεις

9.1 Μαθήματα που χρησιμοποιήθηκαν από το πρόγραμμα του ΠΜΣ-Ρομποτικής.σελ 192	σελ 192
9.2 Μελλοντικές Επεκτάσεις Βελτιώσεις.....σελ192	σελ192
9.3 Συμπεράσματα Δυσκολίες κατα την ολοκλήρωση- Δοκιμές.....σελ.193	σελ.193

Κεφάλαιο 10

Κώδικες που αναπτύχθηκαν.....σελ.194-208	σελ.194-208
--	-------------

Κεφάλαιο 11

Βιβλιογραφία.....	σελ.209-214
Κατάλογος υλικών.....	σελ.215
Πίνακας Σχημάτων-Συντομογραφίες.....	σελ.216-217

Περίληψη

Η συγκεκριμένη μεταπτυχιακή εργασία, πραγματεύεται την κατασκευή και υλοποίηση ενός συστήματος ρομποτικού βραχίονα, 3 βαθμών ελευθερίας, ο οποίος ελέγχεται άμεσα από Arduino και έμμεσα από PC με τη χρήση του προγράμματος Matlab, επιτρέποντας έτσι την υλοποίηση αλγορίθμων υψηλού επιπέδου για Σχεδιασμό Διαδρομής του βραχίονα και αποφυγή εμποδίων. Ο βραχίονας αποτελείται από μία μεταλλική αρθρωτή κατασκευή και η κίνηση των βραχιόνων δίνεται από ειδικούς σερβομηχανισμούς υψηλής ροπής, με έλεγχο θέσης. Ο έλεγχος της κίνησης των σερβομηχανισμών του βραχίονα γίνεται άμεσα από σύστημα Arduino. Ο Arduino επικοινωνεί μέσω σειριακής θύρας με το PC από το οποίο δέχεται άμεσες εντολές κίνησης των βραχιόνων, μέσω ειδικού πρωτοκόλλου που έχει σχεδιαστεί για την περίπτωση. Στο PC έχει αναπτυχθεί το μοντέλο του Χώρου Διατάξεων του βραχίονα καθώς και αλγόριθμοι Σχεδιασμού Διαδρομής για την κίνηση του άκρου του βραχίονα εντός του Χώρου Διατάξεων αλλά και αποφεύγοντας τα εμπόδια.

Abstract

The purpose of this Diploma thesis , deals the construction and the implementation of a robot arm manipulator, three degrees of freedom, that is directly controlled by Arduino and indirectly by PC using Matlab , this allowing the implementation of high-level algorithms for the path planning design of manipulator and obstacle avoidance. The robotic arm is made up of a metal modular structure and the movement of joints given by high-torque special-purpose servomotors with position control. The movement control of servo motors directly comes from arduino. Arduino communicates through a serial port with the computer , from which it receives direct commands for the movement of the robotic manipulator , through a special protocol designed for the occasion. On PC, the Arrangement Space Model has been developed , as well as path planning algorithms for moving the tip of the arm into the Arrangement Space as well as avoiding obstacles.

Εισαγωγή

Ρομποτική είναι η επιστήμη η οποία συνδυάζει αρχές από τις επιστήμες των μαθηματικών, της φυσικής, της μηχανολογίας, της ηλεκτρολογίας, της πληροφορικής, των τηλεπικοινωνιών, της βιολογίας. Εμπριέχει επίσης την θεωρία συστημάτων αυτομάτου ελέγχου, την επεξεργασία ψηφιακών σημάτων, την υπολογιστική όραση και τα νευρωνικά δίκτυα. Η ρομποτική πραγματεύεται τον σχεδιασμό, την υλοποίηση, τον έλεγχο και την λειτουργία ενός ρομποτικού συστήματος. Η λέξη ρομπότ προέρχεται από τη Σλαβική *robota* που σημαίνει εξαναγκασμένη εργασία. Καθιερώθηκε ως όρος με την σημερινή του έννοια το 1920 από τον Τσέχο θεατρικό συγγραφέα Karel Čapek στο έργο του "R.U.R." (Rossum's Universal Robots), όπου σατιρίζει την εξάρτηση της κοινωνίας από τους μηχανικούς εργάτες (ρομπότ), που τελικά εξοντώνουν τους δημιουργούς τους. Στην προσπάθεια να δοθεί ένας ορισμός ως προς το τι είναι ρομπότ, δύο ορισμοί επικράτησαν, σύμφωνα με το Robot Institute of America (1979):

«Ένα αναπρογραμματιζόμενο, πολυλειτουργικό χειριστήριο σχεδιασμένο να μετακινεί υλικά, εργαλεία ή συσκευές μέσα από διάφορες προγραμματισμένες κινήσεις για την εκπλήρωση διαφόρων εργασιών».

Σύμφωνα με το λεξικό Webster (1993): **«Μια αυτόματη συσκευή που πραγματοποιεί λειτουργίες υπό τις οδηγίες ενός ανθρώπου ή μιας μηχανής με ανθρώπινη μορφή».** Μπορούμε να ταξινομήσουμε τους διάφορους τύπους ρομπότ με βάση την εφαρμογή τους. Οπότε τα Ρομπότ ταξινομούνται σε διάφορες κατηγορίες , όπως: Βιομηχανικά ρομπότ, ρομποτικά οχήματα εξερεύνησης, Ρομπότ στην επιστήμη της ιατρικής , για διασκέδαση και εκπαιδευτικούς σκοπούς και τέλος για στρατιωτικές και αστυνομικές εφαρμογές και διάσωση.

Key worlds: Robotics, Servo-motors, Robot arm, Arduino, Matlab , Path Planning , Algorithms, Kinematics, Mathematics, Mechanical engineering, Electronics, Electrical Engineering, Programming, Computer science, obstacle avoidance.

Λέξεις κλειδιά: Ρομποτική, Σέρβο-κινητήρας, Ρομποτικός Βραχίονας, Μικροελεγκτής Arduino, Λογισμικό Matlab, Κινηματική , Αλγόριθμοι αποφυγής εμποδίων, Μαθηματικά,

ΤΕΙ ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ | ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ | ΠΜΣ ΣΤΗ ΡΟΜΠΟΤΙΚΗ
Μηχανολογία, Ηλεκτρονική, Ηλεκτρολογία, Προγραμματισμός, Επιστήμη της
Πληροφορικής, αποφυγή εμποδίων.

Ιστορικές Αναφορές

Το πρώτο γνωστό ρομπότ στην ιστορία της ανθρωπότητας κατασκευάστηκε περίπου στο 400-350 π.Χ. από τον αρχαίο Έλληνα μαθηματικό Αρχύτα και ήταν ένα ατμοκίνητο περιστέρι. Ο Αρχύτας είναι γνωστός ως ο «πατέρας της μηχανολογίας». Κατασκεύασε το πουλί-ρομπότ από ξύλο και χρησιμοποίησε ατμό για να τροφοδοτήσει τις κινήσεις του. Κατάφερε να το κάνει να πετάξει για περίπου 200 μέτρα, προτού ξεμείνει από ατμό. Ο Ήρων* ο Αλεξανδρεύς, Έλληνας σοφός του 1ου αιώνα, κατασκεύασε μεγάλο αριθμό αυτόματων μηχανισμών που βασίζονταν στις αρχές της στατικής και δυναμικής των ρευστών και στην κλασική μηχανική. Κατά την παράδοση, που ίσως να περιλαμβάνει και υπερβολές, κατασκεύασε μηχανικά πουλιά που κελαηδούσαν, έπιναν νερό και πετούσαν. Οπωσδήποτε, επί πολλούς αιώνες δεν φαίνεται να υπήρξαν μιμητές του. Ο Τάλως ήταν μυθικός χάλκινος γίγαντας, το πρώτο ρομπότ στην ιστορία, που προστάτευε την μινωική Κρήτη από κάθε επίδοξο εισβολέα, λέγεται ότι κατασκευάστηκε από τον Ήφαιστο. Στην Ευρώπη του 18ου αιώνα εκδηλώθηκε, ωστόσο, ξαφνικό ενδιαφέρον περί τα αυτόματα μεταξύ πολλών επιτήδειων τεχνιτών. Σε μουσείο της Βιέννης διατηρείται ένας «γραφέας» (1753), μηχανισμός με ικανότητα γραφής και σχεδίασης. Το μηχανικό σύστημα ελέγχου του χεριού περικλείεται στο εσωτερικό σφαίρας που αποτελεί μέρος της βάσης.

Ο Πιερ Ζα-κέ-Ντροζ (γεννήθηκε το 1721) και ο γιος του Ανρί-Λουί Ζακέ-Ντροζ (γεννήθηκε το 1752), Γάλλοι ωρολογοποιοί, κατασκεύασαν πολλούς μηχανικούς ανθρώπους, που έγραφαν, σχεδίαζαν ή έπαιζαν μουσικά όργανα. Ενώ μόλις το 1938 κατασκευάζεται ο πρώτος προγραμματιζόμενος μηχανισμός βαφής από τους Αμερικανούς Willard Pollard και Harold Roselund. Έκτοτε και κυρίως από την δεκαετία του '50 και έπειτα η ρομποτική επιστήμη κάνει μεγάλα άλματα. Το 1962 η General Motors εισάγει στην γραμμή παραγωγής το πρώτο βιομηχανικό ρομπότ το Unimate. Το 1964 ιδρύονται στο Massachusetts Institute of Technology και στο Stanford University τα πρώτα εργαστήρια τεχνητής νοημοσύνης ενώ ακολουθούν και άλλα πανεπιστήμια και ερευνητικοί οργανισμοί.

Κεφάλαιο 1

1.Εισαγωγή στην επιστήμη της Ρομποτικής

1.1 Εισαγωγικές έννοιες

Ρομποτική είναι ο σύγχρονος τεχνολογικός κλάδος της αυτοματοποίησης, που έχει ως αντικείμενο τη μελέτη, το σχεδιασμό και τη λειτουργία των ρομπότ, καθώς και την έρευνα για την περαιτέρω ανάπτυξή τους, είναι ένας συνδυασμός των αρχών από τις επιστήμες των μαθηματικών, της φυσικής, της μηχανολογίας, της ηλεκτρολογίας, της πληροφορικής, των τηλεπικοινωνιών και της βιολογίας. Εμπεριέχει επίσης την θεωρία του αυτόματου ελέγχου. Ένα ρομπότ συγκροτείται από δύο συστήματα, το μηχανικό (στο οποίο περιλαμβάνεται το σύστημα κίνησης) και το ηλεκτρονικό (στο οποίο υπάγεται και η επαναπρογραμματιζόμενη μνήμη του). Υπάρχουν διάφορα κριτήρια διάκρισης και αντίστοιχες κατηγοριοποιήσεις των ρομπότ. Μία από αυτές είναι η διάκρισή τους σε τρεις, επί του παρόντος, "γενιές". Στην πρώτη γενιά κατατάσσονται ρομπότ με περιορισμένη ευελιξία, που διευθύνονται από τον άνθρωπο, όπως, για παράδειγμα, οι απλοί "χειριστές", σχετικά απλά εργαλεία που επιτρέπουν, για παράδειγμα, τη μετακίνηση επικίνδυνων αντικειμένων (π.χ. ραδιενεργών υλικών). Στη δεύτερη γενιά κατατάσσονται τα ρομπότ που είναι εφοδιασμένα με σταθερό πρόγραμμα δράσης και ρομπότ που λαμβάνουν εντολές από κάποιο σύστημα αριθμητικού ελέγχου και στην τρίτη γενιά κατατάσσονται ρομπότ , που είναι εφοδιασμένα με :

- με αισθητήρες ικανά να λαμβάνουν "πληροφορίες" από το περιβάλλον
- με διάφορες διατάξεις επεξεργασίας πληροφοριών
- με κινητήριο σύστημα εκτέλεσης εργασιών

Υπάρχουν τρεις βασικές ιδιότητες σε ένα ρομπότ:

- Η δυνατότητα επαναπρογραμματισμού (**programmability**)
A robot is a computer (information/ data processing)
- Δυνατότητα μηχανικής δράσης (**mechanical abilities**), εκτέλεση φυσικών εργασιών στο περιβάλλον **A robot is a machine(mechatronic device)**

- Προσαρμοστικότητα , ευελιξία , πολυσχιδής λειτουργικότητα (adaptability, versatility, flexibility)

.....**Adapt to different environment and task requirements**

Η ρομποτική πραγματεύεται τον σχεδιασμό, την υλοποίηση, τον έλεγχο και την λειτουργία ενός ρομποτικού συστήματος. Η επιστήμη της Ρομποτικής μελετά τις μηχανές εκείνες οι οποίες μπορούν να αντικαταστήσουν τον άνθρωπο στην εκτέλεση μιας εργασίας, η οποία συνδυάζει τη φυσική δραστηριότητα με τη διαδικασία λήψης αποφάσεων. Τα έργα επιστημονικής φαντασίας έχουν επηρεάσει τους περισσότερους στον τρόπο με τον οποίο φαντάζονται τα ρομπότ. Το 1984 με την πρώτη ταινία ο εξολοθρευτής και λίγο αργότερα στα τέλη του 2000 , με την ταινία Matrix γίνεται μια αναθεώρηση στην μέχρι τότε αντίληψη για τις μηχανές, ότι οι υπολογιστές παύουν να είναι μεγάλα κουτιά που απλά αντιλαμβάνονται μηδέν και ένα και εκτελούν εντολές , αλλά αποκτούν ευφυία και ανθρώπινη μορφή, όλα αυτά χάρη στον άνθρωπο, διότι ο άνθρωπος είναι η πιο ισχυρή μηχανή αναζήτησης λύσεων , σύλληψης ιδεών , εφευρέσεων και θεωριών σε όλο το σύμπαν. Οπότε θα μπορούσαμε να πούμε οτι ρομπότ θεωρούνται οι μηχανές αυτές οι οποίες , ανεξάρτητα απο την εμφάνισή τους , είναι ικανές να αλλάξουν το περιβάλλον στο οποίο λειτουργούν μέσα από δράσεις που ακολουθούν κάποιους εγγενείς κανόνες και με βάση τα δεδομένα για το ίδιο το ρομπότ και για το ίδιο το περιβάλλον του , που αποκτώνται από τα αισθητήρια με τα οποία είναι εφοδιασμένο. Έτσι , η ρομποτική έχει πρόσφατα οριστεί **ως η επιστήμη εκείνη η οποία μελετά την ευφυή διασύνδεση της αντίληψης με τη δράση.**

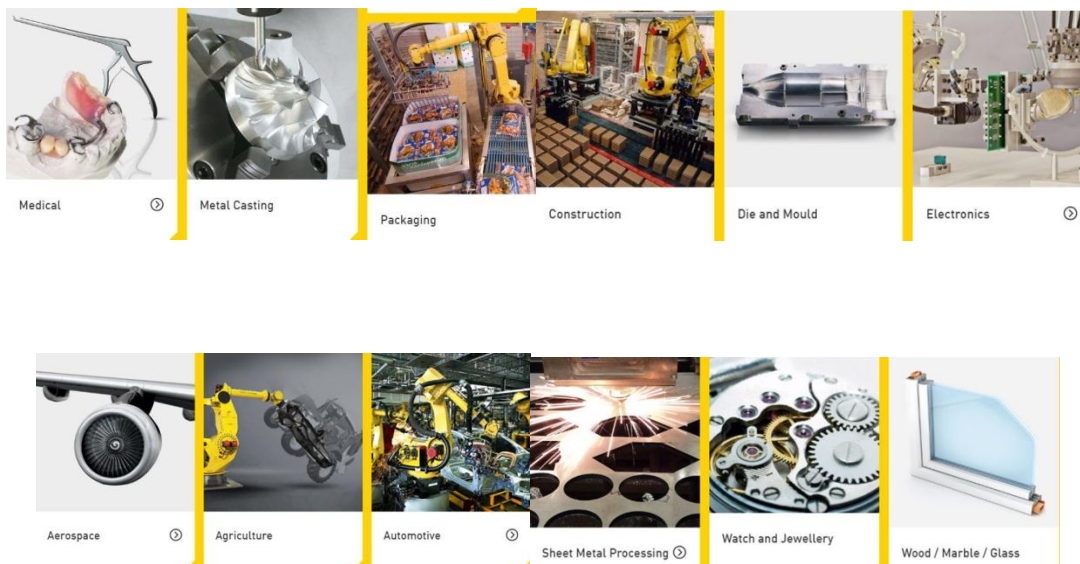
Επομένως ,θα μπορούσαμε να δώσουμε τον ακριβή ορισμό για το τί ακριβώς είναι ρομπότ. Σύμφωνα με το Ινστιτούτο Ρομπότ των ΗΠΑ, "**ρομπότ είναι μια επαναπρογραμματιζόμενη πολυλειτουργική χειριστική διάταξη, σχεδιασμένη για τη μετακίνηση υλικών, εξαρτημάτων, εργαλείων και εξειδικευμένων διατάξεων, μέσω μεταβλητών, προγραμματισμένων κινήσεων για την εκτέλεση μιας σειράς εργασιών**".

1.2 Εφαρμογές της ρομποτικής.

Ευρύτατη χρήση ρομπότ γίνεται σε πάρα πολλούς παραγωγικούς τομείς και κυρίως στη βιομηχανία (βιομηχανική ρομποτική), στην ιατρική, την αεροναυπηγική, την αεροδιαστημική, υποθαλάσσιες έρευνες κ.ά, γεγονός που έδωσε σημαντική ώθηση στην περαιτέρω ανάπτυξη της βιομηχανίας των ρομπότ, ιδιαίτερα στην Ιαπωνία και τις ΗΠΑ. Οι κυριότερες εφαρμογές των βιομηχανικών ρομπότ, μέχρι σήμερα, ήταν οι ηλεκτροσυγκολλήσεις, οι εφαρμογές σε εργασίες πρεσαρίσματος, εργασίες κοπής και διάτρησης, οι συναρμολογήσεις, σε βαφές με ψεκασμό και η επεξεργασία επιφανειών σε τροφοδοτήσεις εργαλειομηχανών, σε μορφοποιήσεις πλαστικών σε μήτρες κ.ά. Από τα μέσα περίπου της δεκαετίας του 1980 η χρήση των ρομπότ γενικεύτηκε στο πλαίσιο της ανάπτυξης των Ολοκληρωμένων Συστημάτων Παραγωγής (Computer-Integrated Manufacturing), αυτοματοποιημένων και ευέλικτων εργοστασίων, στα οποία οι εργαλειομηχανές μπορούν να επαναπρογραμματίζονται ταχύτατα για την παραγωγή νέων ή διαφοροποιημένων προϊόντων. Πέρα, όμως, από τη βιομηχανία ευρύτατη χρήση ρομπότ γίνεται και σε μη μεταποιητικές εφαρμογές, όπως, για παράδειγμα, σε πυρηνικούς σταθμούς, υποθαλάσσιες έρευνες, σε ιατρικές εφαρμογές, στην εξόρυξη πετρελαίου κ.λπ. Τα χειρουργικά ρομπότ εισέβαλλαν δυναμικά στο πεδίο της ιατρικής και νοσηλευτικής επιστήμης μέσα στην τελευταία δεκαετία. Συστήματα ρομποτικής τηλεχειρουργικής έχουν ήδη χρησιμοποιηθεί για υπερατλαντικές επεμβάσεις (επέμβαση Lindbergh 2001). Ρομποτικοί βραχίονες που ενεργοποιούνται με τη φωνή μπορούν και χειρίζονται τη λαπαροσκοπική κάμερα. Το αυτόνομο υποβρύχιο όχημα έχει την κωδική ονομασία «Εξερευνητής» και είναι ικανό να καταδύεται σε βάθη 5.000 μέτρων, ενώ μπορεί να ταξιδέψει σε απόσταση μεγαλύτερη των 100 χλμ., κάτω από θαλάσσια επιφάνεια που είναι καλυμμένη με πυκνό πάγο. Επίσης για υποθαλάσσιες εξερευνήσεις χρησιμοποιούνται, υποβρύχια οχήματα, έχοντας τη δυνατότητα να συλλέγουν πληροφορίες από τα κομμάτια πάγου που επιπλέουν και βρίσκονται στη θαλάσσια περιοχή της Ανταρκτικής, βοηθώντας τους επιστήμονες να κατανοήσουν την περιβαλλοντολογική δυναμική της Ανταρκτικής και την επίδρασή της στην κλιματική αλλαγή.

Σημαντικό ρόλο παίζουν και σε στρατιωτικές επιχειρήσεις, έναν τομέα όπου απαιτείται προσοχή, διότι η εξέλιξη της μηχανικής ευφυΐα και της τεχνητής νοημοσύνης μπορεί να αποβεί καταστροφική. Όσο αναφορά την εξερεύνηση του διαστήματος και άλλων πλανητών έχουν δημιουργηθεί από επιστήμονες ρομποτικοί βομβίνοι (οι γνωστοί μπούμπουρες), ένα είδος αγριομέλισσας που είναι μεγαλύτερο σε μέγεθος από τα υπόλοιπα είδη. Αυτά τα ρομποτικά έντομα θα είναι εφοδιασμένα με διάφορους αισθητήρες και συστήματα ασύρματης επικοινωνίας για να μεταδίδουν τα δεδομένα τους, ενώ θα μπορούν να φορτίζονται πάνω στα ρομπότ εξερεύνησης που βρίσκονται στην επιφάνεια του Άρη. Αν και τα οχήματα εξερεύνησης, πιο γνωστά ως rover, παρέχουν πλήθος σημαντικών πληροφοριών στους επιστήμονες εντούτοις κινούνται με αρκετά αργό ρυθμό. Τα ρομποτικά έντομα από την άλλη πλευρά θα μπορούν να μετακινούνται γρήγορα σε διάφορες περιοχές του Κόκκινου Πλανήτη και να μεταδίδουν στο rover τα δεδομένα τους ώστε να έχουν πρόσβαση σε αυτά οι επιστήμονες στην συνέχεια. Επειδή η ατμόσφαιρα του Άρη είναι πολύ πιο αραιή από αυτή της Γης θα πρέπει οι ρομποτικές μέλισσες να διαθέτουν μεγαλύτερα σε μέγεθος και ισχυρότερα φτερά από εκείνα των πραγματικών για να μπορούν να πετούν χωρίς προβλήματα στον πλανήτη. Επίσης ρομπότ διασώστες έχουν κάνει την εμφάνισή τους, με κύριο στόχο την αντιμετώπιση έκτακτων καταστάσεων όπως πυρκαγιές, διαρροή αερίων, ραδιενέργεια κτλ., απεγκλώβιση σε περιπτώσεις σεισμού και γενικά παροχή βοήθειας σε καταστάσεις έκτακτης ανάγκη όπου δεν μπορούν να πλησιάσουν οι διασώστες ή άλλοι άνθρωποι. Ρομπότ επίσης με την ευρεία έννοια μπορούν να θεωρηθούν και τα αυτοματοποιημένα διαστημόπλοια (μη επανδρωμένα), που χρησιμοποιούνται για διαστημικές έρευνες, καθώς και ειδικές κατασκευές όπως, για παράδειγμα, η σοβιετική σεληνάκατος Lunokhod-1, η οποία καθοδηγούνταν με ασύρματο από τη Γη. Τέλος, σε πειραματική και όχι εμπορική βάση έχουν κατασκευαστεί οικιακά ρομπότ που καθαρίζουν το σπίτι, σερβίρουν ποτά ή "παίζουν" με τα παιδιά.

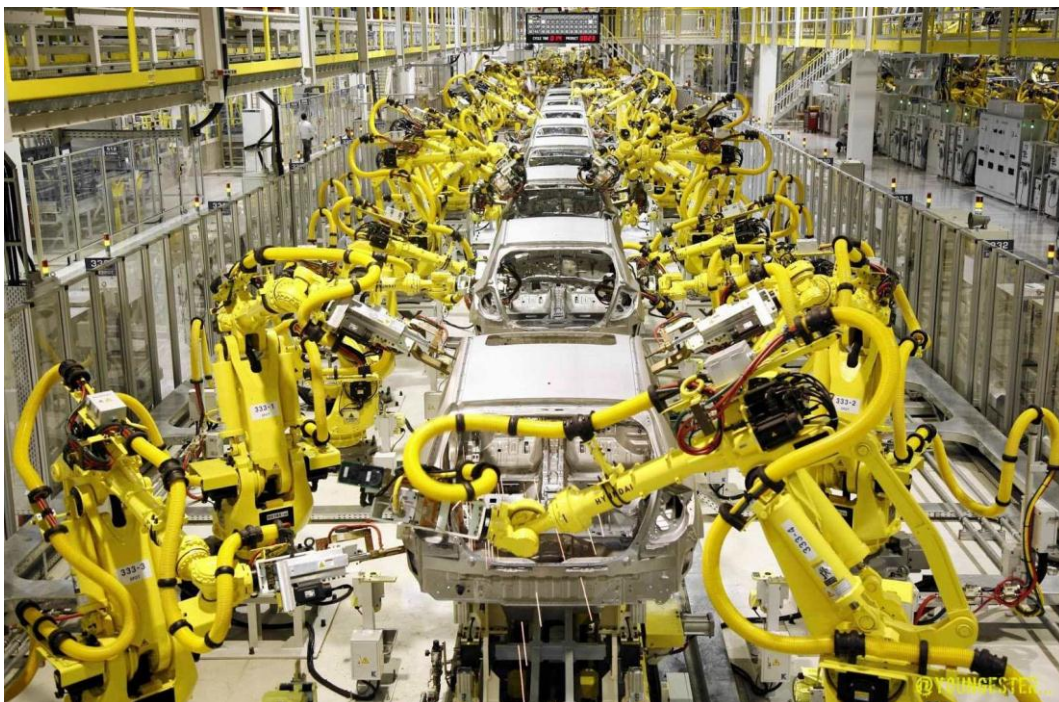
Η ανάπτυξη του κλάδου της τεχνητής νοημοσύνης (artificial intelligence) κατά τη δεκαετία του 1980 άνοιξε ευρύτατες προοπτικές εφαρμογής της στη ρομποτική. Όπως είναι γνωστό, η τεχνητή νοημοσύνη αποτελεί ένα από τα σημαντικότερα πεδία έρευνας της πληροφορικής και αφορά την κατασκευή συστημάτων αυτοματισμού εφοδιασμένων με ικανότητα μάθησης, δυνατότητα κατανόησης της φυσικής γλώσσας, ικανότητα αξιολόγησης στοιχείων, λήψης αποφάσεων κ.ά. Οι σχετικές έρευνες στον τομέα της ρομποτικής αφορούν την κατασκευή ρομπότ τα οποία πέρα από τις βασικές αισθήσεις, όπως η αφή και η όραση, θα είναι εφοδιασμένα με αντιληπτικές ικανότητες (για παράδειγμα, αντίληψη σχημάτων, μορφών, εικόνων κ.λπ.), με ικανότητα διεξαγωγής λογικών συνειρμών και εξαγωγής συμπερασμάτων, καθώς και με δυνατότητες ανακατανομής δεδομένων ανάλογα με τη χρήση για την οποία προορίζονται και με ικανότητα αυτοδιόρθωσης. Η ανάπτυξη της προηγμένης τεχνολογίας ρομπότ αναμένεται ότι θα βοηθήσει σημαντικά στην επίλυση προβλημάτων και στην ολοκλήρωση εργασιών σε χώρους που είναι δύσκολα προσπελάσιμοι για τον άνθρωπο. Ωστόσο εκφράζονται και φόβοι για τη δυσκολία ελέγχου των συστημάτων αυτών, καθώς θα είναι εφοδιασμένα με ικανότητα ανάπτυξης σχετικής αυτονομίας κατά τη λειτουργία τους.



1.3 Αναφορά στους διάφορους τύπους ρομπότ

Κατά την πολυετή εξέλιξη της επιστήμης της ρομποτικής προέκυψαν διάφορα είδη ρομποτικών μηχανισμών, οι οποίοι διαφέρουν σημαντικά στη μορφή και την κατασκευή αποτελούνται όμως από αντίστοιχα επιμέρους υποσυστήματα, έχουν κοινά όμως το μηχανολογικό υποσύστημα, το υποσύστημα αίσθησης και το σύστημα ελέγχου. Μερικά από τα σπουδαιότερα είδη ρομπότ θα αναφερθούν παρακάτω.

1.3.1 Ρομπότ Σταθερής Βάσης: τα ρομπότ αυτά αποτελούνται από διαδοχικά στερεά σώματα (σύνδεσμοι) που συνδέονται μέσω αρθρώσεων σχηματίζοντας μία κινηματική αλυσίδα. Η αλυσίδα αυτή έχει το ένα άκρο της (βάση) σταθερά συνδεδεμένο με κάποιο σημείο του περιβάλλοντος χώρου. Η μορφή αυτή ρομπότ είναι η παραδοσιακή μορφή ενός βιομηχανικού ρομποτικού βραχίονα, και περιλαμβάνει το βραχίονα, τον καρπό και το εργαλείο.



Εικόνα 1.1 Βιομηχανική παραγωγή με ρομποτικούς βραχίονες



Εικόνα 1.2. Χρήση ρομποτικών βραχιόνων-φανοποιών για βαφή αυτοκινήτων



Εικόνα 1.3 Η σύγχρονη μορφή ενός ρομποτικού συστήματος σταθερής βάσης

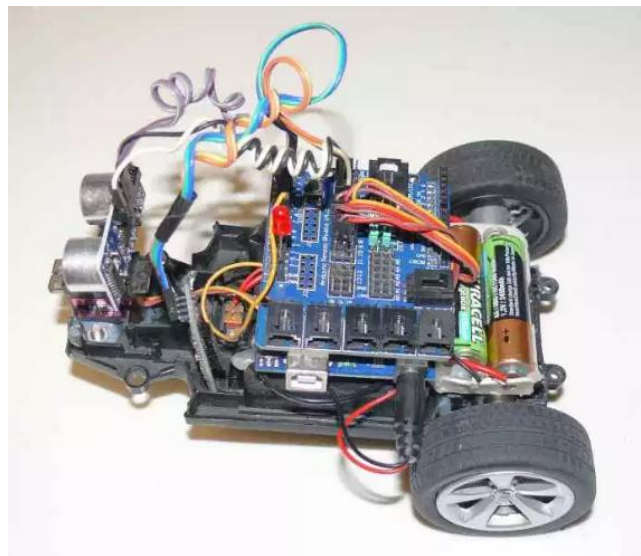
1.3.2 Κινούμενα Ρομπότ: ως κινητά ρομπότ χαρακτηρίζονται όλα εκείνα τα ρομπότ που έχουν τη δυνατότητα να μετακινήσουν όλα τα σημεία του μηχανισμού τους. Η δυνατότητα αυτή προσφέρεται από ειδικά συστήματα προώθησης, τα οποία μπορεί να είναι είτε απλά (όπως τροχοί, ερπύστριες) είτε πολύπλοκα (όπως jet, προπέλες, μηχανικά πόδια κτλ.). Τα κινούμενα ρομπότ διακρίνονται σε επιμέρους κατηγορίες ανάλογα με το βαθμό αυτονομίας τους. Έτσι έχουμε:

- Τα **AGVs** (Automatic Guided Vehicles) έχουν περιορισμένη αυτονομία κίνησης, δεδομένου ότι η τροχιά τους είναι προκαθορισμένη μέσω καλωδίων στο έδαφος ή πομπών στον περιβάλλοντα χώρο.(εικ 1.4.)



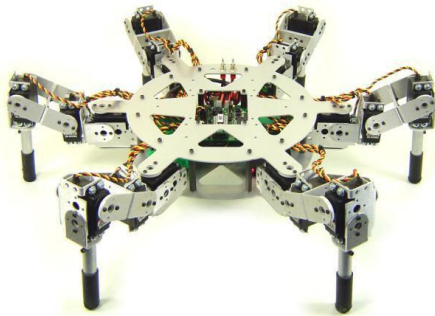
Εικόνα 1.4 AGV robot στη βιομηχανία

•Αυτόνομα Έντροχα Ρομπότ: τα ρομπότ αυτά λειτουργούν με αρκετά υψηλό βαθμό αυτονομίας. Πιο συγκεκριμένα μπορούν και λειτουργούν χωρίς συνεχή εξωτερική επίβλεψη και είναι ικανά να εκτελούν εργασίες αυτόνομα δεχόμενα μόνο ορισμένες υψηλού επιπέδου εντολές. (εικ. 1.5)



Εικόνα 1.5 Αυτόνομο ρομπωτικό όχημα με ρόδες

•Βαδίζοντα Ρομπότ: τα ρομπότ αυτά χρησιμοποιούν μηχανικά πόδια για την κίνησή τους και όχι συμβατικούς τροχούς όπως στις προηγούμενες δύο κατηγορίες. Τα κυριότερα πλεονεκτήματα της συγκεκριμένης υλοποίησης είναι η μεγάλη δυνατότητα αποφυγής εμποδίων και η ικανότητα αναρρίχησης σε ανώμαλα εδάφη και μη επίπεδες επιφάνειες. Από τα πιο συνηθισμένα ρομπότ αυτής της κατηγορίας είναι τα ανθρωπόμορφα , τα δίποδα ενώ δεν αποκλείονται και εφαρμογές με περισσότερα από δύο πόδια, π.χ. ρομπότ που μοιάζουν και κινούνται όπως οι αράχνες.(εικ. 1.6 και 1.7)



Εικ1.6 Ρομπότ αράχνη για κίνηση σε ανώμαλα εδάφη Εικ 1.7 Βαδίζοντα Ρομπότ Στρατιωτικές, αστυνομικές εφαρμογές και διάσωση

- **ROVs:** τα ROVs (Remotely Operated Vehicles) ανήκουν στην κατηγορία των μη επανδρωμένων υποβρύχιων ρομπότ. Όπως υποδηλώνει το όνομά τους δεν έχουν μεγάλο βαθμό αυτονομίας, μιας και είναι συνδεδεμένα με το μητρικό πλοίο μέσω καλωδίου, το οποίο και καλύπτει τις ανάγκες του ρομπότ σε ενέργεια και επικοινωνίες. Τα ρομπότ αυτού του τύπου έχουν σχήμα κουτιού και κινούνται γενικά σε χαμηλές ταχύτητες με τη βοήθεια πτερυγίων προσομοιώνοντας την κίνηση των υποθαλάσσιων οργανισμών. (εικ 1.8.)



Εικόνα 1.8 ROVs εξερευνώντας τον υποθαλάσσιο κόσμο

• **Εναέρια ρομπότ (UAV):** πρόκειται για μη επανδρωμένα ιπτάμενα ρομπότ, όπως ελικόπτερα και αεροπλάνα. Τα ρομπότ αυτά έχουν διαρκώς αυξανόμενες εφαρμογές και χρησιμοποιούνται για στρατιωτικούς κυρίως σκοπούς. Τα τελευταία χρόνια τα τετρακόπτερα (quadcopters) έχουν γίνει αρκετά δημοφιλή καθώς αποτελούν σχετικά φθηνές λύσεις για χόμπι, κινηματογράφιση, μεταφορά προϊόντων κ.α. (εικ. 1.9)



Εικόνα 1.9.UAV-Τετρακόπτερο



Εικ.1.10 Ιπτάμενο Ρομπότ drone , με κάμερα , κατάλληλο για κατασκοπία.

1.4 Χρήση των ρομπότ και η επιρροή τους στον άνθρωπο

1.4.1 Πλεονεκτήματα: Τα ρομπότ έχουν συμβάλει αρκετά στη ζωή του ανθρώπου σε διάφορους τομείς, στο επίπεδο της καθημερινής ζωής τα ρομπότ είναι ως επί το πλείστον μηχανικές συσκευές προγραμματισμένες να εκτελούν συγκεκριμένες επαναλαμβανόμενες λειτουργίες, να χρησιμοποιούνται για εργασίες επικίνδυνες ή δύσκολα πραγματοποιήσιμες από τον άνθρωπο, καθώς και για οικιακές εργασίες. Έτσι, υπάρχουν ρομπότ ικανά να καθαρίσουν το σπίτι, να μαγειρέψουν ή να μας διασκεδάσουν. Οι ρομποτικές συσκευές χρησιμοποιούνται συνήθως για την εκτέλεση πολλών εργασιών, που οι άνθρωποι είτε δεν μπορούν να κάνουν, επειδή είναι ιδιαίτερος πολύπλοκες, είτε δεν θέλουν, επειδή είναι βαρετές, βρώμικες ή επικίνδυνες. Ένα κλασικό παράδειγμα ρομποτικών εφαρμογών που έχουμε σήμερα βρίσκεται στην κατασκευή και συναρμολόγηση των αυτοκινήτων. Τα ρομπότ παίρνουν τη θέση των εργαζομένων στη γραμμή συναρμολόγησης των εργοστασίων, όπου εκτελούνται εξειδικευμένες εργασίες, όπως η τοποθέτηση καρφιών, η συναρμολόγηση βαρέων εξαρτημάτων, η βαφή κ.λπ.

Επίσης η εξερεύνηση των πλανητών του Ηλιακού μας Συστήματος έχει πραγματοποιηθεί από ρομπότ. Τις τελευταίες δεκαετίες, πολλές ρομποτικές διαστημικές συσκευές (όπως είναι οι τροχιακοί δορυφόροι, οι συσκευές προσεδάφησης και εξερεύνησης εδάφους κ.λπ.) έχουν επισκεφθεί τη Σελήνη, τους πλανήτες και τους δορυφόρους τους, αστεροειδείς και κομήτες. Σύμφωνα με μία μερίδα ειδικών, ένα ρομπότ μπορεί να κάνει στο Διάστημα ό,τι και ένας άνθρωπος και μάλιστα φθηνότερα και χωρίς τον κίνδυνο να χαθούν ανθρώπινες ζωές. Οι άνθρωποι ως διαστημικοί ταξιδιώτες απαιτούν εκτενή συστήματα υποστήριξης ζωής. Εξάλλου, με τις σημερινές τεχνολογίες πρόωσης, χρειάζεται πολύς χρόνος για να φθάσουμε σε οποιονδήποτε προορισμό πέρα από τη Σελήνη. Τα ρομπότ μπορούν να επιβιώσουν σε μακροχρόνια ταξίδια στο Διάστημα και να επιτύχουν τους στόχους της αποστολής εξερεύνησης, ακριβώς όπως και οι άνθρωποι. Η χρήση των ρομπότ εμφανίζεται πλέον κατά κόρον και στην Ιατρική, προσφέροντας σημαντική βοήθεια στη Χειρουργική και στην Ορθοπαιδική, επιτρέποντας τον λεπτομερή έλεγχο και την τεράστια ακρίβεια χρήσης των χειρουργικών εργαλείων.

Ως παραδείγματα μπορούμε να αναφέρουμε την αντιμετώπιση της γαστροοισοφαγικής παλινδρόμησης, καθώς και χειρουργικές επεμβάσεις στη χοληδόχο κύστη, με τη βοήθεια ενός ενδοσκοπίου που εισάγεται στο σώμα των ασθενών. Η «Νευρορομποτική» προσπαθεί να συνδέσει αποτελεσματικά το ανθρώπινο νευρικό σύστημα με προσθετικά ρομποτικά μέλη, όπως χέρια και πόδια, κατασκευάζει τεχνητά μέλη για παραπληγικούς ασθενείς. Η εξέλιξη της νευρορομποτικής έχει φτάσει σε επίπεδο όπου, οι επιστήμονες παίρνουν τα σήματα κατευθείαν από τον εγκέφαλο, τοποθετώντας βιοτσιπάρια στον φλοιό του, αυτά κινούν τεχνητά μέλη, όπως χέρια και πόδια. Καθώς τα ρομπότ γίνονται ακόμη μικρότερα και οι προγραμματιστές προσπαθούν να τα εξοπλίζουν με τεχνητή νοημοσύνη, η ιατρική κοινότητα θα διευρύνει συνεχώς τους τρόπους με τους οποίους χρησιμοποιεί αυτή την τεχνολογία, με στόχο τη θεραπεία των ασθενών, τη βελτίωση της ποιότητας της ζωής και την πρόληψη των προβλημάτων υγείας.

1.4.2 Μειονεκτήματα: Τα ρομπότ δεν είναι ικανά να ανταποκριθούν σε έκτακτες περιπτώσεις εκτός εάν η κατάσταση έχει συμπεριληφθεί στην εκπαίδευση και στο σύστημα του ρομπότ, οπότε τα καθήκοντα τους είναι πολύ περιορισμένα. Επίσης χρειάζεται να λαμβάνονται αυστηρά μέτρα ασφαλείας για να εξασφαλίζεται η ασφάλεια αυτών των ανθρώπων που συνεργάζονται μαζί τους ή τα χειρίζονται. Το κόστος εγκατάστασης, συντήρησης και εκμάθησης είναι πάρα πολύ μεγάλο και μερικές φορές απαγορευτικό. Σε μερικά χρόνια από τώρα τα ρομπότ θα είναι παντού, είναι πιθανό ο παγκόσμιος πληθυσμός να περιλαμβάνει δισεκατομμύρια ανθρώπων και δισεκατομμύρια ρομπότ, με τα τελευταία να κάνουν σχεδόν όλη τις δουλειές ρουτίνας ή τις βαριές εργασίες. Οι άνθρωποι θα εργάζονται για τη βελτίωση του λογισμικού των ρομπότ και η βιομηχανία της Πληροφορικής θα αποτελεί το «σπίτι» των εταιρειών που αναπτύσσουν προγράμματα για ρομπότ, όπως έγινε περίπου στις μέρες μας με τις εταιρείες που αναπτύσσουν εφαρμογές, τις οποίες οι χρήστες «κατεβάζουν» και εγκαθιστούν. Σε κάποιο βαθμό, τα όρια μεταξύ ανθρώπων και ρομπότ θα γίνουν θολά. Στις μεταμοσχεύσεις θα αρχίσουν να χρησιμοποιούνται ηλεκτρονικά ελεγχόμενα τεχνητά όργανα και η προσθετική μελών θα αποτελεί ρουτίνα στη χειρουργική διαδικασία. Νανορομπότ θα «ταξιδεύουν» βαθιά μέσα στο ανθρώπινο σώμα για την έγχυση φαρμάκων σε ασθενή κύτταρα ή για την εκτέλεση μικροχειρουργικών επεμβάσεων.

Ειδικά εγκατεστημένοι αισθητήρες θα παρακολουθούν την υγεία των ανθρώπων και θα μεταφέρουν τα ευρήματά τους σε cloud «αποθήκες», στις οποίες θα έχουν πρόσβαση οι γιατροί. Όλα αυτά, λογικά μπορεί να οδηγήσουν σε σημαντική αύξηση του προσδόκιμου ζωής. Επιπλέον, οι άνθρωποι θα ζουν σε «έξυπνα» σπίτια, όπου οι περισσότερες ανέσεις θα είναι πλήρως αυτοματοποιημένες. Το λογισμικό που θα «τρέχει» το σπίτι, θα φροντίζει για την κατανάλωση και την αναπλήρωση ενέργειας, νερού, τροφίμων και προμηθειών. Η μόνη ανησυχία των κατοίκων θα είναι να εξασφαλίσουν ότι διαθέτουν αρκετά χρήματα στην τράπεζα για να πληρώσουν τους λογαριασμούς τους. Το μόνο σίγουρο είναι πως θα εμφανιστεί η τεchnοφοβία σε μεγάλο ποσοστό ανθρώπων, για το λόγο αυτό δεν θα είναι όλοι ενθουσιασμένοι με τον νέο «ρομποτικό» κόσμο. Είναι πιθανόν να προκύψουν νέου τύπου «Λουδίτες», οι οποίοι θα αντιπαχθούν στην ανάπτυξη των «έξυπνων» σπιτιών, στον αυτοματοποιημένο τρόπο ζωής και στα ρομπότ. Η αντίθεση προς τις εξελίξεις της Πληροφορικής θα οδηγήσει κάποιους στην αποφυγή χρήσης έξυπνων συστημάτων, συσκευών και ρομπότ για ορισμένους τύπους εργασιών και αυτοί δεν θα έχουν καμία ψηφιακή ταυτότητα.

1.4.3 Σύνοψη

Στον κόσμο της Ρομποτικής δεν υπάρχουν πλεονεκτήματα και μειονεκτήματα, όταν η χρήση των ρομπότ γίνεται με σκοπό την διευκόλυνση των ανθρώπων σε διάφορους τομείς της καθημερινότητας, βασικός πυλώνας κατά την εξέλιξη τους θα πρέπει να είναι η εξυπηρέτηση των ανθρώπινων αναγκών και η υπακοή στους δημιουργούς τους, αλλά το μέλλον είναι αβέβαιο καθώς η μηχανική ευφυΐα και η τεχνητή νοημοσύνη εξελίσσονται όλο και περισσότερο, ρομπότ εξυπηρετούν πλέον και στρατιωτικούς σκοπούς. Όλο αυτό φαίνεται να είναι παράδοξο, αλλά σύντομα θα γίνει πραγματικότητα, ο ρυθμός με τον οποίο εξαφανίζονται οι υπολογιστές θα συνοδεύεται από το ρυθμό με τον οποίο η τεχνολογία της πληροφορίας θα μεγαλώνει όλο και περισσότερο και θα καθορίζει τις ζωές μας.

Κεφάλαιο 2

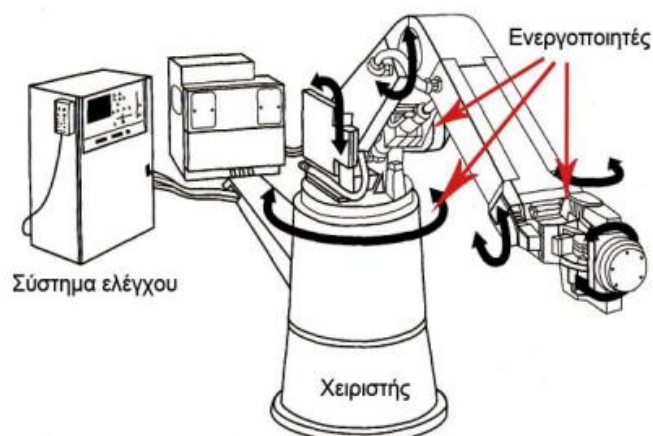
2.1 Ρομποτικοί Βραχίονες

2.1.1 Ορισμός

Ρομποτικός Βραχίονας είναι ένας τύπος μηχανικού βραχίονα, μια μηχανική κατασκευή που συνήθως είναι προγραμματιζόμενη, με λειτουργίες παρόμοιες με του ανθρώπινου βραχίονα. Ρομποτικός βραχίονας μπορεί να είναι το άθροισμα ενός μηχανισμού ή μπορεί να είναι και ένα μέρος ενός πιο πολυσύνθετου ρομπότ. Οπότε πιο αναλυτικά το μηχανικό αυτό σύστημα προσομοιώνει τις λειτουργίες ενός ανθρώπινου χεριού, γι' αυτό ονομάζεται και μηχανικός βραχίονας. Ο κάθε βραχίονας αποτελείται από τρία μέρη: α) τη βάση, β) τον κορμό και γ) τον καρπό. Η βάση είναι το στήριγμα του βραχίονα όπου πάνω σε αυτή βρίσκεται ο κορμός του, ο οποίος αποτελείται από διαδοχικούς συνδέσμους και αρθρώσεις. Το εργαλείο τελικής δράσης κινείται με μια ομάδα αρθρώσεων που ονομάζεται «καρπός». Η κάθε άρθρωση αντιστοιχεί σε έναν βαθμό ελευθερίας, οπότε ένας βραχίονας ο οποίος έχει n αριθμό αρθρώσεων θα έχει και n βαθμούς ελευθερίας. Η βάση είναι στερεωμένη στον χώρο εργασίας του ρομποτικού βραχίονα και πάνω σε αυτή συνδέονται διαδοχικά οι αρθρώσεις, οι σύνδεσμοι με κατάληξη το τελικό εργαλείο δράσης.

ΡΟΜΠΟΤΙΚΗ ΚΑΙ ΒΡΑΧΙΟΝΕΣ

Δομή ρομποτικού βραχίονα:



Εικόνα 2.1 Δομή του Ρομποτικού Βραχίονα

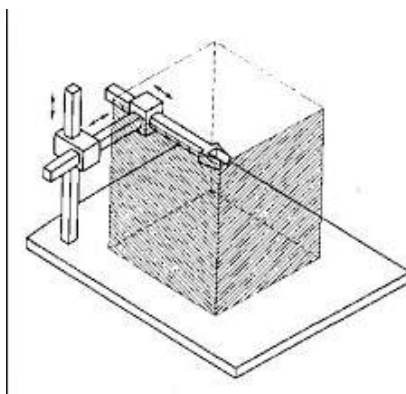
Οι σύνδεσμοι είναι ο «σκελετός» των ρομπότ και είναι στερεά σώματα. Οι αρθρώσεις είναι οι μηχανισμοί κίνησης μεταξύ των συνδέσμων. Τέλος, το εργαλείο τελικής δράσης είναι το εργαλείο με το οποίο το ρομπότ εκτελεί τη δοσμένη εργασία. Ανάλογα με την εργασία που εκτελεί ένας ρομποτικός βραχίονας, το εργαλείο τελικής δράσης μπορεί να προσαρμοστεί σε αυτή (π.χ. μπορεί να είναι εργαλείο ηλεκτροσυγκόλλησης, κατσαβίδι ή το συνηθέστερο να είναι μια αρπάγη). Επειδή η θέση ενός αντικειμένου και ο προσανατολισμός του καθορίζεται με έξι συντεταγμένες, τρεις για τον καθορισμό της θέσης ενός σημείου του αντικειμένου και τρεις για τον καθορισμό του προσανατολισμού του αντικειμένου γύρω από αυτό το σημείο, ο μηχανισμός καθοδήγησης πρέπει να έχει έξι ανεξάρτητες κινήσεις. Βάσει αυτού, ο μηχανισμός καθοδήγησης αποτελείται από τον μηχανισμό παραγωγής τροχιάς και το μηχανισμό προσανατολισμού. Ο μηχανισμός παραγωγής τροχιάς πραγματοποιεί την μετατόπιση ενός σημείου του αντικειμένου από μία ορισμένη θέση σε μία άλλη, ενώ ο μηχανισμός προσανατολισμού πραγματοποιεί τον προσανατολισμό του αντικειμένου για να καταλάβει την τελική του θέση.

2.2 Ταξινόμηση των Ρομποτικών Βραχιόνων βάσει της Γεωμετρικής τους διαμόρφωσης

Ο τύπος και η διαδοχή των αρθρώσεων ενός βραχίονα επιτρέπει την ταξινόμησή τους σε διάφορες κατηγορίες. Παρακάτω θα γίνει αναλυτική αναφορά στον κάθε τύπο.

2.2.1 Βραχίονες Καρτεσιανής Γεωμετρίας

Η καρτεσιανή γεωμετρία υλοποιείται με τρεις διαδοχικές πρισματικές αρθρώσεις. Οι άξονες των αρθρώσεων αυτών είναι ανα δύο κάθετοι μεταξύ τους (εικ 2.1). Η καρτεσιανή δομή παρέχει μεγάλη δυσκαμψία και σταθερή ακρίβεια σε ολόκληρο το χώρο εργασίας που ουσιαστικά είναι ένα παραλληλεπίπεδο. Βασικό μειονέκτημα της κατασκευής είναι η μειωμένη επιδεξιότητα κίνησης, λόγω της πρισματικής φύσης των αρθρώσεων. Ένα καρτεσιανό ρομπότ ή γραμμικό ρομπότ όπως αλλιώς ονομάζεται είναι ένα βιομηχανικό ρομπότ του οποίου οι τρεις κύριοι άξονες ελέγχου είναι γραμμικοί (δηλαδή μετακινούνται σε ευθεία γραμμή και δεν γυρίζουν) και είναι σε γωνία 90 μοιρών ο ένας με τον άλλο. Μεταξύ άλλων προτερημάτων, αυτή η μηχανική διάταξη απλοποιεί τον έλεγχο ρομπότ και την επίλυση του βραχίονα. Τα ρομπότ καρτεσιανών συντεταγμένων με τον οριζόντιο άξονα στηριγμένο και στα δύο άκρα του μερικές φορές ονομάζονται και ρομπότ γκάντρι (gantry robots). Μια συνηθισμένη εφαρμογή αυτού του τύπου ρομπότ είναι η μηχανή CNC ελέγχου με υπολογιστή, όπως επίσης χρησιμοποιείται και σε μηχανές επιλογής όπου ένας δείκτης μετακινείται σε ένα πεδίο X-Y ενώ ένα εργαλείο υψώνεται ή κατεβαίνει σε ένα επίπεδο για να ζωγραφίσει ένα ακριβές σχέδιο.



Εικ 2.2 Καρτεσιανός Ρομποτικός βραχίονας

2.2.2 Βραχίονες τύπου Gantry

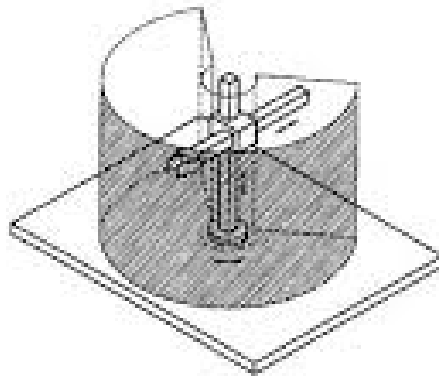
Οι βραχίονες τύπου Gantry είναι στην ουσία καρτεσιανοί βραχίονες, διαφέρουν όμως σημαντικά στον τρόπο προσεγγισμού του αντικείμενου ενδιαφέροντος. Ο βραχίονας τύπου Gantry προσεγγίζει το αντικείμενο από πάνω, τη στιγμή που ο απλός καρτεσιανός βραχίονας το προσεγγίζει από το πλάι. Άμεσες συνέπειες της διαφοροποίησης αυτής είναι η αύξηση του χώρου εργασίας και της δυσκαμψίας, καθώς επίσης και η δυνατότητα χειρισμού μεγάλων και βαριών αντικειμένων.



Εικ 2.3 Βραχίονας τύπου Gantry

2.2.3 Κυλινδρικοί Βραχίονες

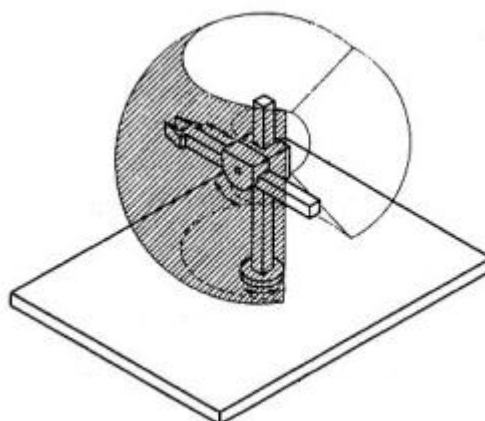
Στους κυλινδρικούς βραχίονες η πρώτη πρισματική άρθρωση της καρτεσιανής δομής έχει αντικατασταθεί από μια περιστροφική άρθρωση. Οι συγκεκριμένοι βραχίονες χαρακτηρίζονται από καλή δυσκαμψία, όμως η ακίβεια της θέσης του καρπού μειώνεται καθώς η οριζόντια μετατόπιση αυξάνεται. Στην περίπτωση αυτή ο χώρος εργασίας είναι τμήμα κυλίνδρου, σημαντικό μειονέκτημα της συγκεκριμένης γεωμετρίας είναι ότι ο βραχίονας εισέρχεται στο χώρο εργασίας και τον περιορίζει.



Εικ. 2.4 Κυλινδρικός Βραχίονας

2.2.4 Σφαιρικοί Βραχίονες

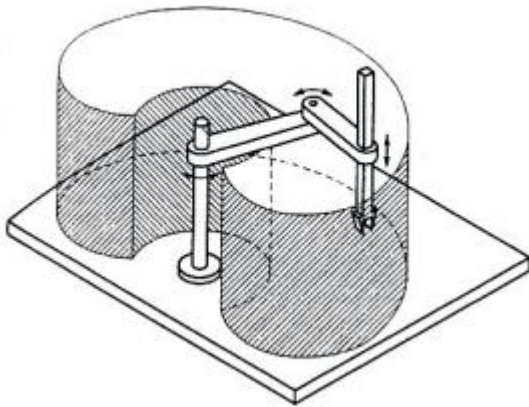
Στους βραχίονες αυτούς αντικαθίσταται πλέον και η δεύτερη πρισματική άρθρωση της καρτεσιανής δομής με περιστροφική (εικ 2.4) . Η μηχανολογική πολυπλοκότητα αυξάνει , ενώ η δυσκαμψία μειώνεται . Επιπλέον η ακρίβεια του καρπού μειώνεται με την αύξηση της ακτινικής απόστασης. Ο χώρος εργασίας είναι ένα τμήμα σφαίρας το οποίο εμπεριέχει και ένα μέρος της βάσης με άμεση συνέπεια τη δυνατότητα χειρισμού αντικειμένων που βρίσκονται στο έδαφος.



Εικ 2.5 Σφαιρικός βραχίονας

2.2.5 Βραχίονες τύπου SCARA

Η γεωμετρία τύπου SCARA παρουσιάζει κάποιες ιδιομορφίες, είναι ειδική περίπτωση και περιλαμβάνει δύο περιστροφικές και μια πρισματική άρθρωση, τοποθετημένες κατά τέτοιο τρόπο ώστε οι άξονες της κίνησης να είναι παράλληλοι μεταξύ τους. Το όνομα SCARA προέρχεται από τα αρχικά των λέξεων selective compliance assembly robot arm. Η συγκεκριμένη γεωμετρία παρέχει μεγάλη δυσκαμψία σε κατακόρυφη φόρτιση και ελαστικότητα σε οριζόντια. Η ακρίβεια τοποθέτησης του καρπού μειώνεται με την αύξηση της απόστασης του από τον άξονα της πρώτης άρθρωσης.



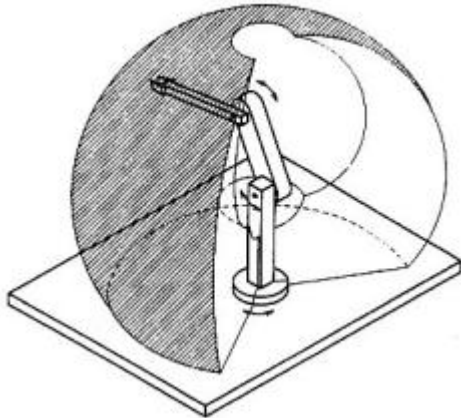
Εικ. 2.6 Βραχίονας τύπου SCARA



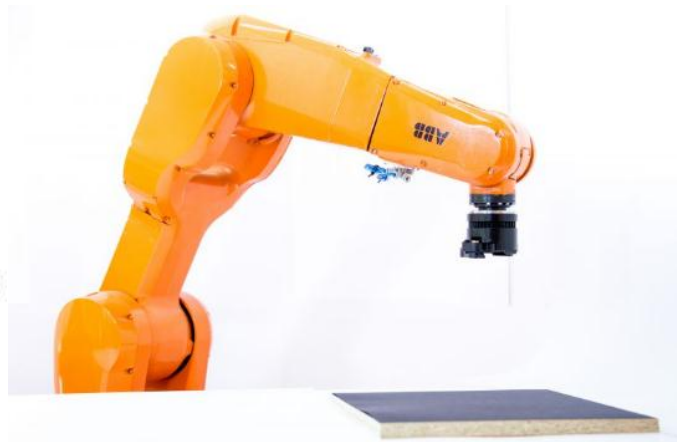
Εικ. 2.7 Βραχίονας SCARA από την Mitsubishi

2.2.6 Ανθρωπομορφικοί Βραχίονες

Η ανθρωπομορφική γεωμετρία υλοποιείται με τρεις διαδοχικές περιστροφικές αρθρώσεις. Πιο συγκεκριμένα ο άξονας περιστροφής της πρώτης άρθρωσης είναι κατακόρυφος και κάθετος στους άξονες περιστροφής των επόμενων δύο αρθρώσεων, οι οποίοι είναι παράλληλοι μεταξύ τους. Η συγκεκριμένη δομή παρέχει τη μεγαλύτερη επιδεξιότητα από όλες τις προηγούμενες, καθώς όλες οι αρθρώσεις είναι περιστροφικές. Ωστόσο η ακρίβεια του καρπού δεν είναι σταθερή εντός του χώρου εργασίας που έχει τη μορφή σφαίρας.



Ανθρωπομορφικός Βραχίονας εικ 2.8

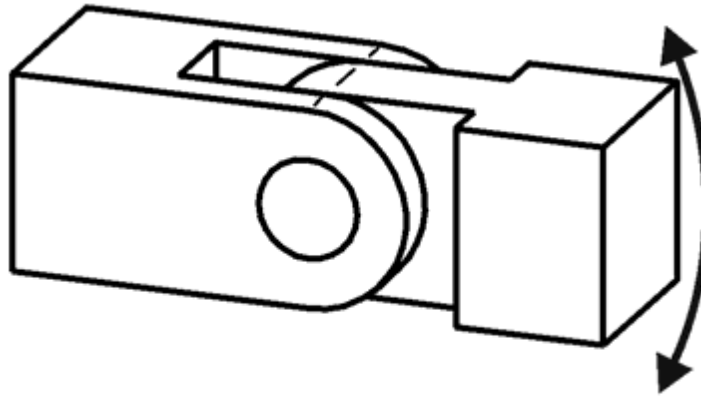


Ανθρωπομορφικός Βραχίονας από την ABB εικ 2.9

2.3 Είδη αρθρώσεων Ρομποτικού βραχίονα

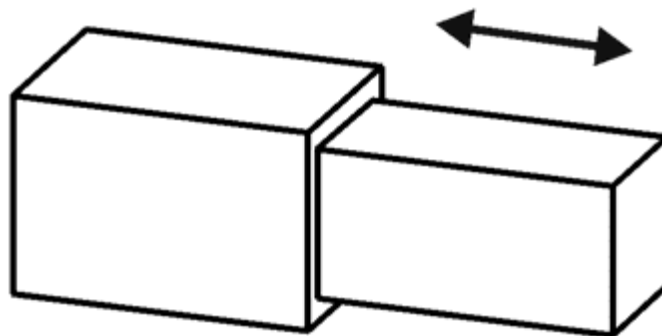
Αρθρώσεις είναι οι διατάξεις οι οποίες συνδέουν δύο συνδέσμους μεταξύ τους, με τη βοήθεια των ενεργοποιητών κινούν τους συνδέσμους και κατα συνέπεια τον βραχίονα. Στο χώρο των τριών διαστάσεων (μήκος, πλάτος και ύψος) οι κινήσεις που μπορούν να πραγματοποιηθούν μεταξύ δύο στερεών σωμάτων εκ των οποίων το ένα θεωρείται ακίνητο, είναι τρεις μεταφορικές κινήσεις στις διευθύνσεις των αξόνων του καρτεσιανού συστήματος και τρεις περιστροφικές κινήσεις. Ένας βραχίονας με τη χρήση δύο αρθρώσεων μπορεί να κινηθεί μεταξύ των δύο αξόνων X, Y (μήκος και πλάτος), ενώ με τρεις αρθρώσεις μπορεί να κινηθεί στο χώρο μεταξύ τριών αξόνων X, Y, Z (μήκος, πλάτος και ύψος).

Οι αρθρώσεις ενός βραχίονα χωρίζονται σε δύο κατηγορίες: τις στροφικές και τις πρισματικές. Υπάρχουν και άλλα είδη αρθρώσεων όπως είναι οι κυλινδρικές, οι ελεύθερες, οι σφαιρικές και οι αρθρώσεις κύλησης οι οποίες προκύπτουν από τη σύνδεση των δύο βασικών αρθρώσεων. Στα παρακάτω σχήματα που ακολουθούν μπορούμε να διακρίνουμε τα διάφορα είδη των αρθρώσεων.



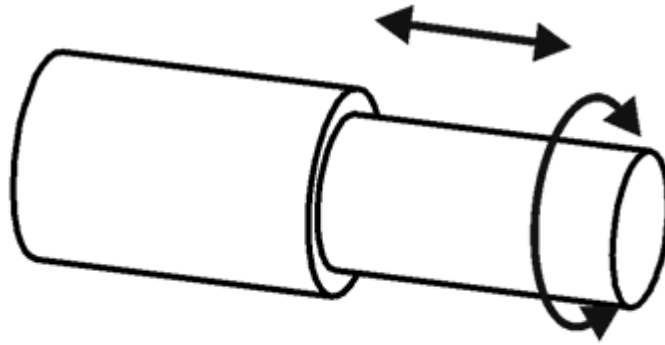
Στροφική

Σχήμα 2.1. Στροφική άρθρωση



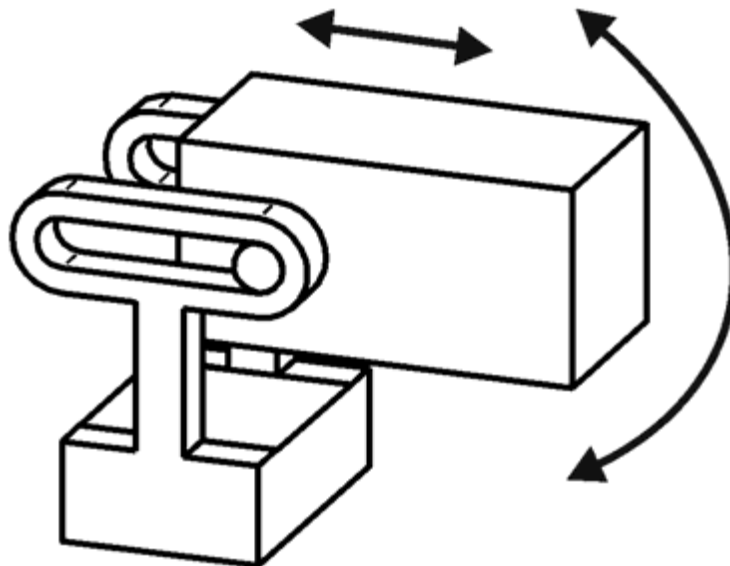
Πρισματική

Σχήμα 2.2 Πρισματική άρθρωση



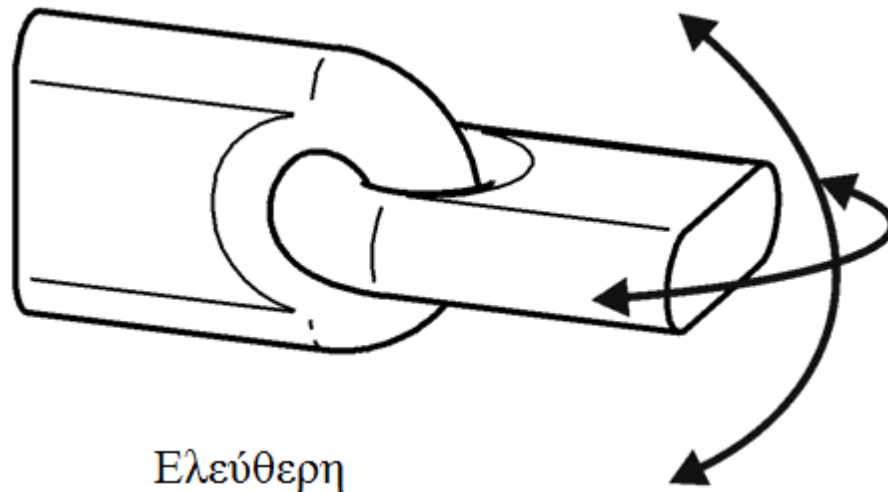
Κυλινδρική

Σχήμα 2.3. Κυλινδρική άρθρωση

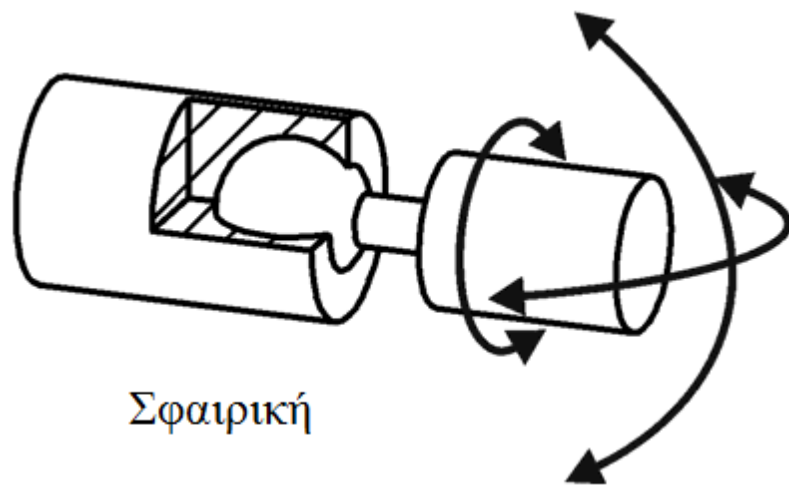


Κύληση

Σχήμα 2.4. Άρθρωση Κύληση



Σχήμα 2.5 Ελεύθερη Άρθρωση



Σχήμα 2.6. Σφαιρική άρθρωση

Στην πλειοψηφία τους οι ρομποτικοί μηχανισμοί κατασκευάζονται με συνδυασμό απλών στροφικών ή πρισματικών αρθρώσεων. Αυτό οφείλεται στο ότι είναι πολύ δύσκολο να κατασκευαστεί άρθρωση με πολλούς βαθμούς ελευθερίας και επιπλέον να ελεγχθεί στη συνέχεια με κάποιον επενεργητή .

2.4 Το σύστημα ελέγχου του Ρομποτικού βραχίονα

Το σύστημα ελέγχου μας δίνει τη δυνατότητα να προγραμματίσουμε την εργασία την οποία θα εκτελεί ο βραχίονας και ελέγχει την κίνησή του κατά τη διάρκεια της εκτέλεσης της εργασίας που έχουμε προγραμματίσει. Αποτελείται από δύο μέρη το υλικό (hardware) και το λογισμικό (software).

2.4.1 Υλικό τμήμα

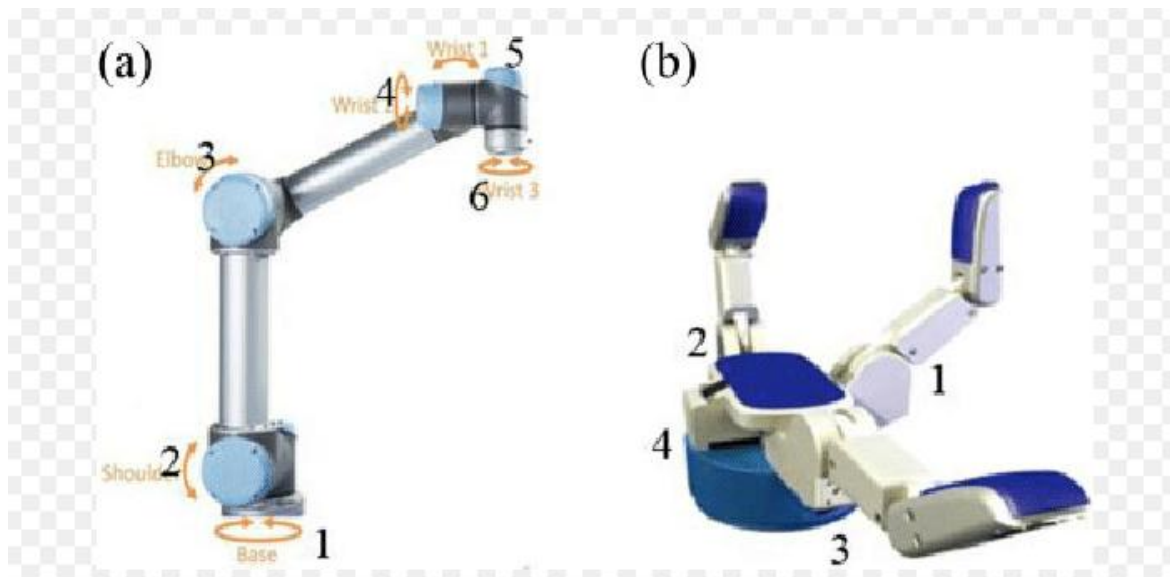
Το υλικό τμήμα περιλαμβάνει : α) έναν υπολογιστή στον οποίο εγκαθίσταται το λογισμικό του ρομποτικού ελεγκτή και ο οποίος αποθηκεύει το πρόγραμμα των εντολών που πρέπει να εκτελεστεί για μια συγκεκριμένη εργασία β) τα ηλεκτρονικά ελέγχου, τα οποία υλοποιούν το βρόχο ελέγχου για τον κινητήρα της κάθε άρθρωσης και γ) τα ηλεκτρονικά επικοινωνίας, τα οποία δίνουν τη δυνατότητα σύνδεσης του ελεγκτή με τα αισθητήρια των αρθρώσεων και του άκρου, δ) τους ενισχυτές ισχύος, οι οποίοι χρησιμοποιούνται για την ενίσχυση του σήματος ελέγχου στο επίπεδο που απαιτείται από τους κινητήρες των αρθρώσεων.

2.4.2 Λογισμικό τμήμα

Το Λογισμικό τμήμα περιλαμβάνει : α) ένα προγραμματιστικό περιβάλλον με χρήση κάποιας γλώσσας υψηλού επιπέδου, β) το κινηματικό λογισμικό, το οποίο με γνωστή τη θέση των αρθρώσεων καθορίζει τη θέση του άκρου και αντίστροφα, δηλαδή με δεδομένη την επιθυμητή θέση του άκρου καθορίζει τις αντίστοιχες θέσεις κάθε κινητήρα άρθρωσης, γ) το λογισμικό ελέγχου, το οποίο υλοποιεί τον αλγόριθμο ελέγχου της κάθε άρθρωσης, δηλαδή με γνωστή την κατάσταση του κάθε κινητήρα (θέση, ταχύτητα), καθορίζει το απαιτούμενο σήμα ελέγχου ώστε να κινηθεί ο κινητήρας προς την επιθυμητή θέση, δ) το λογισμικό δυναμικών όρων, το οποίο χρησιμοποιεί δεδομένα για τις δυναμικές παραμέτρους του βραχίονα και την κατάστασή του, ώστε να προσαρμόσει το σήμα ελέγχου με στόχο την καλύτερη λειτουργία του ρομποτικού βραχίονα, ε) το βοηθητικό λογισμικό, το οποίο φροντίζει για την παρακολούθηση της λειτουργίας του βραχίονα και ενημερώνει το χρήστη για λάθη.

2.5 Βαθμοί ελευθερίας

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. Σε αυτό το σημείο κρίνεται σκόπιμο να επισημάνουμε τη διαφορά που υπάρχει ανάμεσα στους βαθμούς κινητικότητας του βραχίονα και τους βαθμούς ελευθερίας που απαιτούνται για την εκτέλεση της εργασίας. Για έναν βραχίονα το πλήθος των βαθμών κινητικότητας είναι σταθερό και ισούται 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 βραχίονας. Για την γενική περίπτωση που θέλουμε να τοποθετήσουμε και να προσανατολίσουμε ένα αντικείμενο στο χώρο απαιτούνται 6 βαθμοί ελευθερίας (οι 3 για να τοποθετήσουμε ένα σημείο του αντικειμένου στο χώρο και 3 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 απαιτείται μέχρι 6 βαθμούς ελευθερίας.



Σχήμα 2.7 Κάθε αριθμός αντιστοιχεί σε ένα βαθμό ελευθερίας

2.6 Ενεργοποιητές

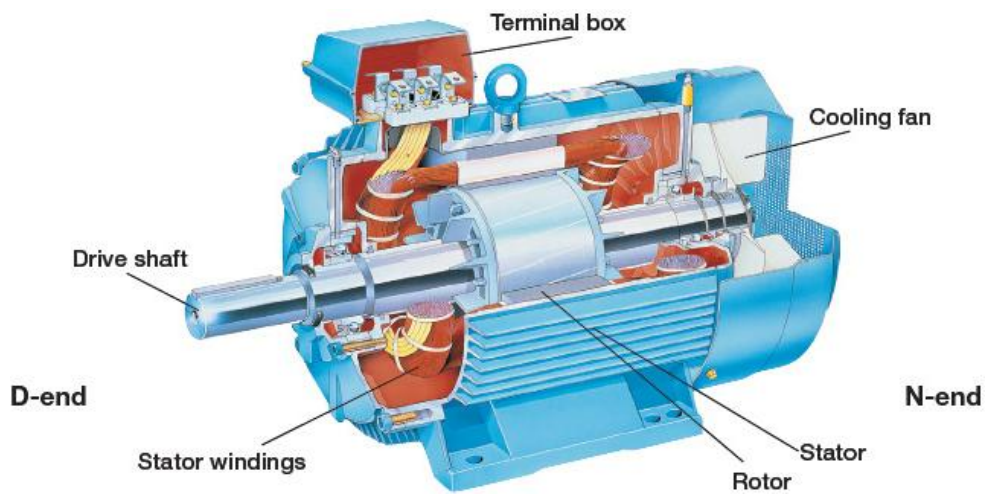
Οι ενεργοποιητές είναι συσκευές που ενεργοποιούν την κίνηση των τμημάτων του βραχίονα. Δέχονται σαν είσοδο ένα ηλεκτρικό σήμα από έναν ηλεκτρονικό υπολογιστή προκειμένου να ξεκινήσει η λειτουργία τους. Τέτοιες συσκευές είναι τα υδραυλικά συστήματα, τα πνευματικά συστήματα, οι κινητήρες συνεχούς ρεύματος, οι βηματικοί κινητήρες και οι servo-κινητήρες. Κάθε κατηγορία έχει πλεονεκτήματα και μειονεκτήματα, γι' αυτό σε κάθε διεργασία χρησιμοποιείται ο κατάλληλος τύπος. Έτσι υπάρχουν διάφοροι ενεργοποιητές των οποίων τη λειτουργία θα αναλύσουμε σε αυτό το κεφάλαιο. Οι κινητήρες συνεχούς ή εναλλασσόμενου ρεύματος (AC-DC Motors) με γωνία στρέψης τις 360 μοίρες, έχουν μεγάλη ροπή. Ένα μεγάλο μειονέκτημα όμως είναι ότι δεν υπάρχει έλεγχος της θέσης του άξονά τους και γι' αυτό χρειάζονται εξωτερικό κύκλωμα οδήγησης. Οι βηματικοί κινητήρες έχουν και αυτοί γωνία στρέψης τις 360 μοίρες, κι έχουν μικρή ροπή. Όπως και στην προηγούμενη κατηγορία κινητήρων έτσι και οι βηματικοί ή step-motors δεν έχουν κάποιο κύκλωμα προκειμένου να ελεγχτεί η θέση τους, έτσι χρειάζονται και αυτοί εξωτερικό κύκλωμα οδήγησης. Οι κινητήρες servo έχουν γωνία περιστροφής τις 180 μοίρες, έχουν μεγάλη ροπή και διαθέτουν ενσωματωμένο κύκλωμα οδήγησης ώστε να μπορούμε να ελέγξουμε την θέση τους. Αυτό γίνεται με τη χρήση ενός ποτενσιόμετρου το οποίο μαζί με το κύκλωμα είναι ενσωματωμένα στον σερβοκινητήρα. Επίσης υπάρχουν και τα υδραυλικά συστήματα όπως και τα πνευματικά τα οποία είναι έμβολα που με τη χρήση υγρού κυρίως λαδιού ή πεπιεσμένου αέρα στην περίπτωση των πνευματικών συστημάτων, κινούν την άρθρωση κατάλληλα. Στην περίπτωση των υδραυλικών ενεργοποιητών, λόγω της υδραυλικής πίεσης η ροπή τους είναι μεγαλύτερη από αυτή των προηγούμενων ενεργοποιητών-κινητήρων και χρησιμοποιούνται συνήθως σε εργασίες, όπου ο βραχίονας έχει να κάνει με βαριά αντικείμενα.

2.6.1 Κινητήρες AC

Οι κινητήρες AC είναι οι ηλεκτρικοί κινητήρες εναλασσόμενου ρεύματος . Οι πολυφασικοί κινητήρες εναλασσόμενου ρεύματος είναι κατά βάση μηχανές σταθερής ταχύτητας, αλλά διαφοροποιούνται ως προς κάποια σχεδιαστικά στοιχεία τους με συνέπεια να διαμορφώνονται τέσσερις βασικές υποκατηγορίες των κινητήρων εναλασσόμενου ρεύματος. Στον πιο συνήθη τύπο (DESIGN A,B) έχουμε κανονικές τιμές ροπής και ρεύματος εκκίνησης και χαμηλή ολίσθηση. Στον δεύτερο τύπο (DESIGN C) έχουμε υψηλή ροπή εκκίνησης με κανονική ένταση ρεύματος εκκίνησης και χαμηλή ολίσθηση. Στον τρίτο τύπο (DESIGN D) έχουμε επίσης υψηλή ροπή εκκίνησης αλλά χαμηλό ρεύμα εκκίνησης, ενώ η ολίσθηση είναι υψηλή. Στον τέταρτο τύπο (DESIGN F) έχουμε χαμηλή ροπή και ρεύμα εκκίνησης αλλά και χαμηλή ολίσθηση. Οι κινητήρες εναλασσομένου ρεύματος καλύπτουν το μεγαλύτερο όγκο εφαρμογών στη βιομηχανία. Το μεγάλο τους πλεονέκτημα σε σχέση με τους κινητήρες συνεχούς ρεύματος είναι μπορούν να προσφέρουν πολύ μεγαλύτερη ισχύ για το ίδιο μέγεθος κινητήρα. Επίσης ο βαθμός απόδοσης τους είναι αρκετά μεγάλος και η οικονομικότητά τους ως προς την κατανάλωση ρεύματος ικανοποιητική, με συνέπεια να επιλέγονται για εφαρμογές όπου έχουμε μηχανήματα πολύ μεγάλης ισχύος που εργάζονται στο βιομηχανικό πεδίο σε συνεχή βάση. Τέτοια παραδείγματα αποτελούν οι αντλίες και οι συμπιεστές με πιο χαρακτηριστικές περιπτώσεις τα πολύ μεγάλα φυγοκεντρικά μηχανήματα (είτε αντλίες είτε συμπιεστές) η ισχύς των οποίων μπορεί να είναι της τάξης μέχρι και ενός (η περισσότερων) MW. Ωστόσο και τα βιομηχανικά μηχανήματα μικρής ισχύος και όχι ειδικών απαιτήσεων στην πλειοψηφία τους ενεργοποιούνται από ηλεκτροκινητήρες εναλασσομένου ρεύματος.



Εικόνα 2.10 AC motor



Σχήμα 2.8 AC-motor



Εικόνα 2.11 AC κινητήρες σε διάφορα μεγέθη

2.6.2 Κινητήρες DC-Κινητήρες συνεχούς ρεύματος

Οι DC κινητήρες (εικόνες 2.12 και 2.13) έχουν μεγάλη ροπή στρέψης και μπορούν να διαχειριστούν βαρέα φορτία, δεν έχουν μειωτήρα για μείωση των στροφών και δεν υπάρχει έλεγχος θέσης του άξονα που σημαίνει ότι χρειάζεται κύκλωμα ελέγχου κλειστού βρόχου για τον έλεγχο θέσης του άξονα. Οι Κινητήρες DC είναι συνεχείς ενεργοποιητές που μετατρέπουν την ηλεκτρική ενέργεια σε μηχανική ενέργεια. Ο κινητήρας DC επιτυγχάνει αυτό με τη δημιουργία μιας συνεχούς γωνιακής περιστροφής που μπορεί να χρησιμοποιηθεί για την περιστροφή αντλιών, ανεμιστήρων, συμπιεστών, τροχών κλπ. Ο κινητήρας DC ή κινητήρας συνεχούς ρεύματος για να του δώσουμε τον πλήρη τίτλο του είναι ο συνηθέστερα χρησιμοποιούμενος κινητήρας για την παραγωγή συνεχούς κίνηση και της οποίας η ταχύτητα περιστροφής μπορεί εύκολα να ελεγχθεί, καθιστώντας τους ιδανικά για χρήση σε διάφορες εφαρμογές. Ένας ηλεκτροκινητήρας συνεχούς ρεύματος αποτελείται από δύο μέρη, έναν "στάτορα" που είναι το σταθερό τμήμα και έναν "περιστροφέα" που είναι το περιστρεφόμενο τμήμα.

Οι κινητήρες DC έχουν σχεδόν γραμμικά χαρακτηριστικά με την ταχύτητα περιστροφής τους να καθορίζεται από την εφαρμοζόμενη τάση συνεχούς ρεύματος και η ροπή εξόδου τους να καθορίζεται από το ρεύμα που ρέει μέσω των περιελίξεων του κινητήρα. Η ταχύτητα περιστροφής κάθε κινητήρα DC μπορεί να ποικίλει από μερικές περιστροφές ανά λεπτό (σ.α.λ.) σε πολλές χιλιάδες στροφές ανά λεπτό, καθιστώντας τις κατάλληλες για ηλεκτρονικές, αυτοκινητοβιομηχανικές ή ρομποτικές εφαρμογές. Συνδυάζοντας τους με κιβώτια ταχυτήτων (Gearbox), η ταχύτητα εξόδου τους μπορεί να μειωθεί, ενώ παράλληλα αυξάνει την ισχύ ροπής του κινητήρα σε υψηλή ταχύτητα.



Εικόνα 2.12 DC Motor



Εικόνα 2.13 DC Motor



Εικόνα 2.14 DC Motor



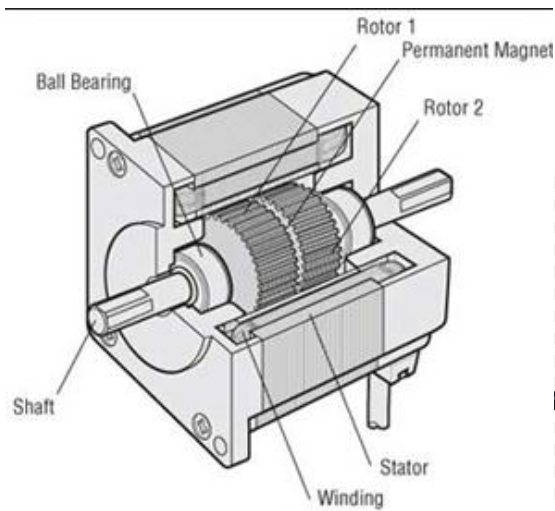
Εικόνα 2.15 Βιομηχανικός DC Κινητήρας

2.6.3 Βηματικοί Κινητήρες –Stepper Motors

Οι βηματικοί κινητήρες (εικόνες 2.16-2.18) αποτελούνται από έναν ρότορα μαλακού σιδήρου με οδοντώσεις και έναν στάτη με τέσσερα ζεύγη ηλεκτρομαγνητών. Οι βηματικοί κινητήρες σε σχέση με τους κινητήρες συνεχούς τάσης δεν χρειάζονται φρένα για να υπάρξει επιβράδυνση ή ακινητοποίηση, σε χαμηλή ταχύτητα παράγουν μεγάλη ροπή, έχουν αξιόπιστη λειτουργία καθώς δεν απαιτείται αλλαγή στις κινούμενες ηλεκτρικές επαφές (καρβουνάκια), μπορούν επίσης να έχουν μεγάλο εύρος ταχύτητας περιστροφής του άξονα. Έχουν όμως θορυβώδη λειτουργία, αδυναμία περιστροφής του άξονα σε υψηλές ταχύτητες όταν αυτός έχει φορτίο και κατά την μετακίνηση φορτίων μεγάλου βάρους μπορεί να μη σταματήσει άμεσα ο άξονας και να υπάρξει ταλάντωση. Επίσης χρειάζονται κύκλωμα αρχικοποίησης προκειμένου να υπάρξει έλεγχος θέσης του άξονα. Οι βηματικοί κινητήρες χρησιμοποιούν συνδυασμό ηλεκτρικών σημάτων για την περιστροφή τους. Τα ηλεκτρικά σήματα διοχετεύονται στο στάτη που αποτελείται από τέσσερα ζεύγη ηλεκτρομαγνητών A-A' B-B' C-C' D-D'. Για να κινηθεί ο ρότορας τα ηλεκτρικά σήματα εφαρμόζονται διαδοχικά σε καθένα απ' τα ζεύγη του στάτη κάθε φορά που ολοκληρώνεται μια σειρά παλμών προς όλους τους μαγνήτες του στάτη ο ρότορας κάνει μια στροφή X μοιρών.

Στην αγορά υπάρχουν τρία είδη βηματικού κινητήρα:

- Βηματικός μόνιμου μαγνήτη(Permanent magnet)
- Βηματικός μεταβλητής μαγνητικής αντίστασης (Variable reluctance)
- Βηματικός υβριδικού τύπου(Hybrid)



Motor Structural Diagram: Cross-Section Parallel to Shaft

Εικόνα 2.16 stepper motor



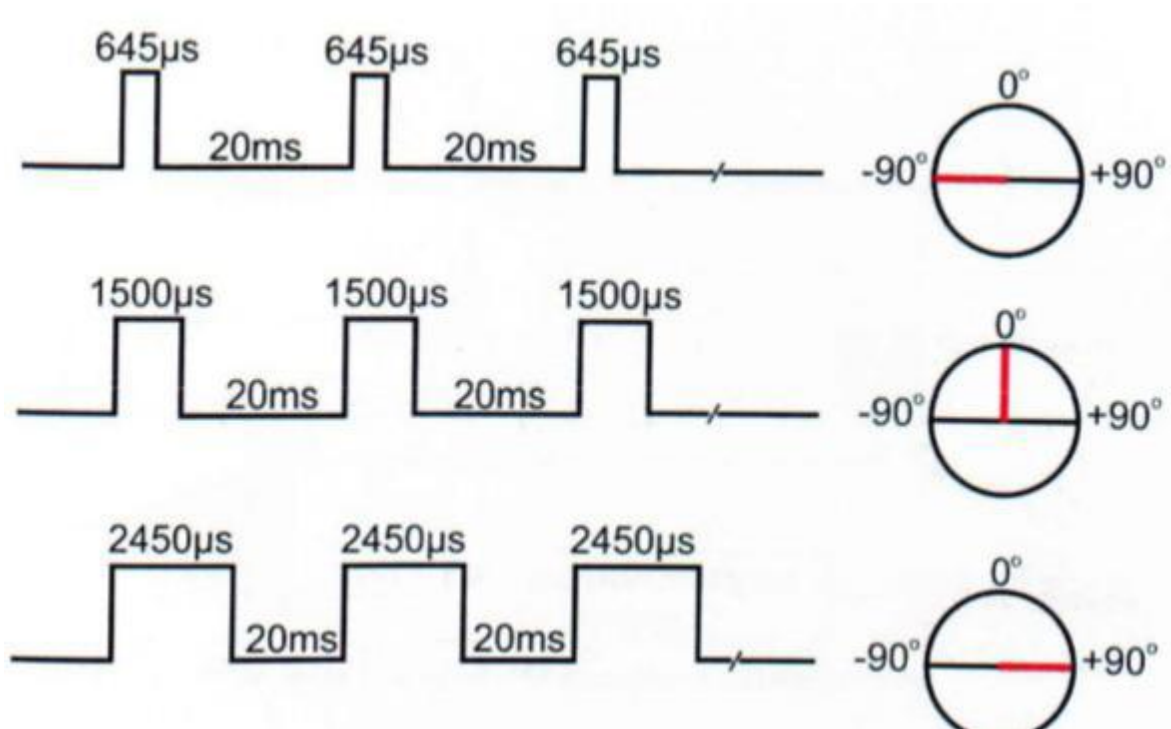
Εικόνα 2.17 stepper motor



Εικόνα 2.18 Διάφορα μεγέθη βηματικών κινητήρων ανάλογα με την εφαρμογή

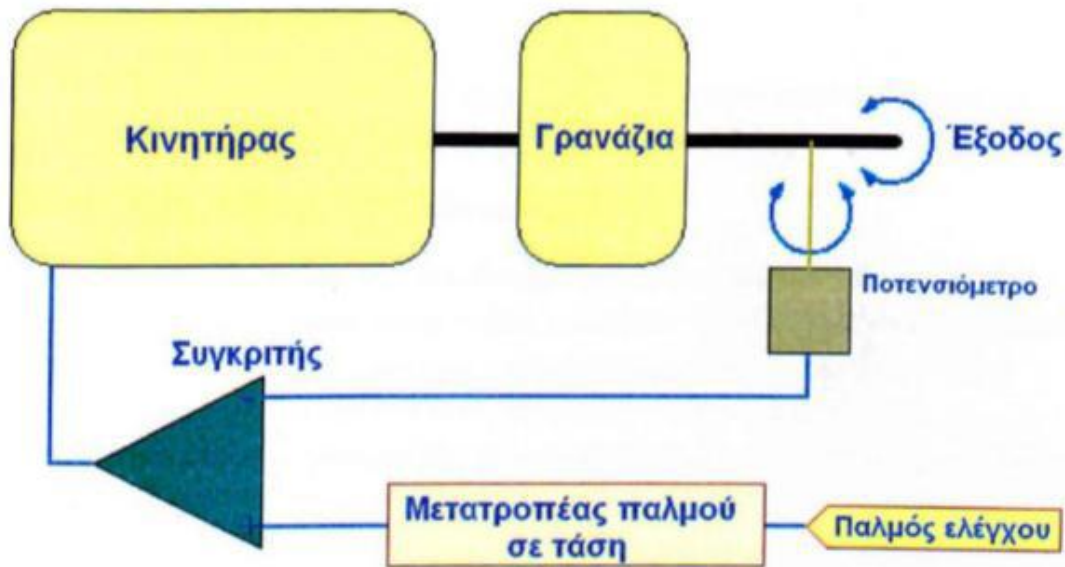
2.6.4 Κινητήρες Σέρβο –Servo Motors

Οι κινητήρες Servo (εικόνες 2.19-2.21) είναι συσκευές που αποτελούνται από ένα ηλεκτροκινητήρα συνεχούς ρεύματος, ένα ηλεκτρονικό κύκλωμα που ελέγχει τη θέση του τελικού άξονα κίνησης και ένα κιβώτιο υποβιβασμού της σχέσης μετάδοσης του κινητήρα όπου φαίνεται αρκετά καλά στην (εικόνα 2.21). Ο άξονας μπορεί να μετακινηθεί σε διάφορες θέσεις αν αποσταλεί στον Servo ένα κωδικοποιημένο σήμα. Πιο συγκεκριμένα, ο ελεγκτής διαμορφώνει και μεταδίδει στον Servo ηλεκτρικούς παλμούς όπου φαίνεται και στο(σχήμα 2.8), ανάλογα με τη θέση στην οποία πρέπει να περιστραφεί ο άξονας του. Οι ηλεκτρικοί παλμοί λαμβάνονται και αποκωδικοποιούνται από το Servo, με τη βοήθεια του κυκλώματος ελέγχου που περιλαμβάνεται σε αυτό.



Σχήμα 2.9 Οριακές τιμές κινητήρα servo

Στη συνέχεια, μετά την αποκωδικοποίηση του ηλεκτρικού σήματος των παλμών, το κύκλωμα ελέγχου του σέρβο οδηγεί τον κινητήρα του στην κατάλληλη θέση (σχήμα 2.9). Όσο υπάρχει αυτό το σήμα στην γραμμή εισόδου του Servo τόσο αυτός θα διατηρεί τον άξονα του σε μια συγκεκριμένη θέση. Όταν αλλάζει το σήμα προκαλεί τον Servo να μεταβάλει την γωνία του άξονα. Στη μέθοδο ελέγχου ανοιχτού βρόχου που χρησιμοποιείται στους κινητήρες Servo, υπάρχει επικοινωνία μιας κατεύθυνσης, από τον ελεγκτή προς το σύστημα που πρόκειται να ελεγχθεί. Δεν υπάρχει ανατροφοδότηση πληροφορίας από το σύστημα προς τον ελεγκτή. Για το λόγο αυτό η μέθοδος ανοιχτού βρόχου δεν χρησιμοποιεί αισθητήρες όπως η μέθοδος κλειστού βρόχου. Παρόλα αυτά, διαβάζοντας την τάση από τον κεντρικό ακροδέκτη του ποτενσιόμετρου που είναι ενσωματωμένο στον κινητήρα, μπορεί να χρησιμοποιηθεί και σαν συσκευή εισόδου, παρέχοντας ένα αισθητήριο θέσης. Αυτό μπορεί να φανεί πολύ χρήσιμο σε εφαρμογές όπως ανίχνευση συγκρούσεων σε ανενεργούς ρομποτικούς βραχίονες, μέτρηση ροπής του κινητήρα / οπτικό έλεγχο με ανάδραση κλπ. Σε πρακτικές εφαρμογές οι servo-κινητήρες χρησιμοποιούνται σε τηλεχειριζόμενα αεροπλάνα, αυτοκίνητα και στην ρομποτική. Οι κινητήρες αυτοί έχουν ένα εξαιρετικά μικρό μέγεθος αλλά είναι αρκετά ισχυροί για το μέγεθος τους.



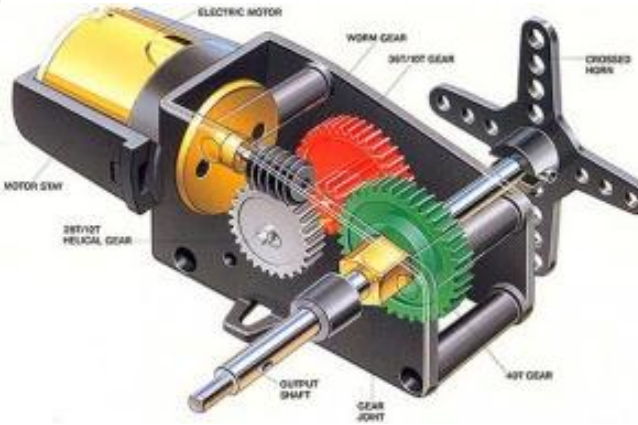
Σχήμα 2.10 Αναπαράσταση εσωτερικού κυκλώματος οδήγησης servo-κινητήρα



Εικόνα 2.19 Διάφορα μεγέθη κινητήρων servo



Εικόνα 2.20 Servo-κινητήρας



Εικόνα 2.21 Το εσωτερικό του servo κινητήρα

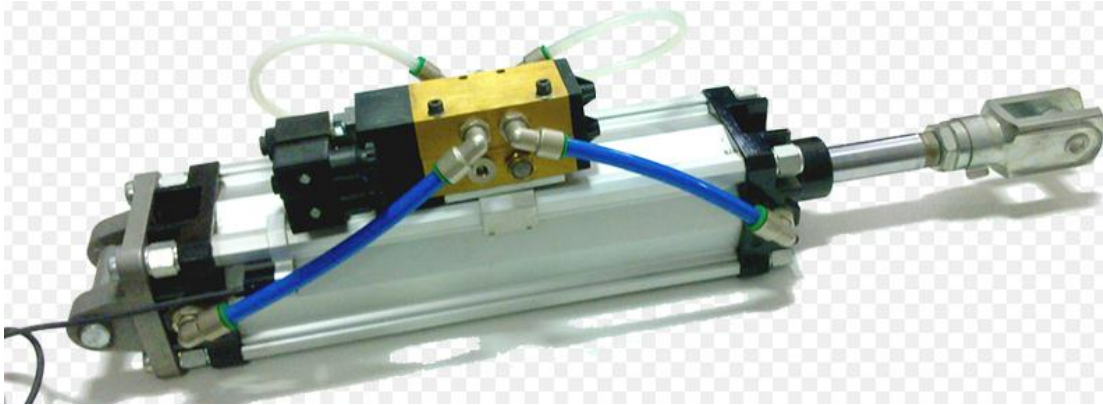
2.6.5 Πνευματικά έμβολα ενεργοποιητές

Τί είναι τα πνευματικά συστήματα? Είναι τα συστήματα αυτά που χρησιμοποιούν πεπιεσμένο αέρα για την εκτέλεση μιας εργασίας, που μπορεί να είναι επαναλαμβανόμενη, δύσκολη, να απαιτεί ακρίβεια ή ταχύτητα. Τα πνευματικά συστήματα είναι ευρέως διαδεδομένα στην βιομηχανία, λόγω του χαμηλού κόστους. Τα πνευματικά συστήματα επιτρέπουν την ανάπτυξη υψηλών δυνάμεων και ροπών, έχουν γρήγορη δράση, εύκολη συντήρηση, ενώ παράγουν μεγάλες δυνάμεις είναι πολύ οικονομικά σε σύγκριση με άλλα συστήματα. Ο τρόπος λειτουργίας τους είναι πολύ απλός, πεπιεσμένος αέρας εισέρχεται από τη μια πλευρά του εμβόλου προκαλώντας την επιμήκυνση του, έπειτα επανέρχεται στην αρχική του θέση, είτε με την εφαρμογή μιας αντίθετης δύναμης, είτε με τη χρήση κάποιου ελατηρίου. Η Ευελιξία είναι ένα σημαντικό πλεονέκτημα των ρομπότ με πνευματικούς ενεργοποιητές. Ένα σημαντικό χαρακτηριστικό οπότε είναι η ευελιξία που είναι εγγενής στο σύστημα. Αυτό συμβαίνει επειδή οι θάλαμοι αέρα μπορούν να συμπιεστούν, αυτό οδηγεί σε ένα κρίσιμο πλεονέκτημα έναντι των ρομπότ που κινούνται με κινητήρες. Σε περίπτωση σύγκρουσης με εμπόδιο, η προαναφερθείσα ικανότητα συμπίεσης του αέρα αρχίζει αμέσως να ισχύει. Η συμπεριφορά του ρομπότ είναι εγγενώς ευέλικτη και συμβάλλει σε έναν ορισμένο βαθμό ασφάλειας.

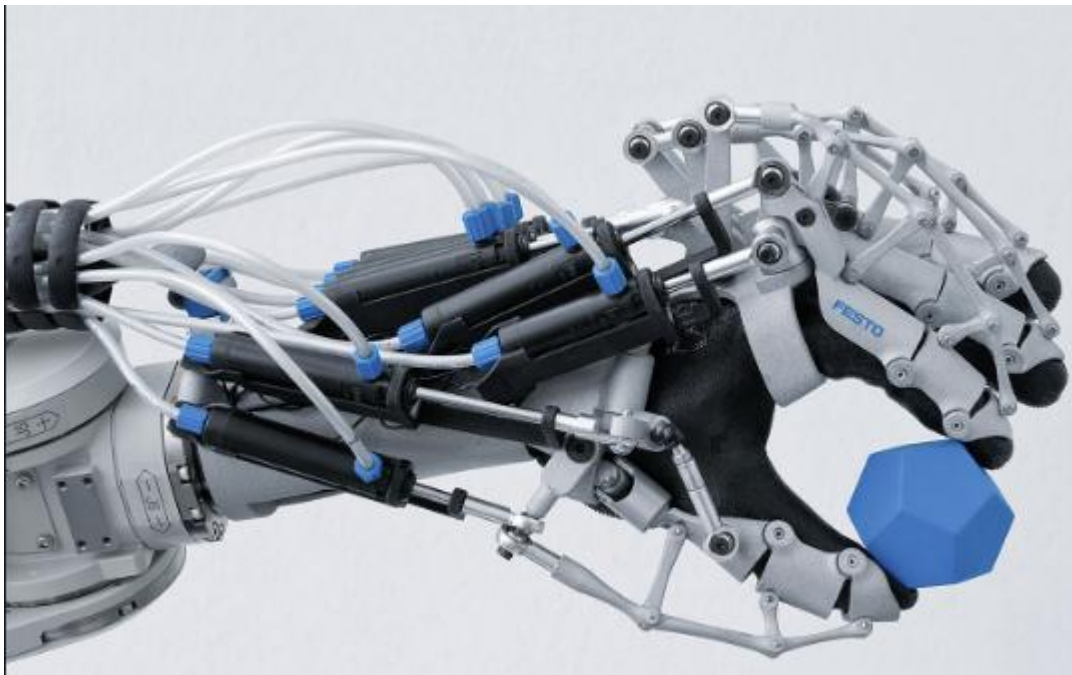
Ο ρυθμιστής θέσης φυσικά θα καταστήσει το πραγματικό εύκαμπτο και δονητικό πνευματικό σύστημα πιο άκαμπτο. Δεδομένου ότι ο ελεγκτής ανταποκρίνεται μόνο μετά από μια σύγκρουση, ο πιθανός κίνδυνος σε περίπτωση σύγκρουσης μειώνεται. Η στάθμη πίεσης του κάθε άξονα μπορεί να ρυθμιστεί. Οι ροπές κάθε άρθρωσης μπορούν να ελέγχονται σύμφωνα με την εφαρμογή. Με αυτό τον τρόπο, το ρομπότ μπορεί να εκπληρώσει τα καθήκοντα του και να είναι αποδοτικό στην κάθε περίπτωση. Για παράδειγμα, εάν υποτίθεται ότι κρατάει ένα ελαφρύ αντικείμενο όπως ένα στυλό σταθερά σε μια θέση, έτσι και στην περίπτωση όπου εκτελεί ισχυρή κίνηση, όμως οι ροπές μπορούν επίσης να ρυθμιστούν ακριβώς έτσι ώστε το ρομπότ να μπορεί να κρατήσει "το ίδιο" και το τεμάχιο εργασίας. Η δύναμη της βαρύτητας του ρομπότ διαφέρει ανάλογα με τη θέση του βραχίονα. Αυτό υπολογίζεται μαθηματικά και αντισταθμίζεται ανάλογα. Το φορτίο που πιάνεται μπορεί να κατευθύνεται σχεδόν χωρίς βαρύτητα από τον χειριστή.



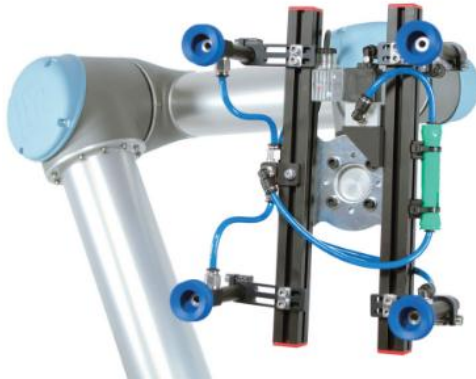
Εικόνα 2.22 Διάφορα πνευματικά έμβολα που μπορούν να χρησιμοποιηθούν ως ενεργοποιητές



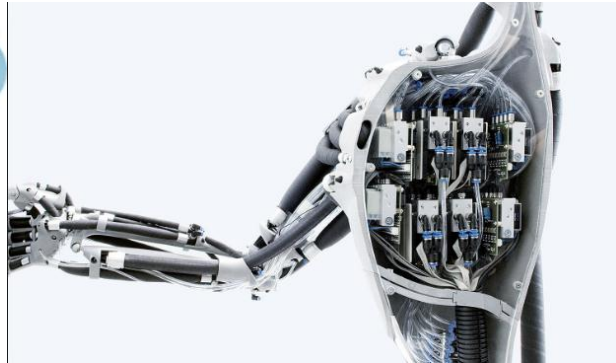
Εικόνα 2.23 Πνευματικό έμβολο Ενεργοποιητής



Εικ. 2.24 Ρομποτικός βραχίονας με τη χρήση πνευματικών ενεργοποιητών , προσομείωση ανθρώπινου χεριού



Εικ. 2.25 πνευματικός ενεργοποιητής



Εικ. 2.26 Ανθρωποειδής Ρομπότ με τη χρήση πνευματικών by Festo

2.6.6 Υδραυλικά έμβολα Ενεργοποιητές

Μερικά από τα πλεονεκτήματα των υδραυλικών συστημάτων για τη χρήσης τους ως επενεργητές:

Μεταβλητή ταχύτητα: Οι κοινοί ηλεκτρικοί κινητήρες και οι μηχανές εσωτερικής καύσεως λειτουργούν με σταθερή ταχύτητα. Η Μεταβολή της ταχύτητας απαιτεί ακριβές και ευαίσθητες βοηθητικές διατάξεις. Ένας υδραυλικός κινητήρας όμως, έχει την δυνατότητα της συνεχούς μεταβολής της ταχύτητας με απλή μεταβολή της παροχής της αντλίας (αντλία μεταβλητής παροχής) ή με την χρήση ενός ρυθμιστή ροής.

Αντιστροφή της περιστροφής ή της κινήσεως: Σε συνήθη μηχανικά ή ηλεκτρικά συστήματα κινήσεως, ή αντιστροφή της περιστροφής είναι δυνατή με βαθμιαία επιβράδυνση, μηδενισμό της κινήσεως και αντίστροφη κίνηση. Ο υδραυλικός κύλινδρος και ο υδραυλικός κινητήρας όμως, μπορούν να αντιστρέψουν την περιστροφή σχεδόν ακαριαία χωρίς ουσιαστικό πρόβλημα. Μια τετραοδηγική βαλβίδα κατευθύνσεως μπορεί να αντιστρέψει ακαριαία την ροή ενώ η βαλβίδα ανακουφίσεως και κατάλληλη διάταξη βαλβίδων για απόσβεση των κρούσεων και της αδράνειας του ρευστού προστατεύουν το σύστημα από σπηλαίωση και υπερβολική πίεση.

Προστασία από υπερβολικό φορτίο και έλεγχος των φορτίων: Η ανακουφιστική βαλβίδα του υδραυλικού συστήματος προστατεύει το σύστημα από ανεξέλεγκτη αύξηση του φορτίου. Η παρουσία της ανακουφιστικής βαλβίδας και ο έλεγχος των δυνάμεων που επιτυγχάνουμε με τις διάφορες βαλβίδες πίεσεως εξασφαλίζουν απόλυτη προστασία και ακριβή έλεγχο σε ένα υδραυλικό σύστημα.....

Μικρός και περιορισμένος όγκος: Τα υδραυλικά στοιχεία και ολόκληρο το υδραυλικό σύστημα προσφέρουν υψηλή απόδοση συνδυάζοντας μικρό όγκο και μικρό βάρος. Μια ενδιαφέρουσα πρόσφατη εξέλιξη είναι οι ένθετες βαλβίδες ή λογικά στοιχεία με τις οποίες επιτυγχάνεται πληθώρα εναλλακτικών ελέγχων με ελάχιστο όγκο και βάρος. (Εικόνα 2.25)

Δυνατότητα ακαριαίας στάσεως: Η αδράνεια των ηλεκτρικών ή μηχανικών συστημάτων είναι τέτοια ώστε η ακαριαία στάση τους σε κίνηση του συστήματος μπορεί να είναι καταστροφική. Επί πλέον τα συστήματα αυτά απαιτούν νέα εκκίνηση μετά την στάση.

Ο υδραυλικός κύλινδρος και ο υδραυλικός κινητήρας ωστόσο έχουν την δυνατότητα ακαριαίας στάσεως και εκκινήσεως. Η αντλία του συστήματος είναι δυνατόν να συνεχίσει να λειτουργεί χωρίς να υπάρχει ανάγκη ακινητοποιήσεως της εκ νέου εκκινήσεως.

Μεγάλη ποικιλία ελέγχων: Υπάρχουν μεγάλες δυνατότητες ελέγχου σε ένα υδραυλικό σύστημα. Ο έλεγχος μπορεί να είναι αυτόματος, με χειρισμό, μηχανικός, υδραυλικός, πνευματικός, ηλεκτρικός, ηλεκτρονικός, ή συνδυασμοί όλων. Στις ένθετες βαλβίδες οι έλεγχοι μπορούν να πάρουν την μορφή λογικών απαντήσεων στο αλγεβρικό άθροισμα δυνάμεων σε επιφάνειες εμβόλων. Μια πρόσφατη εξέλιξη είναι ο έλεγχος με μικροϋπολογιστές.



Εικόνα 2.27 Υδραυλικά έμβολα



Εικόνα 2.28 Υδραυλικό έμβολο σε βραχίονα



Εικόνα 2.29 Hydraulic manipulator από την KNR

2.7 Χώρος εργασίας

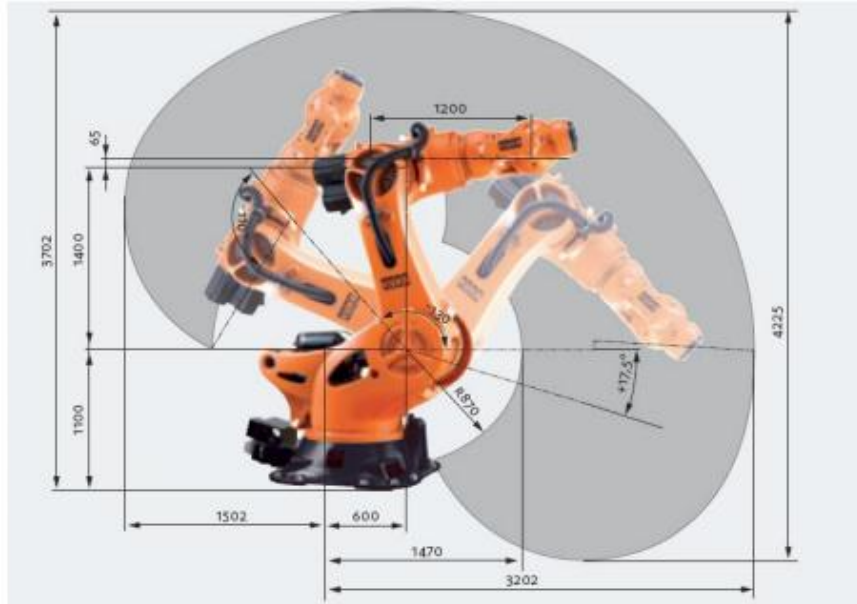
Ως χώρος εργασίας ορίζεται ο γεωμετρικός τόπος των σημείων του χώρου τα οποία μπορεί να τα προσεγγίσει το εργαλείο τελικής δράσης του ρομποτικού βραχίονα. Ανάλογα με το είδος του ρομποτικού βραχίονα, ο χώρος μπορεί να είναι ακίνητος ως προς αυτόν ή όχι, ο χώρος εργασίας ενός βραχίονα μπορεί να είναι πολύ περιορισμένος, λιγότερο περιορισμένος ή και καθόλου, ανάλογα με τη χρήση την οποία έχει επιλεγεί να κάνει. Στην εικόνα 2.30 μπορούμε να παρατηρήσουμε ένα χειριστή εν δράσει στο χώρο εργασίας του.



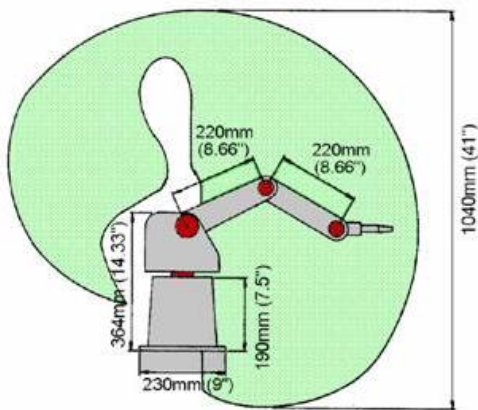
Εικ. 2.30 Ρομποτικός βραχίονας Γλύπτης

Ο χώρος εργασίας μπορεί να διαφέρει από βραχίονα σε βραχίονα με βάση του βαθμούς ελευθερίας, τις διαστάσεις των αρθρώσεων και την φύση της επιθυμητής εργασίας.

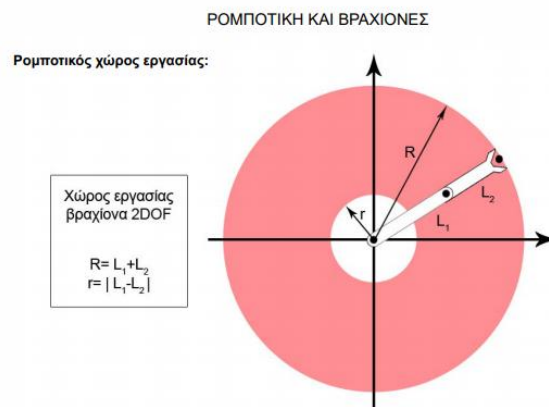
Ο χώρος εργασίας ενός ανθρωπομορφικού βραχίονα της KUKA:



Εικ. 2.31 Ανθρωπομορφικός βραχίονας από την KUKA Robotics



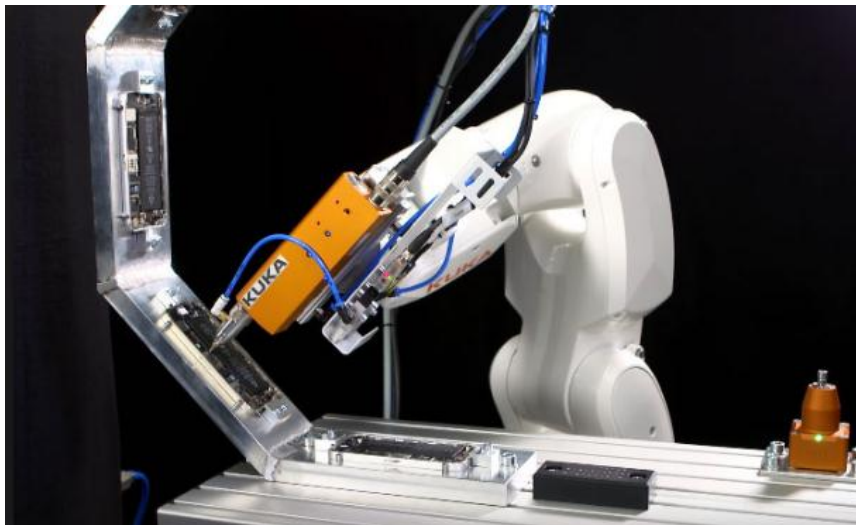
Εικ. 2.32 Χώρος εργασίας και distάνσεις αρθρώσεων



Εικ. 2.33 Χώρος εργασίας

2.8 Διαχωρισμός με βάση το τελικό εργαλείο δράσης

Διάφορα είδη ρομποτικών βραχιόνων οι οποίοι διαχωρίζονται με βάση το άκρο και διαφορετική εφαρμογή στη σύγχρονη βιομηχανία:



Εικ. 2.34 Ρομποτικός Βραχίονας με άκρο κάποιο είδος κατασβιδιού κατα την εκτέλεση εργασίας συναρμολόγησης



Εικ 2.35 Ρομποτικός βραχίονας κατα την εκτέλεση εργασίας ηλεκτροσυσκόλλησης

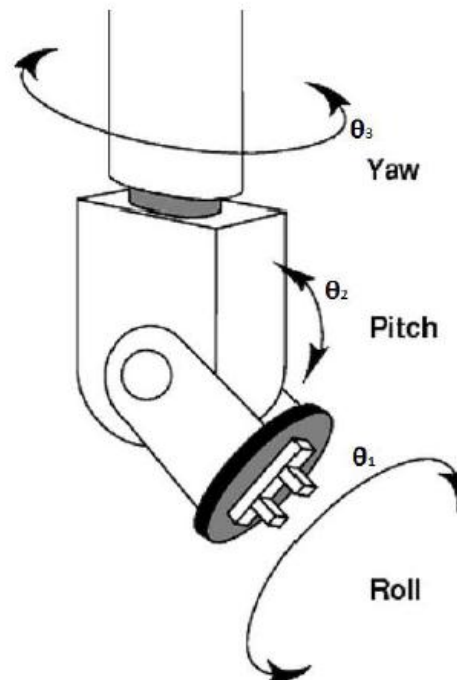


Εικ 2.36 Ρομποτικός βραχίονας σε κάποιο φανοποιείο κατά την εκτέλεση εργασίας βαφής



Εικ 2.37 Ρομποτικοί βραχίονες σε γραμμή παραγωγής κατά την εκτέλεση εργασιών παλετοποίησης

Ο καρπός είναι υπεύθυνος για τον προσανατολισμό του άκρου του βραχίονα και γι' αυτό είναι συνήθως συμπαγής . Ο καρπός στην ανθρωπομορφική μορφή του αποτελείται από τρεις περιστροφικές αρθρώσεις που επιτρέπουν τη στροφή (roll) την κλίση (pitch) και την εκτροπή (yaw) όπως και στον άνθρωπο (Σχ. 2.11) Η μεγαλύτερη επιδεξιότητα του καρπού επιτυγχάνεται όταν οι άξονες των τριών περιστροφικών αρθρώσεων τέμνονται σε ένα σημείο , τότε αποκαλείται σφαιρικός καρπός (εικ 2.38).



Σχήμα 2.11 οι γωνίες RPY (roll – pitch – yaw).

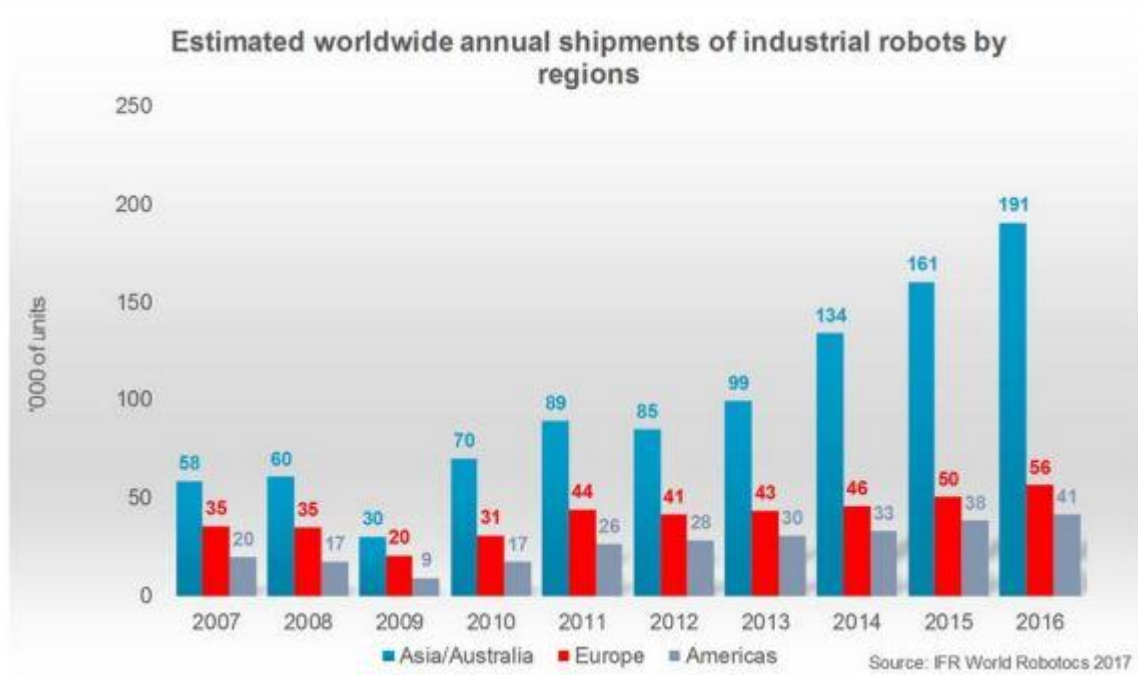


Εικ. 2.38 Σφαιρικός καρπός

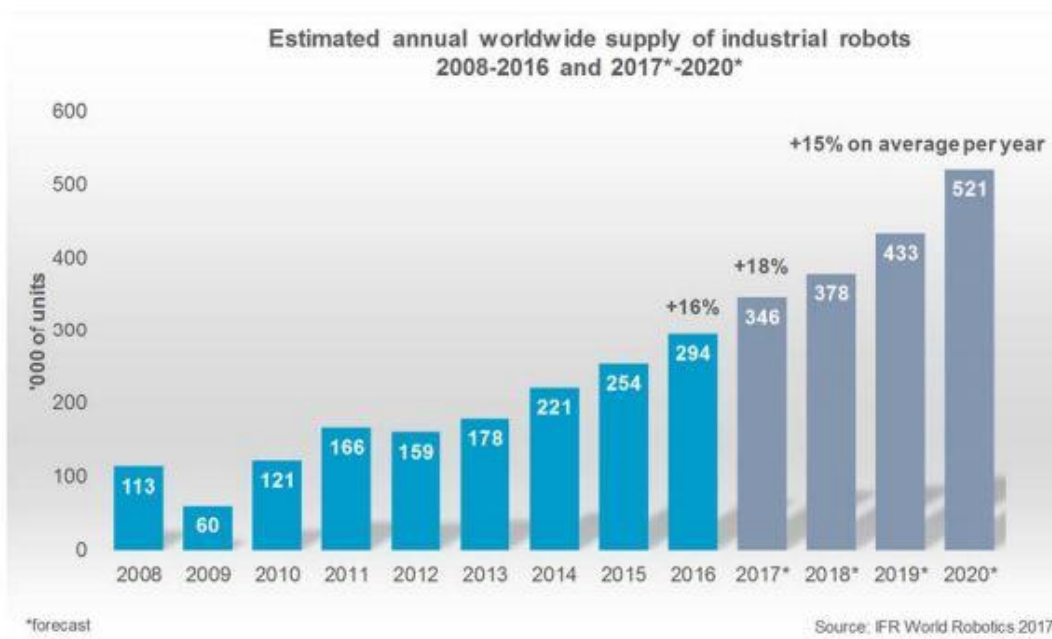
2.9 Εταιρίες κατασκευής Ρομποτικών Βραχιόνων-Στατιστικά

Παρακάτω έχουμε τη δυνατότητα να δούμε τα Λογότυπα των μεγαλύτερων εταιριών κατασκευής ρομποτικών βραχιόνων στον κόσμο.

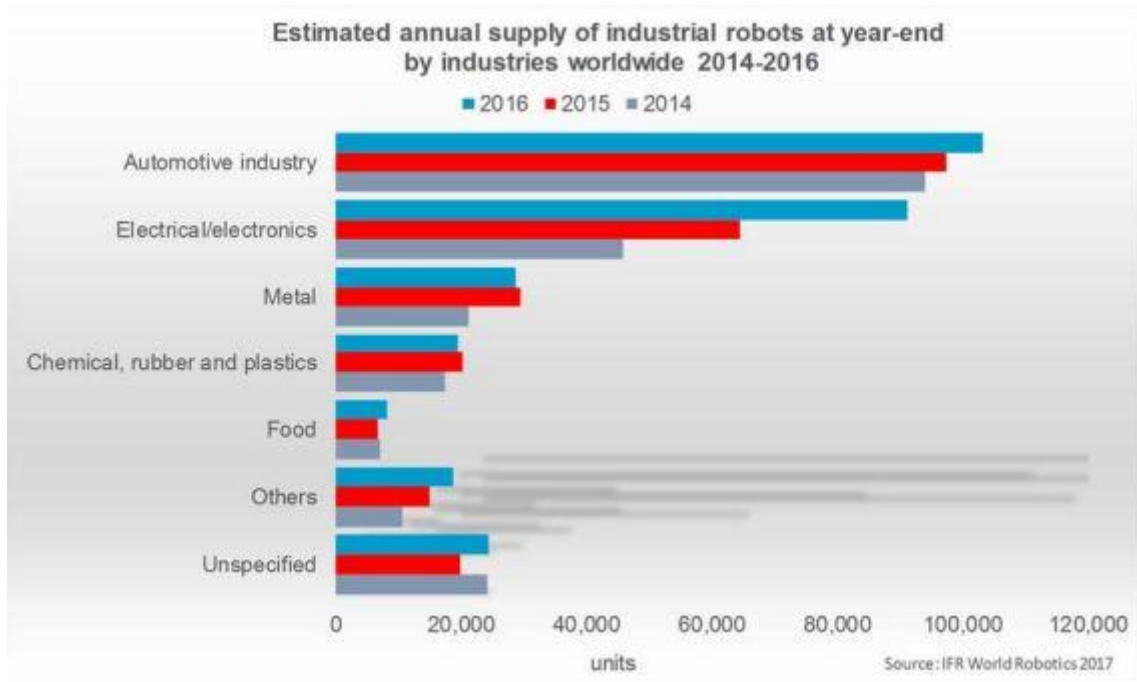




Εικ. 2.39 Ετήσιες διεθνής αποστολές βιομηχανικών Ρομπότ ανά Ήπειρο



Εικ. 2.40 Εκτιμώμενη ετήσια παγκόσμια προμήθεια ανα έτος



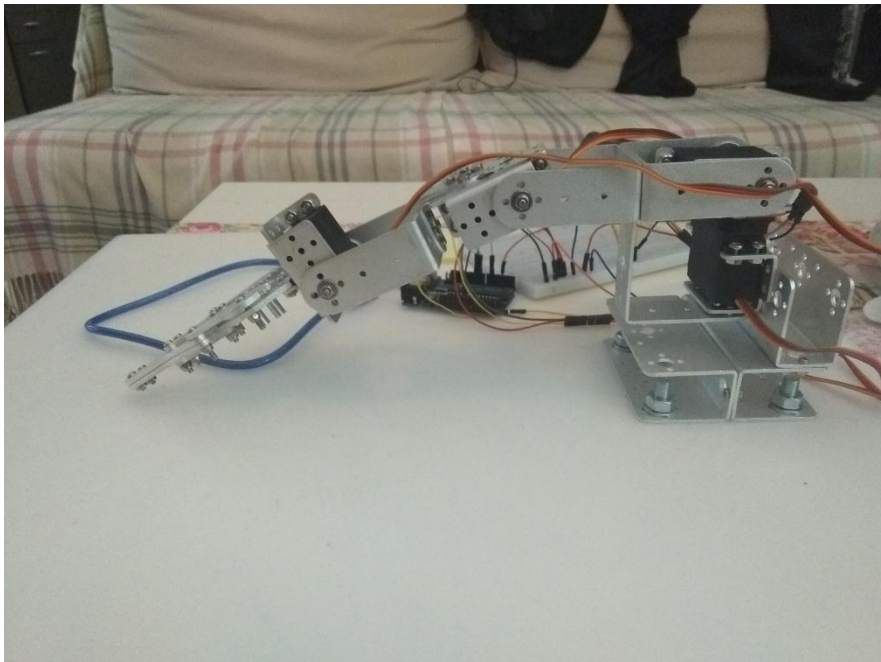
Εικ. 2.41 Ετήσια προμήθεια βιομηχανικών ρομπότ ανά τύπο βιομηχανίας

Κεφάλαιο 3

3.1 Κατασκευή και συναρμολόγηση Μηχανικού μέρους Ρομποτικού Βραχίονα -υλικά και τροφοδοσία

Υλικά που χρησιμοποιήθηκαν για το hardware

Για την κατασκευή του μηχανικού μέρους χρησιμοποιήθηκαν ειδικά εξαρτήματα αλουμινίου , βίδες παξιμάδια και ροδέλλες. Χρησιμοποιήθηκε κατσαβίδι και ειδική κολλά για βίδες ,αξίζει να σημειώσουμε πως για τη κατασκευή της βάσης, των συνδέσμων και των αρθρώσεων απαιτείται μεγάλη προσοχή και λεπτομέρεια ,ώστε να μπορεί ο βραχίονας να εκτελεί εύκολα τις επιθυμητές κινήσεις. Κατασκευάσαμε αρχικά τη βάση κι έπειτα όλες τις αρθρώσεις κι τους συνδέσμους του βραχίονα, καθώς και την αρπάγη. Στην εικόνα 3.1 φαίνεται ξεκάθαρα η κατασκευή και διάταξη του βραχίονα.



Εικ. 3.1 Υλικά για τη συναρμολόγηση του βραχίονα

Για την κατασκευή της βάσης χρησιμοποιήθηκε ερταλόν , υλικό ειδικό με μεγάλη αντοχή στις μηχανικές καταπονήσεις. Τρυπήθηκε με σκοπό τη δημιουργία τεσσάρων οπών για τη στήριξη της βάσης του βραχίονα, επιπλέον χρησιμοποιήθηκαν βίδες M6 X60 παξιμάδια και ροδέλλες για την πάκτωση του βραχίονα επάνω στη βάση .



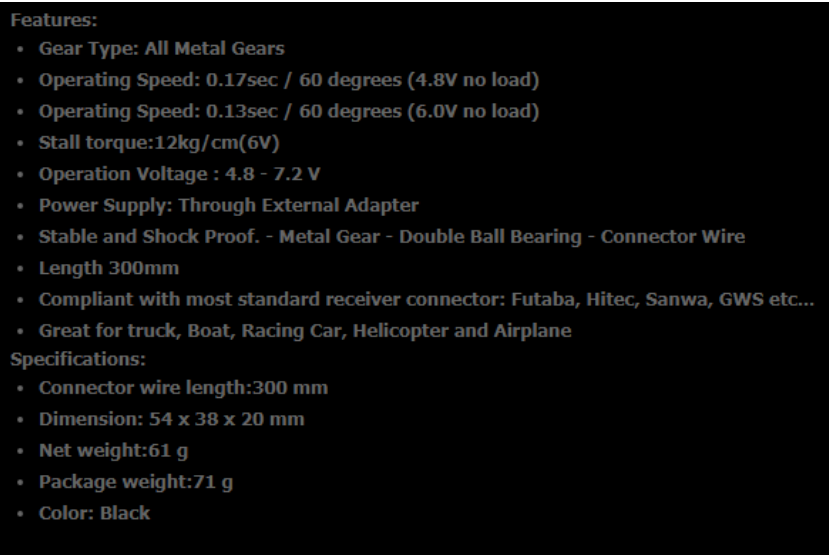
Εικ. 3.2 Βάση κατασκευασμένη από Ερταλόν υψηλής αντοχής

Για την κίνηση των αρθρώσεων χρησιμοποιήθηκαν servo-κινητήρες υψηλής ροπής τύπου MG996R. Οι οποίοι τοποθετήθηκαν στα κατάλληλα σημεία για την κίνηση της κάθε άρθρωσης ξεχωριστά. Στην εικόνα 3.3 μπορούμε να δούμε τον τύπο των servo-κινητήρων που χρησιμοποιήθηκαν και στην εικόνα 3.4 τα τεχνικά χαρακτηριστικά τους.



MG996R Servo - Metal Gear / High Torque Motor for Arduino
Προσθήκη στη λίστα αγαθών

Εικόνα 3.3 Servo-κινητήρας υψηλής ροπής MG996R



Εικόνα 3.4 Τεχνικά χαρακτηριστικά Servo MG996R

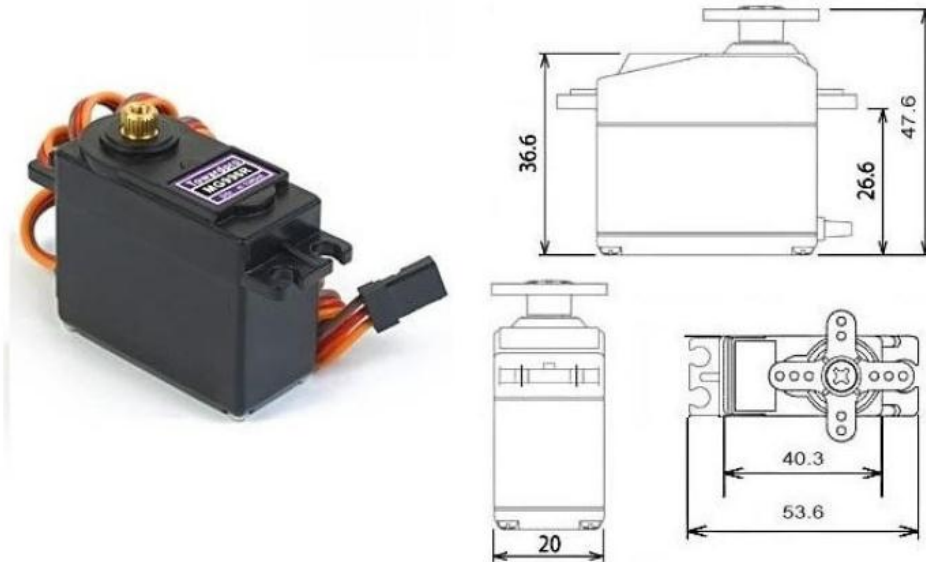
Features:

- **Gear Type:All metal gears**
- **Operation Speed:0.17sec/60degrees(4.8v no load)**
- **Operation Speed:0.13sec/60degrees(6.0v no load)**
- **Stall torque:12kg/cm(6v)**
- **Operation voltage:4.8-7.2v**
- **Power Supply: Through external adapter**
- **Stable and stock proof-metal gear-double ball bearing-connector wire**
- **Length:300mm**
- **Compliant with most standard receiver connector:Futaba,Hitec,Sanwa,GWS etc.**
- **Great for truck, Boat , Racing car,Helicopter and Airplane**

Specifications

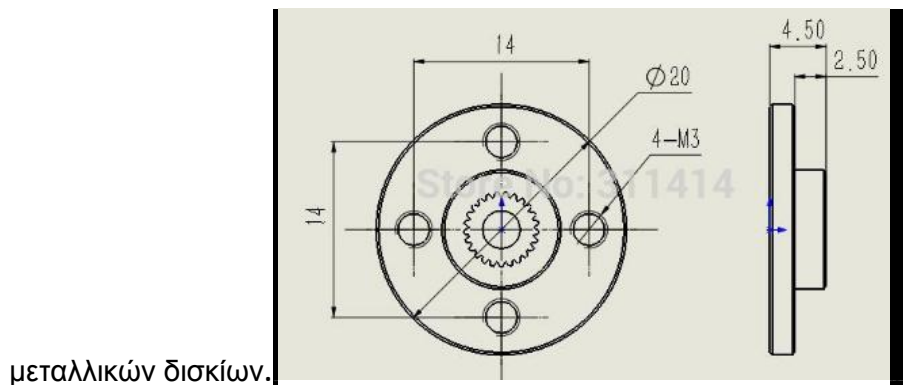
- **Connector wire length :300mm**
- **Dimension : 54 x 38 x 20mm**
- **Net weight: 61gr**
- **Package weight: 71gr**
- **Color: Black**

MG996R High Torque Metal Gear Dual Ball Bearing Servo

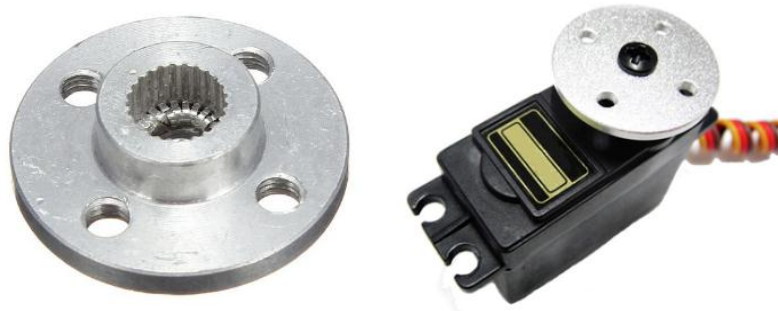


Εικόνα 3.5 Διαστάσεις servo MG966R

Επιπλέον για τη μετάδοση της κίνησης από τα servo στις αρθρώσεις ήταν απαραίτητη η χρήση μεταλλικών δίσκων για τους servo-κινητήρες. Στις εικόνες 3.6-3.7 έχουμε τη δυνατότητα να παρατηρήσουμε αναλυτικές διαστάσεις και φωτογραφίες των ειδικών



Εικόνα 3.6 Διαστάσεις servo MG966R



Εικόνα 3.7 Metal disc for servo MG996R

Για την τροφοδοσία των servo χρησιμοποιήθηκε εξωτερικό τροφοδοτικό και μετατροπέας DC-DC για την σταθεροποίηση της τάσης, διότι σε διάφορες δοκιμές που πραγματοποιήθηκαν με μπαταρίες υπήρχε συνεχής αστάθεια στην τάση.



Εικόνα 3.8 Σταθεροποιημένο τροφοδοτικό 12V/10A/120W, Power Plus, CP1207-10A

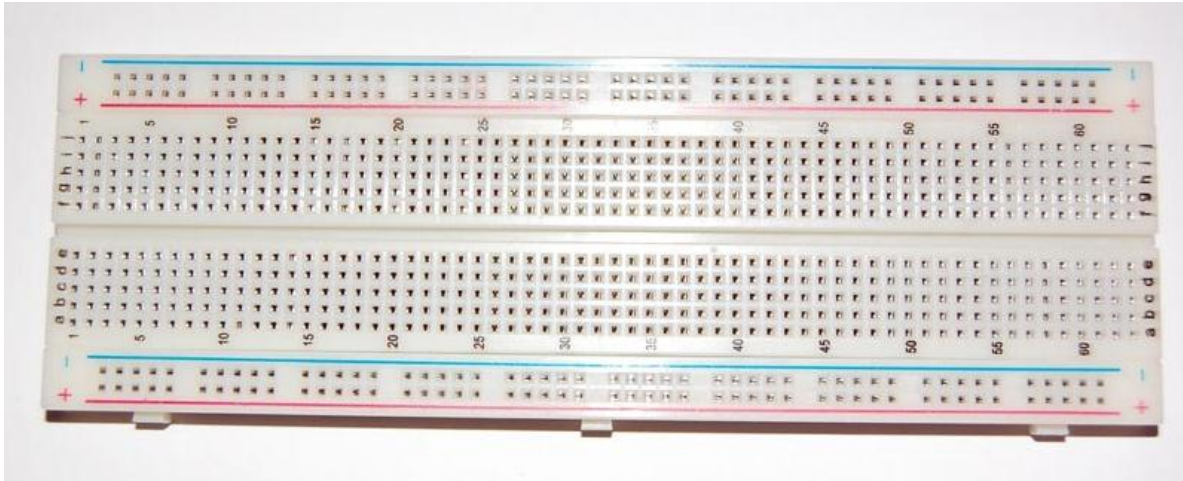


Εικόνα 3.9 DC-DC 10A 200W Adjustable Buck Converter Module-Power Supply

- Module Name: 200W adjustable buck module
- Model: ZS-PD8
- Type: Non-isolated buck module (BUCK)
- Input voltage: 4.5-32V (maximum can not exceed 32V)
- Output voltage: 0.8-30V continuously adjustable
- Output current: MAX. 12A (There are associated between current, voltage and temperature. When in high current, the chip overheating and automatically reduced flow, in the general ambient temperature naturally cooling current can in 9-12A, stable working strength when cooling can up to 9-12A)
- Output power: Natural cooling power 100W, enhance cooling can up to 200W
- Working temperature: -40 to + 85 degrees
- Operating frequency: 300KHz
- Conversion efficiency: Up to 95% (efficiency is related with input / output voltage, current and temperature)
- Short circuit protection: Yes
- Overtemperature protection: Yes (automatically shut off the output after overtemperature)
- Input reverse polarity protection: Yes (please do not reverse for a long time)
- Transient pulse voltage protection: Yes (OVP voltage 34V)
- Wiring: High current terminals, free welding, IN+ IN- for input positive and negative, OUT+ OUT- for output positive and negative
- Aluminum Shell Size: 70 x 39 x 31mm

Εικόνα 3.10 Τεχνικά χαρακτηριστικά του μετατροπέα DC-DC

Η χρήση breadboard 630 Tie points 6.5 x 2.1 in , ήταν απαραίτητη διότι όλες οι συνδεσμολογίες όσο αναφορά τους servo κινητήρες την εξωτερική τροφοδοσία , την πλακέτα arduino και τις δοκιμές με τα ποτενσιόμετρα έγιναν εκεί επάνω.



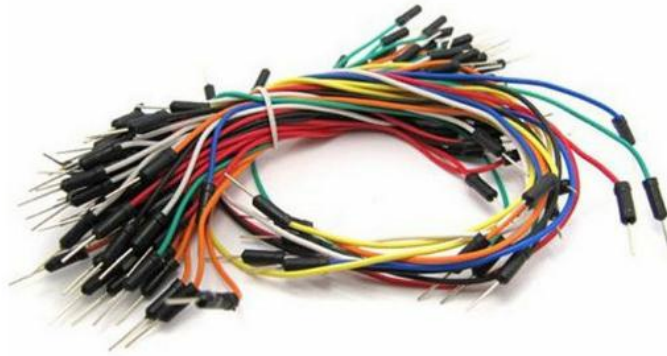
Εικόνα 3.11 Bread board

Επίσης στην παρούσα Διπλωματική εργασία χρησιμοποιήθηκαν ποτενσιόμετρα για τον έλεγχο των servo-κινητήρων κατά την διάρκεια δοκιμών.



Εικόνα 3.12 Ποτενσιόμετρο 10K Ohm

Για τις συνδεσμολογίες των ηλεκτρονικών μερών χρησιμοποιήθηκαν διάφορα είδη καλώδιων

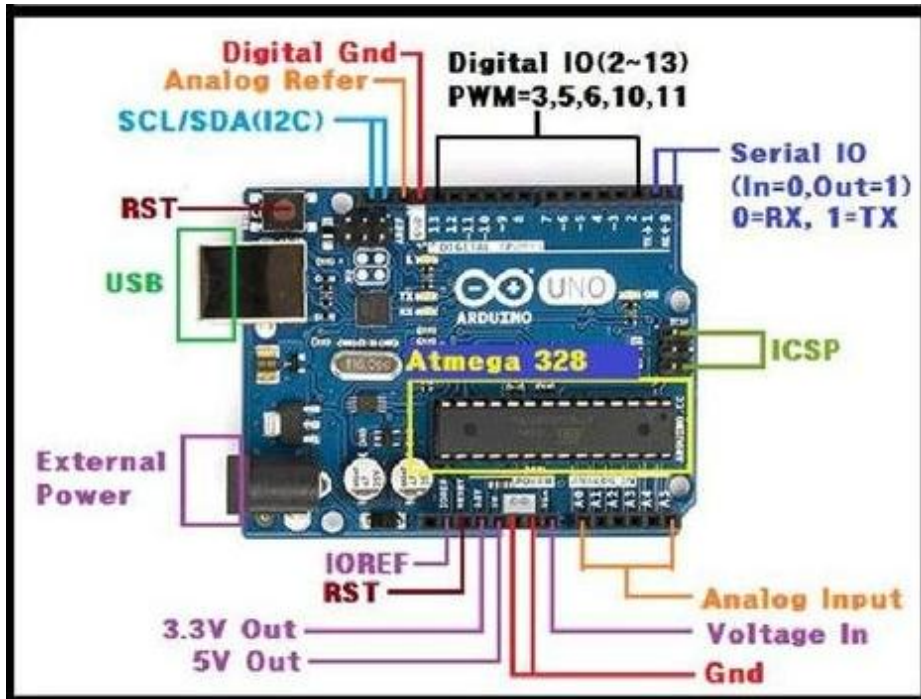


Εικόνα 3.13 Καλώδια για τις συνδέσεις

Ο έλεγχος των servo θα γίνει με την πλακέτα Arduino Uno R3, αναλυτική περιγραφή της οποίας θα γίνει στο 5^ο κεφάλαιο.



Εικόνα 3.13 Arduino Uno R3



Εικόνα 3.14 Arduino Uno R3 Pins

Στον παρακάτω πίνακα 3.1 παρουσιάζονται αναλυτικά όλα τα υλικά που χρησιμοποιήθηκαν στην κατασκευή του ρομποτικού βραχίονα, όπως επίσης αναγράφεται και η τιμή για το κάθε υλικό ξεχωριστά.

3.2 Πίνακας εξαρτημάτων και Υλικών συναρμολόγησης

Υλικό	Χαρακτηριστικά/διαστάσεις	Τιμή
Βάση Ερταλόν	50cm x 50cm x 2cm	100 ευρώ
6 x Servo	MG996R	22 ευρώ
Μπουλόνια πάκτωσης βάσης	M8 x 60	4 ευρώ
Κατασκευή αλουμινίου για το σκελετό	Επιπλέον παξιμάδια και βίδες	40 ευρώ
6 x Metal discs για τα Servo	Επιπλέον παξιμάδια, βίδες, ροδέλλες.	7 ευρώ
Εξωτερικό τροφοδοτικό	CP-1027 12V/10A-120W	12 ευρώ
Adjustable converter	DC-DC converter 12A/200w	10 ευρώ
Bread board	1600 οπών	8 ευρώ
Πλακέτα Arduino	Arduino Uno R3	8 ευρώ
Καλώδια	Διάφορα μεγέθη	5 ευρώ
6 x Ποτενσιόμετρα	10K Ohm	1,20 ευρώ

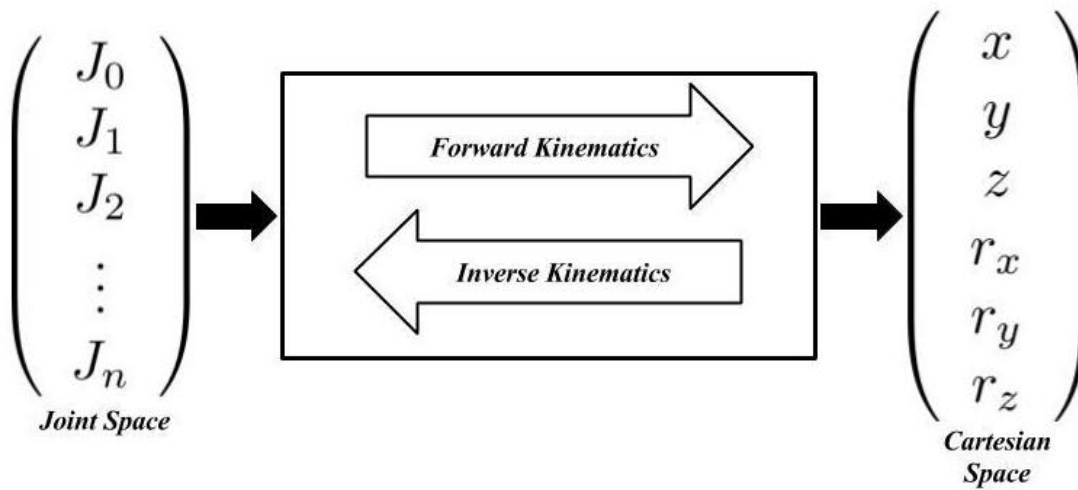
Πίνακας 3.1 Εξαρτήματα και τιμές

Συνολικό κόστος υλικών: 217,20 ευρώ

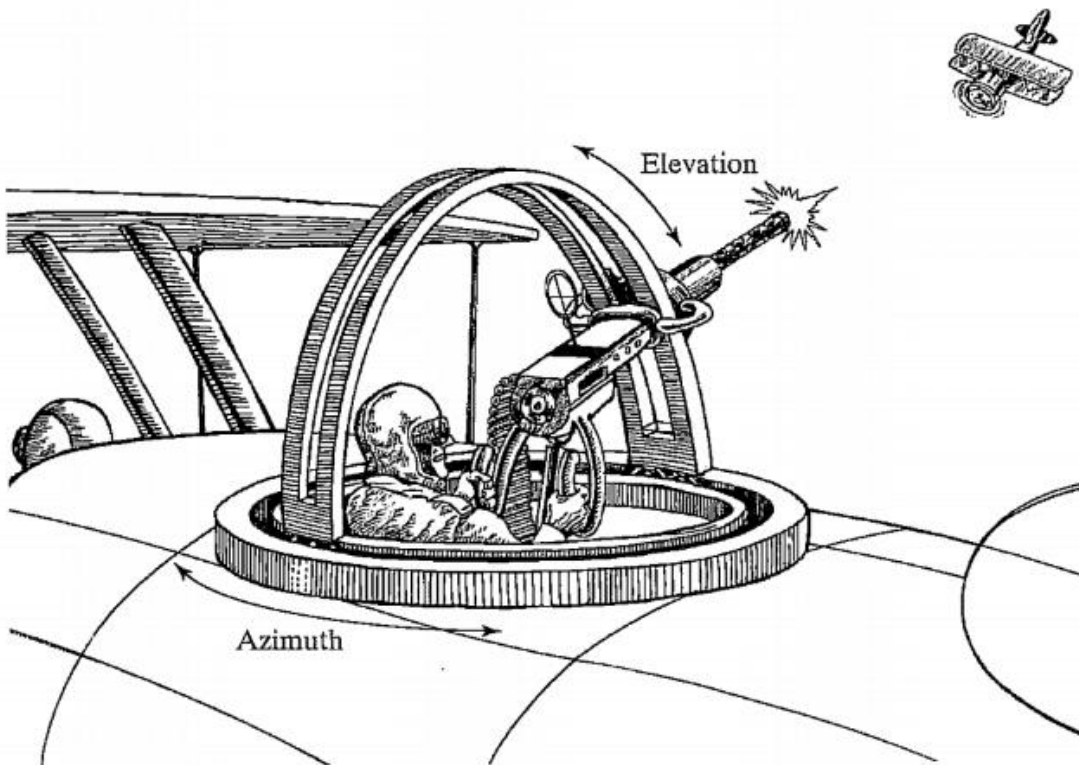
Κεφάλαιο 4

4.1 Εισαγωγή στην Κινηματική Θεωρία

Κινηματική είναι η επιστήμη που μελετά την κίνηση χωρίς να λαμβάνει υπόψη τις δυνάμεις που τις προκαλούν. Στην κινηματική εστιάζουμε στη θέση, την ταχύτητα, την επιτάχυνση και όλες τις μεταβλητές που περιγράφουν τη θέση του ρομποτικού βραχίονα. Ο επιστημονικός αυτός κλάδος, επικεντρώνεται στην επίλυση δύο προβλημάτων στην κινηματική ανάλυση ενός βραχίονα. Αυτά είναι το ευθύ κινηματικό και το αντίστροφο κινηματικό πρόβλημα. Στο ευθύ κινηματικό πρόβλημα συνιστάται η εύρεση της θέσης και ο προσανατολισμός του εργαλείου τελικής δράσης, ως προς το σύστημα συντεταγμένων του χώρου εργασίας, γνωρίζοντας τη θέση κάθε άρθρωσης. Στο αντίστροφο κινηματικό πρόβλημα προσδιορίζονται οι θέσεις των αρθρώσεων ως προς το σύστημα συντεταγμένων του χώρου εργασίας, γνωρίζοντας σε αυτή την περίπτωση τη θέση και τον προσανατολισμό του εργαλείου τελικής δράσης ή ενδιάμεσων σημείων. Όσο αναφορά την κινηματική μελέτη αρθρωτών βραχιόνων των ρομπότ, θα λέγαμε ότι είναι μια αλληλουχία στερεών σωμάτων, τα οποία ονομάζονται σύνδεσμοι και συνδέονται μεταξύ τους με αρθρώσεις. Η κάθε άρθρωση συνήθως αντιστοιχεί σε ένα βαθμό ελευθερίας. Διακρίνουμε δύο βασικά είδη αρθρώσεων: αυτές επιτρέπουν στροφή γύρω από έναν άξονα, οι οποίες ονομάζονται περιστροφικές αρθρώσεις και αυτές που επιτρέπουν ολίσθηση κατά μήκος ενός άξονα και ονομάζονται πρισματικές αρθρώσεις.



Σχήμα 4.1 Κινηματική θεωρία



Εικόνα 4.1 Ένας διπλότυπος του Β 'Παγκοσμίου Πολέμου με πιλότο και οπίσθιο πυροβόλο. Το οπίσθιο πυροβόλο υπόκειται στο πρόβλημα των μοναδικών θέσεων.

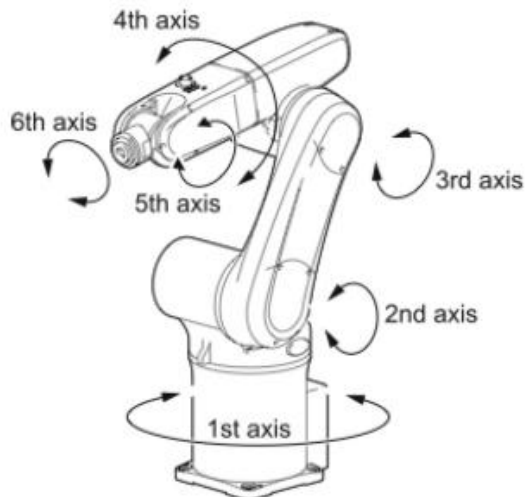
4.1.1 Χαρακτηριστικά μεγέθη Ρομποτικού Βραχίονα

Ο αριθμός των ανεξάρτητων κινήσεων που μπορεί να κάνει ένα αντικείμενο στον τρισδιάστατο χώρο ονομάζεται βαθμός ελευθερίας (β.ε) της κίνησής του. Ένα στερεό σώμα που κινείται ελεύθερα στο χώρο έχει έξι βαθμούς ελευθερίας. Συγκεκριμένα οι τρεις είναι για τη θέση και οι άλλοι τρεις είναι για τον προσανατολισμό. Εάν θεωρήσουμε ένα καρτεσιανό σύστημα συντεταγμένων ως σύστημα αναφοράς των θέσεων και του προσανατολισμού του σώματος, τότε οι τρεις ανεξάρτητες κινήσεις για την θέση και οι τρεις ανεξάρτητες κινήσεις για τον προσανατολισμό μπορούν να οριστούν ως τρεις μεταφορικές κινήσεις κατά μήκος των αξόνων και ως τρεις περιστροφές γύρω από τους άξονες του συστήματος αναφοράς.

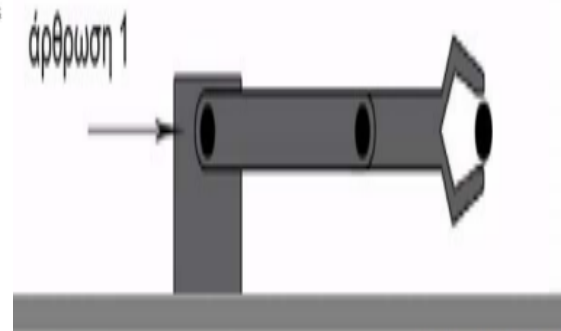
Κάθε άρθρωση ενός ρομποτικού βραχίονα, είτε είναι πρισματική, είτε περιστροφική, συμβάλλει με ένα βαθμό κινητικότητας στον βραχίονα, ο οποίος μπορεί να αντιπροσωπεύει έναν βαθμό ελευθερίας όταν η κίνησή του είναι ανεξάρτητη.

Βασικές έννοιες:

- Σύνδεσμος (μήκος L)
- Άρθρωση (γωνία θ , απόσταση d)
- Τελικό σημείο δράσης(συντεταγμένες p_x, p_y, p_z)
- Βαθμοί κινητικότητας DOM (Degrees of Mobility)
- Βαθμοί ελευθερίας DOF (Degrees of freedom)
- Ρομποτικός χώρος εργασίας



Εικ. 4.2 Κατανόηση Βασικών εννοιών-α



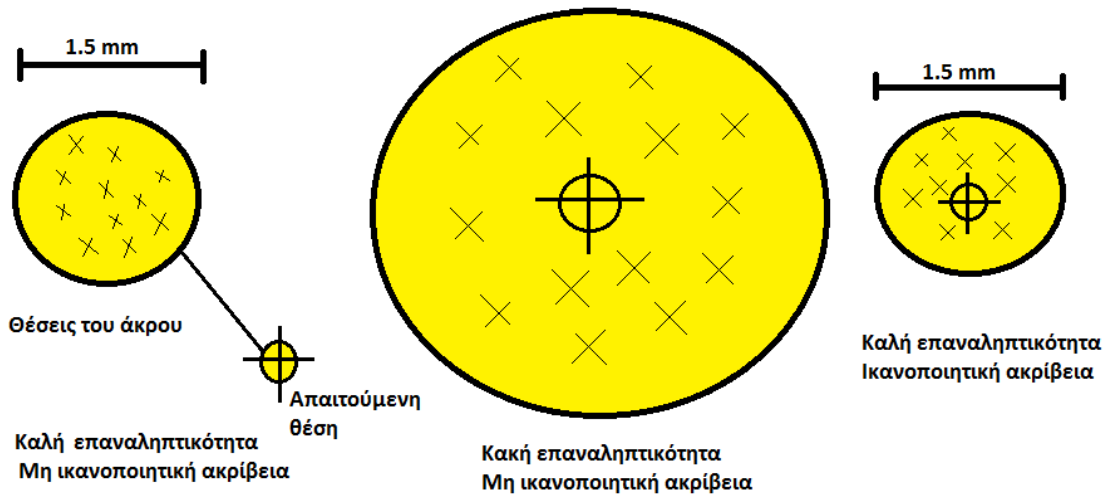
Εικ 4.3 Κατανόηση Βασικών εννοιών-β

Οι πρώτες τρεις αρθρώσεις στις βασικές γεωμετρίες υλοποιούν τους τρεις αναγκαίους βαθμούς ελευθερίας για την τοποθέτηση ενός σώματος στο χώρο, ενώ οι οι τρεις περιστροφικές αρθρώσεις ενός σφαιρικού καρπού υλοποιούν τους τρεις αναγκαίους βαθμούς ελευθερίας για τον προσανατολισμό του σώματος. Επομένως για ελεύθερη κίνηση στο χώρο χρειαζόμαστε, ρομποτικούς βραχίονες με έξι βαθμούς ελευθερίας, των οποίων η υλοποίηση επιτυγχάνεται με έξι τουλάχιστον αρθρώσεις. Υπάρχουν ρομπότ με λιγότερες ή περισσότερες από έξι αρθρώσεις, αυτά που έχουν λιγότερες αρθρώσεις έχουν και περιορισμένη κίνηση, αλλά μπορεί να επαρκούν για κάποιες εργασίες, οπότε λογικό είναι να υπάρχουν διαφοροποιήσεις ανάλογα με τη φύση της εργασίας την οποία έχουν επιλεχθεί να εκτελούν. Για παράδειγμα ένα ρομπότ με τέσσερις βαθμούς ελευθερίας με τις τρεις πρώτες αρθρώσεις σε γεωμετρία SCARA και την τέταρτη περιστροφική και με άξονα περιστροφής τον άξονα της τρίτης πρισματικής άρθρωσης, μπορεί να χρησιμοποιηθεί με μεγάλη επιτυχία σε εργασίες κάθετης συναρμολόγησης. Έτσι θα μπορούσαμε να πούμε με βεβαιότητα ότι ένας βραχίονας με περισσότερες από έξι αρθρώσεις, βελτιώνει κατα πολύ την κινητική του ευελιξία αλλά παρουσιάζει περίπλοκο κινηματικό λογισμικό.

Δύο πολύ σημαντικά χαρακτηριστικά των ρομποτικών βραχιόνων είναι η ακρίβεια και η επαναληπτικότητα. Οι προδιαγραφές των βιομηχανικών Ρομπότ αναφέρονται συνήθως στην επαναληπτικότητα σε χιλιοστά (mm).

Επαναληπτικότητα είναι η χαρακτηριστική διάσταση του όγκου μέσα στον οποίο το ρομπότ θα τοποθετήσει το άκρο του όταν κινηθεί κατ' επανάληψη σε μία επιθυμητή θέση από διάφορες αρχικές θέσεις. Έτσι ένας βραχίονας με επαναληπτικότητα 1,5 χιλιοστό τοποθετεί κάθε φορά το άκρο του μέσα σε σφαίρα διαμέτρου 1,5 χιλιοστό (Εικ. 3.1). Εάν τώρα επιπλέον το κέντρο της σφαίρας συμπίπτει ή είναι κοντά στην επιθυμητή θέση , τότε ο βραχίονας έχει ικανοποιητική ακρίβεια. Εάν όμως αυτό δεν ισχύει, είτε διότι η επαναληπτικότητα είναι κακή είτε διότι το κέντρο της σφαίρας απέχει σημαντικά από την επιθυμητή θέση , τότε ο βραχίονας είναι ανακριβής (Εικ. 3.1).

Η ακρίβεια και η επαναληπτικότητα διαφέρουν για τις διάφορες θέσεις του άκρου στις οποίες μπορεί να κινηθεί ο βραχίονας . Οι διαφορές εξαρτώνται τόσο από την επιθυμητή θέση όσο και από την κατεύθυνση κίνησης του άκρου. Η ακρίβεια είναι μικρότερη για θέσεις κοντά στα όρια του χώρου εργασίας του βραχίονα , όπου ο βραχίονας είναι σχεδόν εκτεταμένος. Τέλος , ένα άλλο πολύ σημαντικό χαρακτηριστικό των βραχιόνων είναι το ωφέλιμο φορτίο που μπορούν να διαχειριστούν ή να μετακινήσουν. Φορτίο είναι το μέγιστο βάρος του αντικειμένου που μπορεί να χειριστεί το ρομπότ με τις προδιαγραφές απόδοσης που δίνει ο κατασκευαστής .Το φορτίο στα βιομηχανικά ηλεκτρικά ρομπότ κυμαίνεται από 80kg έως 100 kg , ενώ τα υδραυλικά ρομποτικά συστήματα μπορούν να σηκώσουν βάρη που φτάνουν τα 300 kg και να τα μεταφέρουν με ταχύτητες που φτάνουν τα 3m/s.



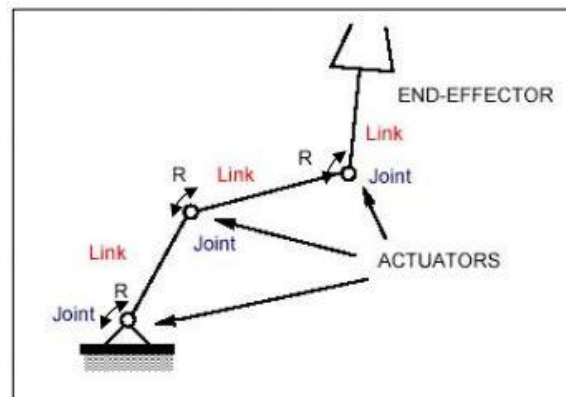
Σχήμα 4.2 Σφαίρα Επαναληπτικότητας και Ακρίβειας του άκρου

4.2 Ευθύ κινηματικό πρόβλημα

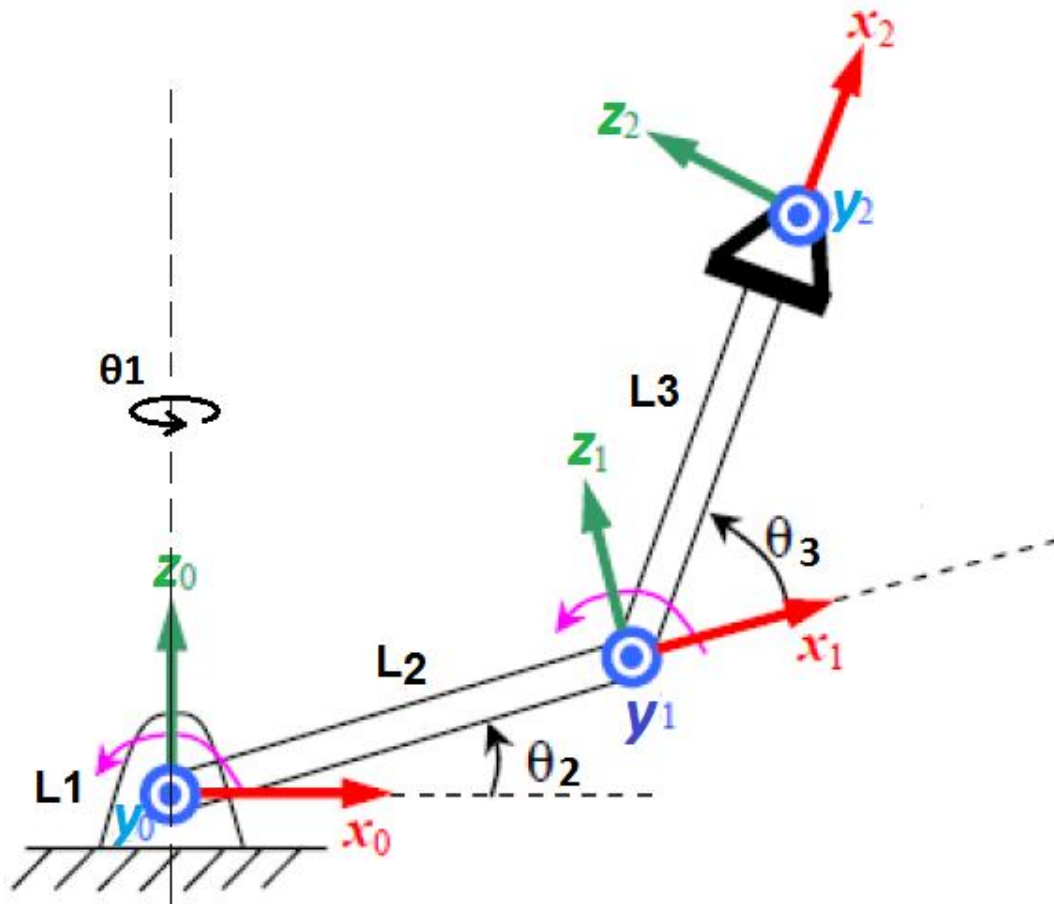
Στην ευθεία κινηματική ανάλυση ενός βραχίονα με n αρθρώσεις, υπολογίζουμε τη θέση και τον προσανατολισμό του άκρου ενός βραχίονα ως προς ένα αδρανειακό πλαίσιο που τοποθετείται στη βάση του βραχίονα, ως συνάρτηση των μεταβλητών των αρθρώσεων $q \in \mathbb{R}$. Υπολογίζουμε δηλαδή τον ομογενή μετασχηματισμό: $g(q): Q \rightarrow SE(3)$ που απεικονίζει το χώρο των αρθρώσεων Q στο χώρο $SE(3)$ της θέσης και του προσανατολισμού ενός στερεού σώματος (στην περίπτωση του άκρου του βραχίονα). Η ευθεία κινηματική ανάλυση γίνεται με βάση ένα σύνολο συμβατικών κανόνων που αφορούν τους τρόπους τοποθέτησης των πλαισίων στους συνδέσμους, έτσι ώστε να απλοποιούνται οι κινηματικές εξισώσεις του βραχίονα. Οι σύνδεσμοι του βραχίονα ξεκινώντας πάντα από τη βάση του βραχίονα, η οποία θεωρείται ως ο σύνδεσμος μηδέν. Ο πρώτος κινούμενος σύνδεσμος είναι ο σύνδεσμος 1 κ.ο.κ., έως το σύνδεσμο n . Σε κάθε σύνδεσμο επισυνάπτουμε ένα πλαίσιο το οποίο και αριθμείται σύμφωνα με το σύνδεσμο στον οποίο ανήκει. Συνοψίζοντας όλα τα παραπάνω, η ευθεία κινηματική ανάλυση περιλαμβάνει τα παρακάτω:

- Ποιά είναι η θέση και ο προσανατολισμός του άκρου (εργαλείο, αρπάγη) , όταν ξέρω τις γωνίες των αρθρώσεων του ρομπότ.
- Η οποία βοηθάει στην προσομείωση.
- Για την κινηματική ανάλυση ενός ρομποτικού χειριστή είναι πολύ σημαντικό να γνωρίζουμε καλά τις παρακάτω πολύ βασικές έννοιες:
 - Link (σύνδεσμος)
 - Joint (άρθρωση)
 - Actuator(κινητήρας-κινητήριο στοιχείο-ενεργοποιητής)
 - End Effector (τελικό στοιχείο δράσης)

Όλα τα παραπάνω μπορούν να γίνουν καλύτερα κατανοητά στην παρακάτω εικόνα 4.3



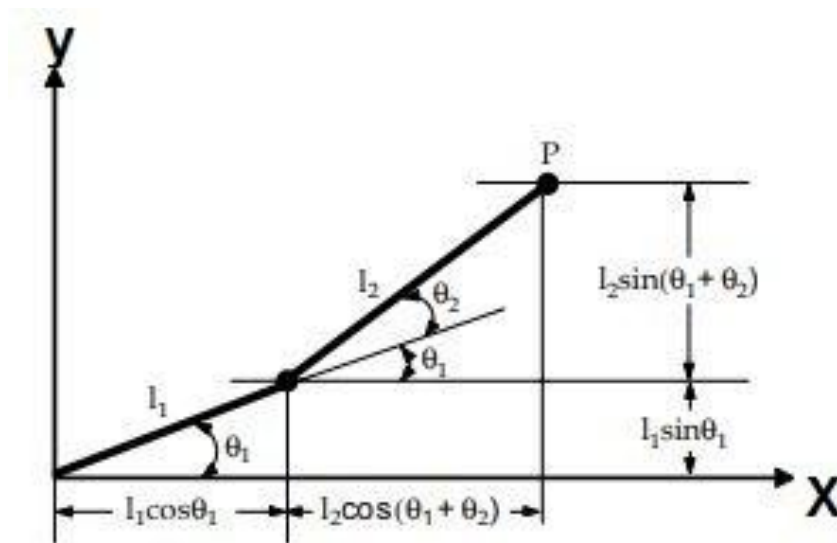
Εικ. 4.3 Link-joint-actuator-end effector



Εικόνα 4.4 Τοποθέτηση αξόνων επάνω στο βραχίονα

Για την περίπτωση της κινηματικής ανάλυσης του βραχίονα στην παρούσα διπλωματική εργασία, έχουμε δημιουργήσει ειδικό πρόγραμμα στο excel, όπου εισάγουμε τα δεδομένα και παίρνουμε την ανάλογη λύση με βάση αυτά (μήκη αρθρώσεων και μεταβολές των γωνιών). Η επίλυση θα μπορούσε να γίνει με διάφορους τρόπους όπως με πίνακες D-H ή Denavit-Handenberg, στην παρούσα εργασία όμως η κινηματική επίλυση θα γίνει με τη χρήση τριγωνομετρίας. Με βάση τα μήκη των συνδέσμων και τα είδη των αρθρώσεων έχουμε σχηματίσει τις εξισώσεις κίνησης του βραχίονα στους 3 άξονες X Y Z, κι έπειτα εισάγοντας τη γωνία που επιθυμούμε για την κίνηση της κάθε άρθρωσης ξεχωριστά, προκύπτει η τελική θέση στην οποία θα βρεθεί το άκρο του βραχίονα.

Οπότε η προσέγγιση της γεωμετρικής λύσης βασίζεται στην αποσύνθεση της χωρικής γεωμετρίας του Ρομποτικού Χειριστή, σε διάφορα προβλήματα γεωμετρίας επιπέδων. Εφαρμόζεται σε απλές δομές ρομποτικών χειριστών όπως για παράδειγμα στην περίπτωση 2 Dof. (βλέπε σχήμα 4.5)



Σχήμα 4.3 Κινηματική ανάλυση με τριγωνομετρία

Οπότε με βάση το σχήμα 4.5 και χρησιμοποιώντας τριγωνομετρική επίλυση, οι σχέσεις που προκύπτουν πολύ απλά είναι οι παρακάτω:

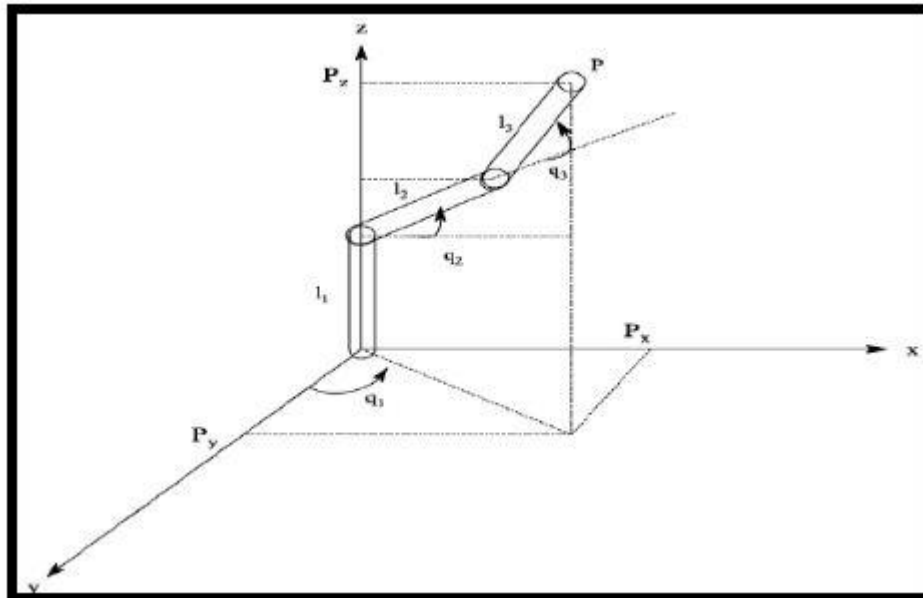
$$P_x = l_1 \cdot \cos\theta_1 + l_2 \cdot \cos(\theta_1 + \theta_2)$$

$$P_y = l_1 \cdot \sin\theta_1 + l_2 \cdot \sin(\theta_1 + \theta_2)$$

, οι σχέσεις αφορούν κίνηση του ρομποτικού χειριστή 2 Dof στους 2 άξονες X και Y. Στην περίπτωση μας όπου οι βαθμοί ελευθερίας γίνονται 3, όπου προστίθεται και ο υποβραχίονας της βάσης, θα έχουμε κίνηση και στους 3 άξονες X-Y-Z, οι σχέσεις τροποποιούνται και γίνονται ως έχει παρακάτω: (1) (2) και (3) είναι αυτές που θα μας δώσουν τη λύση στην επίλυση του ευθέως κινηματικού προβλήματος του ρομποτικού χειριστή.

- $P_x = (L_2 \cdot \cos(\theta_2) + L_3 \cdot \cos(\theta_2 + \theta_3)) \cdot \sin(\theta_1)$ (Σχέση 1),
- $P_y = (L_2 \cdot \cos(\theta_2) + L_3 \cdot \cos(\theta_2 + \theta_3)) \cdot \cos(\theta_1)$ (Σχέση 2)
- $P_z = L_2 \cdot \sin(\theta_2) + L_3 \cdot \sin(\theta_2 + \theta_3) + L_1$ (Σχέση 3)

,όπου **L1**, **L2** και **L3** είναι τα μήκη των συνδέσμων. Η διάταξη του βραχίονά μας φαίνεται στο παρακάτω σχήμα 4.6



Σχήμα 4.4 Βραχίονας περιστροφικών αρθρώσεων 3 βαθμών ελευθερίας

Επεξήγηση εξισώσεων:

Px , για τη μετατόπιση στον άξονα X πολλαπλασιάζω το μήκος του δεύτερου συνδέσμου (**L2**) με το συνημίτονο της γωνίας που σχηματίζει ο δεύτερος σύνδεσμος σε σχέση με τον πρώτο (**cosθ2**) κι έπειτα το προσθέτω με το γινόμενο του μήκους του τρίτου συνδέσμου με το συνημίτονο του αθροίσματος των γωνιών που σχηματίζει ο δεύτερος σύνδεσμος σε σχέση με τον πρώτο και ο τρίτος σύνδεσμος σε σχέση με τον δεύτερο, (**L3* cos(θ2+θ3)**), τέλος το άθροισμα αυτών των πράξεων θα το πολλαπλασιάσω με ο ημίτονο της πρώτης γωνίας , η οποία είναι η γωνία περιστροφής της βάση του βραχίονα (**sinθ1**). (**Σχέση 1**)

Py , για τη μετατόπιση στον άξονα Y πολλαπλασιάζω το μήκος του δεύτερου συνδέσμου (**L2**) με το συνημίτονο της γωνίας που σχηματίζει ο δεύτερος σύνδεσμος σε σχέση με τον πρώτο (**cosθ2**) κι έπειτα το προσθέτω με το γινόμενο του μήκους του τρίτου συνδέσμου με το συνημίτονο του αθροίσματος των γωνιών που σχηματίζει ο δεύτερος σύνδεσμος σε σχέση με τον πρώτο και ο τρίτος σύνδεσμος σε σχέση με τον δεύτερο, (**L3* cos(θ2+θ3)**), τέλος το άθροισμα αυτών των πράξεων θα το πολλαπλασιάσω με το συνημίτονο της πρώτης γωνίας , η οποία είναι η γωνία περιστροφής της βάση του βραχίονα (**cosθ1**). (**Σχέση 2**)

Pz , για τη μετατόπιση στον άξονα Z πολλαπλασιάζω το μήκος του δεύτερου συνδέσμου (**L2**) με το ημίτονο της γωνίας που σχηματίζει ο δεύτερος σύνδεσμος σε σχέση με τον πρώτο (**sinθ2**) κι έπειτα το προσθέτω με το γινόμενο του μήκους του τρίτου συνδέσμου

με το ημίτονο του αθροίσματος των γωνιών που σχηματίζει ο δεύτερος σύνδεσμος σε σχέση με τον πρώτο και ο τρίτος σύνδεσμος σε σχέση με τον δεύτερο.

($L3 * \sin(\theta2+\theta3)$), τέλος το άθροισμα αυτών των πράξεων θα το προσθέσω με το μήκος του πρώτου συνδέσμου ($L1$) . (Σχέση 3)

Σύνδεσμος	Μήκος
L1	114 mm
L2	104 mm
L3	223 mm

Πίνακας 4.1 Μήκη συνδέσμων

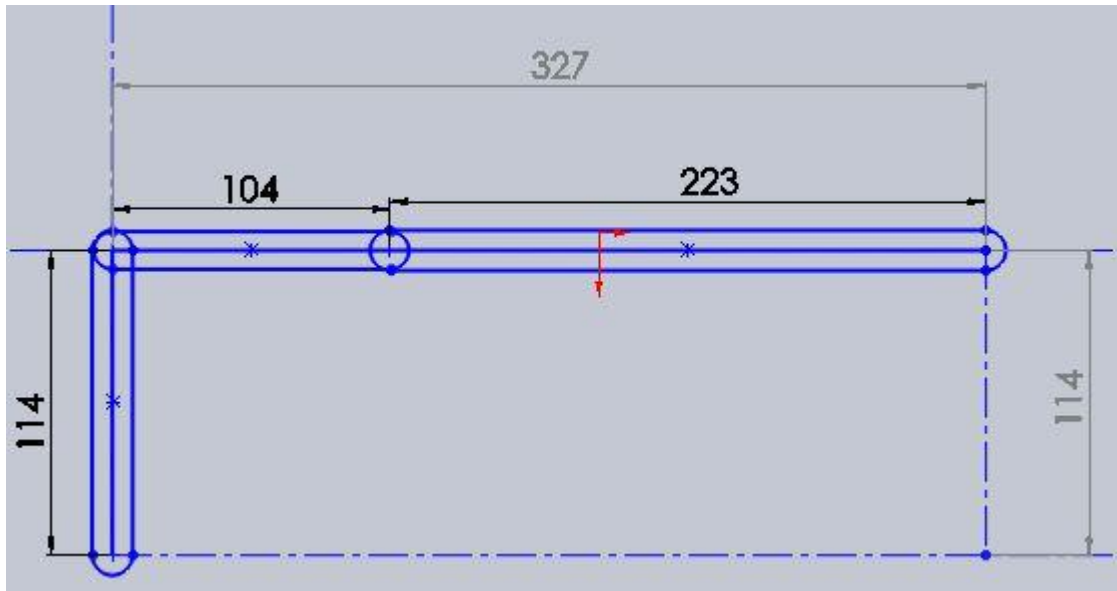
Οπότε για να γίνει πιο κατανοητή η επίλυση , του ευθέως κινηματικού προβλήματος θα προχωρήσουμε σε επίλυση μερικών περιπτώσεων , εισάγωντας στο πρόγραμμα διάφορες γωνίες , έτσι θα λάβουμε σαν αποτέλεσμα τα διάφορα P_x , P_y , P_z . Κι έπειτα θα κάνουμε επαλήθευση των αποτελεσμάτων σε σχεδιαστικό πρόγραμμα όπου θα μας δώσει τις πραγματικές μεταβολές των αρθρώσεων.

Αρχικά θα τοποθετήσουμε όλες τις γωνίες ίσες με μηδέν $\theta1=0$, $\theta2=0$, $\theta3=0$.

(παράδειγμα 1)

Corners	Deg	Rad	SOLUTION		A/A
theta1=	0	0	Px=	0	σχέση1
theta2=	0	0	Pγ=	327	σχέση2
theta3=	0	0	Pz=	114	σχέση3
Links mm			Forward kinematic		
L1=	114				
L2=	104				
L3=	223				

Πίνακας 4.2 Excel Επιβεβαίωσης αποτελέσματος παρ.1



Σχήμα 4.5 Επιβεβαίωση παρ.1 στο σχεδιαστικό πρόγραμμα

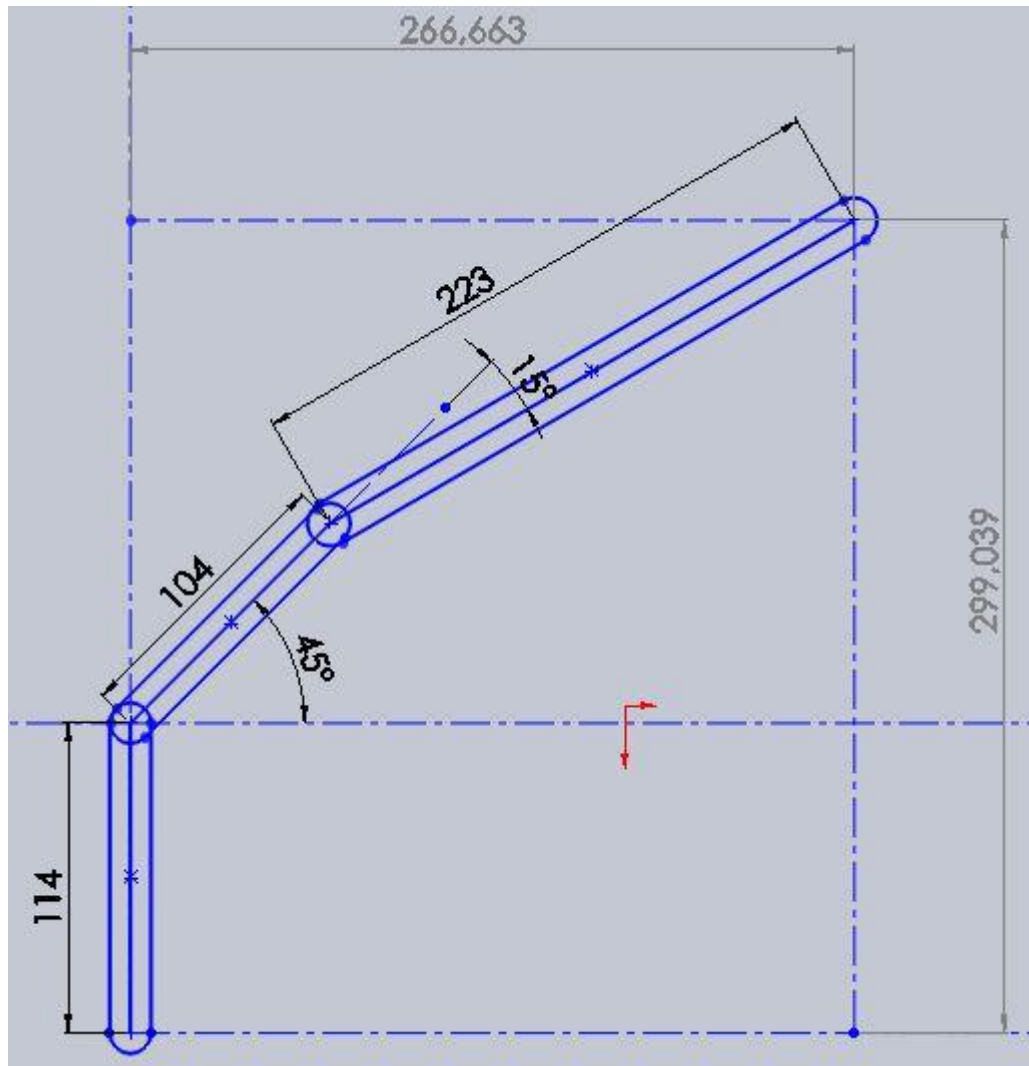
Στο σχήμα 4.5 γίνεται επιβεβαίωση του αποτελέσματος του πίνακα 4.2 και της πραγματικότητας με βάση τη γεωμετρία του βραχίονα, εάν τοποθετήσω όλες τις γωνίες ίσες με μηδέν τότε θα έχω $P_z=114\text{mm}$ και $P_y=327\text{mm}$, όπου στον άξονα Z έχω το μήκος του υποβραχίονα L1 και στον άξονα Y έχω το άθροισμα των μηκών των 2 υποβραχιόνων L1 και L2, με μετατόπιση μόνο στον άξονα των Y.

Αποτέλεσμα : $P_x=0\text{mm}$, $P_y=327\text{mm}$, $P_z=114\text{mm}$

Έπειτα θα τοποθετήσω $\theta_1=0$, $\theta_2=45$ και $\theta_3=-15$ (παράδειγμα 2)

Corners	Deg	Rad		
theta1=	0	0		
theta2=	45	0,785398		
theta3=	-15	-0,2618		
			SOLUTION	
			Px=	0
			Py=	266,6628
			Pz=	299,0391
			Forward kinematic	

Πίνακας 4.3 Excel Επιβεβαίωση αποτελέσματος παρ.2



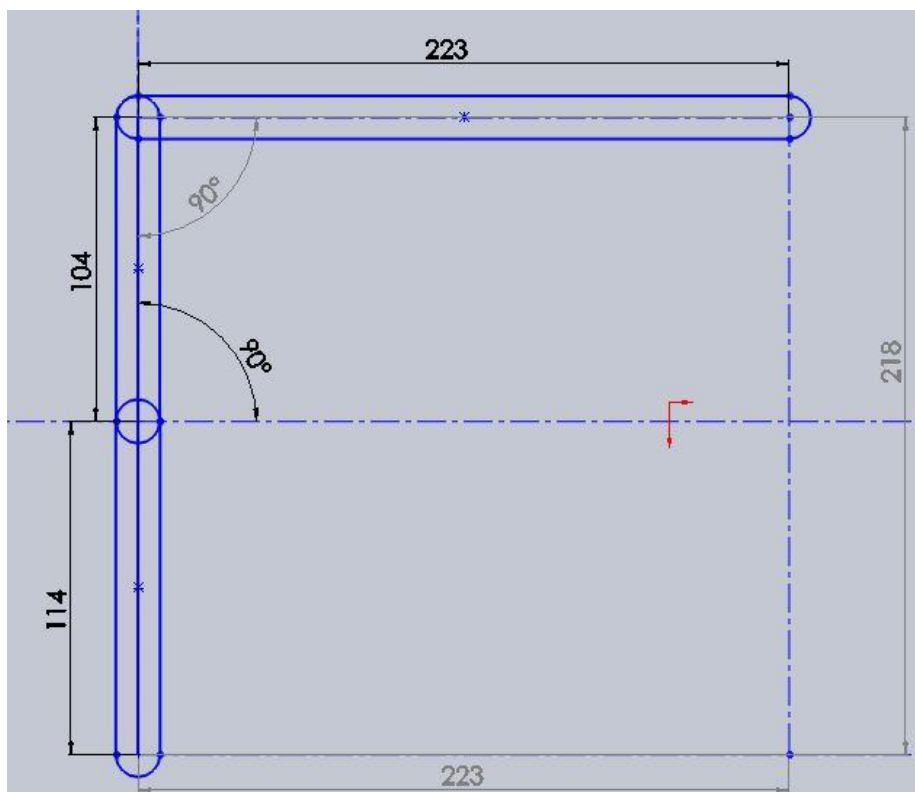
Σχήμα 4.6 Επιβεβαίωση παρ.2 στο σχεδιαστικό πρόγραμμα

Αποτέλεσμα : $P_x=0\text{mm}$, $P_y=266,663\text{mm}$, $P_z=299,039\text{mm}$

Για τις γωνίες $\theta_1=0$, $\theta_2=90$ και $\theta_3=-90$, έχουμε: (παράδειγμα 3)

	A	B	C	D	E	F	G	H
1	Corners	Deg	Rad					
2	theta1=	0	0					
3	theta2=	90	1,570796					
4	theta3=	-90	-1,5708					
5					SOLUTION		A/A	
6	Links	mm			Px=	0	σχέση1	
7	L1=	114			Pγ=	223	σχέση2	
8	L2=	104			Pz=	218	σχέση3	
9	L3=	223						
10					Forward kinematic			
11								
12								

Πίνακας 4.4 Excel Επιβεβαίωση αποτελέσματος παρ.3



Σχήμα 4.6 Επιβεβαίωση παρ. 3 στο σχεδιαστικό πρόγραμμα

Εάν βέβαια τοποθετήσω $\theta_1=0$, $\theta_2=90$ και $\theta_3=-90$, έχω λάβει το σωστό αποτέλεσμα διότι η μετατόπιση στον άξονα Z είναι το άθροισμα των μηκών του 1^{ου} και του 2^{ου} συνδέσμου

και η μετατόπιση στον Y είναι το μήκος του 3^{ου} συνδέσμου.
Αποτέλεσμα : : P_x=0mm, P_y=223mm , P_z=218mm

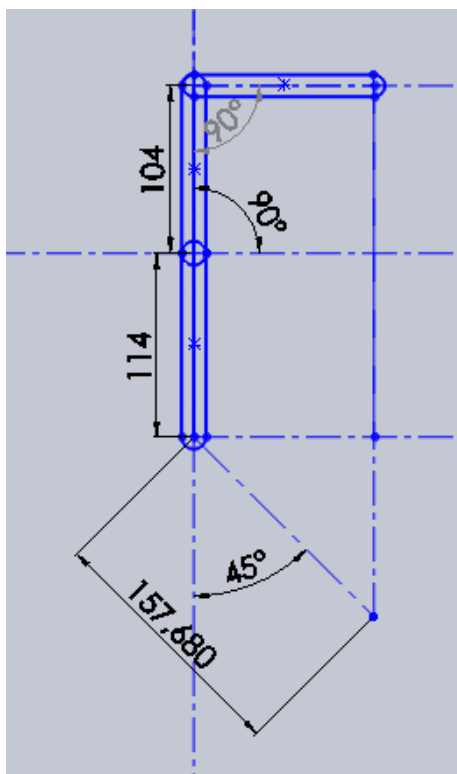
Για καλύτερο έλεγχο του προγράμματος που έχουμε δημιουργήσει θα εισάγω κίνηση και στις 3 γωνίες , οπότε αναμένω μετατόπιση και στους 3 άξονες X Y Z.Οπότε έχω θ₁=45, θ₂=90. θ₃=-90. (παράδειγμα 4)

	A	B	C	D	E	F	G	H
1	Corners	Deg	Rad					
2	theta1=	45	0,785398					
3	theta2=	90	1,570796					
4	theta3=	-90	-1,5708					
5								
6	Links	mm						
7	L1=	114						
8	L2=	104						
9	L3=	223						
10								
11								
12								

SOLUTION		A/A
P _x =	157,6848	σχέση1
P _y =	157,6848	σχέση2
P _z =	218	σχέση3

Forward kinematic

Πίνακας 4.5 Excel Επιβεβαίωση αποτελέσματος παρ.4



Σχήμα 4.7 Επιβεβαίωση παρ.4 στο σχεδιαστικό πρόγραμμα

Αποτέλεσμα : $P_x=157,6848\text{mm}$ $P_y=157,6848\text{mm}$, $P_z=218\text{mm}$

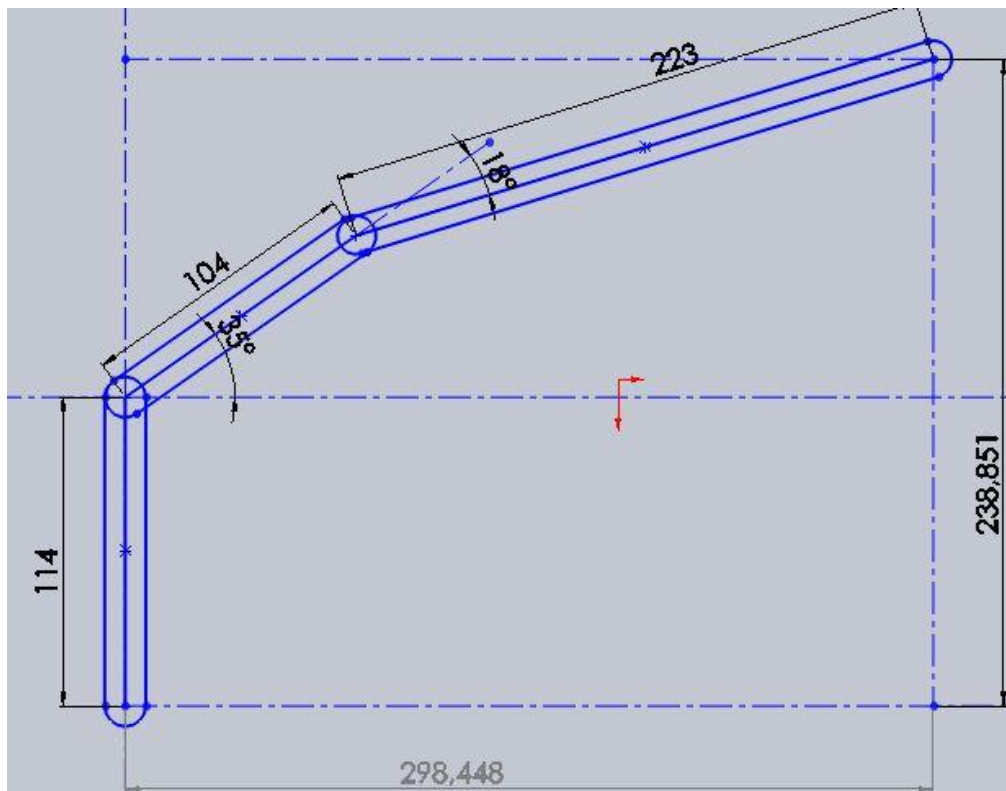
Για τις γωνίες $\theta_1=0$, $\theta_2=35$ και $\theta_3=-18$, έχουμε: (παράδειγμα 5)

	A	B	C	D	E	F	G	H
1	Corners	Deg	Rad					
2	theta1=	0	0					
3	theta2=	35	0,610865					
4	theta3=	-18	-0,31416					
5								
6	Links	mm						
7	L1=	114						
8	L2=	104						
9	L3=	223						
10								
11								
12								

SOLUTION		A/A
$P_x=$	0	σχέση1
$P_y=$	298,4478	σχέση2
$P_z=$	238,8508	σχέση3

Forward kinematic

Πίνακας 4.6 Excel Επιβεβαίωση αποτελέσματος παρ.5



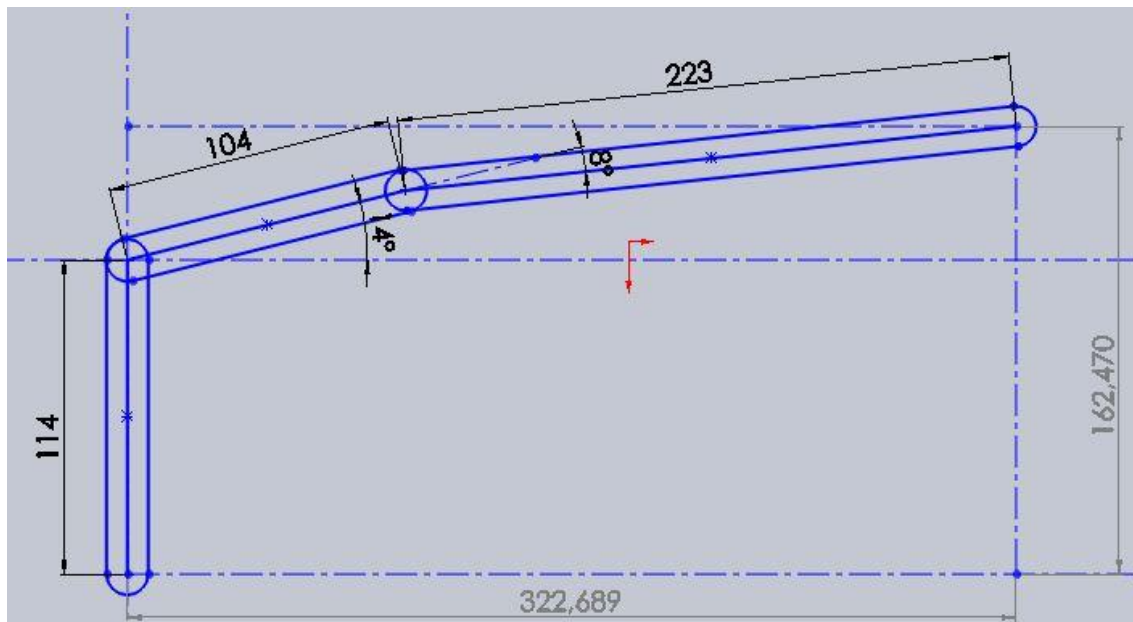
Σχήμα 4.8 Επιβεβαίωση παρ. 5 στο σχεδιαστικό πρόγραμμα

Αποτέλεσμα : $P_x=0\text{mm}$ $P_y=298,448\text{mm}$, $P_z=238,851\text{mm}$

Για τις γωνίες $\theta_1=0$, $\theta_2=14$ και $\theta_3=-8$, έχουμε: (παράδειγμα 6)

	A	B	C	D	E	F	G	H
1	Corners	Deg	Rad					
2	theta1=	0	0					
3	theta2=	14	0,244346					
4	theta3=	-8	-0,13963					
5					SOLUTION		A/A	
6	Links	mm			$P_x=$	0	σχέση1	
7	L1=	114			$P_y=$	322,6891	σχέση2	
8	L2=	104			$P_z=$	162,4697	σχέση3	
9	L3=	223						
10					Forward kinematic			
11								
12								

Πίνακας 4.7 Excel Επιβεβαίωση αποτελέσματος παρ.6



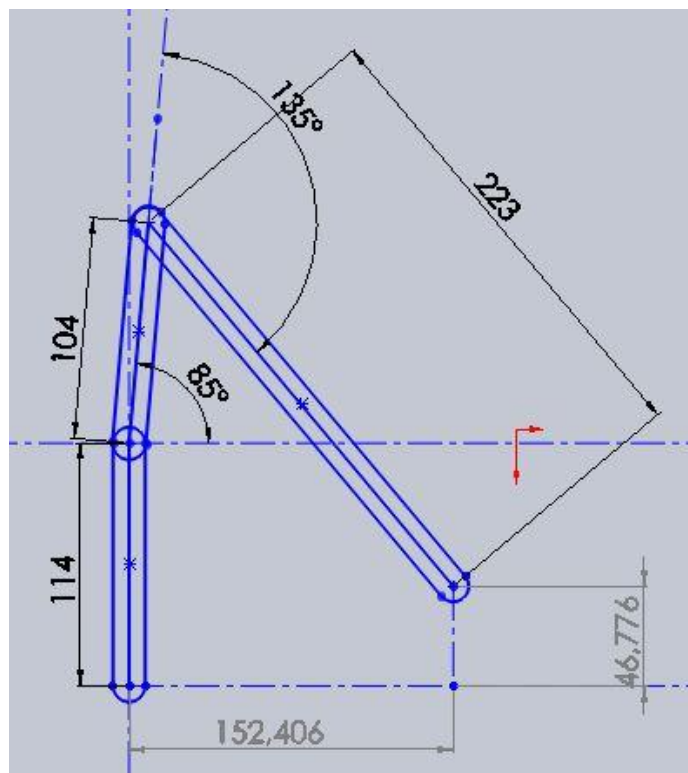
Σχήμα 4.9 Επιβεβαίωση παρ. 6 στο σχεδιαστικό πρόγραμμα

Αποτέλεσμα : $P_x=0\text{mm}$ $P_y=322,689\text{mm}$, $P_z=162,470\text{mm}$

Για τις γωνίες $\theta_1=0$, $\theta_2=85$ και $\theta_3=-135$, έχουμε: (παράδειγμα 7)

	A	B	C	D	E	F	G	H
1	Corners	Deg	Rad					
2	theta1=	0	0					
3	theta2=	85	1,48353					
4	theta3=	-135	-2,35619					
5					SOLUTION		A/A	
6	Links	mm			Px=	0	σχέση1	
7	L1=	114			Py=	152,4058	σχέση2	
8	L2=	104			Pz=	46,77634	σχέση3	
9	L3=	223						
10					Forward kinematic			
11								
12								

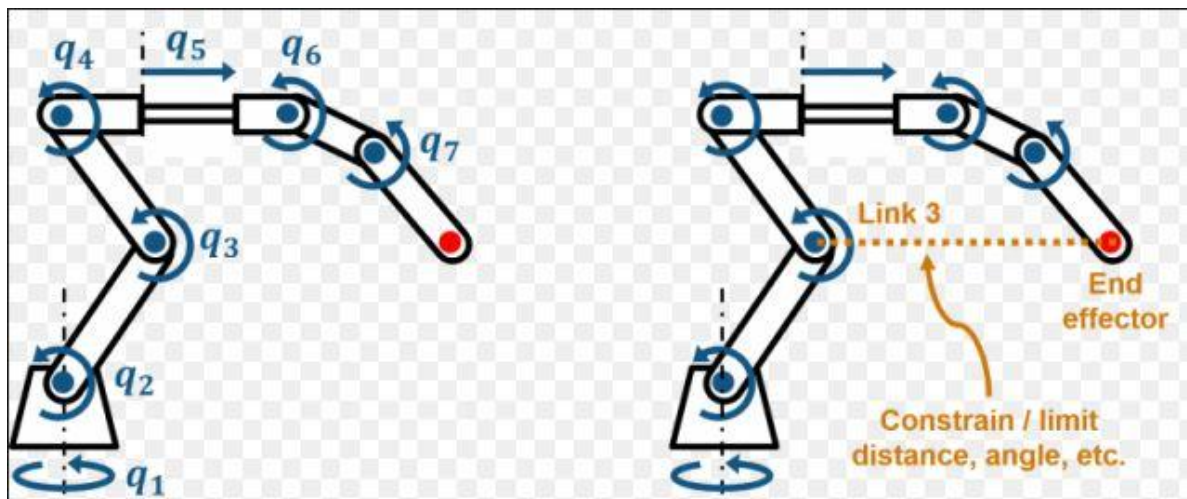
Πίνακας 4.8 Excel Επιβεβαίωση αποτελέσματος παρ.7



Σχήμα 4.10 Επιβεβαίωση παρ.7 στο σχεδιαστικό πρόγραμμα

Αποτέλεσμα : $P_x=0\text{mm}$ $P_y=152,406\text{mm}$, $P_z=46,776\text{mm}$

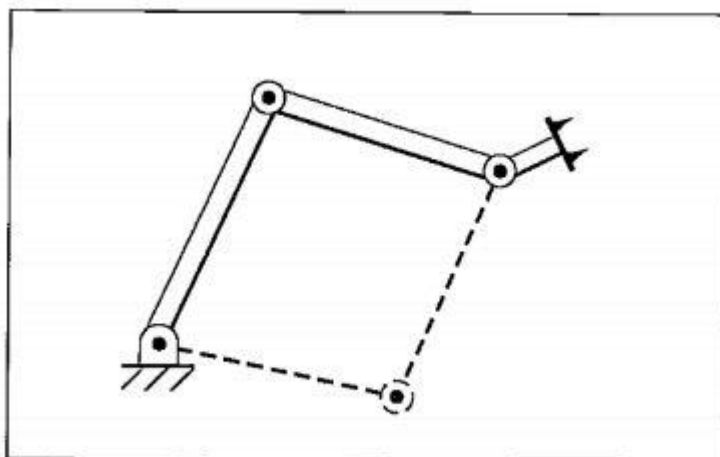
Συμπεράσματα: Το ευθύ κινηματικό πρόβλημα συνίσταται στην εύρεση της θέσης και του προσανατολισμού του άκρου του βραχίονα ως προς τη βάση του, όταν γνωρίζουμε τη θέση της κάθε άρθρωσης. Η δημιουργία των εξισώσεων του χώρου για τους 3 άξονες ξεχωριστά, μας βοήθησε στο να κατανοήσουμε καλύτερα την κινηματική ευθεία ανάλυση του ρομποτικού χειριστή 3 βαθμών ελευθερίας στο χώρο και το ειδικό πρόγραμμα που δημιουργήσαμε στο excel μας έδωσε τη δυνατότητα κάνουμε ανα πάσα στιγμή επιβεβαίωση της ορθότητας των αποτελεσμάτων. Εφόσον γνωρίζουμε το μήκος των συνδέσμων και τον τύπο της κάθε άρθρωσης, μπορούμε να δώσουμε διαφορετικές τιμές στις γωνίες της κάθε άρθρωσης κι έτσι είναι πολύ εύκολο να βρούμε το τελικό σημείο δράσης του άκρου στο χώρο. Σε οποιοδήποτε πρόβλημα κινηματικής ρομποτικού βραχίονα το πιο σημαντικό είναι η ορθή κατασκευή των εξισώσεων κίνησης κι έπειτα η επίλυση είναι πολύ εύκολη. Τέλος θα ήθελα να επισημάνω την ακρίβεια των αποτελεσμάτων σε κάθε μετατόπιση, επιβεβαιωμένων και σε σχεδιαστικό πρόγραμμα, επιβεβαίωση ότι οι κινηματικές εξισώσεις είναι απόλυτα σωστές.



Εικόνα 4.21 Η σπουδαιότητα της Κινηματική ανάλυσης Ρομποτικών Χειριστών για την εύρεση του τελικού σημείου του άκρου

4.3.1 Ανάστροφο κινηματικό πρόβλημα

Η αντίστροφη κινηματική ανάλυση ενός αρθρωτού βραχίονα αφορά την επίλυση ενός δυσκολότερου και πιο σύνθετου προβλήματος όπου, λαμβάνοντας υπόψη την επιθυμητή θέση και τον προσανατολισμό του εργαλείου του άκρου σε σχέση με τον σταθμό, πώς υπολογίζουμε το σύνολο των γωνιών των αρθρώσεων που επιτυγχάνουν το επιθυμητό αποτέλεσμα. Δεδομένης της επιθυμητής θέσης και του προσανατολισμού του άκρου του βραχίονα σε σχέση με τη βάση του, ζητάμε να βρούμε τις τιμές των μεταβλητών των αρθρώσεων. Δηλαδή, θέλουμε τις λύσεις της κινηματικής εξίσωσης. **Υπαρξη λύσεων**: Το ερώτημα αν υπάρχει κάποια λύση δημιουργεί το ζήτημα του χώρου εργασίας του χειριστή. Συνολικά, ο χώρος εργασίας είναι αυτός ο όγκος χώρου που μπορεί να φτάσει ο τελικός τελεστής του χειριστή. Για να υπάρξει λύση, το καθορισμένο σημείο στόχου πρέπει να βρίσκεται μέσα στο χώρο εργασίας. Μερικές φορές, είναι χρήσιμο να λάβουμε υπόψη δύο ορισμούς του χώρου εργασίας: Ο δεξιός χώρος εργασίας είναι αυτός ο όγκος χώρου στον οποίο ο τελικός τελεστής ρομπότ μπορεί να φτάσει με όλους τους προσανατολισμούς. Δηλαδή, σε κάθε σημείο του δεξιού χώρου εργασίας, ο τελικός τελεστής μπορεί να είναι αυθαίρετα προσανατολισμένος. Ο προσπελάσιμος χώρος εργασίας είναι αυτός ο όγκος χώρου που μπορεί να φτάσει το ρομπότ σε τουλάχιστον έναν προσανατολισμό. Είναι σαφές ότι ο δεξιός χώρος εργασίας είναι ένα υποσύνολο του προσπελάσιμου χώρου εργασίας. **Πολλαπλές λύσεις**: Ένα άλλο πιθανό πρόβλημα που συναντάται στην επίλυση κινηματικών εξισώσεων είναι αυτές των πολλαπλών λύσεων. Ένας επίπεδος βραχίονας με τρεις περιστροφικές αρθρώσεις έχει μεγάλο δεξιό χώρο εργασίας στο επίπεδο (δεδομένου του καλού μήκους των συνδέσμων και των μεγάλων αρθρώσεων), καθώς μπορεί να επιτευχθεί οποιαδήποτε θέση στο εσωτερικό του χώρου εργασίας του με οποιοδήποτε προσανατολισμό. Το σχήμα 4.11 δείχνει ένα επίπεδο τριών περιστροφικών αρθρώσεων βραχίονα με την οριστική θέση του τελεστικού εργαλείου.



Σχήμα 4.11 Βραχίονας 3 Dof

Περιορίζουμε την προσοχή μας λοιπόν σε μεθόδους λύσεων κλειστού τύπου, την **Γεωμετρική επίλυση** και την **Αλγεβρική επίλυση**.

Το ανάστροφο κινηματικό πρόβλημα :

1. Δεν είναι απλό
2. Δεν υπάρχει γενική προσέγγιση για την επίλυσή του
3. Για τις απλές κινηματικές δομές αρκεί ο σχεδιασμός του για να πάρουμε τη λύση.
4. Δυστυχώς η λύση δεν είναι μοναδική
5. Επίσης μπορεί να έχουμε: α) μια ή περισσότερες λύσεις β) άπειρες λύσεις γ) το πρόβλημα να μην έχει λύση
6. Οι αναλυτικές λύσεις είναι πιο αποτελεσματικές από υπολογιστική άποψη.

Οι δυσκολίες κατά την επίλυση:

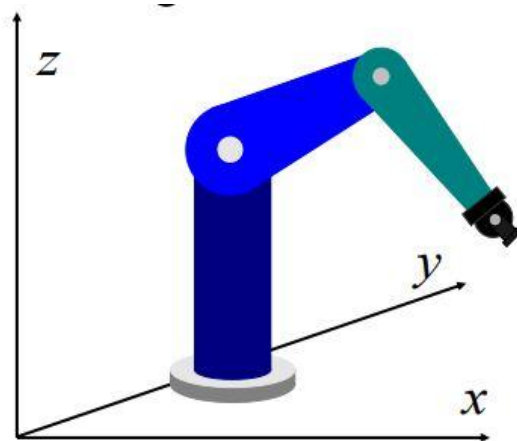
1. Περιστροφικές αρθρώσεις-Ανάστροφη τριγωνομετρία
2. Ασυνέχεια και ιδιαιτερότητες
3. Μπορεί να χαθεί ένας ή και περισσότεροι βαθμοί ελευθερίας σε μερικές διαμορφώσεις
4. Πολλαπλές λύσεις για μια καρτεσιανή θέση
5. Άπειρες λύσεις
6. Μερικές φορές μπορεί να μην υπάρχει δυνατότητα επίλυσης

Συγκεκριμένα :

Δεδομένης της επιθυμητής θέσης (**p**) και του προσανατολισμού (**R**) του τελικού σημείου του άκρου του βραχίονα

$$q = (q_1, q_2, \dots, q_n)$$

Βρίσκουμε τις μεταβλητές των αρθρώσεων , που μπορούν να φέρουν το τελικό σημείο δράσης στην επιθυμητή θέση.



Σχήμα 4.12 Άξονες κίνησης του βραχίονα

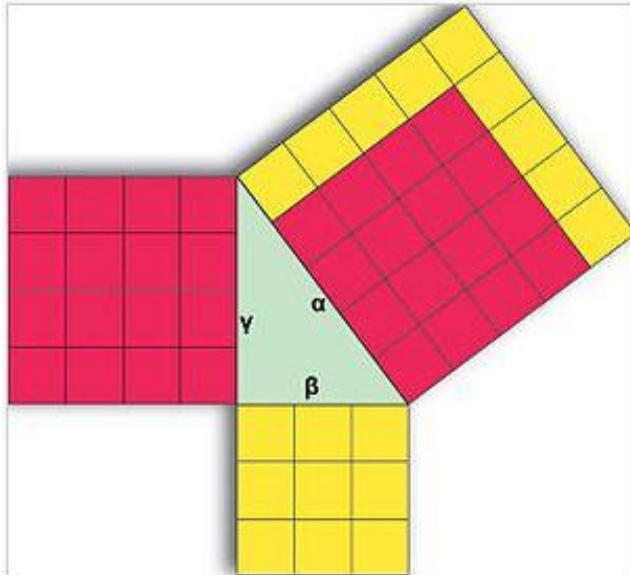
Η αντίστροφη κινηματική ανάλυση περιλαμβάνει:

- Ποιές γωνίες αρθρώσεων επιτυγχάνουν μια επιθυμητή θέση του άκρου.
- Βοηθάει στον έλεγχο

Με εφαρμογή του πυθαγορείου θεωρήματος, που εξ ονόματος αποδίδεται στον αρχαίο Έλληνα φιλόσοφο Πυθαγόρα: **«έν τοῖς ὀρθογωνίοις τριγώνοις τὸ ἀπὸ τῆς τῆν ὀρθὴν γωνίαν ὑποτείνουσας πλευρᾶς τετράγωνον ἴσον ἐστὶ τοῖς ἀπὸ τῶν τῆν ὀρθὴν γωνίαν περιεχουσῶν πλευρῶν τετραγώνοις.»**

Δηλαδή: «το τετράγωνο της υποτινούσης (της πλευράς που βρίσκεται απέναντι από την ορθή γωνία) ενός ορθογώνιου τριγώνου ισούται με το άθροισμα των τετραγώνων των δύο κάθετων πλευρών». Το θεώρημα μπορεί να γραφεί ως εξίσωση συσχετίζοντας τα μήκη των πλευρών α,β και γ,

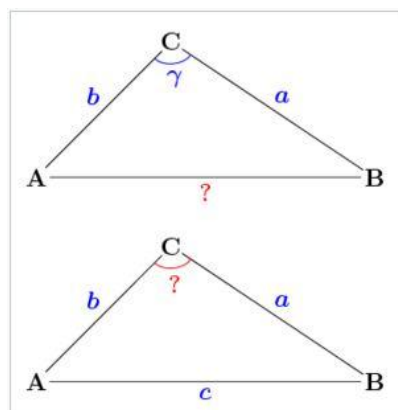
που ονομάζεται πυθαγόρεια εξίσωση: $\gamma^2 + \beta^2 = \alpha^2$ (όπου β και γ τα μήκη των δύο κάθετων πλευρών και α το μήκος της υποτεινουσας)



Σχήμα 4.13 Χρήση νόμων γεωμετρίας (α)

Το τετράγωνο της υποτεινούςας ισούται με το άθροισμα των τετραγώνων των 2 κάθετω πλευρών

Επίσης ο νόμος των συνημιτόνων είναι μια ειδική εφαρμογή του πυθαγορείου θεωρήματος όπου, βλέποντας το παρακάτω σχήμα 4.14



Σχήμα 4.14 Χρήση νόμων γεωμετρίας(β)

παρατηρούμε ότι ισχύουν τα εξής: Η γωνία (γ) έχει συνημίτονο ίσο με μηδέν ($\cos\gamma=0$), άν και μόνο αν $C^2=a^2+b^2$ Επομένως το θεώρημα χρησιμοποιείται στον τριγωνισμό, για να λύσει ένα τρίγωνο(βλέπε σχήμα 4.13), δηλαδή να προσδιορίσει:

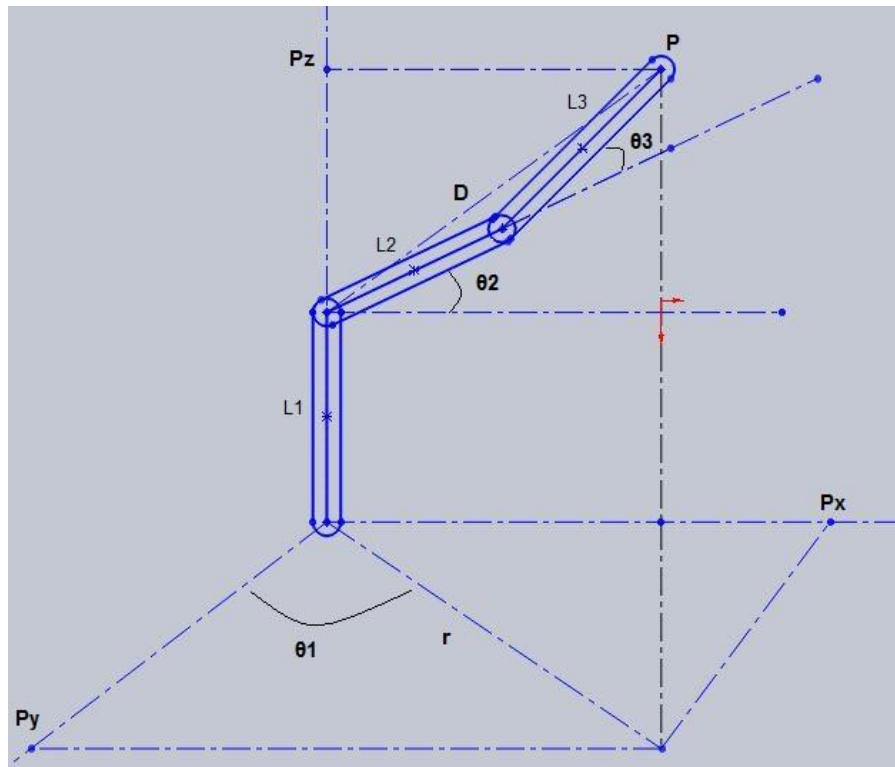
1. μια πλευρά ενός τριγώνου γνωρίζοντας την απέναντί της γωνία και τις παρακείμενες πλευρές

$$c = \sqrt{a^2 + b^2 - 2ab \cos \gamma}$$

2. μια γωνία ενός τριγώνου γνωρίζοντας τις τρεις πλευρές του

$$\gamma = \arccos \frac{a^2 + b^2 - c^2}{2ab}$$

Οπότε αντιστρέφοντας τον ορισμό της προς τα εμπρός κινηματικής ανάλυσης, έχουμε τον ορισμό της αντίστροφης κινηματικής. Με τις εξισώσεις της ανάστροφης κινηματικής μπορούμε να βρούμε τις κατάλληλες γωνίες για τις επιθυμητές συντεταγμένες τελεστή άκρου. Σύμφωνα με τους 2 ορισμούς της κινηματικής, είναι σαφές ότι η αντίστροφη κινηματική είναι πιο πολύπλοκη από την ευθεία κινηματική ανάλυση. Οπότε αυτό που μένει είναι να δημιουργήσουμε τις σχέσεις που θα μας δώσουν την ανάστροφη κινηματική ανάλυση του ρομποτικού βραχίονα που μελετάμε. Αρχικά χρειαζόμαστε σχέδιο με τη διάταξη του χειριστή για να δημιουργήσουμε τις εξισώσεις. Κι έπειτα με τη βοήθεια της τριγωνομετρίας θα επιλύσουμε γεωμετρικά το ανάστροφο κινηματικό πρόβλημα του βραχίονα.



Σχήμα 4.15 Διάταξη βραχίονα για την εξαγωγή εξισώσεων ανάστροφης κινηματικής

Με χρήση των νόμων της τριγωνομετρίας έχουμε:

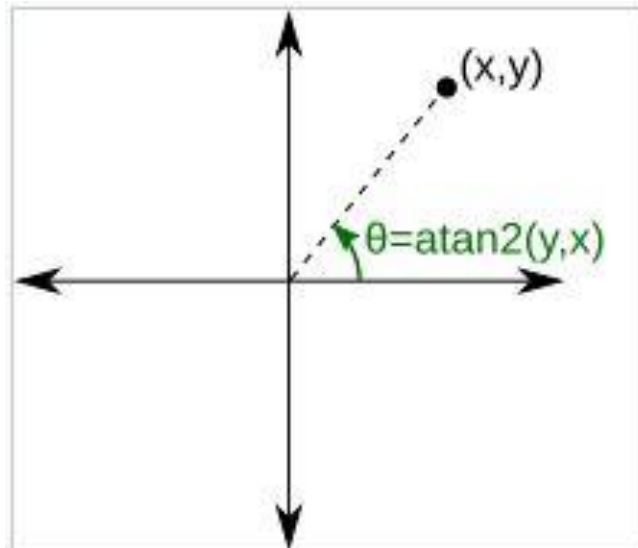
$$\theta_1 = \text{atan2}(P_y, P_x) \quad (\text{σχέση 4})$$

Η λειτουργία της $\text{atan2}(y,x)$ ή $\text{arctan2}(y,x)$ ορίζεται ως η γωνία στο ευκλείδιο επίπεδο, που δίνεται σε ακτίνια μεταξύ της θετικής x -ξιάς και της ακτίνας στο σημείο $(x,y) \neq (0,0)$, επιστρέφει μια μόνο τιμή θ τέτοια ώστε $-\pi < \theta \leq \pi$

Η τριγωνομετρική συνάρτηση $\text{atan2}(X_num; Y_num)$, αποδίδει το τόξο εφαπτομένης ή την αντίστροφη εφαπτομένη των συντεταγμένων x και y που έχουμε καθορίσει. Το τόξο εφαπτομένης είναι η γωνία του άξονα των x με μία γραμμή, η οποία διέρχεται από την αρχή των αξόνων $(0,0)$ και ένα σημείο με συντεταγμένες (x_num, y_num) . Η γωνία εκφράζεται σε ακτίνια μεταξύ $-\pi$ και π , εξαιρούμενης της τιμής $-\pi$.

X_num είναι η συντεταγμένη του x σημείου και

Y_num είναι η συντεταγμένη του y σημείου.



$\text{atan2}(y, x)$ επιστρέφει τη γωνία θ μεταξύ της ακτίνας προς το σημείο (x, y) και την θετική x -αξία, περιοριζόμενη στο $(-\pi, \pi)$.

Σχήμα 4.16 τριγωνομετρική συνάρτηση atan2

Ο υπολογισμός της απόστασης r , επίσης με τους νόμους της τριγωνομετρίας προκύπτει από την τετραγωνική ρίζα του αθροίσματος των τετραγώνων των 2 μετατοπίσεων, της μετατόπισης κατά τον άξονα x (P_x) και της μετατόπισης κατά τον άξονα y (P_y), (το αρνητικό και το θετικό πρόσημο λαμβάνεται ανάλογα με τη θέση του τελικού άκρου του βραχίονα).

$$r = \pm \sqrt{P_x^2 + P_y^2}$$

σχέση 5

Ο υπολογισμός της απόστασης D , η οποία είναι η απόσταση που ενώνει το σημείο χ_1 και το σημείο χ_3 , προκύπτει από την παρακάτω σχέση 8. Η οποία μας δείχνει ότι το D ισούται με :

$$D = \pm \sqrt{(P_z - l_1)^2 + r^2}$$

σχέση 6

, (το αρνητικό και το θετικό πρόσημο λαμβάνεται ανάλογα με τη θέση του βραχίονα (**Elbow Up** ή **Elbow Down** αντίστοιχα). Το D προκύπτει από την τετραγωνική ρίζα του αθροίσματος των παρακάτω πράξεων: της αφαίρεσης της μετατόπισης κατα τον άξονα z (**Pz**) μείον το μήκος του πρώτου συνδέσμου L_1 κι όλο αυτό υψωμένο στο τετράγωνο, προσθέτωντας στο αποτέλεσμα το (r) (**σχέση 7**) υψωμένο στο τετράγωνο, (βλέπε σχήμα 4.11).

$$\theta_3 = \arccos \left(\frac{D^2 - l_3^2 - l_2^2}{2l_2l_3} \right)$$

σχέση 7

Ο υπολογισμός της γωνίας θ_3 προκύπτει με εφαρμογή νόμων της τριγωνομετρίας, από το τόξο συνημιτόνου ή το αντίστροφο συνημίτονο ενός αριθμού, στην συγκεκριμένη περίπτωση η γωνία θ_3 είναι ίση με το αντίστροφο συνημίτονο του αποτελέσματος του παρακάτω κλάσματος: με αριθμητή την απόστασης (D^2) υψωμένης στο τετράγωνο μείον το μήκος του τρίτου συνδέσμου (L_3^2) υψωμένου στο τετράγωνο μείον το μήκος του **δευτέρου συνδέσμου** (L_2^2) επίσης υψωμένου στο τετράγωνο και παρονομαστή το γινόμενο της πράξης του μήκος του δευτέρου συνδέσμου (**L2**) επί το μήκος του τρίτου συνδέσμου (**L3**) επί 2 (δύο).

Επίσης για τον υπολογισμό της γωνίας θ_3 μπορούμε να χρησιμοποιήσουμε και τον παρακάτω τύπο, αυτό βέβαια μπορεί να γίνει και σαν επιβεβαίωση ορθότητας των αποτελεσμάτων μεταξύ (σχέσης 7) και (σχέσης 8).

$$\theta_3 = \arccos \left(\frac{(P_z - l_1)^2 + P_x^2 + P_y^2 - l_3^2 - l_2^2}{2l_2l_3} \right)$$

Σχέση 8

Επίσης το αποτέλεσμα θα είναι ίσο με το τόξο συνημιτόνου της διαίρεσης, με αριθμητή την μετατόπιση του άκρου κατά τον άξονα z (P_z) μείον το μήκος του πρώτου συνδέσμου (L_1) κι όλο αυτό στο τετράγωνο συν την μετατόπιση κατά τον άξονα x (P_x^2) υψωμένης στο τετράγωνο συν την μετατόπιση κατά τον άξονα y (P_y^2) υψωμένης στο τετράγωνο μείον το μήκος του τρίτου συνδέσμου (L_3^2) υψωμένου στο τετράγωνο μείον το μήκος του δεύτερου συνδέσμου (L_2^2) υψωμένου στο τετράγωνο και παρονομαστή το μήκος του τρίτου συνδέσμου (L_3) επί το μήκος του δεύτερου συνδέσμου (L_2) επί 2 (δύο).

Στη σχέση 9 παρατηρούμε τον τύπο που χρησιμοποιήσαμε για να βρούμε τη γωνία θ_2 στην επίλυση της ανάστροφης κινηματικής,

$$\theta_2 = \text{atan2}(r, P_z - l_1) - \text{atan2}(l_2 + l_3 \cos \theta_3, l_3 \sin \theta_3)$$

Σχέση 9

Όπου η γωνία θ_2 ισούται, με τη διαφορά της αντίστοιχης εφαπτομένης της απόστασης r και της διαφοράς P_z (μετατόπιση στον τον άξονα z) μείον το μήκος του πρώτου συνδέσμου και της αντίστροφης εφαπτομένης του μήκους του δεύτερου συνδέσμου συν το γινόμενο του μήκος του τρίτου συνδέσμου επί το συνημίτονο της γωνίας θ_3 και το γινόμενο του ημιτόνου της γωνίας θ_3 επί το μήκος του τρίτου συνδέσμου.

Συγκεντρωτικά οι εξισώσεις που χρησιμοποιήθηκαν για την επίλυση του ανάστροφου κινηματικού:

$$\theta_1 = \text{atan2}(P_y, P_x) \text{ (σχέση 4)}$$

$$r = \sqrt{P_x^2 + P_y^2} \text{ (σχέση 5)}$$

$$D = \sqrt{(P_z - l_1)^2 + r^2} \text{ (σχέση 6)}$$

$$\theta_3 = \arccos\left(\frac{D^2 - l_3^2 - l_2^2}{2 * l_2 * l_3}\right) \text{ (σχέση 7)}$$

$$\theta_2 = \text{atan2}(r, P_z - l_1) - \text{atan2}(l_2 + (l_3 * \cos(\theta_3)), l_3 * \sin(\theta_3)) \text{ (σχέση 9)}$$

4.3.2 Εποπτικά παραδείγματα ανάστροφου κινηματικού βραχίονα 3DOF

Με τη χρήση όλων των προηγούμενων σχέσεων που δημιουργήσαμε για την επίλυση της ανάστροφης κινηματικής, τις οποίες έχουμε εισαγάγει σε αρχείο excel για αυτόματη αναπαραγωγή των αποτελεσμάτων και με τη βοήθεια του σχεδιαστικού προγράμματος Solidworks, θα κάνουμε για τις διάφορες λύσεις του ανάστροφου κινηματικού επιβεβαίωση των αποτελεσμάτων.....

Παράδειγμα 1) Για τη θέση του ευθέως κινηματικού $\theta_1=0^\circ$, $\theta_2=28^\circ$, $\theta_3=-28^\circ$ και τις ακόλουθες μετατοπίσεις

$P_x=0.0\text{mm}$, $P_y=314.82\text{mm}$, $P_z=162.82\text{mm}$ (βλέπε πίνακα 4.9-4.10)

Corners	Deg	Rad		
theta1=	0	0		
theta2=	28	0,488692		
theta3=	-28	-0,48869		
			SOLUTION	A/A
			Px=	0
			Py=	314,8265
			Pz=	162,825
			σχέση1	
			σχέση2	
			σχέση3	
			Forward kinematic	
Links		mm		
L1=	114			
L2=	104			
L3=	223			

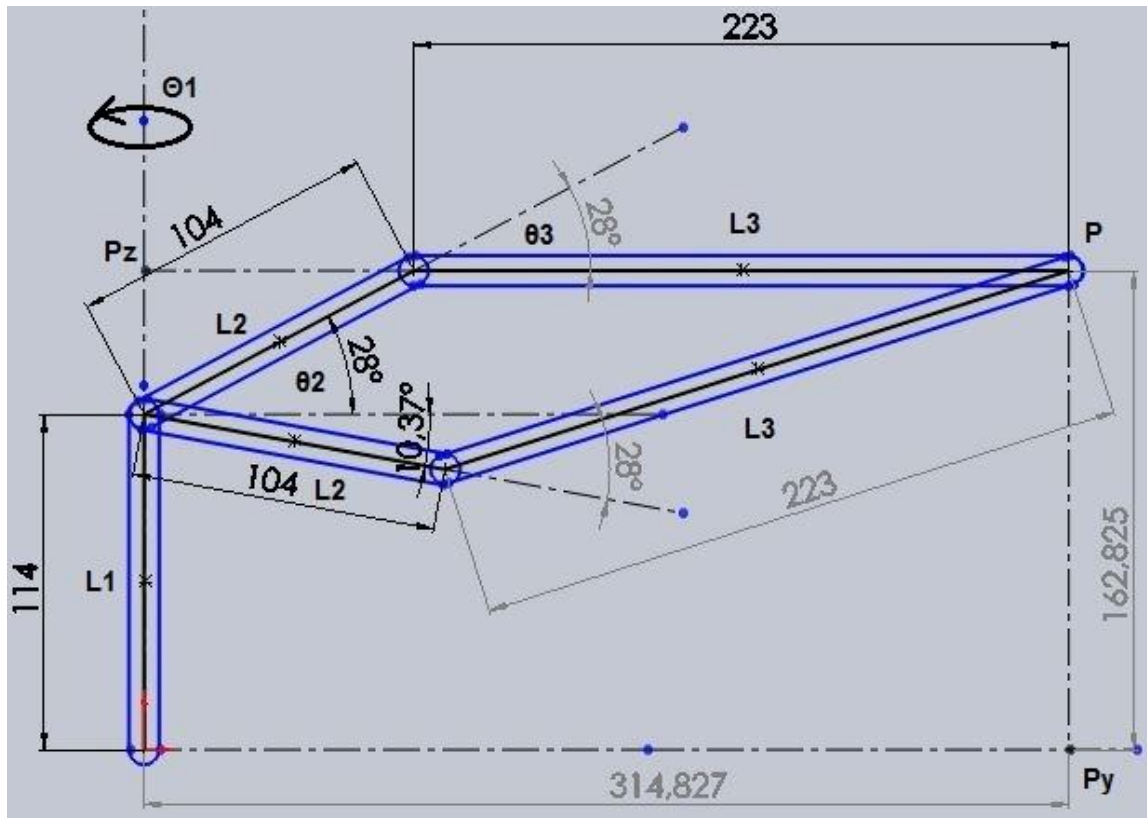
Πίνακας 4.9

Οπότε για το ανάστροφο κινηματικό θα έχουμε 2 λύσεις , για τη θέση του βραχίονα πάνω και βραχίονα κάτω, γνωστές ως **Elbow-up** **Elbow-down**(βλέπε παρακάτω πίνακα 4.10)

I	J	K	L	M	N	O	P	Q	R	S	T	U
Elbow Down		Rad	Deg	A/A				Elbow Up		Rad	Deg	A/A
	theta1=	0	0	σχέση 4					theta1=	0	0	σχέση 4
	r=	-314,827		σχέση 5					r=	314,8265		σχέση 5
	D=	-318,59		σχέση 6					D=	318,5901		σχέση 6
	theta3=	0,488692	28	σχέση 7					theta3=	-0,48869	-28	σχέση 7
	theta3=	0,488692	28	σχέση 8					theta3=	-0,48869	-28	σχέση 8
	theta2=	-0,18097	-10,369	σχέση 9					theta2=	0,488692	28	σχέση 9
				Inverse kinematic								Inverse kinematic
ΕΠΛΗΘΕΥΣΗ ΑΠΟ ΑΝΑΣΤΡΟΦΟ ΣΕ ΕΥΘΥ						ΕΠΛΗΘΕΥΣΗ ΑΠΟ ΑΝΑΣΤΡΟΦΟ ΣΕ ΕΥΘΥ						
Px=	0					Px=	0					
Py=	314,8265					Py=	314,8265					
Pz=	162,825					Pz=	162,825					

Πίνακας 4.10 Παρ.1

Στον πίνακα 4.10 μπορούμε να παρατηρήσουμε τα αποτελέσματα με ακρίβεια χιλιοστού , όπως και την επαλήθευση απο την επίλυση του ανάστροφου κινηματικού ξανά σε ευθύ κινηματικό κινηματικό , αντλώντας τα δεδομένα απο την επίλυση του ανάστροφου. Στο παρακάτω σχήμα 4.17 επίσης παρατηρούμε τις μεταβολές των θέσεων των συνδέσμων του βραχίονα με βάσει τα δεδομένα των πινάκων , απεικονίζεται η ανάστροφη κινηματική επίλυση (elbow-up - elbow-down) και το πιο σημαντικό είναι η ακρίβεια των αποτελεσμάτων του σχεδιαστικού προγράμματος (βλέπε σχ. 4.17) που επιστρέφει πραγματικά δεδομένα.



Σχήμα 4.17 παράδειγμα 1

Παράδειγμα 2) Για τη θέση του ευθέως κινηματικού $\theta_1=28^\circ$, $\theta_2=28^\circ$, $\theta_3=-28^\circ$ και τις ακόλουθες μετατοπίσεις $P_x=147.8\text{mm}$, $P_y=277.97\text{mm}$, $P_z=162.82\text{mm}$ Παρατηρούμε πως η μετατόπιση στον άξονα z παραμένει σταθερή και έχουμε μετατόπιση κατά τον άξονα x, ενώ μεταβάλεται και η μετατόπιση τον y. Αξιοσημείωτη είναι η ορθότητα των αποτελεσμάτων από ευθύ σε ανάστροφο και πάλι σε ευθύ (βλέπε πίνακα 4.11)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Corners	Deg	Rad						Elbow Down	Rad	Deg	A/A				Elbow Up	Rad	Deg	A/A	
2	theta1=	28	0,488692						theta1=	0,488692	28	σχέση 4				theta1=	0,488692	28	σχέση 4	
3	theta2=	28	0,488692																	
4	theta3=	-28	-0,48869																	
5					SOLUTION		A/A													
6	Links	mm			Px=	147,8021	σχέση1		r=	-314,827		σχέση 5				r=	314,8265		σχέση 5	
7	L1=	114			Py=	277,9753	σχέση2		D=	-318,59		σχέση 6				D=	318,5901		σχέση 6	
8	L2=	104			Pz=	162,825	σχέση3		theta3=	0,488692	28	σχέση 7				theta3=	-0,48869	-28	σχέση 7	
9	L3=	223							theta3=	0,488692	28	σχέση 8				theta3=	-0,48869	-28	σχέση 8	
10								Forward kinematic												
11												Inverse kinematic								Inverse kinematic
12																				
13																				
14									ΕΠΙΛΗΘΕΥΣΗ ΑΠΟ ΑΝΑΣΤΡΟΦΟ ΣΕ ΕΥΘΥ							ΕΠΙΛΗΘΕΥΣΗ ΑΠΟ ΑΝΑΣΤΡΟΦΟ ΣΕ ΕΥΘΥ				
15									Px=	147,8021						Px=	147,8021			
16									Py=	277,9753						Py=	277,9753			
17									Pz=	162,825						Pz=	162,825			
18																				

Πίνακας 4.11 παρ.2

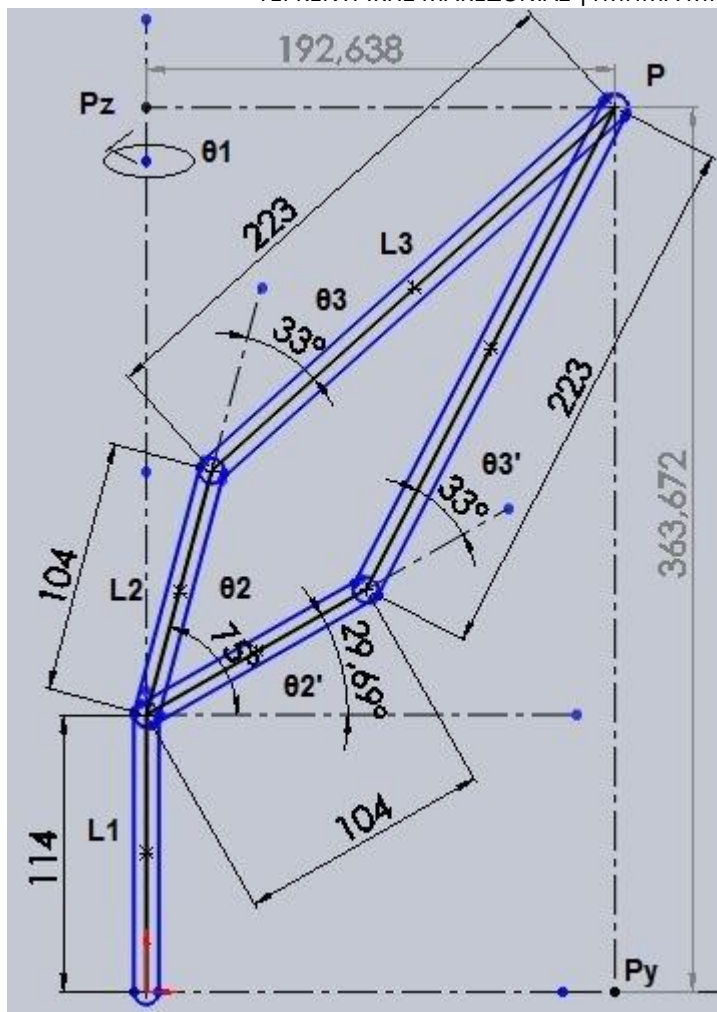
Παράδειγμα 3) Για τη θέση του ευθέως κινηματικού $\theta_1=0^\circ$, $\theta_2=75^\circ$, $\theta_3=-33^\circ$ και τις ακόλουθες μετατοπίσεις $P_x=0.0\text{mm}$, $P_y=192.638\text{mm}$, $P_z=363.672\text{mm}$ Παρατηρούμε επίσης ότι τα αποτελέσματα των μετατοπίσεων του ευθέως κινηματικού (πίνακας 4.12), τα αποτελέσματα –γωνίες του ανάστροφου(πίνακας 4.13) και τα αποτελέσματα της επαλήθευσης συμφωνούν απόλυτα με τις τιμές του σχήματος 4.18

Corners	Deg	Rad			
theta1=	0	0			
theta2=	75	1,308997			
theta3=	-33	-0,57596			
				SOLUTION	A/A
			Px=	0	σχέση1
			Py=	192,6385	σχέση2
			Pz=	363,6724	σχέση3
				Forward kinematic	

Πίνακας 4.12 παρ.3

Elbow Down				Rad	Deg	A/A	Elbow Up				Rad	Deg	A/A
	theta1=	0	0			σχέση 4		theta1=	0	0			σχέση 4
	r=	-192,638				σχέση 5		r=	192,6385				σχέση 5
	D=	-315,35				σχέση 6		D=	315,3504				σχέση 6
	theta3=	0,575959	33			σχέση 7		theta3=	-0,57596	-33			σχέση 7
	theta3=	0,575959	33			σχέση 8		theta3=	-0,57596	-33			σχέση 8
	theta2=	0,518275	29,69496			σχέση 9		theta2=	1,308997	75			σχέση 9
						Inverse kinematic							Inverse kinematic
ΕΠΛΗΘΕΥΣΗ ΑΠΟ ΑΝΑΣΤΡΟΦΟ ΣΕ ΕΥΘΥ						ΕΠΛΗΘΕΥΣΗ ΑΠΟ ΑΝΑΣΤΡΟΦΟ ΣΕ ΕΥΘΥ							
Px=	0						Px=	0					
Py=	192,6385						Py=	192,6385					
Pz=	363,6724						Pz=	363,6724					

Πίνακας 4.13 παρ.3



Σχήμα 4.18 παράδειγμα 3

Παράδειγμα 4) Για τη θέση του ευθέως κινηματικού $\theta_1=0^\circ$, $\theta_2=68^\circ$, $\theta_3=-88^\circ$ και τις ακόλουθες μετατοπίσεις $P_x=0.0$ mm, $P_y=248.51$ mm, $P_z=134.15$ mm Παρατηρούμε επίσης ότι τα αποτελέσματα των μετατοπίσεων του ευθέως κινηματικού (πίνακας 4.14), τα αποτελέσματα του ανάστροφου κινηματικού (πίνακας 4.15) και τα αποτελέσματα της επαλήθευσης συμφωνούν απόλυτα με τις τιμές του σχήματος 4.19

Corners		Deg	Rad
theta1=	0	0	
theta2=	68	1,186824	
theta3=	-88	-1,53589	

Links		mm
L1=	114	
L2=	104	
L3=	223	

SOLUTION		A/A
Px=	0	σχέση1
Py=	248,5105	σχέση2
Pz=	134,1566	σχέση3

Forward kinematic	
-------------------	--

Πίνακας 4.14 παρ.4

Elbow Down				Rad	Deg	A/A
theta1=	0	0	σχέση 4			
r=	-248,511		σχέση 5			
D=	-249,327		σχέση 6			
theta3=	1,53589	88	σχέση 7			
theta3=	1,53589	88	σχέση 8			
theta2=	-1,02496	-58,7258	σχέση 9			

Inverse kinematic	
-------------------	--

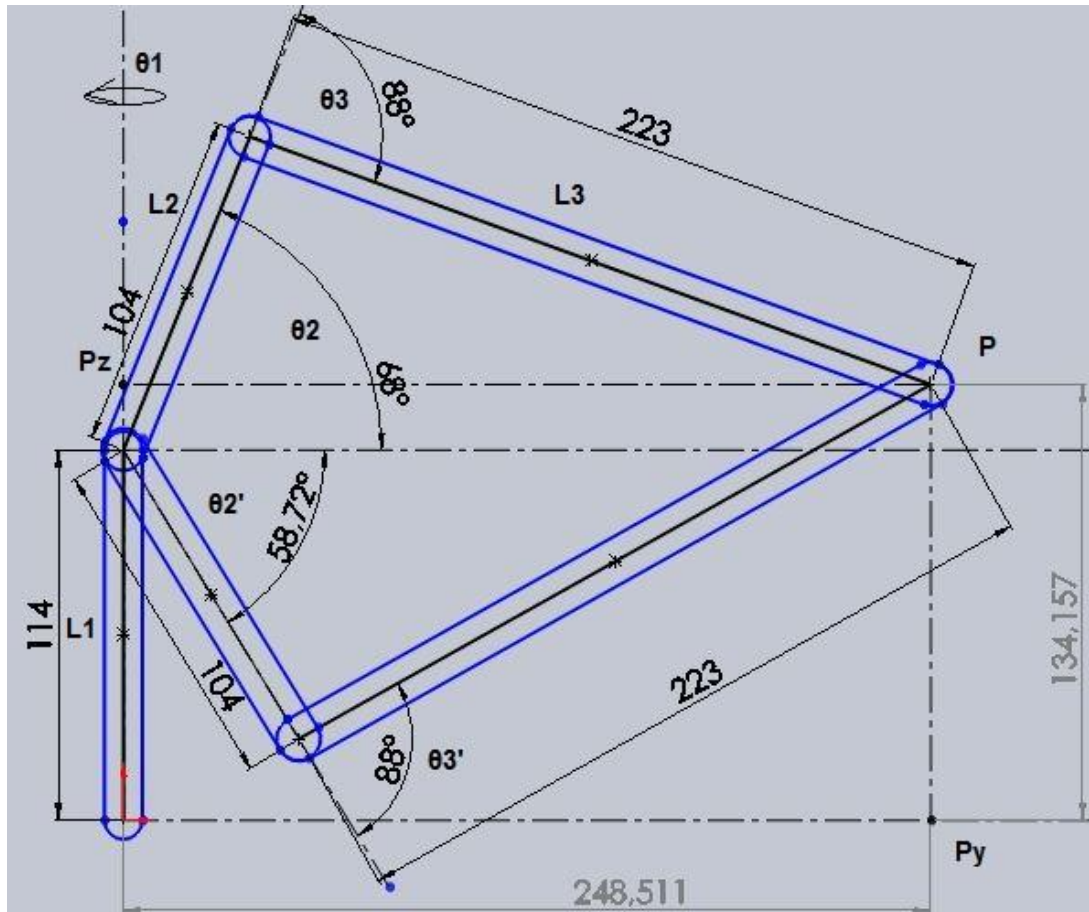
Elbow Up				Rad	Deg	A/A
theta1=	0	0	σχέση 4			
r=	248,5105		σχέση 5			
D=	249,3266		σχέση 6			
theta3=	-1,53589	-88	σχέση 7			
theta3=	-1,53589	-88	σχέση 8			
theta2=	1,186824	68	σχέση 9			

Inverse kinematic	
-------------------	--

ΕΠΛΗΘΕΥΣΗ ΑΠΟ ΑΝΑΣΤΡΟΦΟ ΣΕ ΕΥΘΥ	
Px=	0
Py=	248,5105
Pz=	134,1566

ΕΠΛΗΘΕΥΣΗ ΑΠΟ ΑΝΑΣΤΡΟΦΟ ΣΕ ΕΥΘΥ	
Px=	0
Py=	248,5105
Pz=	134,1566

Πίνακας 4.15 παρ.4



Σχήμα 4.19 παράδειγμα 4

Παράδειγμα 5) Για τη θέση του ευθέως κινηματικού $\theta_1=45^\circ$, $\theta_2=68^\circ$, $\theta_3=-88^\circ$ και τις ακόλουθες μετατοπίσεις $P_x=175.72\text{mm}$, $P_y=175.72\text{mm}$, $P_z=134.15\text{mm}$

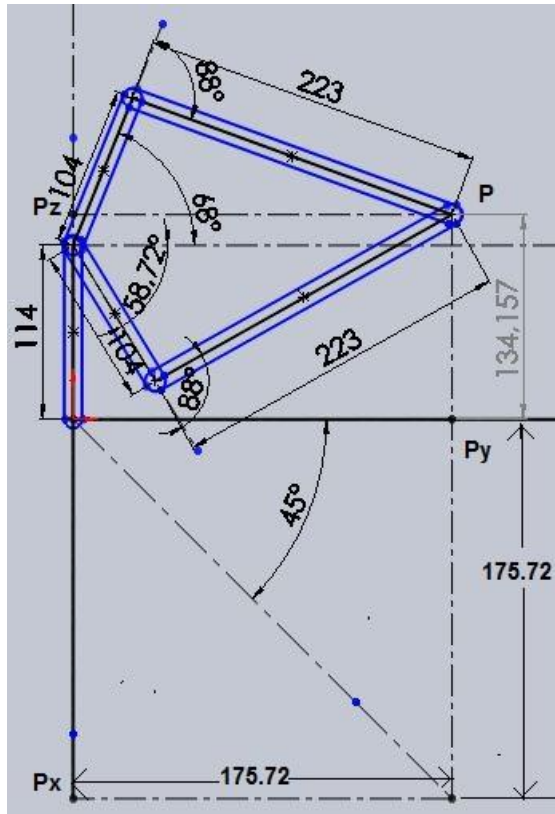
Corners	Deg	Rad		
theta1=	45	0,785398		
theta2=	68	1,186824		
theta3=	-88	-1,53589		
			SOLUTION	A/A
			Px=	175,7235
			Pγ=	175,7235
			Pz=	134,1566
Links		mm		
L1=	114			
L2=	104			
L3=	223			
			Forward kinematic	

Πίνακας 4.16 παρ.5

Περιστρέφοντας και τη γωνία θ_1 45 μοίρες θα έχω μετατόπιση και στον άξονα x πλέον, όπως και μεταβολή στον άξονα y , ενώ η μετατόπιση στον z παραμένει σταθερή διότι οι γωνίες θ_2 και θ_3 παραμένουν οι ίδιες.

Elbow Down				Rad	Deg	A/A	Elbow Up				
	theta1=	0,785398	45	σχέση 4			theta1=	0,785398	45	σχέση 4	
	r=	-248,511		σχέση 5			r=	248,5105		σχέση 5	
	D=	-249,327		σχέση 6			D=	249,3266		σχέση 6	
	theta3=	1,53589	88	σχέση 7			theta3=	-1,53589	-88	σχέση 7	
	theta3=	1,53589	88	σχέση 8			theta3=	-1,53589	-88	σχέση 8	
	theta2=	-1,02496	-58,7258	σχέση 9			theta2=	1,186824	68	σχέση 9	
						Inverse kinematic					
ΕΠΛΗΘΕΥΣΗ ΑΠΟ ΑΝΑΣΤΡΟΦΟ ΣΕ ΕΥΘΥ						ΕΠΛΗΘΕΥΣΗ ΑΠΟ ΑΝΑΣΤΡΟΦΟ ΣΕ ΕΥΘΥ					
Px=	175,7235					Px=	175,7235				
Pγ=	175,7235					Pγ=	175,7235				
Pz=	134,1566					Pz=	134,1566				

Πίνακας 4.17 παρ.5



Σχήμα 4.20 παράδειγμα 5

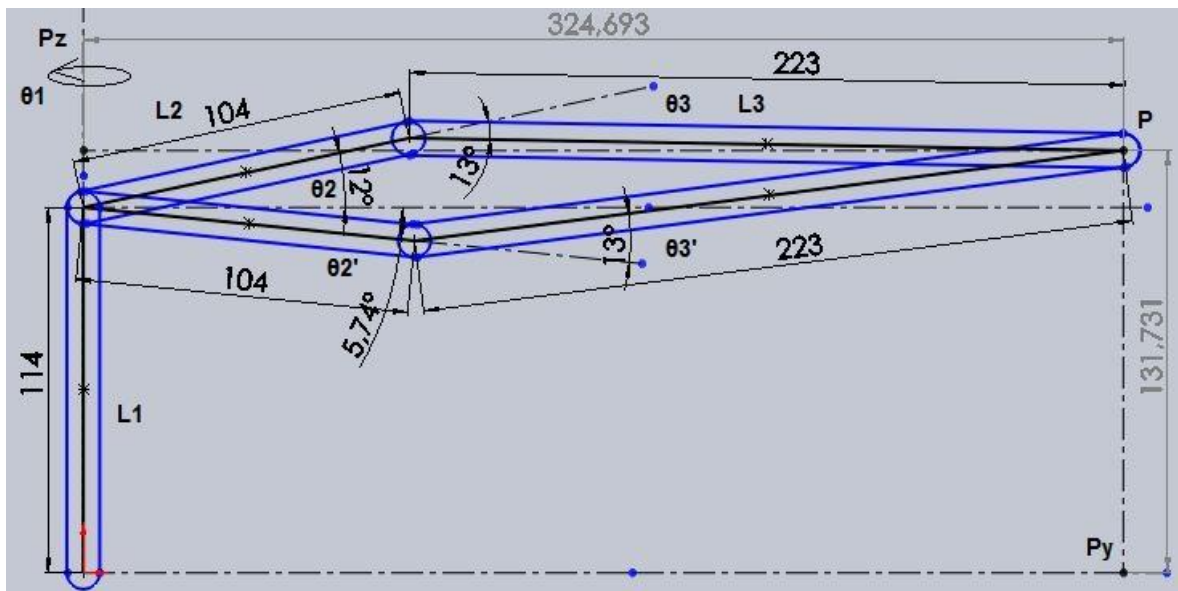
Παράδειγμα 6) Για τη θέση του ευθέως κινηματικού $\theta_1=0.0^\circ$, $\theta_2=12^\circ$, $\theta_3=-13^\circ$ και τις ακόλουθες μετατοπίσεις $P_x=0.0\text{mm}$, $P_y=324.69\text{mm}$, $P_z=131.73\text{mm}$

A	B	C	D	E	F	G
Corners	Deg	Rad				
theta1=	0	0				
theta2=	12	0,20944				
theta3=	-13	-0,22689				
Links	mm					
L1=	114					
L2=	104					
L3=	223					
				SOLUTION		A/A
				Px=	0	σχέση1
				Py=	324,6934	σχέση2
				Pz=	131,7309	σχέση3
				Forward kinematic		

Πίνακας 4.18 παρ.6

Elbow Down				Rad	Deg	A/A	Elbow Up				Rad	Deg	A/A
theta1=	0	0	σχέση 4				theta1=	0	0	σχέση 4			
r=	-324,693		σχέση 5				r=	324,6934		σχέση 5			
D=	-325,177		σχέση 6				D=	325,1772		σχέση 6			
theta3=	0,226893	13	σχέση 7				theta3=	-0,22689	-13	σχέση 7			
theta3=	0,226893	13	σχέση 8				theta3=	-0,22689	-13	σχέση 8			
theta2=	-0,10033	-5,74857	σχέση 9				theta2=	0,20944	12	σχέση 9			
Inverse kinematic						Inverse kinematic							
ΕΠΑΛΗΘΕΥΣΗ ΑΠΟ ΑΝΑΣΤΡΟΦΟ ΣΕ ΕΥΘΥ						ΕΠΑΛΗΘΕΥΣΗ ΑΠΟ ΑΝΑΣΤΡΟΦΟ ΣΕ ΕΥΘΥ							
Px=	0						Px=	0					
Py=	324,6934						Py=	324,6934					
Pz=	131,7309						Pz=	131,7309					

Πίνακας 4.19 παρ.6



Σχήμα 4.21 παράδειγμα 6

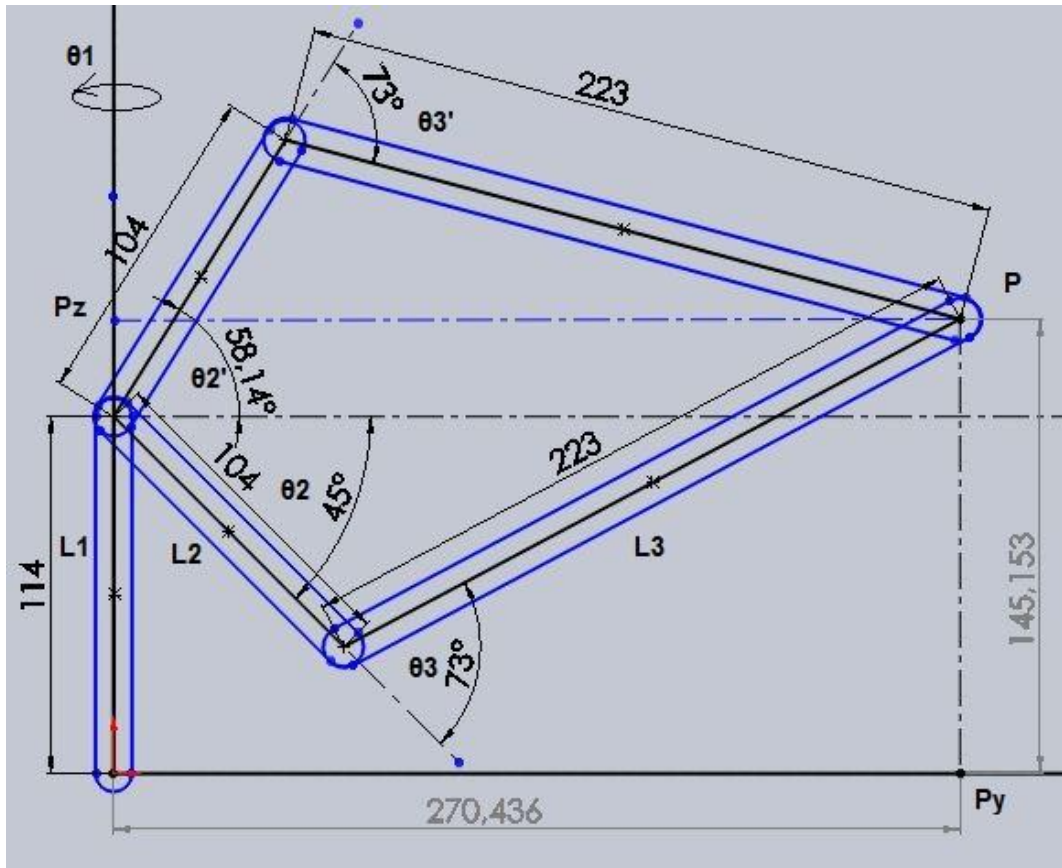
Παράδειγμα 7) Για τη θέση του ευθέως κινηματικού $\theta_1=0.0^\circ$, $\theta_2=-45^\circ$, $\theta_3=73^\circ$ και τις ακόλουθες μετατοπίσεις $P_x=0.0\text{mm}$, $P_y=270.43\text{mm}$, $P_z=145.15\text{mm}$

Corners	Deg	Rad		
theta1=	0	0		
theta2=	-45	-0,7854		
theta3=	73	1,27409		
			SOLUTION	A/A
			Px=	0
			Py=	270,4364
			Pz=	145,1531
				σχέση1
				σχέση2
				σχέση3
Links	mm			
L1=	114			
L2=	104			
L3=	223			
Forward kinematic				

Πίνακας 4.20 παρ.7

Elbow Down				Rad	Deg	A/A	Elbow Up				
	theta1=	0	0			σχέση 4		theta1=	0	0	σχέση 4
	r=	-270,436				σχέση 5		r=	270,4364		σχέση 5
	D=	-272,225				σχέση 6		D=	272,2249		σχέση 6
	theta3=	1,27409	73			σχέση 7		theta3=	-1,27409	-73	σχέση 7
	theta3=	1,27409	73			σχέση 8		theta3=	-1,27409	-73	σχέση 8
	theta2=	1,014778	58,1425			σχέση 9		theta2=	-0,7854	-45	σχέση 9
Inverse kinematic						Inverse kinematic					

Πίνακας 4.21 παρ.7



Σχήμα 4.22 παράδειγμα 7

Παράδειγμα 8) Για τη θέση του ευθέως κινηματικού $\theta_1=0.0^\circ$, $\theta_2=90^\circ$, $\theta_3=-75^\circ$ και τις ακόλουθες μετατοπίσεις $P_x=0.0\text{mm}$, $P_y=215.40\text{mm}$, $P_z=275.71\text{mm}$

Corners	Deg	Rad
theta1=	0	0
theta2=	90	1,570796
theta3=	-75	-1,309

Links	mm
L1=	114
L2=	104
L3=	223

SOLUTION		A/A
Px=	0	σχέση1
Pγ=	215,4015	σχέση2
Pz=	275,7166	σχέση3

Forward kinematic

Πίνακας 4.22 παρ.8

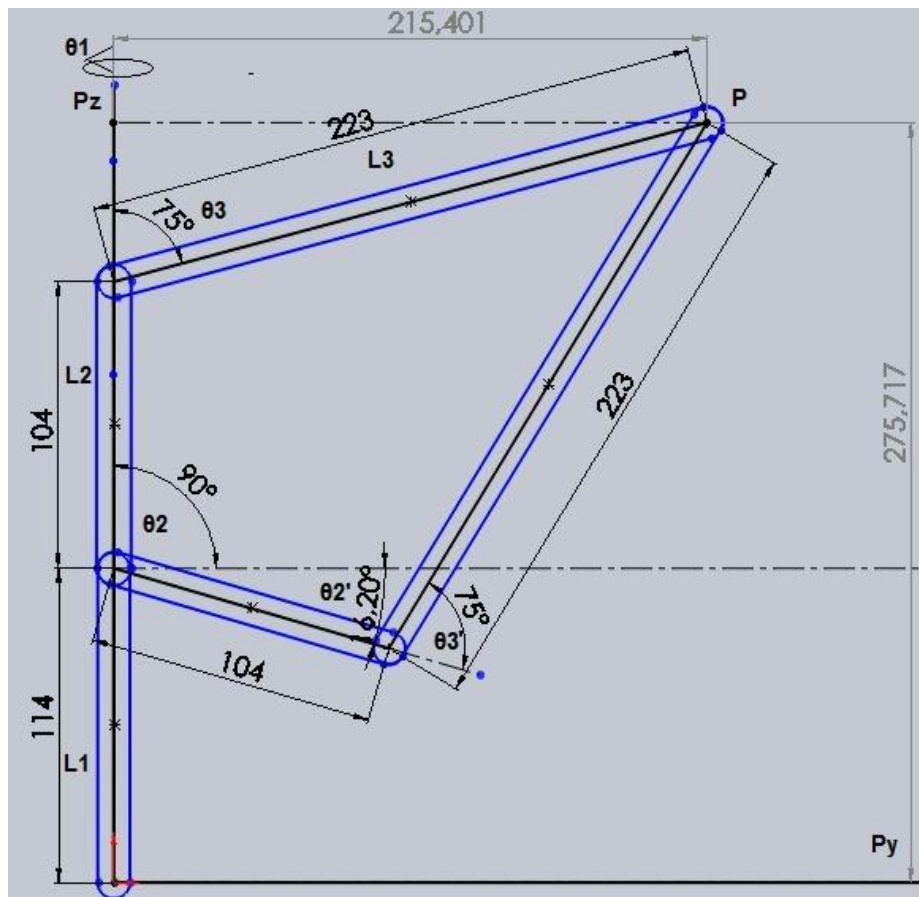
Elbow Down				Rad	Deg	A/A	Elbow Up			
theta1=	0	0	σχέση 4	theta1=	0	0	σχέση 4			
r=	-215,401		σχέση 5	r=	215,4015		σχέση 5			
D=	-269,351		σχέση 6	D=	269,3512		σχέση 6			
theta3=	1,308997	75	σχέση 7	theta3=	-1,309	-75	σχέση 7			
theta3=	1,308997	75	σχέση 8	theta3=	-1,309	-75	σχέση 8			
theta2=	-0,28281	-16,2039	σχέση 9	theta2=	1,570796	90	σχέση 9			

Inverse kinematic				Inverse kinematic			
-------------------	--	--	--	-------------------	--	--	--

ΕΠΛΗΘΕΥΣΗ ΑΠΟ ΑΝΑΣΤΡΟΦΟ ΣΕ ΕΥΘΥ			
Px=	0		
Pγ=	215,4015		
Pz=	275,7166		

ΕΠΛΗΘΕΥΣΗ ΑΠΟ ΑΝΑΣΤΡΟΦΟ ΣΕ ΕΥΘΥ			
Px=	0		
Pγ=	215,4015		
Pz=	275,7166		

Πίνακας Πίνακας 4.23 παρ.8



Σχήμα 4.23 παράδειγμα 8

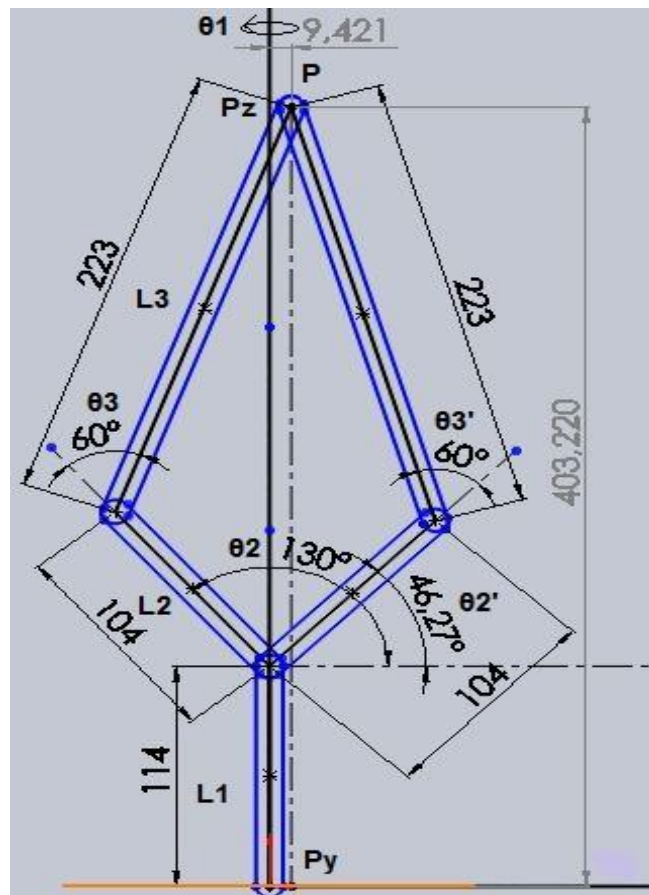
Παράδειγμα 9) Για τη θέση του ευθέως κινηματικού $\theta_1=0.0^\circ$, $\theta_2=130^\circ$, $\theta_3=-60^\circ$ και τις ακόλουθες μετατοπίσεις **$P_x=0.0\text{mm}$, $P_y=9.42\text{mm}$, $P_z=403.22\text{mm}$**

Corners	Deg	Rad		
theta1=	0	0		
theta2=	130	2,268928		
theta3=	-60	-1,0472		
			SOLUTION	A/A
			Px=	0 σχέση1
			Py=	9,420581 σχέση2
			Pz=	403,2201 σχέση3
Links		mm		
L1=	114			
L2=	104			
L3=	223			
			Forward kinematic	

Πίνακας 4.24 παρ.9

Elbow Down		Rad	Deg	A/A	Elbow Up		Rad	Deg	A/A
theta1=	0	0	σχέση 4		theta1=	0	0	σχέση 4	
r=	-9,42058		σχέση 5		r=	9,420581		σχέση 5	
D=	-289,373		σχέση 6		D=	289,3735		σχέση 6	
theta3=	1,047198	60	σχέση 7		theta3=	-1,0472	-60	σχέση 7	
theta3=	1,047198	60	σχέση 8		theta3=	-1,0472	-60	σχέση 8	
theta2=	0,807543	46,2688	σχέση 9		theta2=	2,268928	130	σχέση 9	
Inverse kinematic					Inverse kinematic				
ΕΠΛΗΘΕΥΣΗ ΑΠΟ ΑΝΑΣΤΡΟΦΟ ΣΕ ΕΥΘΥ					ΕΠΛΗΘΕΥΣΗ ΑΠΟ ΑΝΑΣΤΡΟΦΟ ΣΕ ΕΥΘΥ				
Px=	0				Px=	0			
Py=	9,420581				Py=	9,420581			
Pz=	403,2201				Pz=	403,2201			

Πίνακας 4.25 παρ.9



Σχήμα 4.24 παράδειγμα 9

Παράδειγμα 10) Για τη θέση του ευθέως κινηματικού $\theta_1=23.0^\circ$, $\theta_2=130^\circ$, $\theta_3=-60^\circ$ και τις ακόλουθες μετατοπίσεις $P_x=3.68\text{mm}$, $P_y=8.67\text{mm}$, $P_z=403.22\text{mm}$ Παρατηρούμε ότι με την στροφή της γωνίας θ_1 κατά 23 μοίρες η μετατόπιση στον άξονα z παραμένει σταθερή, ενώ έχουμε μεταβολή της μετατόπισης στον άξονα y και φυσικά αποκτά μετατόπιση και στον άξονα x . (βλέπε παρακάτω πίνακες, πίνακα 4.26 και πίνακα 4.27)

A	B	C	D	E	F	G
Corners	Deg	Rad				
theta1=	23	0,401426				
theta2=	130	2,268928				
theta3=	-60	-1,0472				
Links		mm				
L1=	114					
L2=	104					
L3=	223					
				SOLUTION		A/A
				Px=	3,680914	σχέση1
				Py=	8,67169	σχέση2
				Pz=	403,2201	σχέση3
				Forward kinematic		

Πίνακας 4.26 παρ.10

Elbow Down	Rad	Deg	A/A	Elbow Up	Rad	Deg	A/A
theta1=	0,401426	23	σχέση 4	theta1=	0,401426	23	σχέση 4
r=	-9,42058		σχέση 5	r=	9,420581		σχέση 5
D=	-289,373		σχέση 6	D=	289,3735		σχέση 6
theta3=	1,047198	60	σχέση 7	theta3=	-1,0472	-60	σχέση 7
theta3=	1,047198	60	σχέση 8	theta3=	-1,0472	-60	σχέση 8
theta2=	0,807543	46,2688	σχέση 9	theta2=	2,268928	130	σχέση 9
Inverse kinematic				Inverse kinematic			
ΕΠΛΗΘΕΥΣΗ ΑΠΟ ΑΝΑΣΤΡΟΦΟ ΣΕ ΕΥΘΥ				ΕΠΛΗΘΕΥΣΗ ΑΠΟ ΑΝΑΣΤΡΟΦΟ ΣΕ ΕΥΘΥ			
Px=	3,680914			Px=	3,680914		
Py=	8,67169			Py=	8,67169		
Pz=	403,2201			Pz=	403,2201		

Πίνακας 4.27 παρ.10

Κεφάλαιο 5

5.1 Εισαγωγή στα Ηλεκτρονικά μέρη , τον προγραμματισμό μέσω Wiring , το προγραμματιστικό περιβάλλον Arduino IDE και την πλακέτα Arduino

Η **ηλεκτρονική** είναι ένας κλάδος της επιστήμης της φυσικής, όσον αφορά τη θεωρητική μελέτη, που ασχολείται με τη σχεδίαση και κατασκευή πρακτικών κυκλωμάτων και συσκευών που λειτουργούν με τον έλεγχο ροής ηλεκτρονίων και άλλων φορέων ηλεκτρικής αγωγιμότητας, χρησιμοποιώντας ενεργά εξαρτήματα όπως οι ηλεκτρονικές λυχνίες και οι ημιαγωγοί (τρανζίστορ, δίοδοι, ολοκληρωμένα κυκλώματα, κτλ), υποστηριζόμενα και από παθητικά εξαρτήματα. Το κύριο γνωστικό αντικείμενο είναι της επιστήμης ηλεκτρονικού μηχανικού, όσον αφορά την πρακτική εφαρμογή για την επίλυση τεχνολογικών προβλημάτων. Οι κύριες χρήσεις των ηλεκτρονικών κυκλωμάτων είναι ο έλεγχος συστημάτων, η επεξεργασία και διανομή πληροφοριών, η δημιουργία και ο έλεγχος ηλεκτρομαγνητικών κυμάτων και παλμών και η μετατροπή και διανομή ηλεκτρικής ενέργειας. Όλες αυτές οι χρήσεις περιέχουν την δημιουργία και ανίχνευση ηλεκτρομαγνητικών πεδίων και ηλεκτρικού ρεύματος. Τα περισσότερα ηλεκτρονικά συστήματα εμπίπτουν σε δύο μεγάλες κατηγορίες, συστήματα ελέγχου και αυτοματισμού και συστήματα επικοινωνιών.

Προγραμματισμός υπολογιστών (αγγλ.: «computer programming») καλείται το σύνολο των διαδικασιών σύνταξης ενός υπολογιστικού προγράμματος, συνήθως ως υλοποίηση κάποιων αλγορίθμων ύστερα από προσεκτική σχεδίαση, για την αυτοματοποιημένη εκτέλεση εργασιών ή επίλυση κάποιου υπολογιστικού προβλήματος από έναν υπολογιστή. Ο προγραμματισμός περιλαμβάνει επίσης τον έλεγχο του προγράμματος για την επαλήθευση της ακρίβειας και της ορθότητάς του (αποσφαλμάτωση), και την προπαρασκευή των οδηγιών με τις οποίες ένας υπολογιστής θα εκτελέσει τις εντολές που καθορίζονται στις προδιαγραφές του προγράμματος. Θεμελιώδη ρόλο στον υπολογιστικό προγραμματισμό διαδραματίζουν οι χιλιάδες διαφορετικές γλώσσες προγραμματισμού, δηλαδή οι προτυποποιημένες τυπικές γλώσσες απαραίτητες για τη σύνθεση ενός προγράμματος.

Ο πηγαίος κώδικας του προγράμματος αποτελείται από τις εντολές που έχει γράψει ο προγραμματιστής χρησιμοποιώντας μία γλώσσα προγραμματισμού. Το πηγαίο πρόγραμμα για να εκτελεστεί από την Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ) του Η/Υ - τον επεξεργαστή, πρέπει στη συνέχεια να μεταφραστεί σε γλώσσα μηχανής από εξειδικευμένο λογισμικό, τους μεταγλωττιστές (compilers) και τους διερμηνείς (interpreters).

Μικροελεγκτές

Ο μικροελεγκτής (αγγλικά, microcontroller) είναι ένας τύπος επεξεργαστή, ουσιαστικά μια παραλλαγή μικροεπεξεργαστή, ο οποίος μπορεί να λειτουργήσει με ελάχιστα εξωτερικά εξαρτήματα, λόγω των πολλών ενσωματωμένων υποσυστημάτων που διαθέτει. Χρησιμοποιείται ευρύτατα σε όλα τα ενσωματωμένα συστήματα (embedded systems) χαμηλού ελέγχου και μεσαίου κόστους, όπως αυτά που χρησιμοποιούνται σε αυτοματισμούς, ηλεκτρονικά καταναλωτικά προϊόντα (από ψηφιακές φωτογραφικές μηχανές έως και παιχνίδια), ηλεκτρικές συσκευές και κάθε είδους αυτοκινούμενα τροχοφόρα οχήματα.

Αναλυτικά, τα πλεονεκτήματα των μικροελεγκτών είναι:

1. **Αυτονομία**, μέσω της ενσωμάτωσης σύνθετων περιφερειακών υποσυστημάτων όπως μνήμες και θύρες επικοινωνίας. Έτσι πολλοί μικροελεγκτές δεν χρειάζονται κανένα άλλο ολοκληρωμένο κύκλωμα για να λειτουργήσουν.
2. Η **ενσωμάτωση περιφερειακών** σημαίνει ευκολότερη υλοποίηση εφαρμογών λόγω των απλούστερων διασυνδέσεων. Επίσης, οδηγεί σε χαμηλότερη κατανάλωση ισχύος, μεγιστοποιώντας τη φορητότητα και ελαχιστοποιεί το κόστος της συσκευής στην οποία ενσωματώνεται ο μικροελεγκτής.
3. **Χαμηλό κόστος**.
4. **Μεγαλύτερη αξιοπιστία**, και πάλι λόγω των λιγότερων διασυνδέσεων.

5. **Μειωμένες εκπομπές ηλεκτρομαγνητικών παρεμβολών** και μειωμένη ευαισθησία σε αντίστοιχες παρεμβολές από άλλες ηλεκτρικές και ηλεκτρονικές συσκευές. Το πλεονέκτημα αυτό προκύπτει από το μικρότερο αριθμό και μήκος εξωτερικών διασυνδέσεων καθώς και τις χαμηλότερες ταχύτητες λειτουργίας.
6. **Περισσότεροι διαθέσιμοι ακροδέκτες** για ψηφιακές εισόδους-εξόδους (για δεδομένο μέγεθος ολοκληρωμένου κυκλώματος), λόγω της μη δέσμευσής τους για τη σύνδεση εξωτερικών περιφερειακών.
7. **Μικρό μέγεθος** συνολικού υπολογιστικού συστήματος.
8. Η βασική αρχιτεκτονική των μικροελεγκτών δεν διαφέρει από αυτή των κοινών μικροεπεξεργαστών, αν και στους πρώτους απαντάται συχνά η αρχιτεκτονική μνήμης τύπου Harvard, η οποία χρησιμοποιεί διαφορετικές αρτηρίες σύνδεσης της μνήμης προγράμματος και της μνήμης δεδομένων (πχ οι σειρές AVR από την Atmel και PIC από την Microchip). Στους κοινούς μικροεπεξεργαστές συνηθίζεται η ενιαία διάταξη μνήμης τύπου φον Νόιμαν

5.2 Αναφορά στην πλακέτα Arduino

Η Arduino είναι μια Ιταλική εταιρία που κατασκευάζει πλακέτες ενσωματωμένων συστημάτων ανοικτής αρχιτεκτονικής και ανοικτού κώδικα από το 2003, η πιο δημοφιλής πλακέτα είναι η **Arduino Uno R3**. Η οποία βασίζεται στον μικροελεγκτή **Atmega328P** της εταιρίας Atmel και ανήκει στην μεγάλη οικογένεια μικροελεγκτών megaAVR της εταιρίας.



Εικόνα 5.1 Διάφοροι τύποι πλακέτας Arduino

Ο Atmega328P είναι ένας μικροελεγκτής 8-bit αρχιτεκτονικής AVR RISC , με απόδοση 20 MIPS στα 20MHz. Διαθέτει SRAM 2 KB για μεταβλητές , Flash EEPROM 32KB για προγράμματα και έξτρα EEPROM 1 KB για μόνιμη αποθήκευση μεταβλητών. Έχει 23 ψηφιακές γραμμές I/O γενικού σκοπού και 32 καταχωρητές γενικού σκοπού. Έχει 3 ευέλικτους χρονιστές , εσωτερικά και εξωτερικά interrupt , σειριακά interfaces (USART,I2C,SPI), 10 bit μετατροπέα A/D 6 καναλιών για ανάγνωση αναλογικών σημάτων και τάση λειτουργίας στα 5V. Η πλακέτα παρέχει 14 ψηφιακά I/O pins (με αριθμούς 0-13) από τα οποία τα 6 έχουν δυνατότητα εξόδου σήματος PWM(Pulse Width Modulation) και είναι συνήθως τα 3, 5, 6, 9, 10, 11 και συνήθως σημειώνονται στην πλακέτα με περισπωμένη (π.χ ~5).

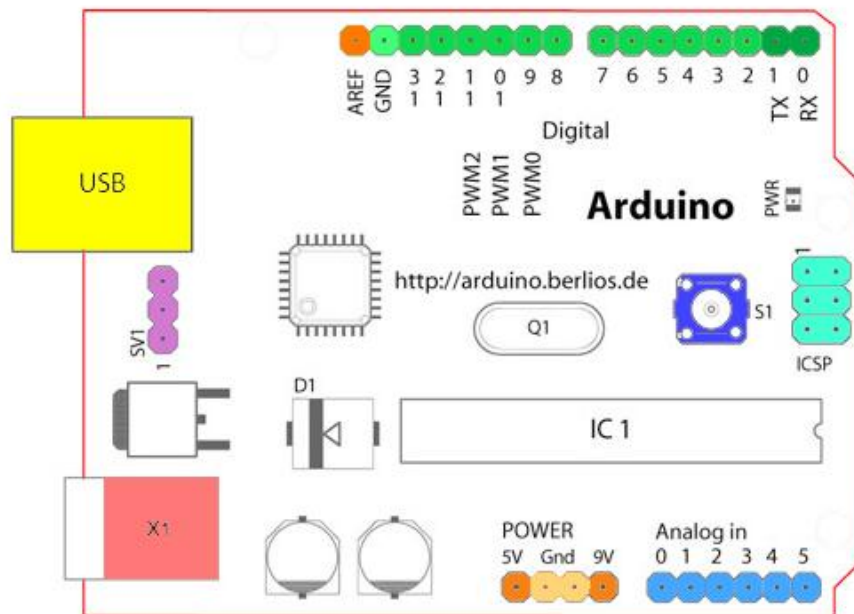
Επίσης παρέχει 6 αναλογικές εισόδους με ονόματα A0-A5 με 10-bit (1024 επίπεδα) μετατροπή αναλογικού σήματος σε ψηφιακό . Διαθέτει USB θύρα για σύνδεση με host PC ως εικονική σειριακή θύρα (virtual COM port).

Το Arduino είναι ένας μικροελεγκτής μονής πλακέτας, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Pure Data, SuperCollider. Οι περισσότερες εκδόσεις του Arduino μπορούν να αγοραστούν προ-συναρμολογημένες· το διάγραμμα και πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους. Το πρόγραμμα Arduino έλαβε τιμητική μνεία στην κατηγορία Digital Communities στο Prix Ars Electronica το 2006 .



Εικόνα 5.2 Η πλακέτα Arduino Uno

Ο Atmega328P διαθέτει ήδη πρόγραμμα bootloader μεγέθους 2KB που διευκολύνει την επικοινωνία με το host PC και το Arduino IDE. Η πλακέτα τροφοδοτείται με τάση 6-20V(συνιστώμενη 7-12V). Όσο για την ανάπτυξη εφαρμογών για τον Arduino γίνεται εύκολα με την Windows εφαρμογή “Arduino IDE” η οποία αποτελεί ελεύθερο λογισμικό. Το Arduino IDE επιτρέπει την ανάπτυξη εφαρμογών (sketch) σε μια C-like γλώσσα προγραμματισμού που βασίζεται στην πλατφόρμα “Wiring”. Στο σχήμα 5.1 και εικόνα 5.4 γίνεται αναλυτική περιγραφή όλων των τμημάτων που απαρτίζουν την πλακέτα Arduino Uno, καθώς και της λειτουργίας τους.



Starting clockwise from the top center:

Σχήμα 5.1 Όλα τα μέρη που απαρτίζουν την πλακέτα Arduino Uno

Παρακάτω θα γίνει αναλυτική περιγραφή του κάθε μέρους με βάση το χρώμα και την αρίθμηση του σχήματος 5.1

- Analog Reference pin (orange)
- Digital Ground (light green)
- Digital Pins 2-13 (green)
- Digital Pins 0-1/Serial In/Out - TX/RX (dark green) - *These pins cannot be used for digital i/o (digitalRead and digitalWrite) if you are also using serial communication (e.g. Serial.begin).*
- Reset Button - S1 (dark blue)
- In-circuit Serial Programmer (blue-green)
- Analog In Pins 0-5 (light blue)
- Power and Ground Pins (power: orange, grounds: light orange)
- External Power Supply In (9-12VDC) - X1 (pink)
- Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SV1 (purple)
- USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow)

Εικ 5.3 Περιγραφή των τμημάτων της πλακέτας Arduino Uno





Εικόνα 5.4 Περιγραφή των μερών της πλακέτας Arduino Uno

5.3 Ιστορικό

Το 2005 ένα σχέδιο κίνησε προκειμένου να φτιαχτεί μία συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν πιο φθηνή από άλλα πρωτότυπα συστήματα διαθέσιμα εκείνη την περίοδο. Οι ιδρυτές Massimo Banzi και David Cueartielles ονόμασαν το σχέδιο από τον Arduino της Ivrea και ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ιβρέα, κωμόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας - την ίδια περιοχή στην οποία στεγαζόταν η εταιρία υπολογιστών Olivetti. Το σχέδιο Arduino είναι μία διακλάδωση της πλατφόρμας Wiring για λογισμικό ανοικτού κώδικα και προγραμματίζεται χρησιμοποιώντας μια γλώσσα βασισμένη στο Wiring (σύνταξη και βιβλιοθήκες), παρόμοια με την C++ με απλοποιήσεις και αλλαγές, καθώς και ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE)

5.4 Εκδόσεις

- Τον Σεπτέμβριο του 2006 ανακοινώθηκε το **Arduino Mini**
- Τον Οκτώβρη του 2008 ανακοινώθηκε το **Arduino Duemilanove**. Αρχικά βασίστηκε στο Atmel Atmega168, αλλά μετά στάλθηκε με το ATmega328.
- Τον Μάρτιο του 2009 ανακοινώθηκε το **Arduino Mega**. Είναι βασισμένο στο Atmel ATmega1280
- Από τον Μάιο του 2011 πάνω από 300.000 Arduino ήταν σε χρήση σε όλο τον κόσμο
- Τον Ιούλιο του 2012 ανακοινώθηκε το **Arduino Leonardo**. Είναι βασισμένο στο Atmel ATmega32u4
- Τον Οκτώβριο του 2012 ανακοινώθηκε το **Arduino Due**. Είναι βασισμένο στο Atmel SAM3X8E, που είχε πυρήνα **ARM Cortex-M3**
- Τον Νοέμβριο του 2012 ανακοινώθηκε το **Arduino Micro**. Είναι βασισμένο στο Atmel ATmega32u4
- Τον Μάιο του 2013 ανακοινώθηκε το **Arduino Robot**. Είναι βασισμένο στο Atmel ATmega32u4 και ήταν το πρώτο επίσημο Arduino με ρόδες
- Τον Μάιο του 2013 ανακοινώθηκε το **Arduino Yun**. Είναι βασισμένο στο ATmega32u4 και στο Atheros AR9331 και ήταν το πρώτο προϊόν wifi που συνδυάζει το Arduino με το Linux

5.5 Ανάπτυξη

Η κύρια ομάδα ανάπτυξης του Arduino αποτελείται από τους: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis και Nicholas Zambetti. Ο Massimo Banzi έδωσε συνέντευξη στις 21 Μαρτίου του 2009 στο επεισόδιο 61 του FLOSS Weekly στο κανάλι TWiT.tv, στο οποίο συζήτησε την ιστορία και τους στόχους του προγράμματος Arduino. Επίσης, έδωσε μια ομιλία στο TEDGlobal 2012 Conference, όπου περιέγραψε διάφορες χρήσεις των πλακετών Arduino σε όλο τον κόσμο.

Το Arduino είναι υλισμικό ανοιχτού λογισμικού: τα σχέδια αναφοράς του υλισμικού του Arduino διανέμονται υπό την Creative Commons Attribution Share-Alike 2.5 άδεια και είναι διαθέσιμα στην ιστοσελίδα του Arduino. Ο σχεδιασμός και η παραγωγή αρχείων για κάποιες εκδόσεις του υλισμικού Arduino είναι επίσης διαθέσιμοι. Ο πηγαίος κώδικας για το IDE είναι διαθέσιμος και διανέμεται υπό την GNU General Public License, έκδοση 2. Παρά το γεγονός ότι το υλισμικό και τα σχέδια του λογισμικού είναι διαθέσιμα ελεύθερα υπό άδειες πνευματικών δικαιωμάτων,

οι προγραμματιστές έχουν ζητήσει η ονομασία "Arduino" είναι αποκλειστική για το επίσημο προϊόν και δεν πρέπει να χρησιμοποιείται για τις εργασίες χωρίς άδεια. Το επίσημο έγγραφο πολιτικής σχετικά με τη χρήση του ονόματος Arduino τονίζει ότι το πρόγραμμα είναι ανοιχτό στη συνεργασία με άλλους στο επίσημο προϊόν. Αρκετά προϊόντα συμβατά με Arduino που κυκλοφορούν στο εμπόριο έχουν αποφύγει το όνομα "Arduino" χρησιμοποιώντας την κατάληξη "-duino" με παραλλαγές στο όνομα.

5.6 Το Περιβάλλον Arduino IDE

Το **Arduino IDE** επιτρέπει την σύνδεση με το Arduino μέσω USB θύρας , καθώς κατά την εγκατάσταση του παρέχει και τους απαραίτητους drivers (οδηγούς). Επιτρέπει τη δημιουργία νέου , το φόρτωμα , το σώσιμο προγράμματος , καθώς και την μεταγλώττιση σε κώδικα μηχανής και την μεταφόρτωση του προγράμματος στην Flash EEPROM του Arduino.



```

3servo-contr.nopot | Arduino 1.8.4
Αρχείο Επεξεργασία Σχέδιο Εργαλεία Βοήθεια
3servo-contr.nopot
#include <Servo.h>

Servo servoA;
Servo servoB;
Servo servoC;

void setup() {
  servoA.attach(10);
  servoB.attach(6);
  servoC.attach(9);
}

void loop() {

  for(int j = 20; j<130; j+=1){
    forward(j);
    delay(200);
  }

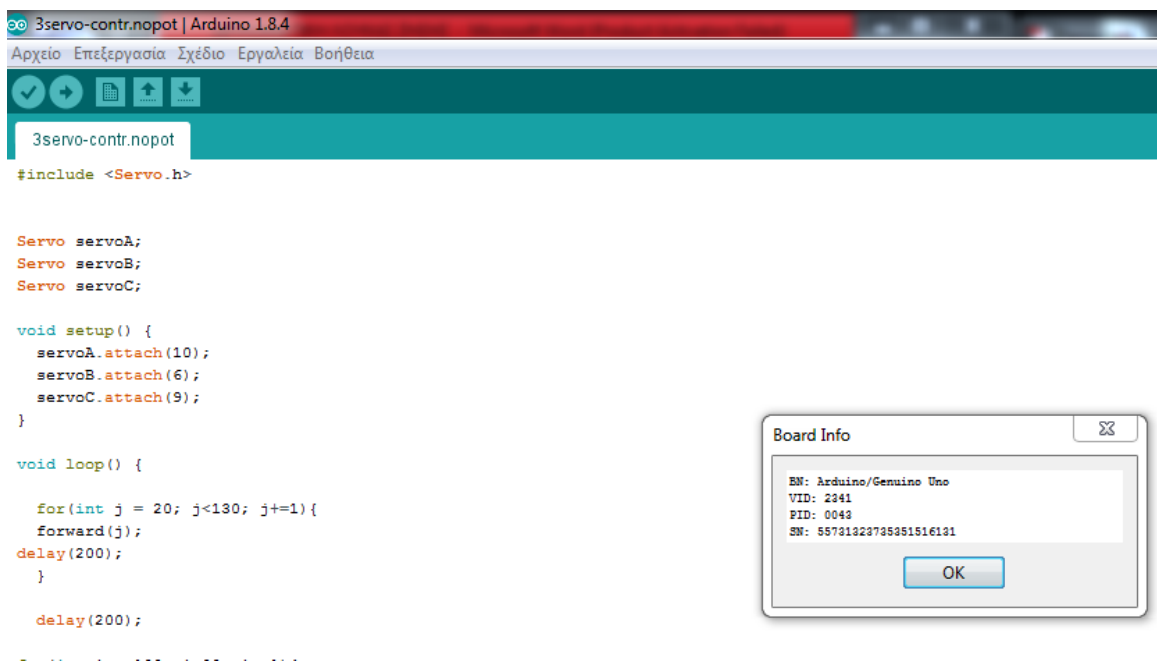
  delay(200);

```

Εικόνα 5.5 Περιβάλλον Arduino IDE στα Windows

Από την επιλογή **Εργαλεία(Tools) -> Board** Ρυθμίζεται το είδος της πλακέτας που θα συνδέσουμε στο Arduino IDE. Έπειτα **Tools->Port** Επιλέγουμε την (εικονική μέσω USB) σειριακή πόρτα που έχουμε συνδέσει στον Arduino. Επιλογή Tools->Get Board Info (εικόνα 5.4) Ανακτά από την πλακέτα και εμφανίζει στην οθόνη στοιχεία για την πλακέτα όπως η ονομασία της πλακέτας(π.χ Arduino Uno) τα στοιχεία της συσκευής(Vendor ID – VID ,Product ID-PID) και τον αριθμό σειράς (Serial Number ή SN). Η εμφάνιση των στοιχείων της πλακέτας επιβεβαιώνει την επιτυχή σύνδεση με αυτήν.

+



```

3servo-contr.nopot | Arduino 1.8.4
Αρχείο Επεξεργασία Σχέδιο Εργαλεία Βοήθεια
3servo-contr.nopot
#include <Servo.h>

Servo servoA;
Servo servoB;
Servo servoC;

void setup() {
  servoA.attach(10);
  servoB.attach(6);
  servoC.attach(9);
}

void loop() {

  for(int j = 20; j<130; j+=1){
    forward(j);
    delay(200);
  }

  delay(200);
}

```

Board Info	
EN:	Arduino/Genuino Uno
VID:	2341
PID:	0043
SN:	55721223735351516131

Εικόνα 5.6 Επιβεβαίωση σύνδεσης και εμφάνιση των στοιχείων της πλακέτας

Το πρόγραμμα του Arduino μπορεί να χωριστεί σε 3 τμήματα που είναι :

ο **Δομή (Structure)**

Περιέχει τον κώδικα του κυρίως προγράμματος της εφαρμογής

ο **Τιμές** (μεταβλητές και σταθερές) **Values**(Variables & Constants)

Αποτελούν το τμήμα δεδομένων της εφαρμογής

ο **Συναρτήσεις**(Functions)

Περιέχουν επαναχρησιμοποιήσιμο κώδικα και συμβάλλουν στο να γίνει ο κώδικας περισσότερο δομημένος.

Τιμές (Values)

```
int PushButton=2;  
boolean running=false;
```

Δομή(Structure)

```
void setup() {.....}  
void loop() {.....}
```

Συναρτήσεις(functions) **long Read_Distance()**

```
{...}
```

Η Δομή αποτελείται από δύο συναρτήσεις με σταθερά ονόματα, τις setup() και loop().

Η συνάστηση setup() συντάσσεται ως εξής:

void setup()

```
{
```

```
//κώδικας που εισάγει ο προγραμματιστής
```

```
}
```

Η συνάρτηση setup() δεν παίρνει ορίσματα και δεν επιστρέφει αποτελέσματα(void). Η setup() καλείται μια φορά, όταν το sketch ξεκινά ή όποτε κάνει επαναφορά (reset) η πλατφόρμα Arduino. Κυρίως, σε αυτήν γίνονται οι αρχικοποιήσεις των μεταβλητών, η ρύθμιση της κατάστασης των ακίδων (pins) και η προετοιμασία των βιβλιοθηκών.

Η συνάρτηση loop () συντάσσεται ως εξής:

Void loop()

{

//κώδικας που εισάγει ο προγραμματιστής

}

Η συνάρτηση void loop() Μετά τη δημιουργία μιας συνάρτησης setup (), η οποία προετοιμάζει και ορίζει τις αρχικές τιμές, η λειτουργία βρόχου () κάνει ακριβώς αυτό που προτείνει το όνομά της και void loop, επιτρέποντας στο πρόγραμμα να αλλάξει και να ανταποκριθεί. Χρησιμοποιείται για να ελεγχθεί ενεργά το Arduino, η συνάρτηση loop() καλείται ξανά και ξανά επιτρέποντας έτσι στο πρόγραμμα να ανταποκριθεί σε εξωτερικά ερεθίσματα. Και οι δυο συναρτήσεις πρέπει να περιλαμβάνονται στο sketch, ακόμα και αν δεν περιέχουν κάτι και να είναι κενές

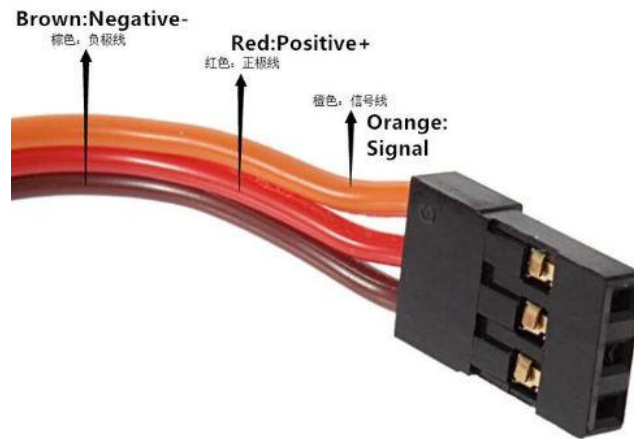
5.7 Έλεγχος servo-κινητήρων με την πλακέτα Arduino Uno

Ο Έλεγχος των 3 servo-κινητήρων σε πρώτη φάση θα γίνεται από την πλάκετα Arduino , στο αρχικό πρόγραμμα γίνεται ανάγνωση των bytes , από τιμές 0-255 κι έπειτα ανάλογα με αυτή την τιμή ο arduino μετατρέπει σε μοίρες τις οποίες θα περιστραφεί ο servo-κινητήρας.

Οι σερβο-μηχανισμοί με έλεγχο θέσης (γωνίας) ή απλά Servo που χρησιμοποιούνται είναι ένα διαδεδομένο είδος ενεργοποιητή (actuator) που περιέχει έναν κινητήρα και έναν αισθητήρα θέσης (γωνίας στροφής) που μαζί αποτελούν ένα σύστημα αυτομάτου ελέγχου κλειστού βρόχου. Έτσι ο σερβομηχανισμός μπορεί να στρέφει την κεφαλή του (Head) συνήθως από -90 έως +90 μοίρες αλλά και να κρατάει την γωνία αυτή υπό φορτίο. Το Servo έχει την δυνατότητα να στρέφει την κεφαλή του (Head) συνήθως 180 μοίρες (από -90 μοίρες έως +90 μοίρες) ενώ με χρήση ντίζας η κίνηση μπορεί να μετατραπεί εύκολα σε γραμμική. Η τυπική τάση τροφοδοσίας του είναι 5V, ενώ ο Servo ελέγχεται μέσω τριών ακροδεκτών – καλωδίων:

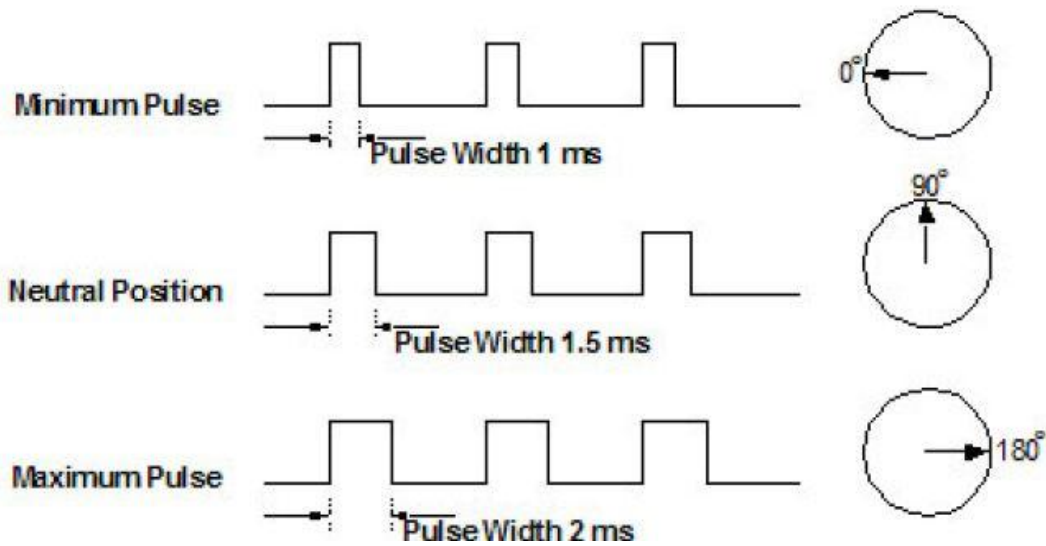
1. Καλώδιο Τροφοδοσίας 5V (**κόκκινο**). Συνδέεται στην τάση τροφοδοσίας 5V.
2. Καλώδιο γείωσης GND (**μαύρο**). Συνδέεται στον αρνητικό πόλο της τροφοδοσίας.
3. Καλώδιο σήματος PIN (άσπρο ή **κίτρινο**).

(βλέπε παρακάτω εικόνα 5.5)



Εικόνα 5.7

Σε αυτό παρέχεται σήμα PWM όπου η διάρκεια του παλμού καθορίζει και την γωνία στροφής της κεφαλής του Servo. Η διάρκεια του παλμού μπορεί να κυμαίνεται από 1 ms έως 2ms ενώ όταν δοθεί παλμός διάρκειας 1.5 ms έρχεται στην ουδέτερη θέση (neutral position). Η συχνότητα του παλμού τυπικά είναι 50Hz (περίοδος 20ms).



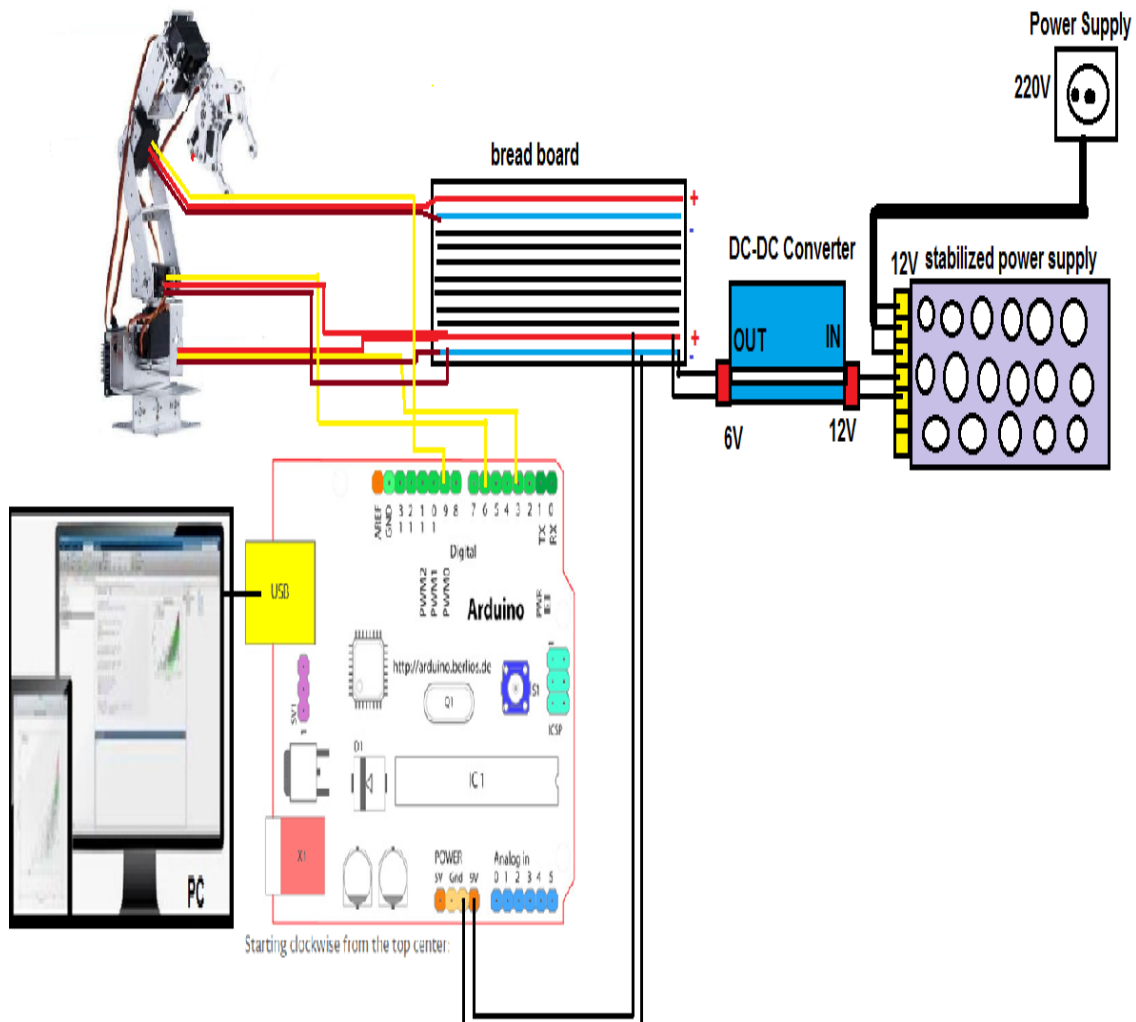
Εικόνα 5.8

Οι servo-κινητήρες απαιτούν σημαντικό ρεύμα, ανάλογα και με τα φορτία που έχουν να μεταφέρουν, γι αυτό το λόγο στην παρούσα διπλωματική εργασία χρησιμοποιούμε σταθερή εξωτερική τροφοδοσία. Ο συγκεκριμένο τύπος σερβοκινητήρα MG-996R υψηλής ροπής απαιτεί ρεύμα από 4.8-7.2V (τα τεχνικά χαρακτηριστικά αναλύθηκαν στο προηγούμενο κεφάλαιο 3).



Εικόνα 5.9 servo-motors τύπου MG996R

Η πιο διαδεδομένη βιβλιοθήκη για έλεγχο των σερβο-κινητήρων είναι η <servo.h>. Στην εργασία αυτή θα χρησιμοποιηθεί η βιβλιοθήκη <VarSpeedServo.h> .



Εικόνα 5.10 Συνδεσμολογία των διαφόρων μερών της κατασκευής και τροφοδοσία

```

Dummy | Arduino 1.8.4
Αρχείο Επεξεργασία Σχέδιο Εργαλεία Βοήθεια

Dummy
#include <VarSpeedServo.h>

//#include <Servo.h> // Εισαγωγή της Βιβλιοθήκης για έλεγχο τω Ser
#include <stdlib.h>
#include<string.h>
#include<math.h>

VarSpeedServo servo1;
VarSpeedServo servo2;
VarSpeedServo servo3;

const int servo1Pin = 3;
const int servo2Pin = 6;
const int servo3Pin = 9;

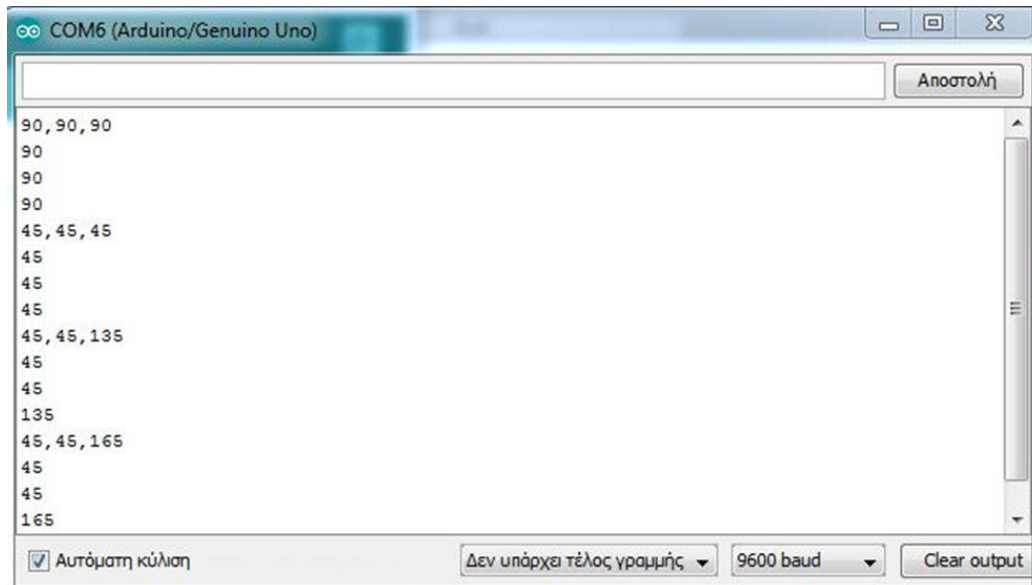
int recValue; // Ορισμός ακέραιας μεταβλητής η οποία θα λαμβάνει τι
String incoming_byte;
int v1,v2,v3 = 0; // Τιμές που θα δώσει ο χρήστης για τις γωνίες
boolean enable = LOW; // Αν είναι HIGH τότε ο χρήστης έχει δώσει c

Ολοκλήρωση ανεβάσματος.
Το σχέδιο χρησιμοποιεί 7918 bytes (24%) του χώρου αποθήκευσης του
Οι καθολικές μεταβλητές χρησιμοποιούν 339 bytes (16%) δυναμικής μν
64 Arduino/Genuino Uno στο COM8

```

Εικόνα 5.11 Φόρτωση της βιβλιοθήκης <VarSpeedServo.h>

Αρχικά οι ενεργοποιητές θα ελεγχθούν από τη σειριακή θύρα που παρέχει το περιβαλλον IDE του Arduino. Ο χρήστης πληκτρολογώντας 3 επιθυμητές τιμές από 0-180 με κόμμα ανάμεσά τους κι έπειτα enter , μπορεί να παρακολουθήσει μέσω της σειριακής θύρας την μεταβολή της γωνίας του κάθε servo ξεχωριστά . (βλέπε παρακάτω εικόνα 5.)



Εικόνα 5.12 Παρακολούθηση σειριακής

Εάν πληκτρολογήσουμε 45,45,135 κι έπειτα αποστολή θα δούμε το servo1 να μετακινείται 45 μοίρες, το servo2 45 μοίρες και το servo3 κατά 135 μοίρες. Στο σημείο αυτό να επισημάνουμε ότι κάναμε χρήση της βιβλιοθήκης <VarSpeedServo.h> , διότι σε αυτή τη βιβλιοθήκη υπάρχει η δυνατότητα ελέγχου της ταχύτητας των σερβο-κινητήρων κι έτσι δεν καταπονούνται από τις απότομες επιταχύνσεις. Έπειτα το κομμάτι του arduino σταματάει εδώ, όλα τα υπόλοιπα θα γίνουν στο προγραμματιστικό περιβάλλον του Matlab σε υψηλό επίπεδο, θα αποστέλλονται στον arduino έπειτα ο arduino θα εκτελεί τις εντολές χαμηλού επιπέδου μέσω ειδικού πρωτοκόλλου επικοινωνίας, το οποίο έχει δημιουργηθεί για την περίπτωση.

Κεφάλαιο 6

6. Εισαγωγή στο Matlab και επικοινωνία μεταξύ Matlab-Arduino



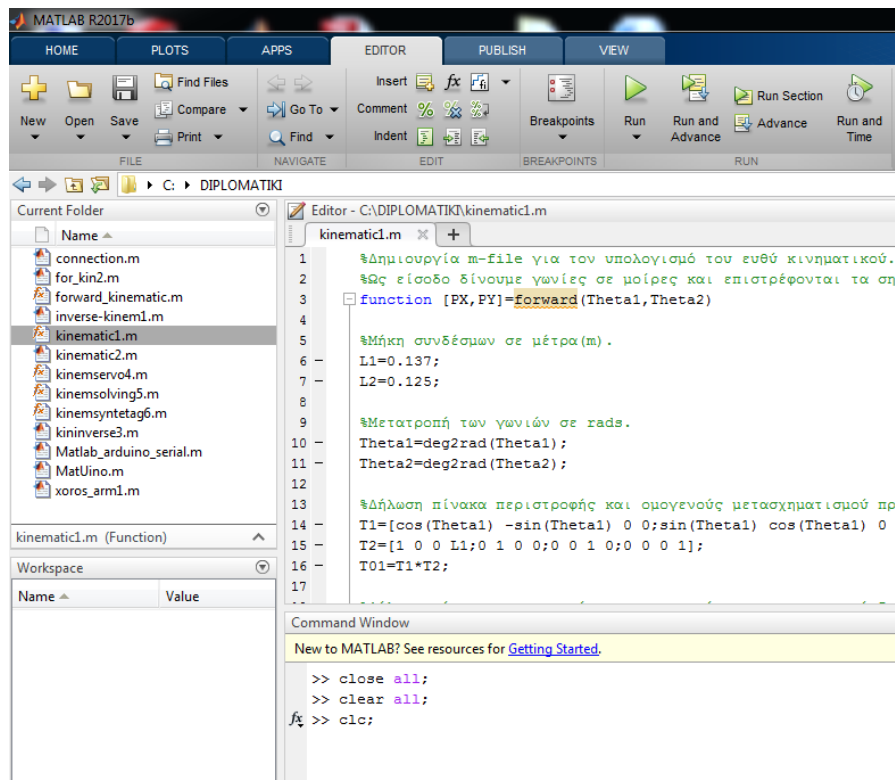
Εικόνα 6.1 Σειριακή επικοινωνία Matlab-Arduino

6.1 Εισαγωγή στο περιβάλλον του Matlab

Το Matlab είναι ένα πρόγραμμα μαθηματικών υπολογισμών που βασίζεται στην διαχείριση πινάκων (matrices). Το όνομα είναι συντομογραφία των λέξεων **MAT**rix-**LAB**oratory. Οι δυνατότητες του περιλαμβάνουν απλές μαθηματικές πράξεις όπως πρόσθεση , αφαίρεση , πολλαπλασιασμό και διαίρεση, προηγμένους μαθηματικούς υπολογισμούς όπως δυνάμεις , ρίζες , τιμές συναρτήσεων (τριγωνομετρικών , εκθετικών , λογαριθμικών κ.α), πράξεις μιγαδικών αριθμών , αποθήκευση και επεξεργασία δεδομένων , παραγωγή και ολοκλήρωση συναρτήσεων, γραφικές παραστάσεις , άλγεβρα πινάκων , συμβολική επίλυση εξισώσεων και πολλά άλλα. Το μεγαλύτερο πλεονέκτημα του Matlab είναι η δυνατότητα επίλυσης σύνθετων αριθμητικών προβλημάτων με μία σειρά από εντολές χωρίς ουσιαστικά να χρειαστεί η ανάπτυξη προγράμματος.

Από την άλλη πλευρά, το Matlab είναι μια φιλική στον χρήστη γλώσσα προγραμματισμού με προηγμένα κι εύχρηστα χαρακτηριστικά που επιτρέπει τον προγραμματισμό πολύπλοκων αλγορίθμων.

Ένα πολύ σημαντικό χαρακτηριστικό του Matlab είναι η διάθεση <<εργαλείων>> για την επίλυση προβλημάτων από συγκεκριμένες περιοχές εφαρμογών. Τα εργαλεία αυτά ονομάζονται Toolboxes και μερικά από αυτά είναι το Control System Toolbox, το Signal Processing Toolbox και το Symbolic Math Toolbox. Η πραγματική και σε βάθος γνωριμία του προγράμματος όμως προϋποθέτει μακρά εργασία, πειραματισμό, πρακτική εξάσκηση και σταδιακή εξερεύνηση των απειριορίστων δυνατοτήτων του. Το Matlab διατίθεται για χρήση σε διαφορετικά περιβάλλοντα (Sun, Apollo, VAXstation, HP workstations, PC and AT compatibles, Apple Macintosh). Το υπολογιστικό περιβάλλον του Matlab λειτουργεί ως εξής: με την ενεργοποίησή του το Matlab εμφανίζει ένα παράθυρο εντολών. Κατά τη διάρκεια της εργασίας το περιβάλλον αποθηκεύει τις εντολές που έχουν δοθεί, όπως και τις μεταβλητές που έχουν δημιουργηθεί και φυσικά μπορεί να τις ανακαλέσει. Οι εντολές και οι μεταβλητές αυτές αποτελούν το χώρο εργασίας (WORKSPACE) του Matlab.



Εικόνα 6.2 Περιβάλλον Matlab.2017b

6.2.1 Σειριακή επικοινωνία Arduino-Matlab

Στο σημείο αυτό είναι απαραίτητη η επικοινωνία του μικροελεγκτή **Arduino** με το προγραμματιστικό περιβάλλον **Matlab** , οπότε θα δημιουργηθεί ειδικό πρωτόκολλο σειριακής επικοινωνίας μεταξύ τους.

Αφού φορτώσουμε κανονικά τον κώδικα στο περιβάλλο IDE του Arduino (βλ. εικόνα 6.3) Έπειτα θα δημιουργήσουμε m.files και κάθε φορά που τρέχουμε τα αρχεία.m στο Matlab θα τρέχει και το πρόγραμμα στο Arduino.

```

Dummy | Arduino 1.8.4
Αρχείο Επεξεργασία Σχέδιο Εργαλεία Βοήθεια
Dummy
#include <VarSpeedServo.h>

// #include <Servo.h> // Εισαγωγή της Βιβλιοθήκης για έλεγχο τω Ser
#include <stdlib.h>
#include <string.h>
#include <math.h>

VarSpeedServo servo1;
VarSpeedServo servo2;
VarSpeedServo servo3;

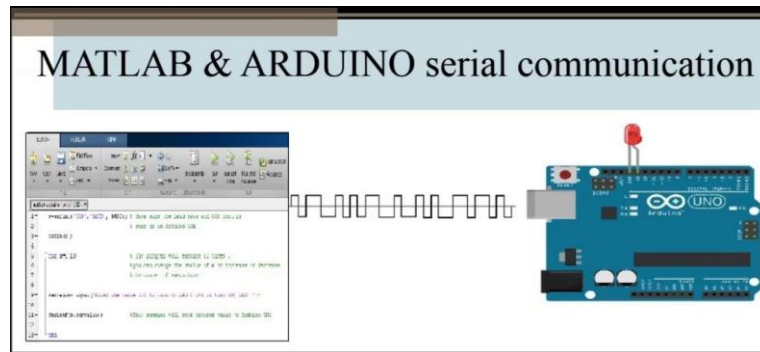
const int servo1Pin = 3;
const int servo2Pin = 6;
const int servo3Pin = 9;

int recValue; // Ορισμός ακέραιας μεταβλητής η οποία θα λαμβάνει τι
String incoming_byte;
int v1,v2,v3 = 0; // Τιμές που θα δώσει ο χρήστης για τις γωνίες
boolean enable = LOW; // Αν είναι HIGH τότε ο χρήστης έχει δώσει c

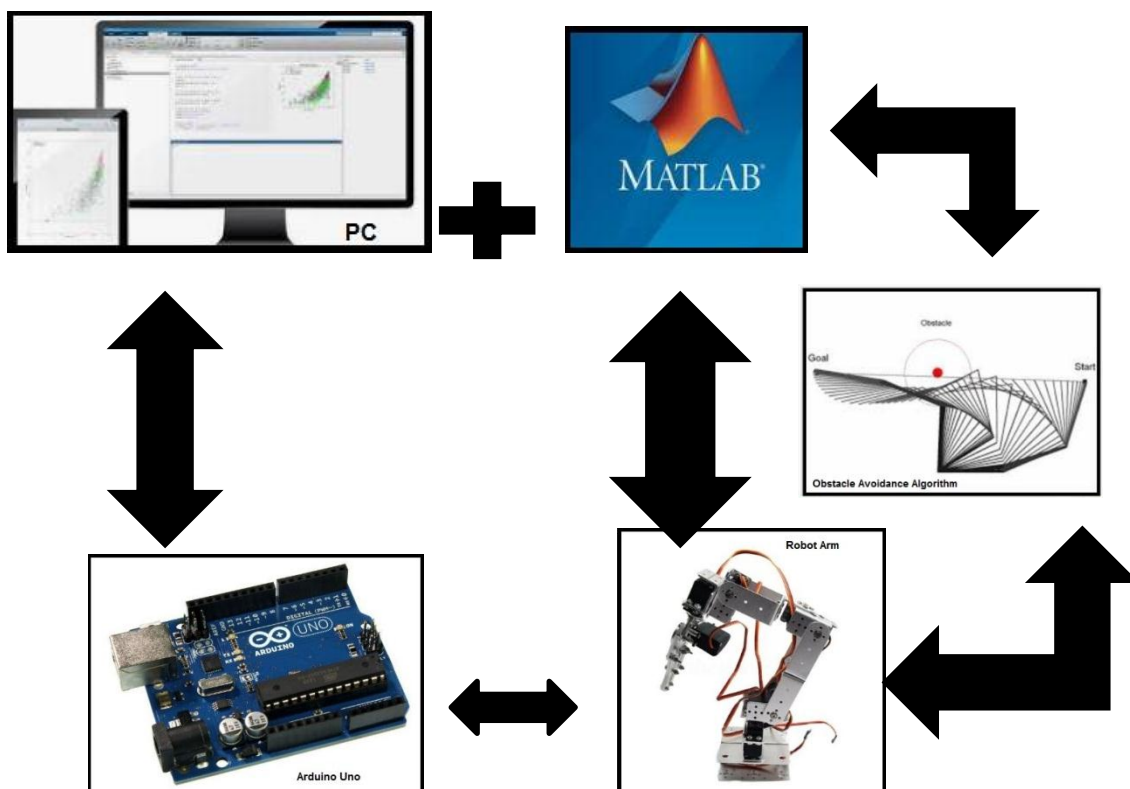
Ολοκλήρωση ανεβάσματος.
Το σχέδιο χρησιμοποιεί 7918 bytes (24%) του χώρου αποθήκευσης του
Οι καθολικές μεταβλητές χρησιμοποιούν 339 bytes (16%) δυναμικής μν
64 Arduino/Genuino Uno στο COM6

```

Εικόνα 6.3 Κώδικας στο Περιβάλλον IDE



Εικόνα 6.4 Σειριακή επικοινωνία Matlab-Arduino



Εικ.6.5 Σειριακή επικοινωνία arduino-Matlab-Βραχίονα

Ουσιαστικά οι εντολές χαμηλού επιπέδου θα εκτελούνται από τον arduino και οι εντολές υψηλού επιπέδου θα εκτελούνται από το Matlab .Στο σημείο αυτό είναι απαραίτητο να δημιουργήσουμε ένα αρχείο.m όπου θα καθορίσουμε την σύνδεση του Matlab με την σειριακή θύρα του arduino που στην περίπτωση μας είναι η 'COM6', θα ρυθμίσουμε το BAUD στις 115200 κι έπειτα με τη εντολή fopen έχουμε ενεργοποιήσει τη σειριακή επικοινωνία.

```
% Έλεγχος αν η COM θύρα είναι ανοικτή και αν είναι την κλείνουμε
if ~isempty(instrfind)
    fclose(instrfind);
    delete(instrfind);
end
%-----
%%
s = serial('COM6','BAUD',115200); % Ορισμός σειριακής στα 9600
fopen(s); % Ενεργοποίηση σειριακής
data_cell = []; % Δημιουργία array για αποθήκευση τιμών που θα σταλούν στο Arduino

% 3 διαφορετικές μεταβλητές για 3 διαφορετικά mode λειτουργίας
angle = 0; % Mode 1 -> Λειτουργία με είσοδο γωνίας
xyz = 0; % Mode 2 -> Λειτουργία με είσοδο σημείων
move = 0; % Mode 3 -> Λειτουργία κίνησης σε προκαθορισμένα σημεία
move2 = 0;
```

Εικ.6.6 Έναρξη σειριακής επικοινωνίας

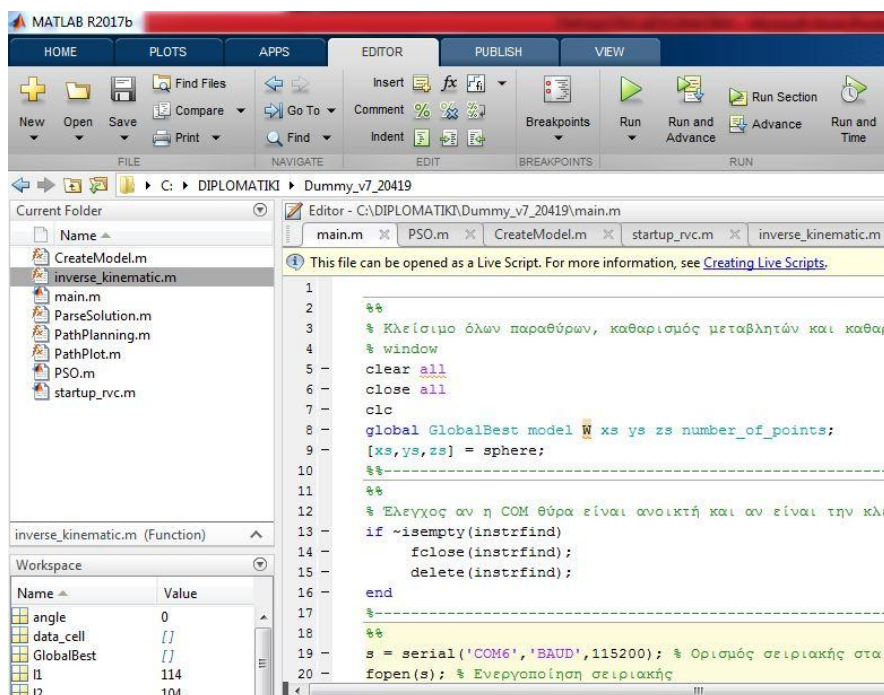
Σε αυτό το σημείο είναι σημαντικό να δημιουργήσουμε ένα αρχείο.m , όπου σε μια function θα γίνεται η επίλυση του ανάστροφου κινηματικού προβλήματος και θα ονομαστεί **inverse_kinematic.m**, καθορίζονται τα μήκη του βραχίονα και με βάση τις κινηματικές εξισώσεις του 4^{ου} κεφαλαίου, αναπτύσσουμε τον κώδικα στο προγραμματιστικό περιβάλλον του Matlab.

```
function [theta1,theta2,theta3] = inverse_kinematic(PX,PY,PZ,elbow)
%This function solves the inverse kinematic model
% Inputs: PX = desired value for x-axis
%         PY = desired value for y-axis
%         PZ = desired value for z-axis
%         elbow =
% Outputs: theta1 = angle for joint 1
%          theta2 = angle for joint 2
%          theta3 = angle for joint 3
l1 = 114; % legth of first arm
l2 = 104; % length of second arm
l3 = 223; % length of third arm
if elbow == 1
    number = 1;
elseif elbow == 0
    number = -1;
end
```

```
theta1= atan2(PY,PX);
r=sqrt(PX^2+PY^2) * number;
D=sqrt((PZ-11)^2+r^2) * number;
theta3=acos((D^2 - 13^2 - 12^2) / (2 * 12 * 13));
theta2=atan2(r,PZ - 11) - atan2(12 + (13 * cos(theta3)), 13 *
sin(theta3));

theta1=rad2deg(theta1);
if abs(theta1) <= 180 && theta1 < 0
    theta1 = 180 + theta1;
end
theta2=rad2deg(theta2);
if abs(theta2) <= 180 && theta2 < 0
    theta2 = 180 + theta2;
end
theta3=rad2deg(theta3)+90;
if abs(theta3) <= 180 && theta3 < 0
    theta3 = 180 + theta3;
end
end
```

Έτσι κάθε φορά που θα τρέχουμε το κυρίως πρόγραμμα θα τρέχει κι αυτό το αρχείο και θα επιλύεται το ανάστροφο κινηματικό για το συγκεκριμένο χειριστή . Οπότε κάθε φορά θα τρέχουμε το κυρίως πρόγραμμα το οποίο έχουμε ονομάσει **main.m** και είναι στο ίδιο path με το **inverse_kinematic.m**. Επιλέγωντας το αρχείο **main.m** και πληκτρολογώντας **RUN** (βλέπε εικ.6.7), στο command widow του Matlab έχουμε 4 επιλογές για κίνηση το βραχίονα(βλέπε εικ6.8).



Εικόνα 6.7 Περιβάλλον matlab έτοιμο για run

```

1 %%
2 % Κλείσιμο όλων παραθύρων, καθαρισμός μεταβλητών και καθαρισμός του comma
3 % window
4 - clear all
5 - close all
6 - clc
7 - global GlobalBest model xs ys zs number_of_points;
8 - [xs,ys,zs] = sphere;
9 %-----
10 %%
11 % Έλεγχος αν η COM θύρα είναι ανοικτή και αν είναι την κλείνουμε
12 - if ~isempty(instrfind)
13 -     fclose(instrfind);

```

Command Window

New to MATLAB? See resources for [Getting Started](#).

Press 1 for angles, 2 for points, 3 for a simple movement, 4 for path planning:

Εικόνα 6.8 Τέσσερις επιλογές από το χρήστη με βάση τον τρόπο που θέλει να κινήσει το βραχίονα

Εφόσον τρέξουμε το αρχείο **main.m** ,στο command window θα εμφανιστεί το παρακάτω μήνυμα:

Press 1 for angles, 2 for points, 3 for a simple movement, 4 for path planning:

Οπότε ο χρήστης με βάση την επιθυμία για τον τρόπο κίνησης του βραχίονα , έχει τη δυνατότητα να επιλέξει ανάμεσα σε 4 τρόπους.

Press 1.....για κίνηση με γωνίες του κάθε σέρβο ξεχωριστά π.χ 90,45,135.. **Press**

2.....για τοποθέτηση του άκρου σε συγκεκριμένα σημεία Px,Py,Pz π.χ 0,327,114

Press 3.....για συνεργασία όλων των αρθρώσεων και των ενεργοποιητών για τον σχηματισμό ευθείας γραμμής , όσο τρέχει η επιλογή 3 ,στο command window ο χρήστης έχει τη δυνατότητα να βλέπει πως μεταβάλλονται οι τιμές των γωνιών των servo.

Press 4.....για κίνηση του βραχίονα από σημείο A(αφετηρία) σε σημείο B(τερματισμό) , επιλέγοντας την καλύτερη δυνατή διαδρομή εντός του χώρου εργασίας και με ταυτόχρονη αποφυγή εμποδίων, η επιλογή αυτή θα αναλυθεί περαιτέρω σε επόμενο κεφάλαιο.

6.2.2 Εγκατάσταση του πακέτου προσομείωσης rvc_tools by Peter Corke

Επίσης έχουμε εγκαταστήσει και χρησιμοποιήσει ένα πακέτο προσομείωσης στο Matlab , δημιουργός του οποίου είναι ο Peter Corke ('Robotics, Vision & Control: (c) Peter Corke 1992-2017 <http://www.petercorke.com>'), το οποίο θα μας βοηθήσει στην σύγκριση της πραγματικής κίνησης και της προσομείωσης . Στο σημείο αυτό για την προσομείωση , αρχικά τρέχουμε το αρχείο startup_rvc.m κι έπειτα το main.m , έτσι ανα πάσα στιγμή θα έχουμε για κάθε live κίνηση του βραχίονα και την κίνηση στην προσομείωση(βλέπε εικ 6.9)

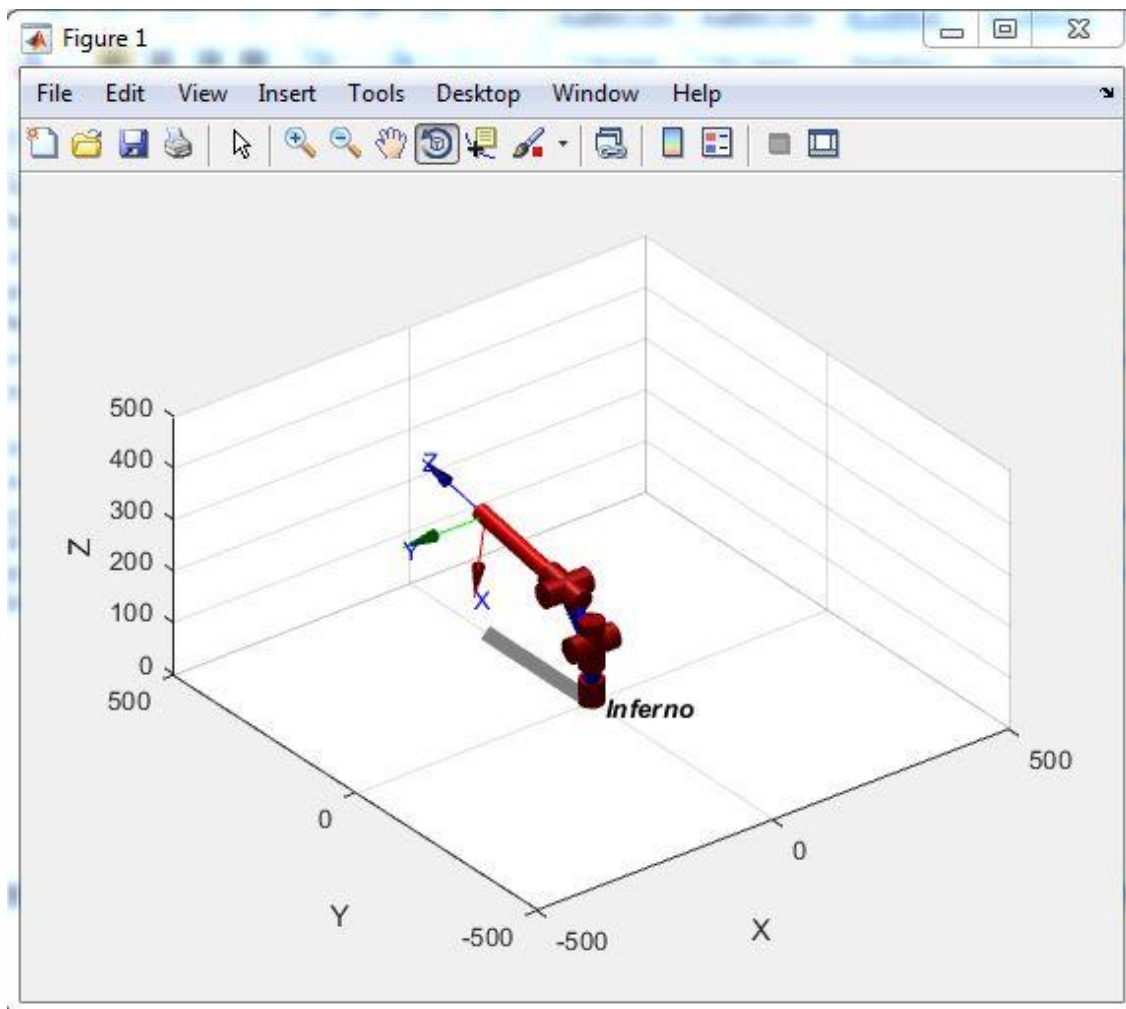
Αφού τρέξουμε το αρχείο `startup_rvc.m`, θα πρέπει να λάβουμε το παρακάτω μήνυμα:
`>> startup_rvc`

Robotics, Vision & Control: (c) Peter Corke 1992-2017 <http://www.petercorke.com>

- **Robotics Toolbox for MATLAB (release 10.3.1)**

- **ARTE contributed code: 3D models for robot manipulators
(C:\DIPLOMATIKI\rvctools\robot\data\ARTE)**

- **pHRIWARE (release 1.2): pHRIWARE is Copyrighted by Bryan Moutrie (2013-2019)
(c)**



Εικ. 6.9 Προσομείωση του βραχίονα στο Matlab

6.3 Κίνηση και χειρισμός του Ρομποτικού βραχίονα με διάφορες επιλογές

6.3.1 Κίνηση και χειρισμός του Ρομποτικού βραχίονα με γωνίες(mode-1)

Αρχικά θα δοκιμάσουμε να κινήσουμε το βραχίονα με την πρώτη επιλογή, τρέχοντας το αρχείο **startup_rvc.m** κι έπειτα το αρχείο **main.m** θα ερωτηθούμε στο command window του matlab, για το πώς θέλουμε να μετακινήσουμε το βραχίονα, εμείς θα πληκτρολογήσουμε1, δηλαδή κίνηση του βραχίονα με εισαγωγή γωνιών.

```

1  %%
2  % Κλείσιμο όλων παραθύρων, καθαρισμός μεταβλητών και καθαρισμός του command
3  % window
4  - clear all
5  - close all
6  - clc
7  - global GlobalBest model xs ys zs number_of_points;
8  - [xs,ys,zs] = sphere;
9  %%-----
10 %%
11 % Έλεγχος αν η COM θύρα είναι ανοικτή και αν είναι την κλείνουμε
12 - if ~isempty(instrfind)
13 -     fclose(instrfind);

```

Command Window

New to MATLAB? See resources for [Getting Started](#).

Press 1 for angles, 2 for points, 3 for a simple movement, 4 for path planning:1
fx Give angles for joints th1,th2,th3 from 16-165 each:

Εικ. 6.10 Επιλογή κίνησης με γωνίες

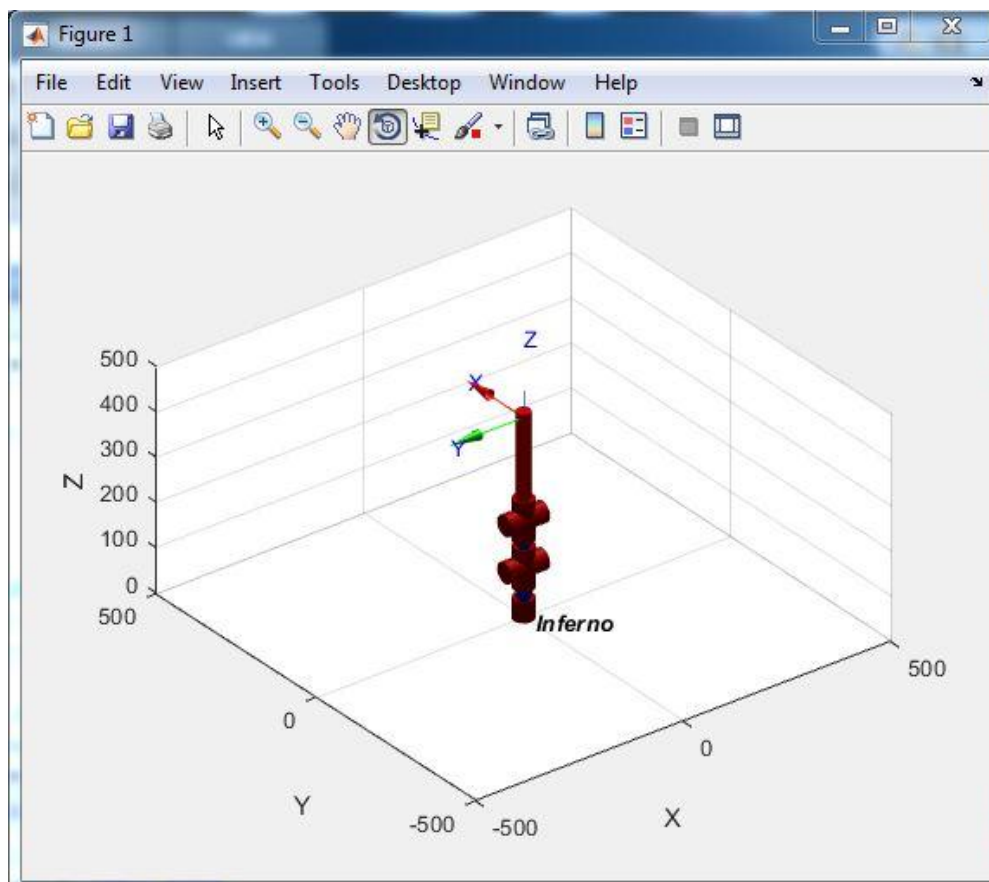
Πολύ σημαντικό είναι το εύρος των γωνιών να κυμένεται από 16-165 μοίρες ώστε να μην επιβαρύνουμε τους σερβο-κινητήρες με απότομα χτυπήματα επάνω στην μεταλλική κατασκευή, ένα άλλο σημαντικό κομμάτι είναι καθώς δίνουμε τις γωνίες να τοποθετούμε κόμμα ανάμεσά τους , αλλιώς δεν θα λάβουμε κάποια κίνηση. Ας προχωρήσουμε στο πρώτο παράδειγμα , πληκτρολογώντας λοιπόν στην επιλογή 1, τις γωνίες 90,90,90,

```
Command Window
New to MATLAB? See resources for Getting Started.

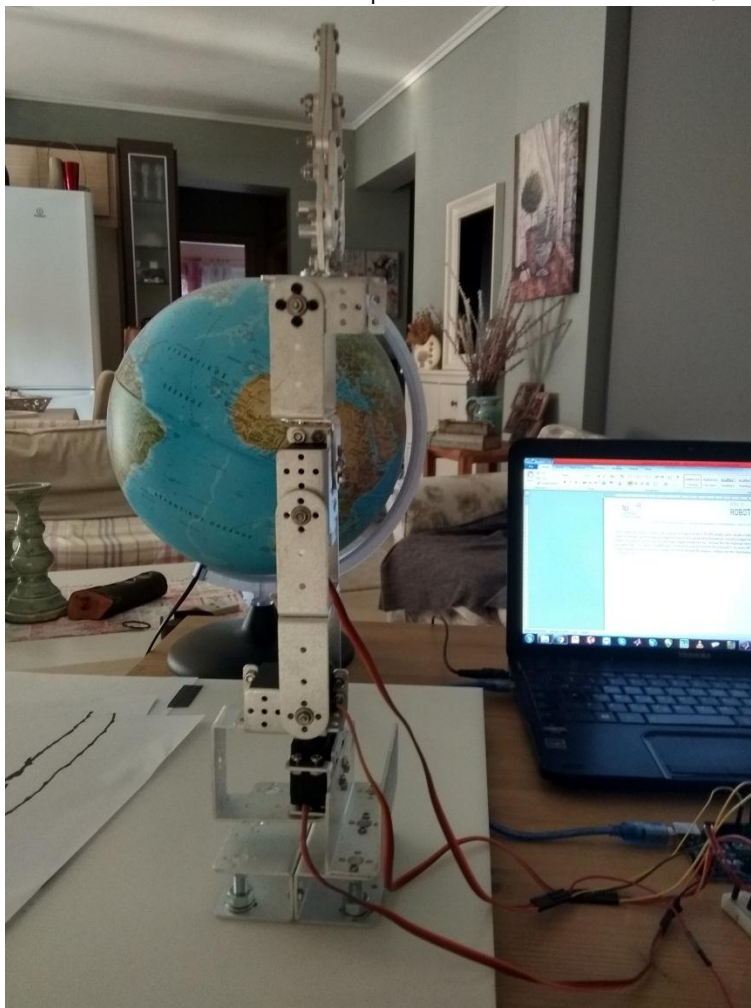
Press 1 for angles, 2 for points, 3 for a simple movement, 4 for path planning:1
Give angles for joints th1,th2,th3 from 16-165 each: 90,90,90
90,90,90
Give angles for joints th1,th2,th3 from 16-165 each:
```

Εικόνα 6.11

(βλ. Εικ 6.11) παρατηρούμε κίνηση όλων των σερβο στη θέση default 90 μοιρών , καθώς και στην προσομείωση.(βλέπε εικόνες 6.12 και 6.13). **Παράδειγμα1.1:**



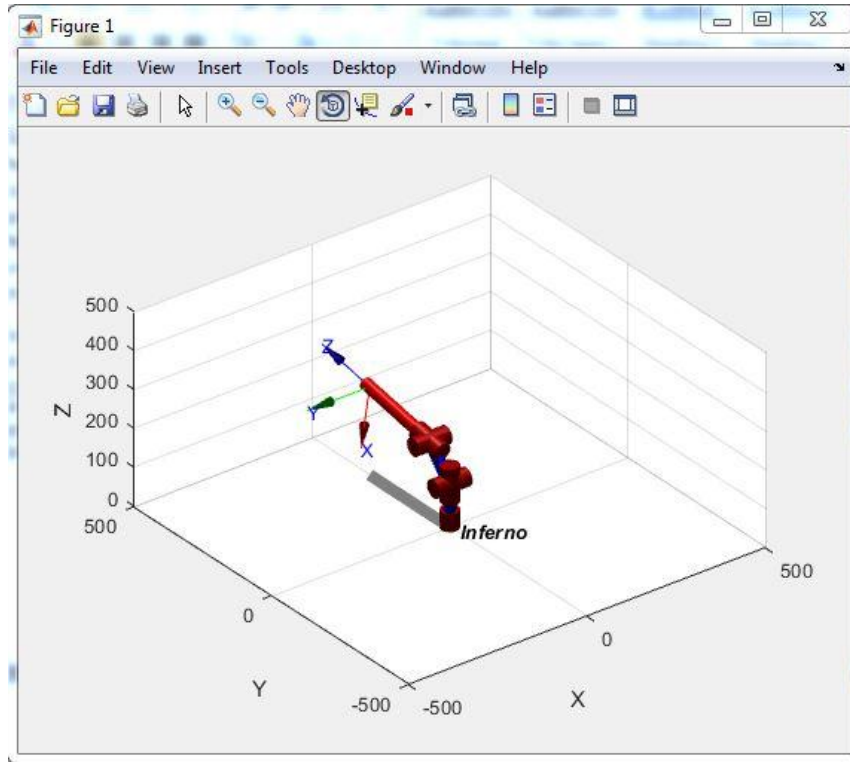
Εικ. 6.12 Προσομείωση Επιλογή 1^η 90,90,90 θέση default



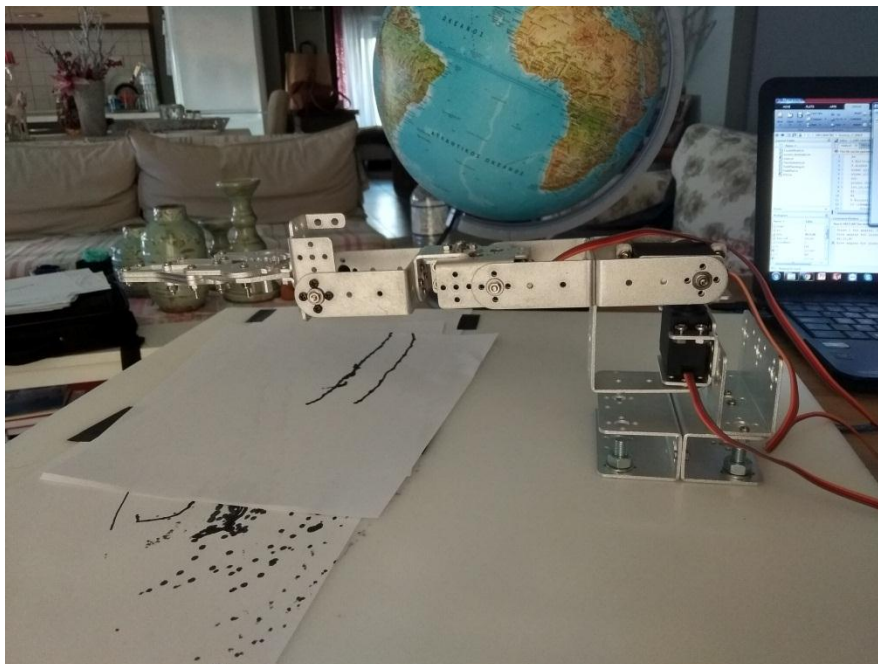
Εικ. 6.13 Πραγματικότητα Επιλογή 1^η 90,90,90 θέση default

Παράδειγμα 2.1:

Έτσι συνεχίζουμε στην 1η επιλογή με εισαγωγή γωνιών , 90,16,90 με το βραχίονα σε πλήρη έκταση.



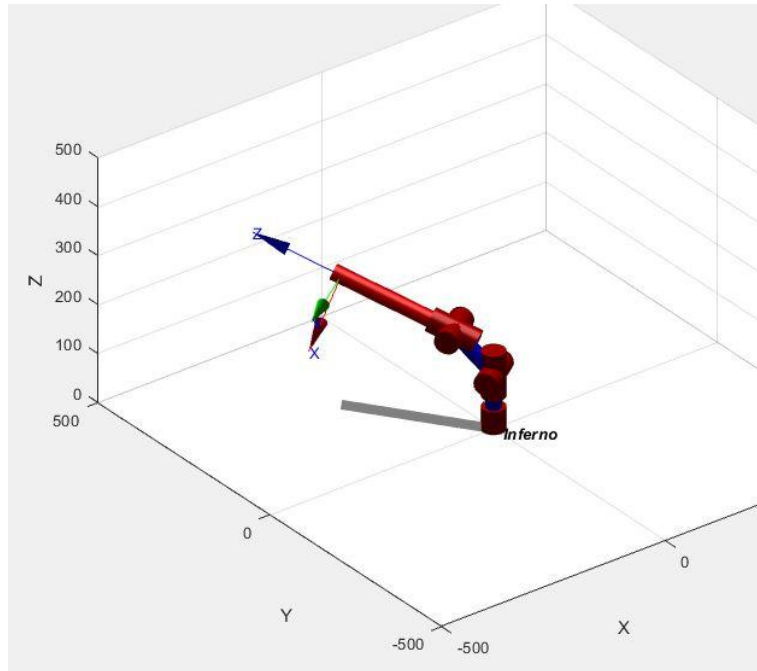
Εικ. 6.14 Προσομείωση Επιλογή 1^η 90,16,90 πλήρη έκταση



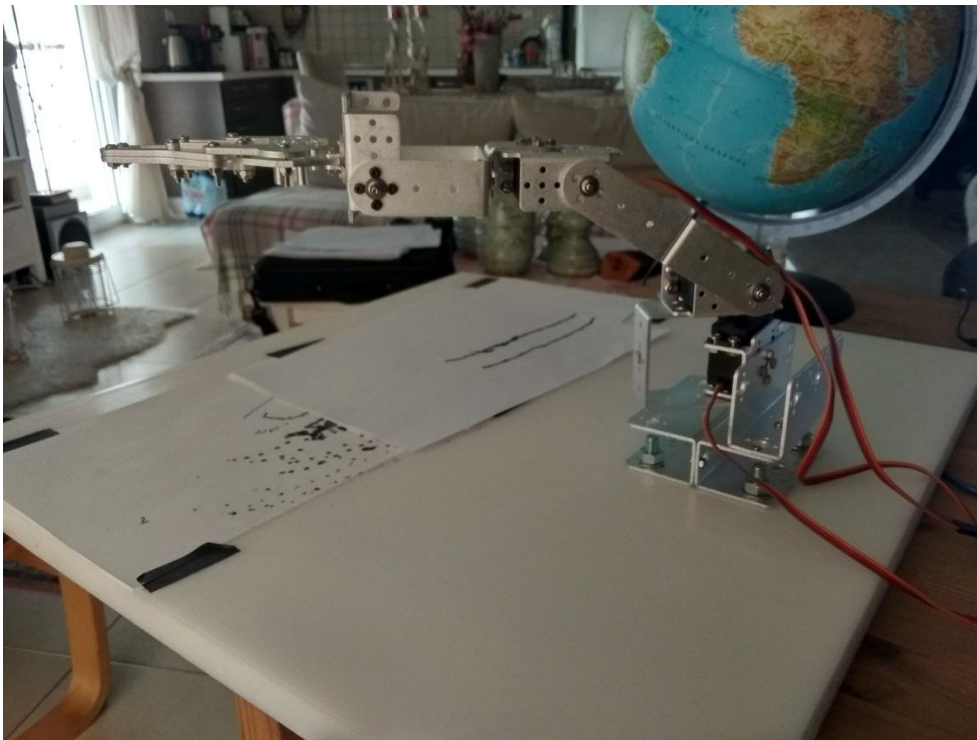
Εικ. 6.15 Πραγματικότητα Επιλογή 1^η 90,16,90 πλήρη έκταση

Παράδειγμα 3.1:

Έπειτα θα συνεχίσουμε στην επιλογή 1 με εισαγωγή γωνιών , 125,45,115 .

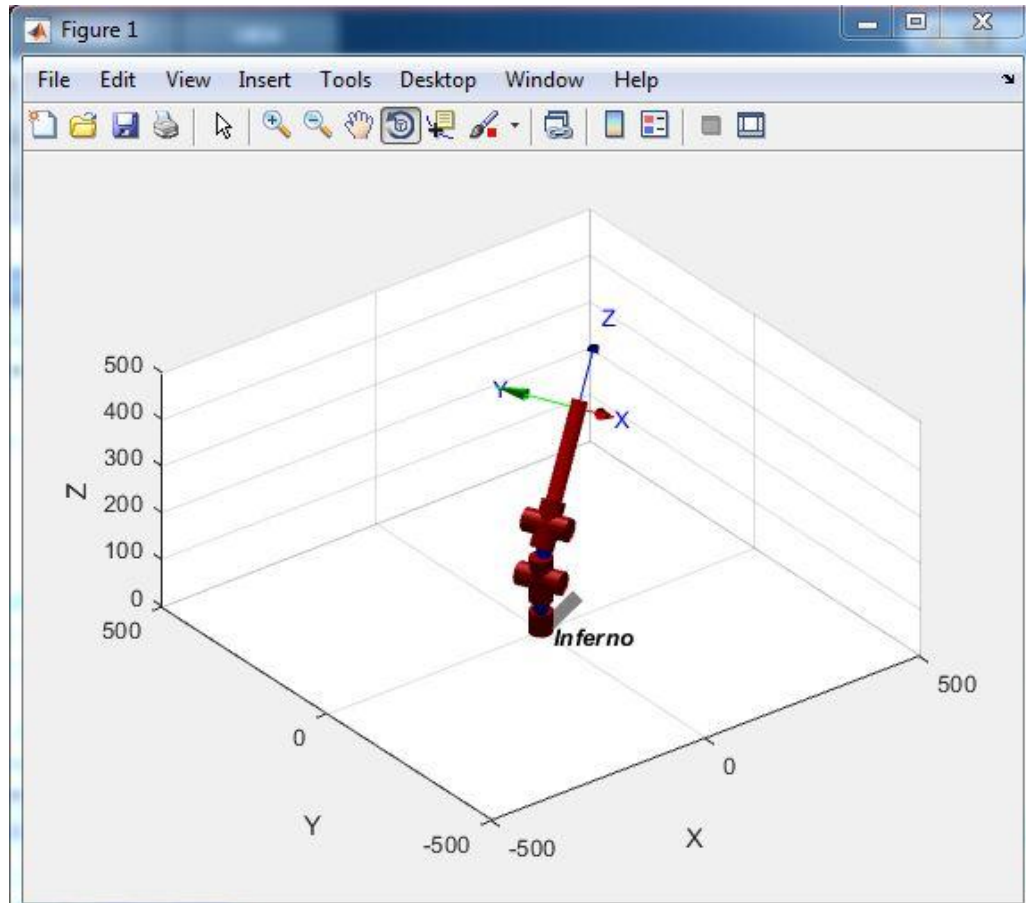


Εικ. 6.16 Προσομείωση Επιλογή 1^η 125,45,115

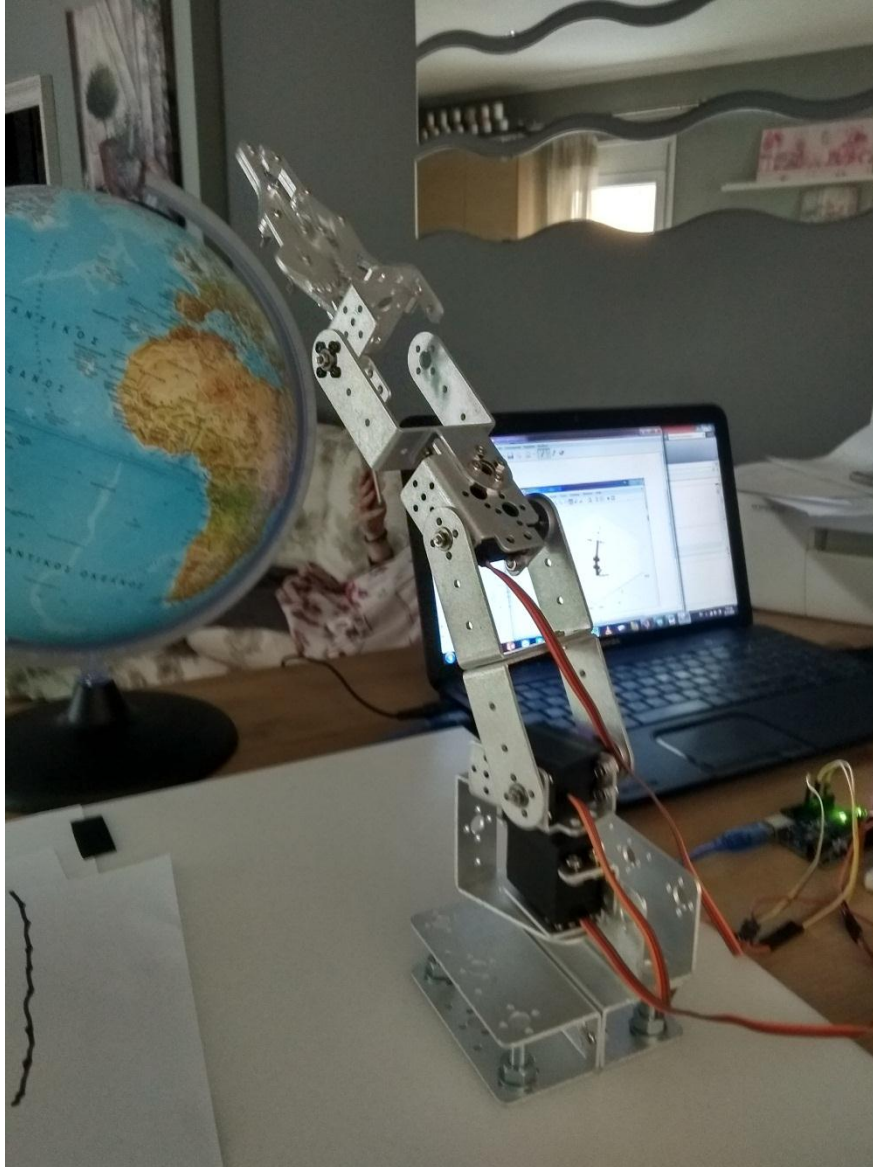


Εικ. 6.17 Πραγματικότητα Επιλογή 1^η 125,45,115

Παράδειγμα 4.1: Έπειτα θα συνεχίσουμε στην επιλογή 1 με εισαγωγή γωνιών , 25,75,108



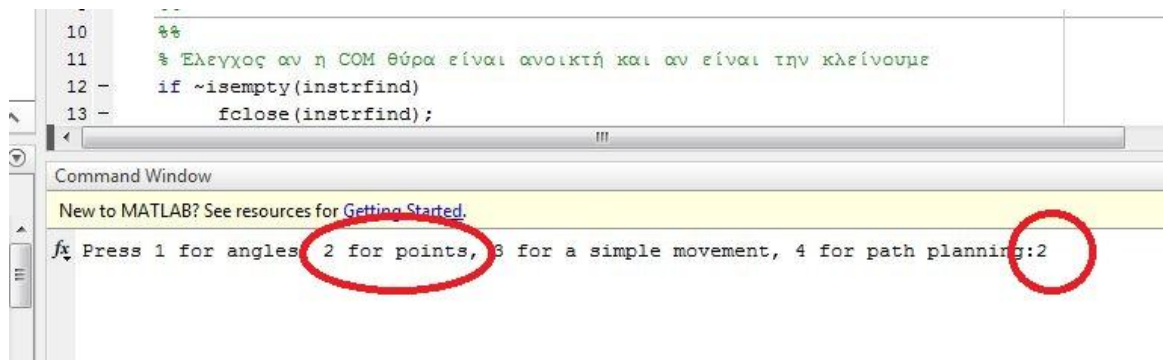
Εικ. 6.18 Προσομείωση Επιλογή 1^η 25,75,108



Εικ. 6.19 Πραγματικότητα Επιλογή 1^η 25,75,108

6.3.2 Κίνηση και χειρισμός του Ρομποτικού βραχίονα με σημεία Px,Py,Pz(mode-2)

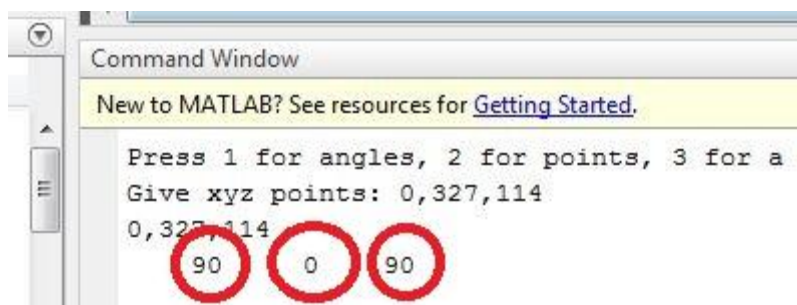
Αφού λοιπόν ξανατρέξουμε το αρχείο **main.m** , θα πληκτρολογήσουμε όταν ερωτηθούμε πως θέλουμε να κινήσουμε το βραχίονα , έτσι θα πληκτρολογήσουμε το 2 (βλέπε εικ 6.19) και Enter και στη συνέχεια το σημεία που επιθυμούμε να κινηθεί ο βραχίονας κατά τους άξονες X-Y-Z , επίσης με κόμμα ανάμεσά .



Εικ. 6.20 Επιλογή 2^η Κίνηση με σημεία εντός του χώρου εργασίας

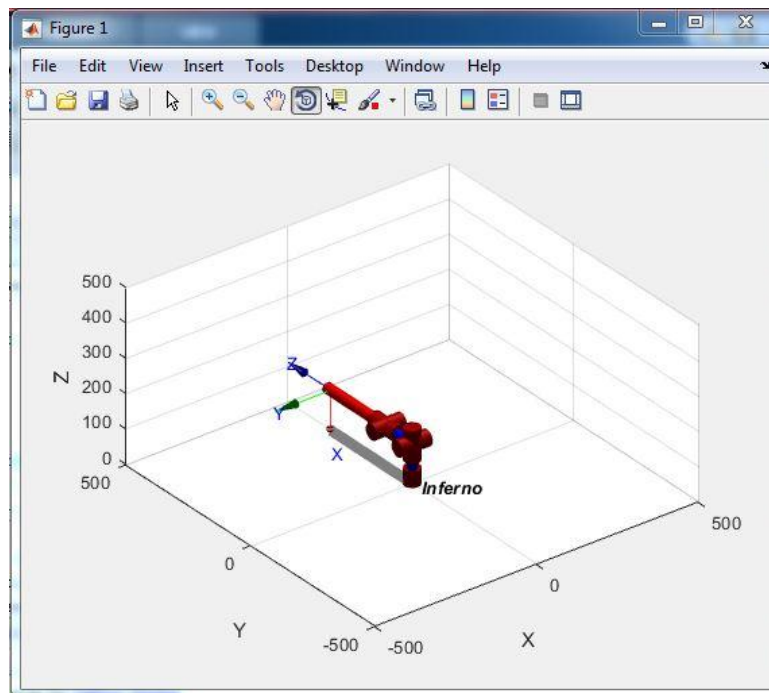
Αφού πληκτρολογήσουμε το 2 και Enter, θα λάβουμε το μήνυμα : **Give xyz points:** έπειτα θα πρέπει να πληκτρολογήσουμε τα επιθυμητά σημεία , όπως αναφέραμε και παραπάνω .

Παράδειγμα1.2: 0, 327, 114 πλήρη έκταση του βραχίονα με ύψος z=114mm, το ύψος δηλαδή της πρώτης άρθρωσης και y=327mm τα μήκη των 2 αρθρώσεων , έτσι το αποτέλεσμα που θα πρέπει να πάρουμε είναι ορθή γωνία. Στο σημείο αυτό θα παρατηρήσουμε στο command window του Matlab την εμφάνιση της μεταβολής των γωνιών που πέρνουν οι σερβοκινητήρες με βάση τα επιθυμητά σημεία. Κατ αρχάς λαμβάνουμε τις γωνίες των σερβο.



Εικόνα 6.21 Γωνίες των σερβοκινητήρων με βάση τα επιθυμητά σημεία

Έπειτα την προσομείωση και την πραγματική κίνηση του βραχίονα στο χώρο.

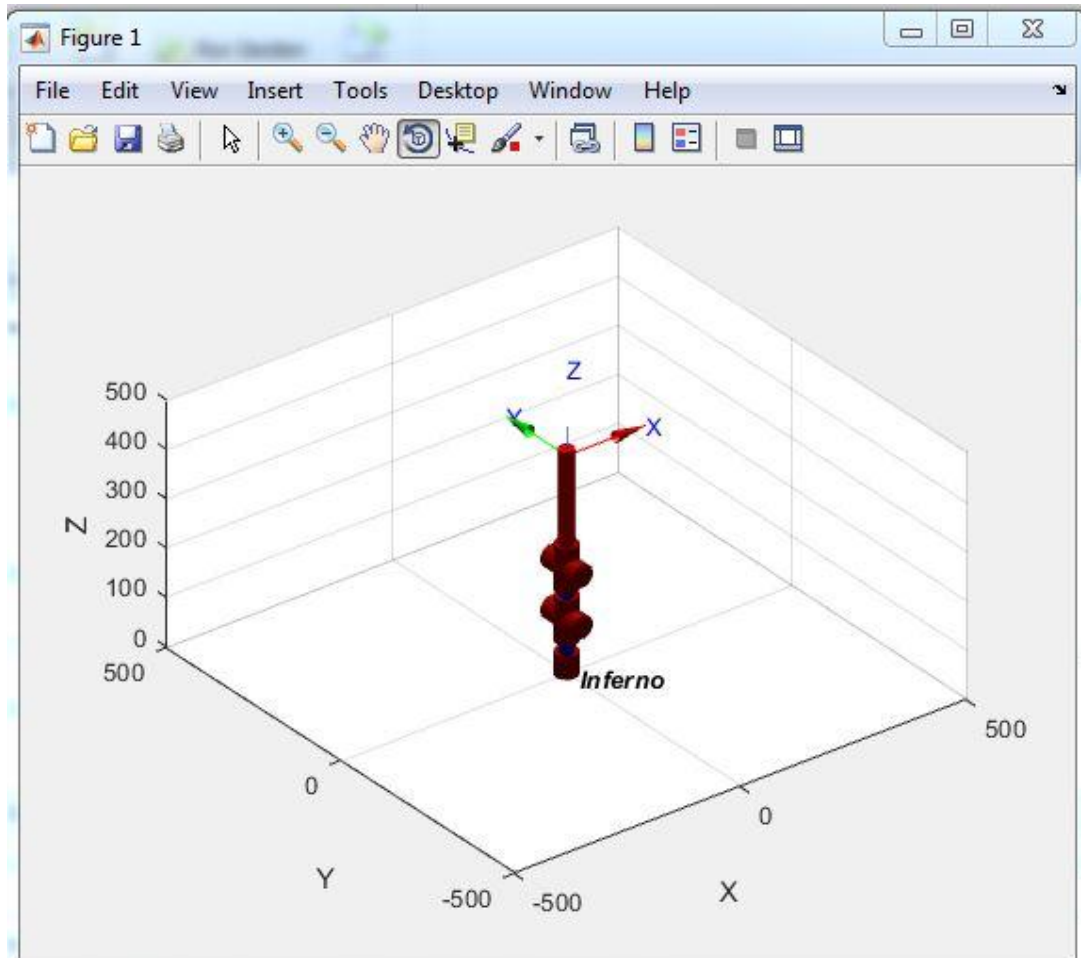


Εικ. 6.22 προσομείωση $X=0$, $Y=327$, $Z=114$

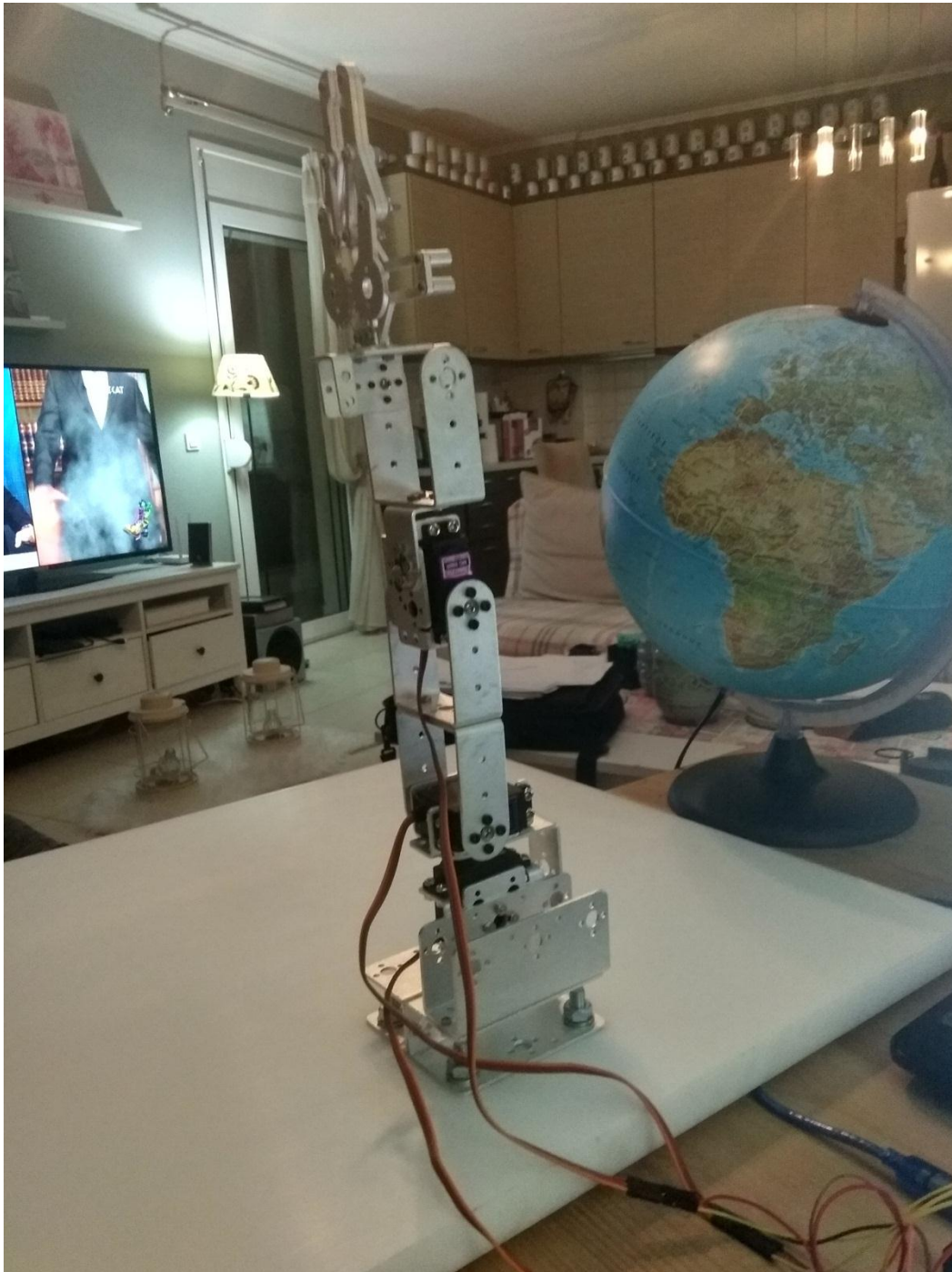


Εικ. 6.23 Πραγματική κίνηση στο χώρο $X=0$, $Y=327$, $Z=114$

Παράδειγμα 2.2 : με τον ίδιο τρόπο θα δώσουμε inputs $X=0$, $Y=0$, $Z=441$, λαμβάνουμε γωνίες 0,90,90.

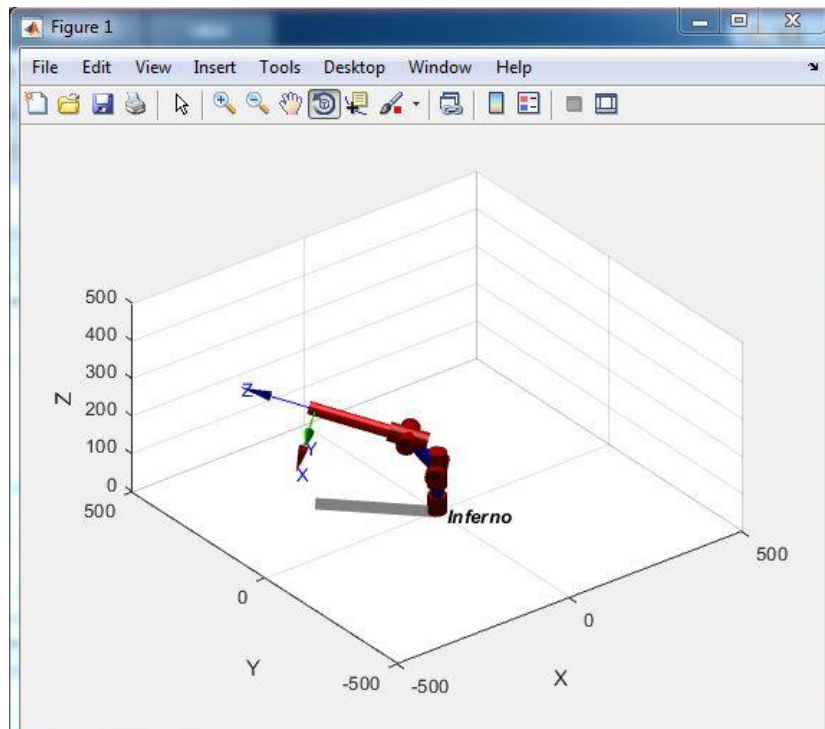


Εικ. 6.24 προσομείωση $X=0$, $Y=0$, $Z=441$



Εικ. 6.25 Πραγματική κίνηση στο χώρο $X=0$, $Y=0$, $Z=441$

Παράδειγμα 3.2 : με τον ίδιο τρόπο θα δώσουμε inputs $X = -200$, $Y = 200$, $Z = 250$, λαμβάνουμε γωνίες 135.....49.79.....125.

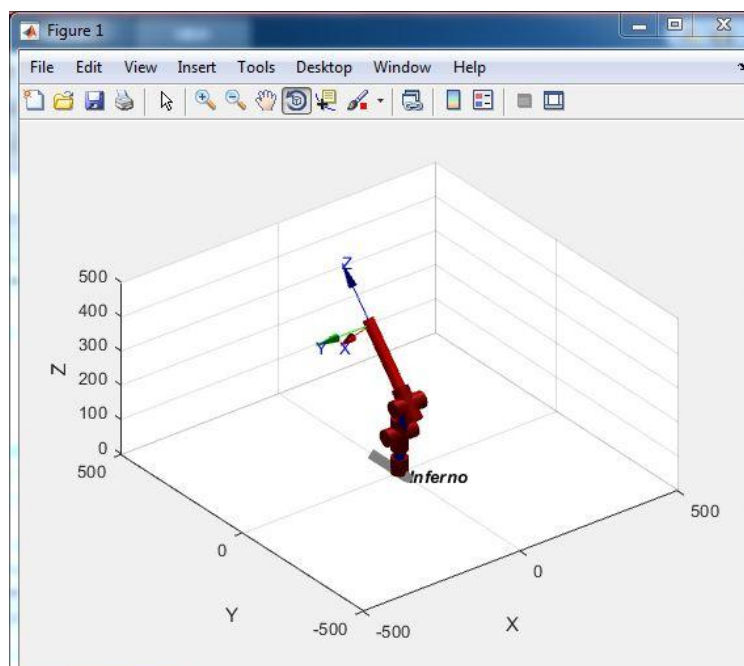


Εικ. 6.26 Προσομείωση $X = -200$, $Y = 200$, $Z = 250$



Εικ. 6.27 Πραγματική κίνηση στο χώρο $X = -200$, $Y = 200$, $Z = 250$

Παράδειγμα 4.2 : με τον ίδιο τρόπο θα δώσουμε inputs $X=0$, $Y=120$, $Z=375$, λαμβάνουμε γωνίες 90.....108.....151

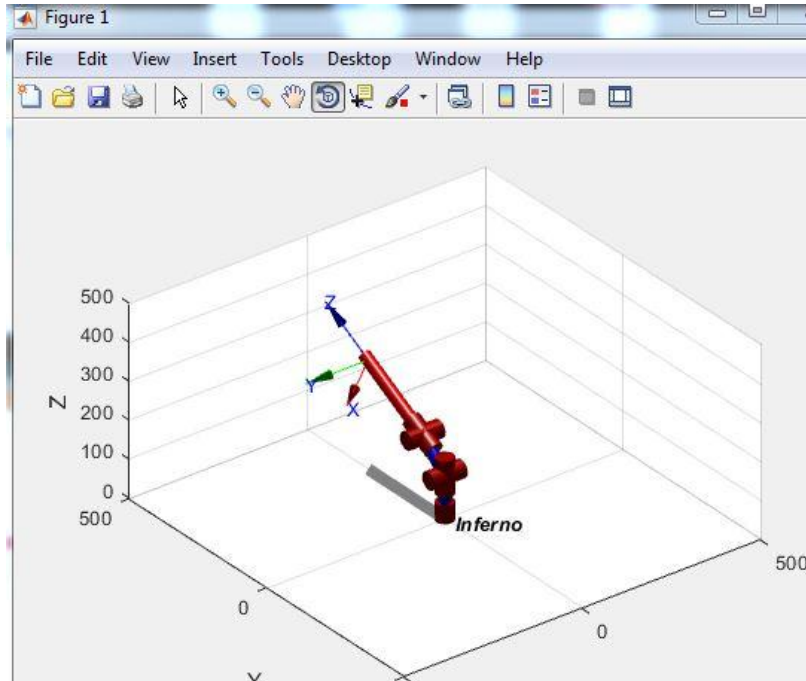


Εικ. 6.28 Προσομείωση $X=0$, $Y=120$, $Z=375$



Εικ. 6.29 Πραγματική κίνηση στο χώρο $X=0$, $Y=120$, $Z=375$

Παράδειγμα 5.2 : με τον ίδιο τρόπο θα δώσουμε inputs $X=0$, $Y=280$, $Z=280$, λαμβάνουμε γωνίες 90.....38,6.....101



Εικ. 6.30 Προσομείωση $X=0$, $Y=280$, $Z=280$



Εικ. 6.31 Πραγματική κίνηση στο χώρο $X=0$, $Y=280$, $Z=280$

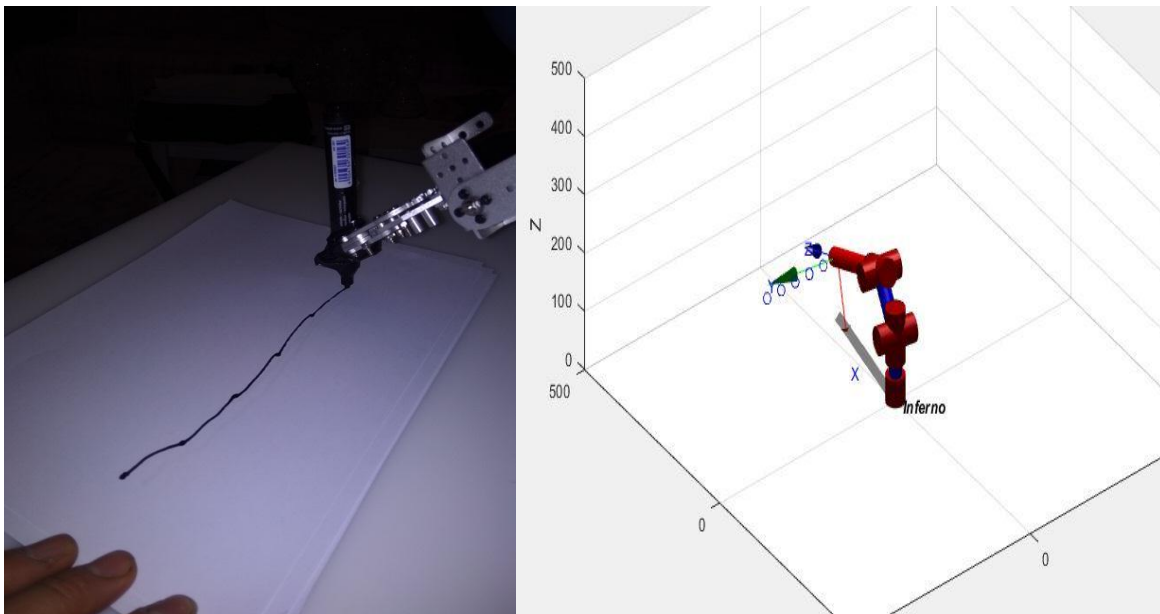
6.3.3 Κίνηση του βραχίονα σε ευθεία γραμμή (mode-3)

Στο σημείο αυτό λοιπόν τρέχουμε το αρχείο **main.m** κι όταν ερωτηθούμε με ποιά από τις τέσσερις επιλογές επιθυμούμε να χειριστούμε το βραχίονα εμείς πληκτρολογούμε την επιλογή 3... Έτσι λοιπόν με τη βοήθεια της `linspace`, θα προσπαθήσουμε με γραμμική παρεμβολή να κινήσουμε το άκρο του βραχίονα σε ευθεία γραμμή, για διευκόλυσή μας τοποθετήσαμε για άκρο εργασίας ένα μαρκαδόρο.

Παράδειγμα 1.3: Στο παράδειγμα αυτό θα κινήσουμε το βραχίονα σε καθορισμένο $Y=280$, $Z=100$ και κίνηση σε 6 διαδοχικά σημεία από την αρχή μέχρι το τέλος. (βλέπε εικ. 6.31)

```
space = linspace(-150, 50, 6);
x = space;
y = 280 * ones(1, length(space));
z = 100 * ones(1, length(space));
i = 1; % Δείχνει σε ποιο σημείο της κίνησης είμαστε
```

Εικ. 6.32 `linspace` για ευθεία γραμμή παραδείγματος 1

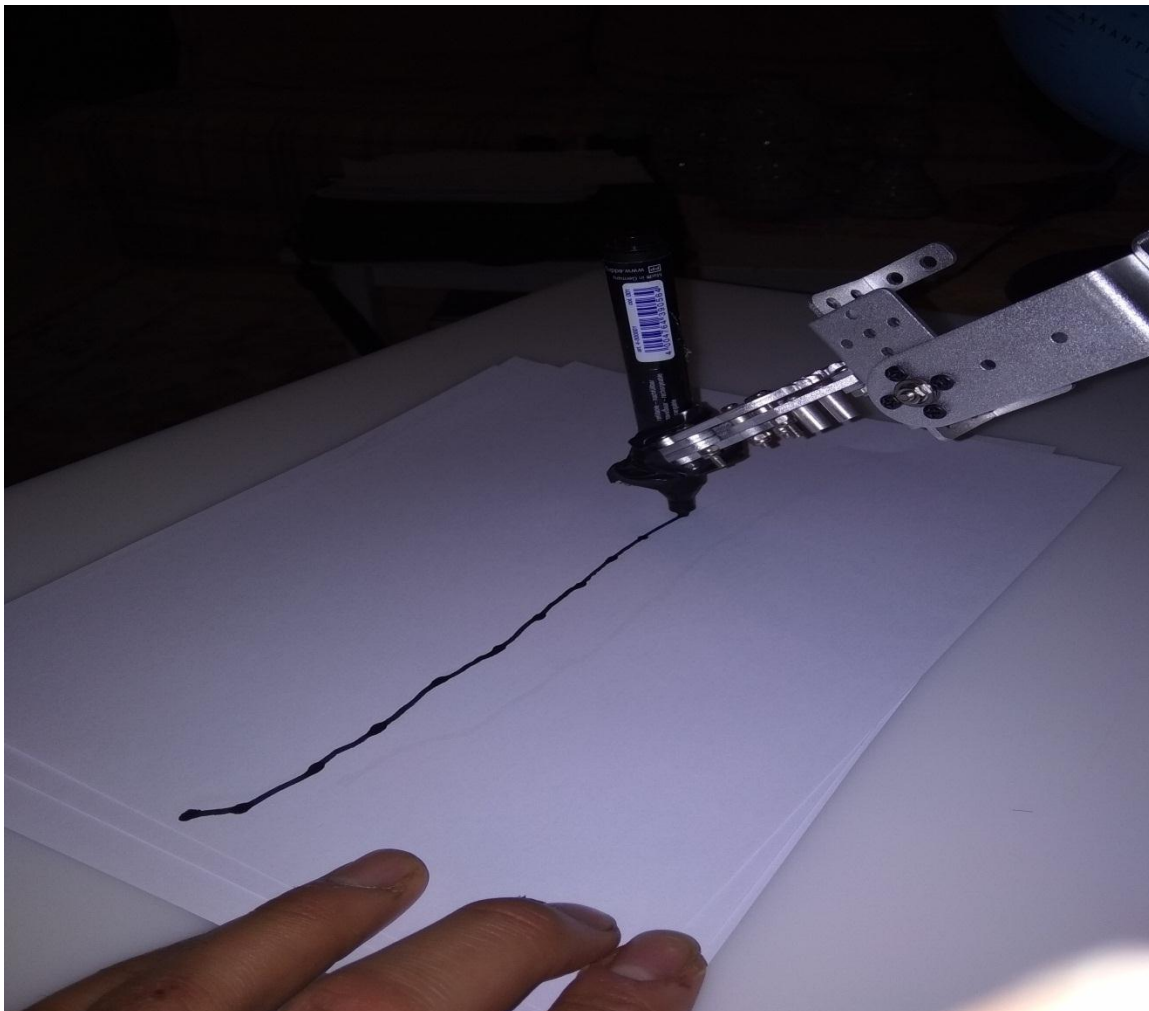


Εικ. 6.33 Αριστερά η πραγματική κίνηση και δεξιά η προσομείωση

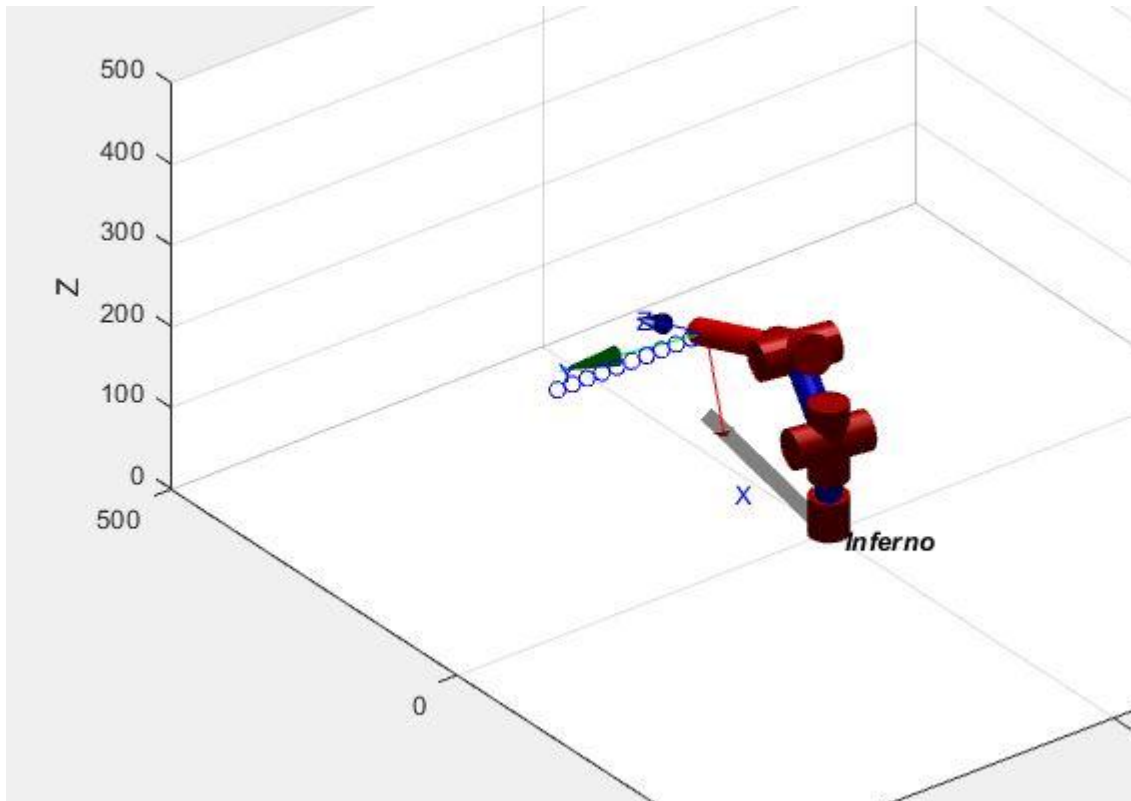
Παράδειγμα 2.3: Στο παράδειγμα αυτό θα κινήσουμε το βραχίονα σε καθορισμένο $Y=280$, $Z=100$ και κίνηση σε 11 σημεία προς σχηματισμό ευθείας.(βλέπε εικ. 6.33)

```
% κίνηση α
space = linspace(-150,50,11);
x = space;
y = 280 * ones(1,length(space));
z = 100 * ones(1,length(space));
i = 1; % Δείχνει σε ποιο σημείο της κίνησης είμαστε
```

Εικ. 6.34 linspace για ευθεία γραμμή παραδείγματος 2

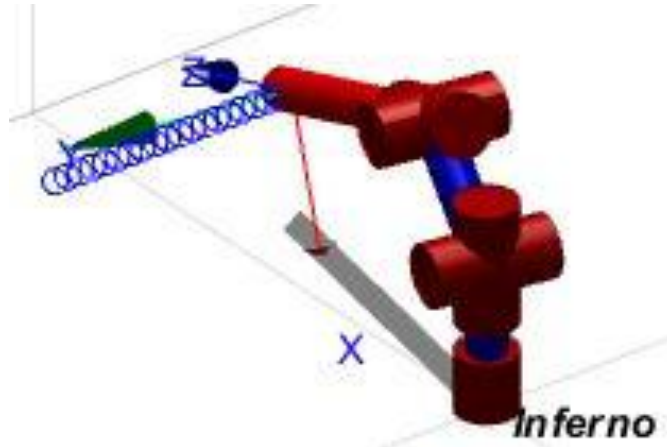


Εικ. 6.35 Πραγματική κίνηση του βραχίονα σε ευθεία γραμμή σε 11 σημεία

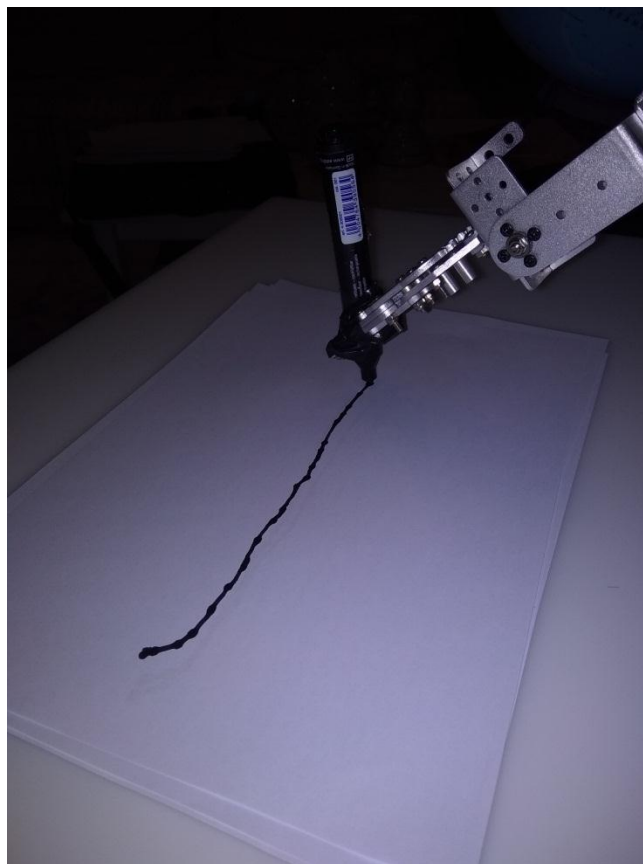


Εικ. 6.36 Προσομείωση κίνησης του βραχίονα σε ευθεία γραμμή 11 σημείων

Παράδειγμα 3.3: Στο παράδειγμα αυτό θα κινήσουμε το βραχίονα σε καθορισμένο $Y=280$, $Z=100$ και κίνηση σε 20 σημεία προς σχηματισμό ευθείας.



Εικ. 6.37 Προσομείωση κίνησης του βραχίονα σε ευθεία γραμμή σε ακολουθία 20 σημείων



Εικ. 6.38 Πραγματική κίνηση του βραχίονα σε ευθεία γραμμή σε ακολουθία 20 σημείων

Κατα το σχηματισμό ευθείας γραμμής , ο βραχίονας κινείται σε μια ακολουθία σημείων, στο command window του Matlab έχουμε τη δυνατότητα παρακολούθησης ανα πάσα στιγμή , την τιμή που παίρνει ο κάθε servo-κινητήρας ξεχωριστά ,από την αρχή έως το τέλος, έως και τον τερματισμό εκτέλεσης της κίνησης.

```

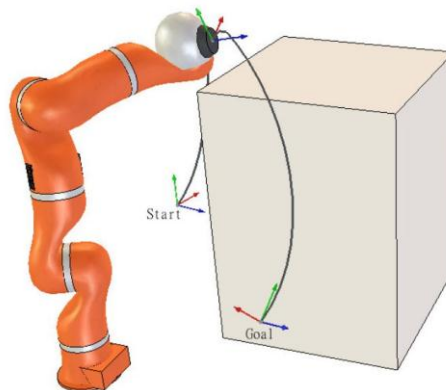
Command Window
New to MATLAB? See resources for Getting Started.

[103]    [40]    [151]
[101]    [41]    [153]
[99]     [42]    [154]
[97]     [43]    [156]
[95]     [44]    [156]
[93]     [44]    [157]
[91]     [44]    [157]
[88]     [44]    [157]
[86]     [44]    [157]
[84]     [43]    [156]
[82]     [43]    [155]
[80]     [42]    [154]
  
```

Εικ. 6.39 Εμφάνιση τιμών στο Command window κατά το σχηματισμό ευθείας γραμμής

Κεφάλαιο 7

7. Εισαγωγή ευφύϊας στο σύστημα για την αποφυγή εμποδίων



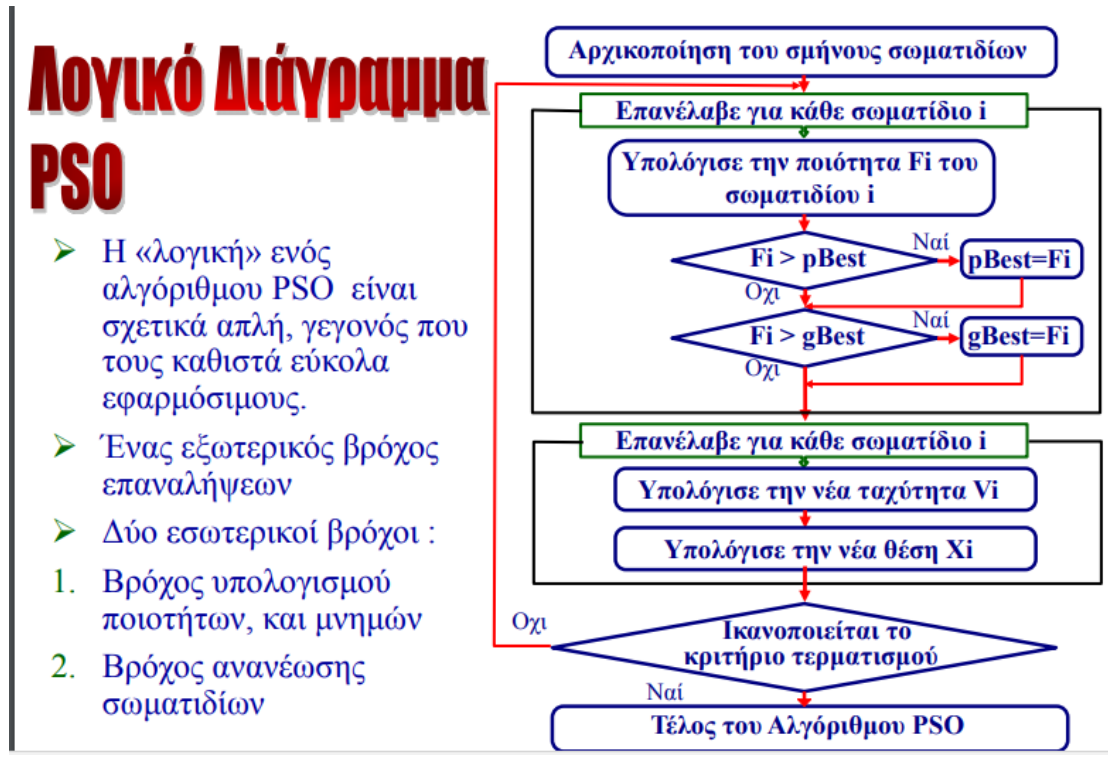
Εικ. 7.1 Εύρεση βέλτιστης διαδρομής αποφεύγωντας εμπόδια

7.1 Ο αλγόριθμος PSO_Particle Swarm Optimization εις την ελληνικήν Βελτιστοποίηση Σμήνους Σωματιδίων

Η μέθοδος αυτή χρησιμοποιεί έναν αριθμό σημείων στο χώρο (λύσεις, σωματίδια) που αποτελούν ένα σμήνος το οποίο κινείται μέσα στον χώρο λύσεων ενός προς-επίλυση προβλήματος, ψάχνοντας για την βέλτιστη λύση. Κάθε σωματίδιο στον χώρο λύσεων προσαρμόζει διαρκώς τον «τρόπο πτήσης» του σύμφωνα με την δική του «εμπειρία πτήσης» αλλά και σύμφωνα με την «εμπειρία πτήσης» και των υπολοίπων σωματιδίων του σμήνους. Τα τελευταία χρόνια ο αλγόριθμος **PSO** έχει γίνει μία εξαιρετικά δημοφιλής εξελικτική μέθοδος με πολλές επιτυχημένες εφαρμογές.



Εικ. 7.2 Μέθοδος βελτιστοποίησης σμήνους σωματιδίων PSO



Εικ. 7.3 Λογικό διάγραμμα του PSO

Πλεονεκτήματα της εφαρμογής PSO έναντι άλλων μεθόδων

- Είναι απλός στην υλοποίηση
- Δεν επηρεάζεται σημαντικά από της παραμέτρους υλοποίησης
- Εύκολα παραλληλοποιείται για παράλληλη επεξεργασία
- Δεν χρησιμοποιεί παραγώγους (derivative free)
- Έχει λίγες παραμέτρους
- Είναι ένας πολύ αποτελεσματικός αλγόριθμος γενικής Βελτιστοποίησης

Μειονεκτήματα του PSO έναντι άλλων μεθόδων

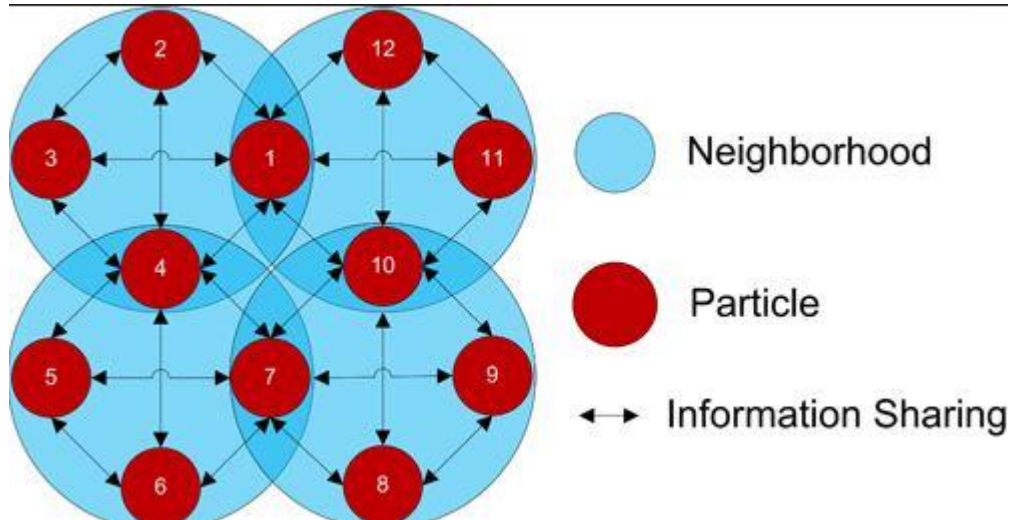
- Τείνει να συγκλίνει πρώιμα σε τοπικά βέλτιστα μεσαίας ποιότητας
- Αργεί να συγκλίνει επακριβώς στο σημείο του βέλτιστου, επιδεικνύοντας ασθενή ικανότητα τοπικής αναζήτησης.

Εφαρμογές του PSO

- Προβλήματα κοπής και συσκευασίας
- Σχεδίαση προσαρμοζόμενων FIR φίλτρων
- Ακύρωση ακουστικής ηχούς (Echo Cancellation)
- Σχεδίαση προσαρμοζόμενων κεραιών
- Ανίχνευση και παρακολούθηση κίνησης (Motion Tracking)
- Συμπύεση εικόνας
- Διαχωρισμός σημάτων (Signal Separation)
- Ανίχνευση ασθενών σημάτων (Weak Signal Detection)
- Συστήματα παραγωγής ηλεκτρικής ενέργειας
- Εκπαίδευση Νευρωνικών Δικτύων
- Προβλήματα σιδηροδρομικών δικτύων και οχημάτων
- Δίκτυα μεταφορών
- Συστημάτων αυτομάτου ελέγχου
- Προβλήματα Ρομποτικής



Εικ. 7.4 PSO-Particle Swarm Optimization Βελτιστοποίηση Σμήνους Σωματιδίων

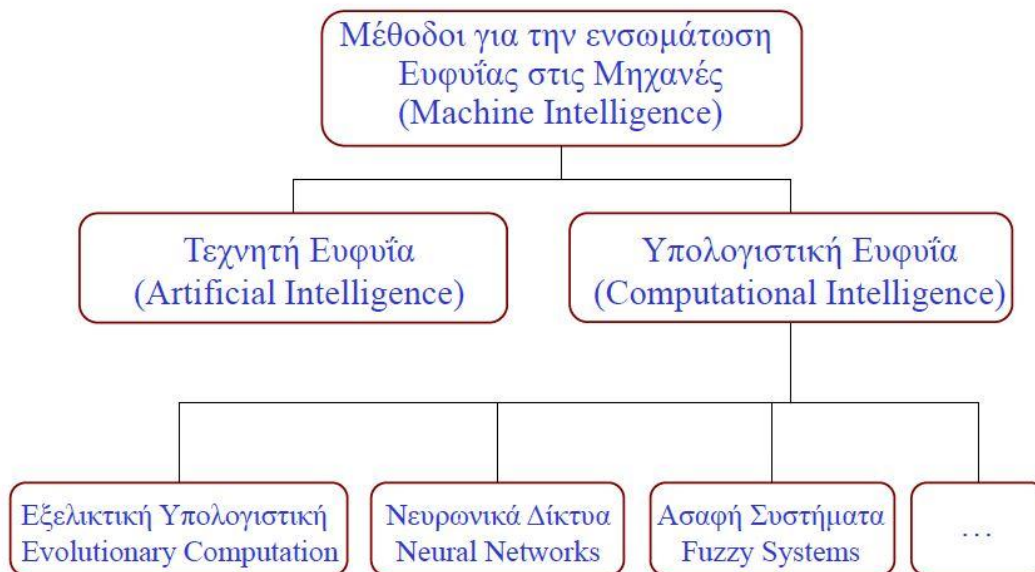


Εικ. 7.5 Η μέθοδος λειτουργίας του Αλγορίθμου PSO

Η τοπολογία του σμήνους καθορίζει το υποσύνολο των σωματιδίων με τα οποία κάθε σωματίδιο μπορεί να ανταλλάξει πληροφορίες. Η βασική έκδοση του αλγορίθμου χρησιμοποιεί την παγκόσμια τοπολογία ως δομή επικοινωνίας σμήνους. Αυτή η τοπολογία επιτρέπει σε όλα τα σωματίδια να επικοινωνούν με τα άλλα σωματίδια, έτσι ολόκληρο το σμήνος μοιράζεται την ίδια καλύτερη θέση από ένα μόνο σωματίδιο. Ωστόσο, αυτή η προσέγγιση μπορεί να οδηγήσει στην ανάπτυξη τοπικών σωματιδίων, έτσι έχουν χρησιμοποιηθεί διαφορετικές τοπολογίες για τον έλεγχο της ροής πληροφοριών μεταξύ σωματιδίων. Για παράδειγμα, στις τοπικές τοπολογίες τα σωματίδια μοιράζονται μόνο πληροφορίες με ένα υποσύνολο σωματιδίων. Αυτό το υποσύνολο μπορεί να είναι γεωμετρικό, για παράδειγμα "τα πλησιέστερα σωματίδια" - ή, πιο συχνά, ένα κοινωνικό, δηλ. ένα σύνολο σωματιδίων που δεν εξαρτάται από οποιαδήποτε απόσταση. Σε αυτές τις περιπτώσεις, η παραλλαγή PSO λέγεται ότι είναι η καλύτερη τοπική (έναντι του παγκόσμιου καλύτερου για το βασικό PSO). Υπάρχουν πολλές σχολές σκέψης για το γιατί και πώς ο αλγόριθμος PSO μπορεί να κάνει βελτιστοποίηση. Μια κοινή πεποίθηση μεταξύ των ερευνητών είναι ότι η συμπεριφορά της σαρκόσ ποικίλλει μεταξύ της διερευνητικής συμπεριφοράς, δηλαδή της αναζήτησης μιας ευρείας περιοχής του χώρου αναζήτησης και της εκμεταλλευτικής συμπεριφοράς, δηλαδή της τοπικά προσανατολισμένης αναζήτησης, ώστε να προσεγγίσει (πιθανώς τοπικά) βέλτιστη. Η σχολή σκέψεων κυριαρχείται από την αρχή του PSO. Ο αλγόριθμος PSO και οι παράμετροί του πρέπει να επιλέγονται έτσι ώστε να υπάρχει ισορροπία μεταξύ της εξερεύνησης και της εκμετάλλευσης, ώστε να αποφευχθεί η πρόωρη σύγκλιση προς την τοπική βέλτιστη αλλά και να εξασφαλιστεί ένας καλός ρυθμός σύγκλισης προς το βέλτιστο. Αυτή η πεποίθηση είναι ο πρόδρομος πολλών παραλλαγών του PSO.

Μια άλλη σχολή σκέψης είναι ότι η συμπεριφορά ενός PSO δεν είναι καλά κατανοητή από την άποψη του πώς επηρεάζει την πραγματική απόδοση, ειδικά για τους υψηλότερους σε διαστάσεις χώρους αναζήτησης και τα προβλήματα βελτιστοποίησης που μπορεί να είναι ασυνεχής, θορυβώδης και χρονικά μεταβαλλόμενος. Η Σχολή σκέψης προσπαθεί απλώς να βρει αλγόριθμους PSO και παραμέτρους που προκαλούν καλές επιδόσεις ανεξάρτητα από τον τρόπο με τον οποίο μπορεί να ερμηνευτεί η επιχείρηση σε σχέση π.χ. εξερεύνηση και εκμετάλλευση.

Ταξινόμηση



Εικ. 7.6 Ταξινόμηση των μεθόδων για την εισαγωγή ευφυΐας στις μηχανές

7.2 Επίλυση προβλημάτων ρομποτικής με τη μέθοδο της Βελτιστοποίησης Σμήνους Σωματιδίων-PSO (Particle Swarm Optimization)

Έτσι λοιπόν θα κάνουμε χρήση του PSO στο πρόβλημα, στο χώρο κίνησης του Ρομποτικού Βραχίονα έχουμε τοποθετήσουμε εμπόδια, σκοπός μας είναι η κίνηση του βραχίονα από ένα συγκεκριμένο σημείο σ' ένα άλλο (τα οποία είναι προκαθορισμένα από το χρήστη), αποφεύγοντας τα εμπόδια κι επιπλέον επιλέγοντας την βέλτιστη διαδρομή, από την αφετηρία προς το τερματισμό. Στη συνέχεια θα δημιουργήσουμε κάποια m.files:

7.2.1 Κατασκευή της συνάρτησης δημιουργίας του μοντέλου του χώρου

Η συνάρτηση αυτή καθορίζει τις τιμές των πεδίων μίας δομής (structure) που ονομάζεται "model" και θα αποτελεί global μεταβλητή του προγράμματος. Στα πεδία της δομής αυτής περιλαμβάνονται τα εξής:

- Ορίζονται οι συντεταγμένες αφετηρίας και προορισμού
- Ορίζεται το πλήθος, οι συντεταγμένες και το μέγεθος των εμποδίων, στην περίπτωση μας σφαιρικά εμπόδια.
- Ο αριθμός των σημείων καμπής της διαδρομής του ρομπότ.
- Ορίζονται τα όρια του χώρου.

Η συνάρτηση θα ονομάστεί «**CreateModel**» και θα αποθηκευτεί στον φάκελο εργασίας σε ένα αρχείο με όνομα «**CreateModel.m**»

```
function model>CreateModel()
```

```
% Αφετηρία  
xs=180; ys=205;
```

```
% Τερματισμός  
xt= -180; yt=205;
```

```
% Συντεταγμένες κυκλικών εμποδίων  
xobs=[65.0 -65.0]  
yobs=[236.0 191.0]
```

```
% Ακτίνες κυκλικών εμποδίων  
robs=[26.0 20.0]
```

```
% Αριθμός σημείων καμπής διαδρομής  
n=3;
```

```
% Όρια του x  
xmin= -280; xmax= 280;
```

```
% Όρια του y
```

ymin = 0 ; ymax= 250;

% Κατασκευή της δομής (structure) "model"

**model.xs=xs;
model.ys=ys;**

**model.xt=xt;
model.yt=yt;**

**model.xobs=xobs;
model.yobs=yobs;**

**model.robs=robs;
model.n=n;**

**model.xmin=xmin;
model.xmax=xmax;**

**model.ymin=ymin;
model.ymax=ymax;**

end

7.2.2 Κατασκευή της συνάρτησης για ανάπτυξη και έλεγχο της εκάστοτε λύσης

Η συνάρτηση αυτή αναπτύσσει την διαδρομή του ρομπότ με βάση τις συντεταγμένες των σημείων καμπής, υπολογίζοντας 100 ενδιάμεσα σημεία με γραμμική παρεμβολή. Στη συνέχεια υπολογίζει την συνολική απόσταση της διαδρομής. Κατόπιν υπολογίζει αν κάποιο σημείο της διαδρομής «τέμνει» κάποιο από τα σφαιρικά εμπόδια. Αν ναι, τότε υπολογίζεται μία τιμή ποινής που προστίθεται στην απόσταση της διαδρομής, χειροτερεύοντας έτσι την ποιότητα της λύσης. Τέλος η συνάρτηση κατασκευάζει μία δομή «sol» με τα στοιχεία της διαδρομής, την οποία και επιστρέφει.

Η συνάρτηση θα ονομάζεται «ParseSolution» και θα αποθηκευτεί στον φάκελο εργασίας σε ένα αρχείο με όνομα «ParseSolution.m»

```
function sol=ParseSolution(position,model)
```

```
% Κατασκευή ανυσμάτων συντεταγμένων ολόκληρης της διαδρομής
```

```
XS=[model.xs position.x model.xt];
```

```
YS=[model.ys position.y model.yt];
```

```
%ZS=[model.zs position.z model.zt];
```

```
% Κατασκευή της διαδρομής με 100 σημεία [xx,yy,zz] με γραμμική παρεμβολή
```

```
k=numel(XS);
```

```
TS=linspace(0,1,k);
```

```
tt=linspace(0,1,100);
```

```
xx=interp1(TS,XS,tt);
```

```
yy=interp1(TS,YS,tt);
```

```
%zz=interp1(TS,ZS,tt);
```

```
% Υπολογισμός των αποστάσεων μεταξύ των σημείων
```

```
dx=diff(xx);
```

```
dy=diff(yy);
```

```
%dz=diff(zz);
```

```
% Υπολογισμός της συνολικής απόστασης της διαδρομής
```

```
%L=sum(sqrt(dx.^2+dy.^2+dz.^2));
```

```
L=sum(sqrt(dx.^2+dy.^2));
```

```
% Έλεγχος επαφής με τα εμπόδια
```

```
nobs = numel(model.xobs); % Αριθμός εμποδίων
```

```
Violation = 0; % ποινή επαφής με εμπόδια
```

```
for k=1:nobs % βρόχος για όλα τα εμπόδια
```

```
% Υπολογισμός αποστάσεων των σημείων της διαδρομής από το κέντρο του εμποδίου
```

```
%d=sqrt((xx-model.xobs(k)).^2+(yy-model.yobs(k)).^2 + (zz-model.zobs(k)).^2);
```

```
d=sqrt((xx-model.xobs(k)).^2+(yy-model.yobs(k)).^2 );
```

```
% Έλεγχος για επαφή με το εμπόδιο
```

```
v=max(1-d/model.robots(k),0);
```

```
% Καθορισμός ποινής
```

```
Violation=Violation+mean(v);
```

```
End
```

```

% Κατασκευή της δομής sol που επιστρέφει η συνάρτηση
sol.TS=TS; % k σημεία μεταξύ 0 και 1
sol.XS=XS; % x συντεταγμένες των σημείων αφετηρίας,καμπής και
τερματισμού
sol.YS=YS; % y συντεταγμένες των σημείων αφετηρίας,καμπής και
τερματισμού
%sol.ZS=ZS; % z συντεταγμένες των σημείων αφετηρίας,καμπής και
τερματισμού
sol.tt=tt; % 100 σημεία μεταξύ 0 και 1
sol.xx=xx; % x συντεταγμένες 100 σημείων της διαδρομής
sol.yy=yy; % y συντεταγμένες 100 σημείων της διαδρομής
%sol.zz=zz; % z συντεταγμένες 100 σημείων της διαδρομής
sol.dx=dx; % x αποστάσεις μεταξύ των 100 σημείων της διαδρομής
sol.dy=dy; % y αποστάσεις μεταξύ των 100 σημείων της διαδρομής
%sol.dz=dz; % z αποστάσεις μεταξύ των 100 σημείων της διαδρομής
sol.L=L; % το μήκος της διαδρομής
sol.Violation=Violation; % η ποινή για επαφή με εμπόδια
sol.IsFeasible=(Violation==0); % boolean πεδίο για επαφή με εμπόδια
end

```

7.2.3 Κατασκευή της συνάρτησης ποιότητας για τις λύσεις του προβλήματος

Η συνάρτηση αυτή υπολογίζει την ποιότητα της κάθε λύσης που παράγει ο αλγόριθμος PSO. Αρχικά ανακτά τις συντεταγμένες των 3 σημείων καμπής της διαδρομής. Στη συνέχεια καλεί την συνάρτηση «ParseSolution» για να αναπτύξει την εκάστοτε λύση και να την ελέγξει για επαφή με τα εμπόδια. Στη συνέχεια υπολογίζει την ποιότητα της λύσης λαμβάνοντας υπ' όψιν για τυχόν ποινές για επαφή με εμπόδια και ενημερώνει την βέλτιστη μέχρι στιγμής λύση «GlobalBest». Η συνάρτηση θα ονομάζεται «PathPlanning» και θα αποθηκευτεί στον φάκελο εργασίας σε ένα αρχείο με όνομα «PathPlanning.m».

```

function scores=PathPlanning(pop)

global model GlobalBest;
% Ανάκτηση των συντεταγμένων των σημείων καμπής
position.x=pop(1:model.n);
position.y=pop(model.n+1:2*model.n);

% Κατασκευή της διαδρομής και έλεγχος αποφυγής εμποδίων
sol=ParseSolution(position,model);

% Υπολογισμός της ποιότητας της λύσης
beta=100; % συντελεστής ποινής για επαφή με εμπόδια

scores=sol.L*(1+beta*sol.Violation);
% Ενημέρωση του GlobalBest
if scores<GlobalBest.Cost

```

```

GlobalBest.Cost=scores;
GlobalBest.Sol=sol;
GlobalBest.Position=position;
end
end

```

7.2.4 Κατασκευή της συνάρτησης που θα σχεδιάζει τις λύσεις του προβλήματος

Η συνάρτηση αυτή θα σχεδιάζει τις θέσεις των εμποδίων της δομής “model” και κατόπιν θα σχεδιάζει την διαδρομή που θα εκφράσει η καλύτερη μέχρι στιγμής λύση που βρίσκεται στην δομή “GlobalBest”.

Η συνάρτηση ονομάζεται «PathPlot» και αποθηκεύτηκε στον φάκελο εργασίας σε ένα αρχείο με όνομα «PathPlot.m»

```

function stop = PathPlot(optimValues,state)
% δήλωση των global μεταβλητών
global GlobalBest model
stop = false;
% παραγωγή 100 τιμών από 0 έως 2π
theta=linspace( 0 , 2*pi , 100 );
% Εμφάνιση των κυκλικών εμποδίων
for k=1 : numel(model.xobs)
    fill(model.xobs(k)+model.robs(k)*cos(theta),...
        model.yobs(k)+model.robs(k)*sin(theta),'r');
    hold on;
end
% Εμφάνιση της διαδρομής
plot(GlobalBest.Sol.xx,GlobalBest.Sol.yy,'k','LineWidth',2);
% Εμφάνιση των σημείων καμπής
plot(GlobalBest.Sol.XS,GlobalBest.Sol.YS,'ro');
% Εμφάνιση της αφητηρίας
plot(model.xs,model.ys,'bs','MarkerSize',12,'MarkerFaceColor','y');
% Εμφάνιση του προορισμού
plot(model.xt,model.yt,'kp','MarkerSize',16,'MarkerFaceColor','g');
hold off;
grid on;
axis equal;

end

```

7.2.5 Κατασκευή του κυρίως προγράμματος για Βέλτιστη σχεδίαση Διαδρομής με χρήση του αλγορίθμου PSO

Η συνάρτηση αυτή θα ορίζει αρχικά τις global μεταβλητές του προγράμματος, και θα δημιουργεί αρχικά το μοντέλο του χώρου κίνησης του ρομπότ και των εμποδίων που βρίσκονται μέσα σε αυτόν. Στη συνέχεια αρχικοποιεί την γεννήτρια τυχαίων αριθμών και τα options «PlotFcn» και «Display» του PSO. Κατόπιν καθορίζει τα άνω και κάτω όρια των μεταβλητών του προβλήματος και στη συνέχεια εκτελεί τον αλγόριθμο PSO. Το πρόγραμμα αποθηκεύτηκε στον φάκελο εργασίας σε ένα αρχείο με όνομα «PSO.m»

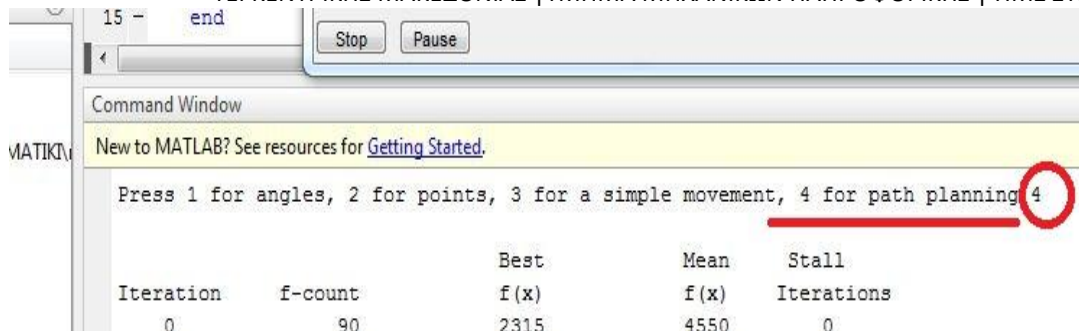
```
% Πρόγραμμα επίλυσης προβλήματος σχεδιασμού διαδρομής
% Global Μεταβλητές
global GlobalBest model;
% Αρχικοποίηση της γεννήτριας τυχαίων αριθμών
GlobalBest.Cost=inf;
GlobalBest.Position=[];
GlobalBest.Sol=[];
%rng default % για αναπαραγωγή των αποτελεσμάτων
rng('shuffle'); % για τυχειότητα
% Καθορισμός options: "PlotFcn" και "Display"
%options = optimoptions(@particleswarm,'PlotFcn',{@pswplotbestf,@PathPlot},...
%'Display','iter');
options =
optimoptions(@particleswarm,'PlotFcn',{@pswplotbestf,@PathPlot},'Display','iter',
'TolFun',0.05,'SwarmSize',500);
% Καθορισμός άνω και κάτω ορίων των μεταβλητών
%lb= [ ones(1,model.n)*model.xmin ones(1,model.n)*model.ymin
ones(1,model.n)*model.zmin];
%ub= [ ones(1,model.n)*model.xmax ones(1,model.n)*model.ymax
ones(1,model.n)*model.zmax];
lb= [ ones(1,model.n)*model.xmin ones(1,model.n)*model.ymin];
ub= [ ones(1,model.n)*model.xmax ones(1,model.n)*model.ymax];
% Εκτέλεση του PSO
[x,fval,exitflag,output] = particleswarm(@PathPlanning,2*model.n,lb,ub,options);
```



Η πιο ισχυρή «μηχανή» αναζήτησης λύσεων , σύλληψης νέων ιδεών , θεωριών και εφευρέσεων σε ολόκληρο το σύμπαν είναι ο ανθρώπινος εγκέφαλος.

7.3 Κίνηση στο χώρο και αποφυγή εμποδίων με τη χρήση του αλγορίθμου PSO (mode-4)

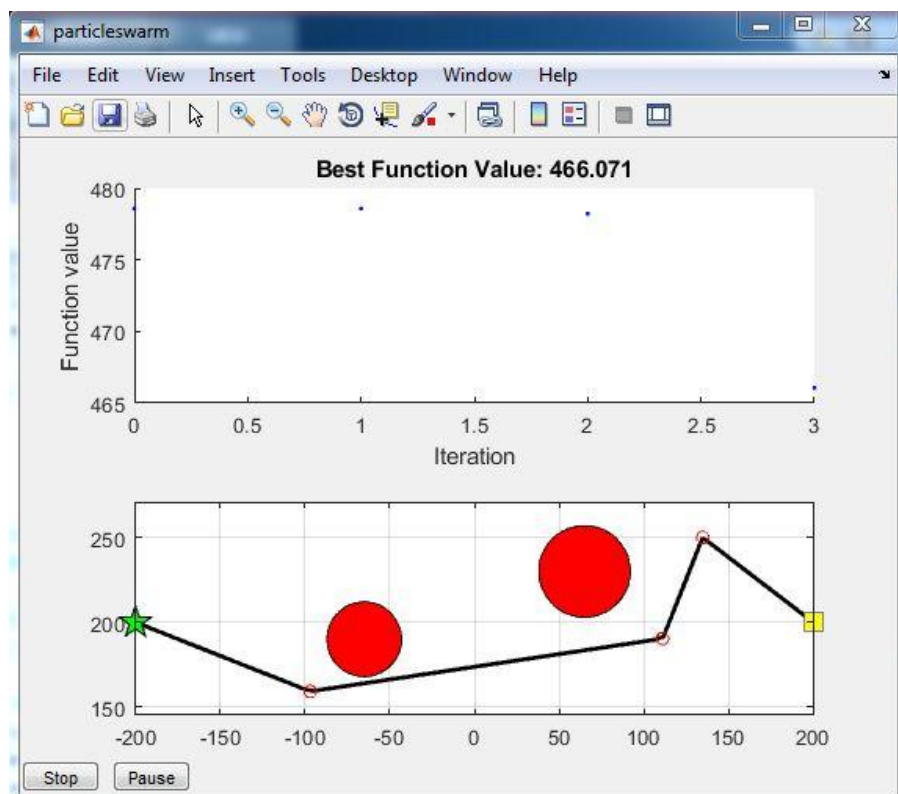
Στο κεφάλαιο αυτό λοιπόν θα ασχοληθούμε με την αποφυγή εμποδίων στο χώρο, έχοντας λοιπόν δημιουργήσει στα προηγούμενα κεφάλαια όλα τα απαραίτητα m.files , θα πάμε λοιπόν να κάνουμε χρήση του αλγορίθμου PSO , έχοντας τη δυνατότητα παρακολούθησης της πορείας του μέσω Matlab , αλλά και στην πραγματική κίνηση του βραχίονα στο χώρο. Αρχικά τρέχουμε στο περιβάλλον του Arduino τον κώδικα του αρχείου Dummy για την οδήγηση των servo , ο οποίος είναι ο ίδιος με τα προηγούμενα κεφάλαια, βάζουμε στο ίδιο path με το **main.m** και το **inverse_kinematic.m** , τα ακόλουθα: **CreateModel.m** , **ParseSolution.m** , **PathPlanning.m** , **PathPlot.m** και **PSO.m** Σκοπός του προβλήματος είναι ο βέλτιστος σχεδιασμός διαδρομής του άκρου του βραχίονα στο χώρο με την ταυτόχρονη αποφυγή εμποδίων (Robot arm path planning and obstacle avoidance). Έτσι λοιπόν η διαδικασία συνίσταται στην εύρεση της βέλτιστης διαδρομής του άκρου το οποίο κινείται σε προκαθορισμένο χώρο με προκαθορισμένα εμπόδια , από ένα σημείο αφετηρίας σε ένα άλλο σημείο τερματισμού , που είναι επίσης προκαθορισμένα. Έτσι ώστε να αποφύγει τα εμπόδια και να φτάσει στον τερματισμό , διανύοντας την ελάχιστη δυνατή διαδρομή. Τα σμήνη σωματιδίων περιφέρονται στο χώρο κίνησης και ανταλλάσσουν πληροφορίες μεταξύ τους , οι γειτονιές με τις υπόλοιπες κι έτσι ο αλγόριθμος καταλήγει στην εύρεση της βέλτιστης λύσης. Η διαδρομή αποτελείται από ευθύγραμμα τμήματα με n σημεία καμπής, όπου το n έχει αρχική τιμή n=3 αλλά είναι παραμετρικό και μπορεί να αλλάξει κατά βούληση. Στη παρούσα εργασία θα έχουμε τη δυνατότητα να παρακολουθήσουμε επίλυση με διάφορες παραλλαγές ως προς τα σημεία καμπής. Ο χώρος περιλαμβάνει 2 κυκλικά εμπόδια, που παρεμβάλλονται μεταξύ αφετηρίας και προορισμού. Η θέση και το μέγεθος των εμποδίων επίσης είναι παραμετρικό και μπορεί να αλλάξει κατά βούληση. Τρέχουμε το αρχείο **main.m** κι όταν ερωτηθούμε με ποιά επιλογή επιθυμούμε να κινήσουμε το βραχίονα , εμείς πληκτρολογούμε το **4** για **path planning**. (βλέπε εικ 7.7) αμέσως βλέπουμε ότι ο Αλγόριθμος PSO έχει ξεκινήσει την διαδικασία επίλυσης του προβλήματος που του έχουμε αναθέσει.



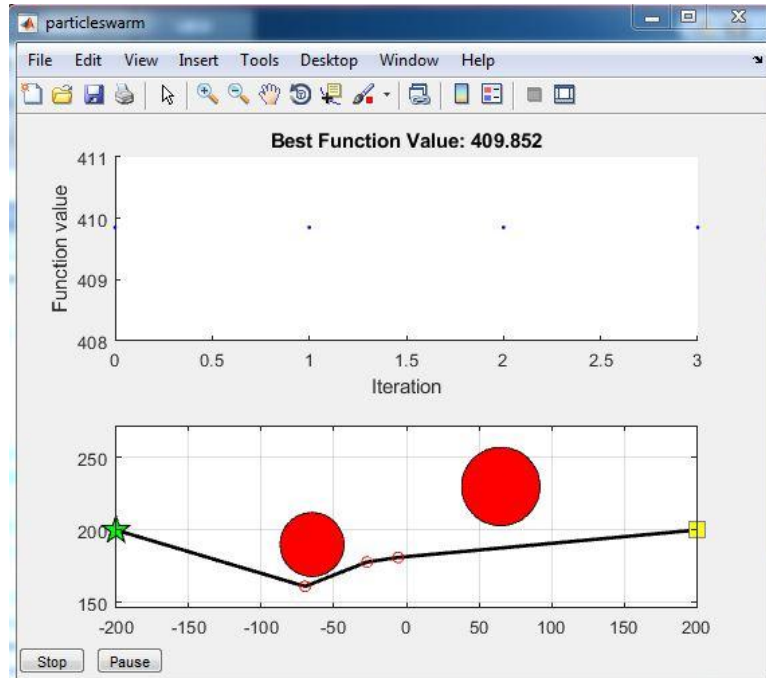
Εικ. 7.7 Κίνηση με την 4^η επιλογή Path Plannig

7.3.1 Επίλυση με 3 σημεία καμπής

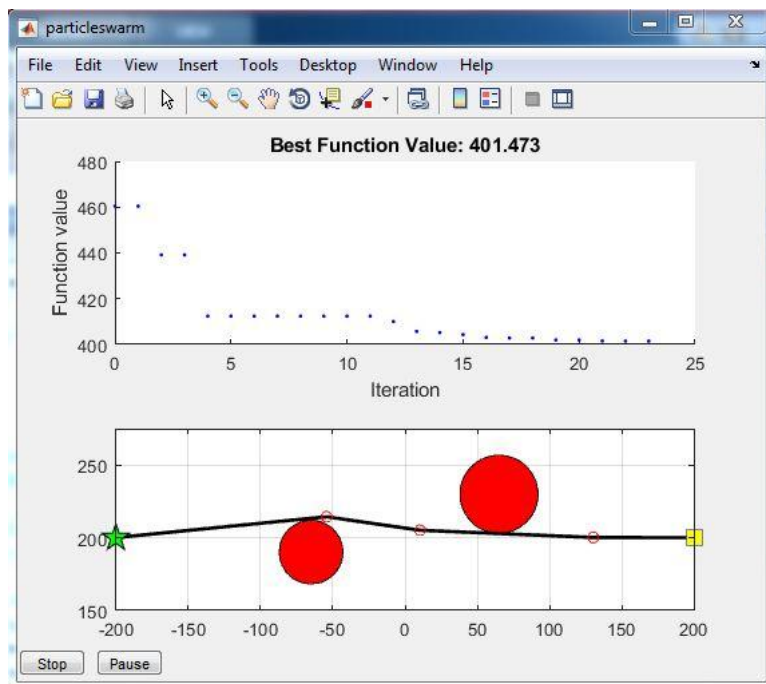
Έτσι λοιπόν ξεκινάει ο PSO για την εύρεση της βέλτιστης λύσης. Στις παρακάτω εικόνες μας δίνεται η δυνατότητα να παρακολουθήσουμε τα διάφορα στάδια αναζήτησης, με 3 σημεία καμπής (n=3).



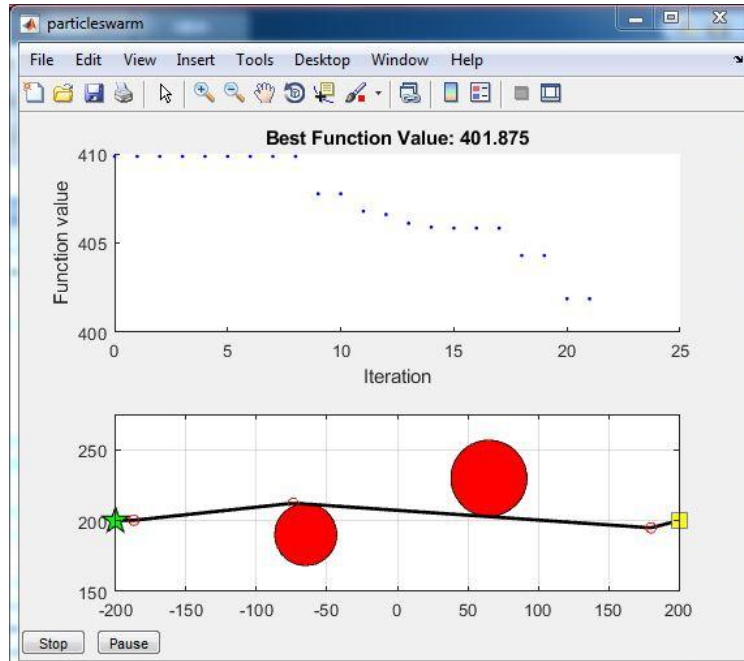
Εικ. 7.8 Στάδιο αναζήτησης βέλτιστης λύσης



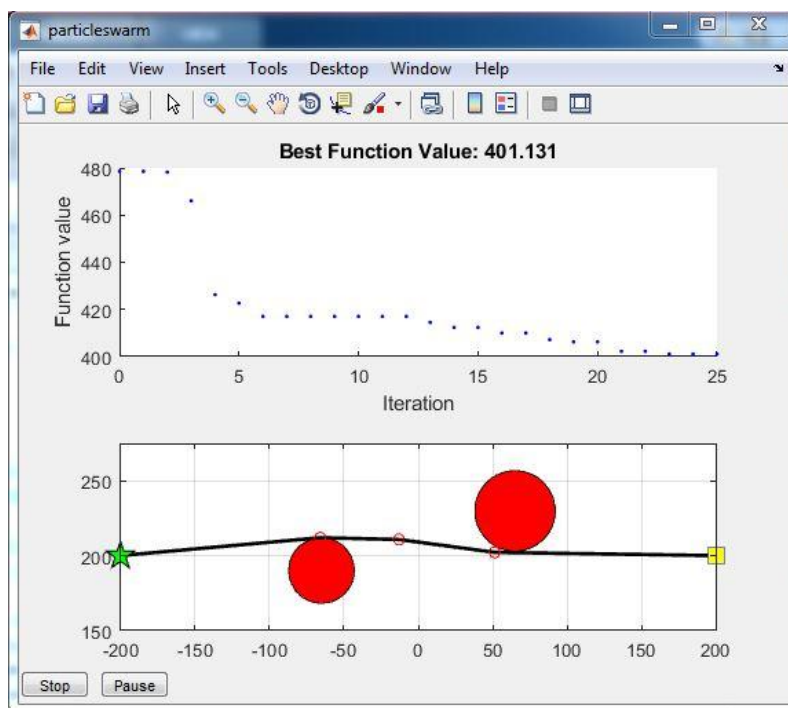
Εικ. 7.9 Στάδιο αναζήτησης βέλτιστης λύσης



Εικ. 7.10 Εύρεση της βέλτιστης λύσης 1^η φορά (Run)



Εικ. 7.11 Εύρεση της βέλτιστης λύσης 2^η φορά(Run)



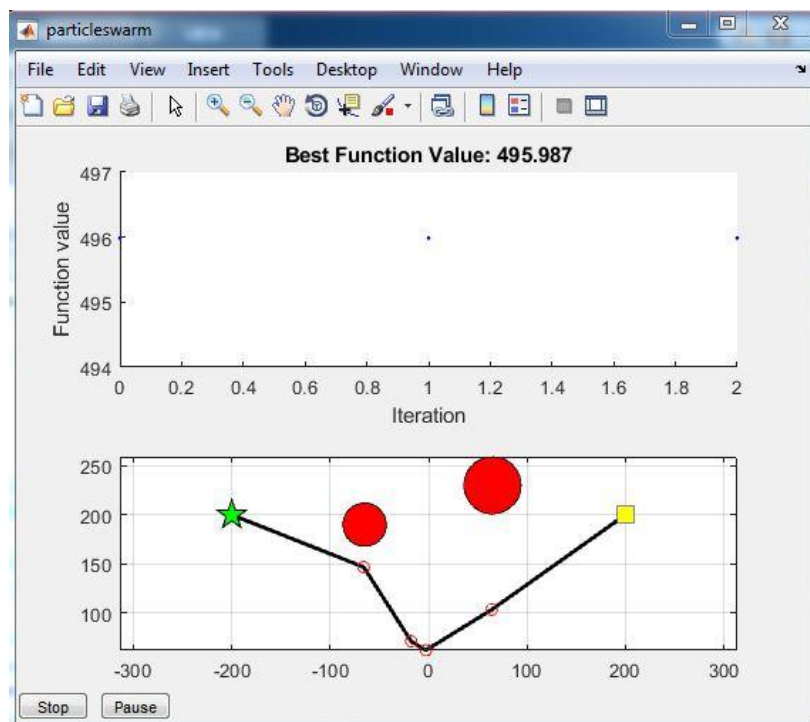
Εικ. 7.12 Εύρεση της βέλτιστης λύσης 3^η φορά(Run)

Παρατηρήσεις αποτελέσματος επίλυσης με 3 σημεία καμπής

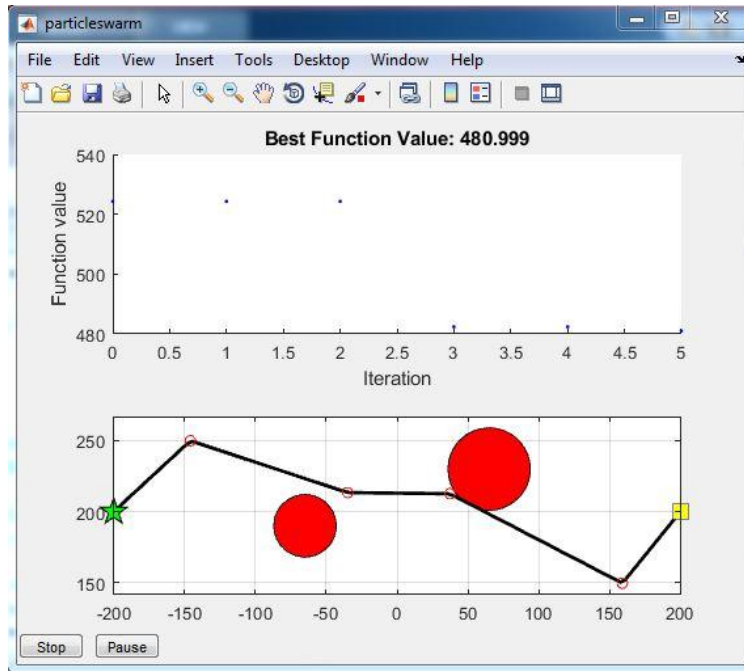
Η μέθοδος PSO βλέπουμε ότι κάθε φορά ξεκινώντας για την εύρεση της βέλτιστης διαδρομής, τοποθετεί αρχικά τα σημεία καμπής σε κάποια τυχαία σημεία αρκετά μακριά από τα εμπόδια, έπειτα μετά από διάφορες μικρο-βελτιώσεις τα σημεία καμπής τοποθετούνται αρκετά κοντά στα εμπόδια, έτσι επέρχεται η βέλτιστη λύση μέσα από αρκετές χιλιάδες λύσεις με αποφυγή των εμποδίων και όπως παρατηρούμε στις εικόνες 7.10, 7.11 και 7.12 οι λύσεις είναι πολύ κοντά ή μια στην άλλη, αλλά με διαφορετική τοποθέτηση των σημείων καμπής.

7.3.2 Επίλυση με 4 σημεία καμπής

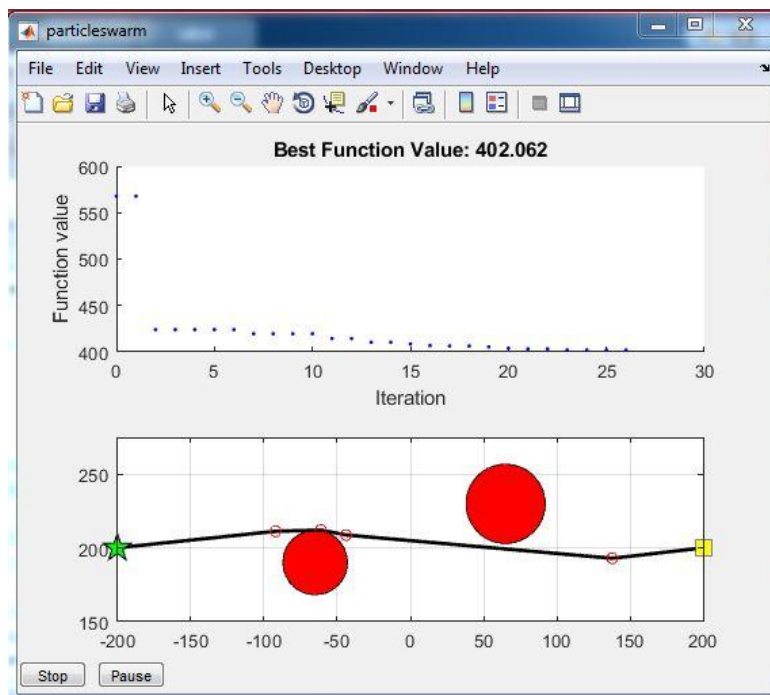
Έπειτα προχωρούμε στην επίλυση της βέλτιστης διαδρομής αυτή τη φορά με 4 σημεία καμπής. (βλέπε παρακάτω εικόνες 7.13-7.17)



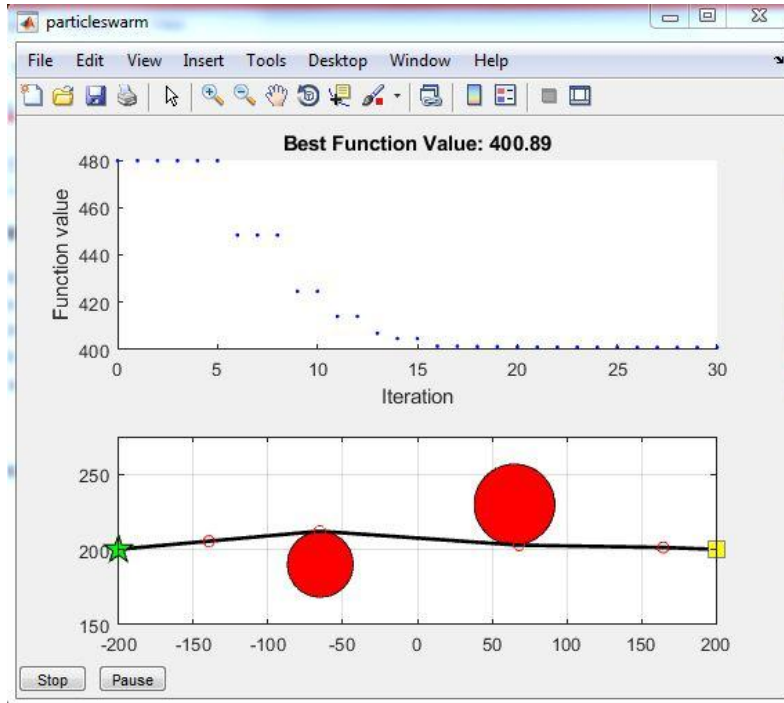
Εικ. 7.13 Στάδιο αναζήτησης βέλτιστης λύσης



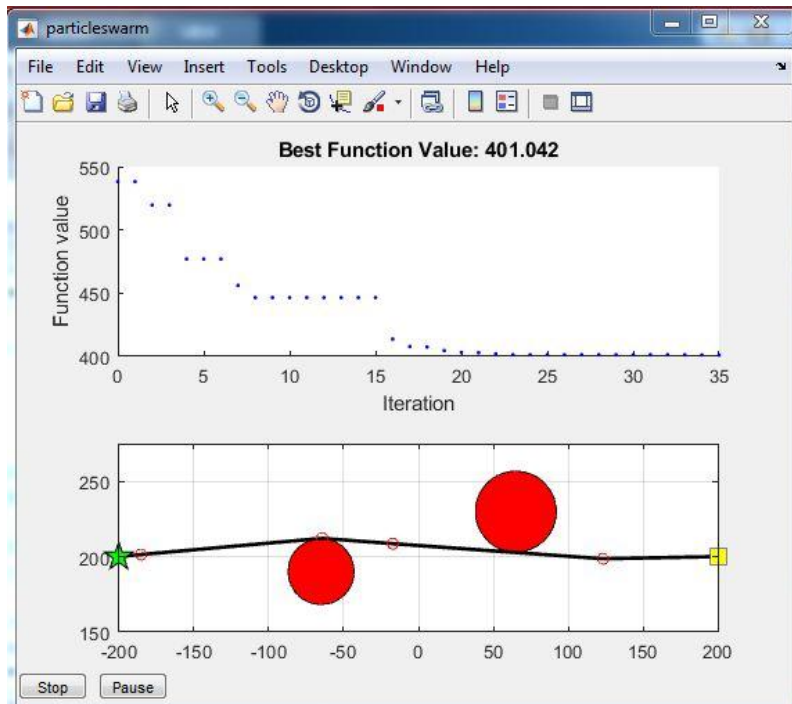
Εικ. 7.14 Στάδιο αναζήτησης βέλτιστης λύσης



Εικ. 7.15 Εύρεσης της βέλτιστης λύσης 1^η φορά(Run)



Εικ. 7.16 Εύρεση της βέλτιστης λύσης 2^η φορά(Run)



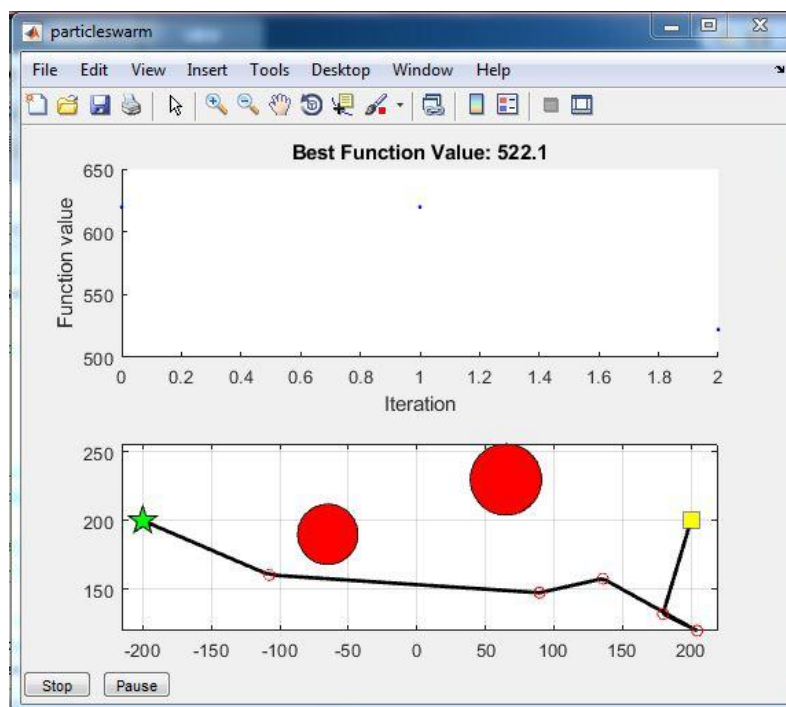
Εικ. 7.17 Εύρεση της βέλτιστης λύσης 3^η φορά(Run)

Παρατηρήσεις αποτελέσματος επίλυσης με 4 σημεία καμπής

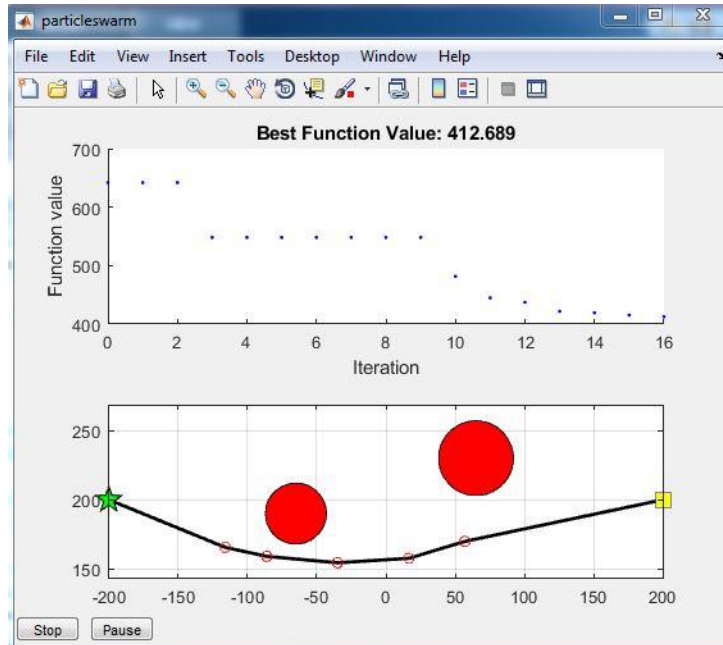
Σε αυτό το σημείο παρατηρούμε πως οι πιθανές λύσεις είναι πιο κοντά η μια στην άλλη κάθε φορά που τρέχουμε τον αλγόριθμο βελτιστοποίησης, κάθε φορά όμως έχουμε διαφορετική επίλυση, με διαφορετική τοποθέτηση των σημείων καμπής. Στην περίπτωση αυτή με 4 σημεία καμπής έχουμε ελαφρώς αυξημένη καθυστέρηση για την εύρεση της βέλτιστης λύσης.

7.3.3 Επίλυση με 5 σημεία καμπής

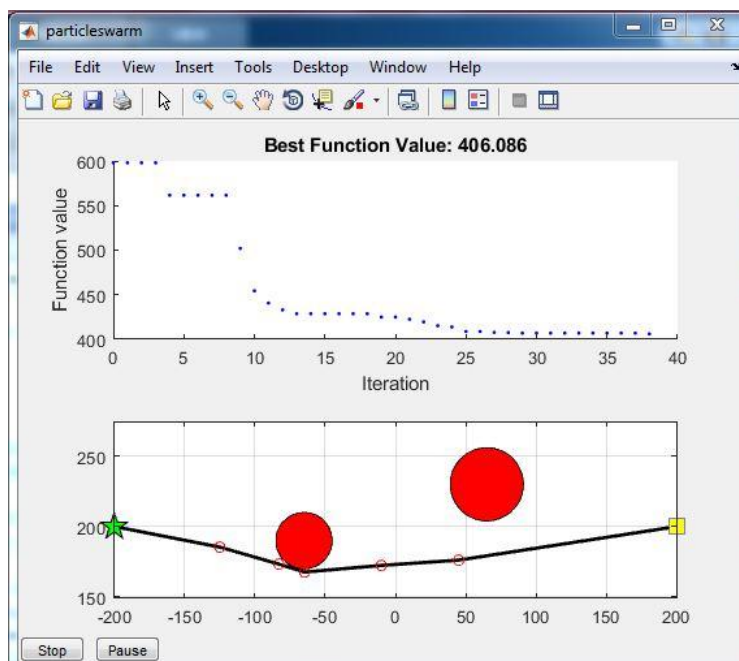
Στην προσπάθεια επίλυσης με 5 σημεία καμπής, θα προσπαθήσουμε να συγκρίνουμε την ποιότητα της λύσης με την χρονική διάρκεια που απαιτείται για την επίλυση, διότι αυξάνεται η πολυπλοκότητα.



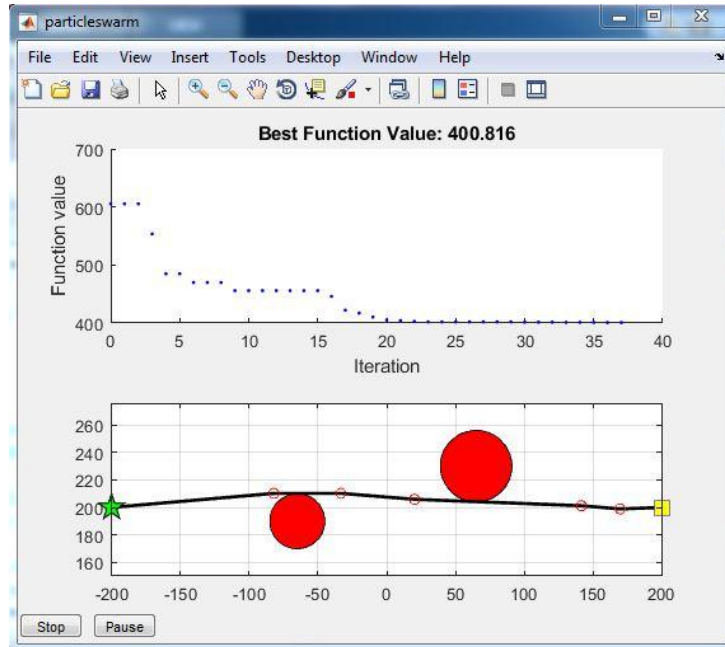
Εικ. 7.18 Στάδιο αναζήτησης βέλτιστης λύσης



Εικ. 7.19 Στάδιο αναζήτησης βέλτιστης λύσης



Εικ. 7.20 Εύρεση της βέλτιστης λύσης 1η φορά(Run)



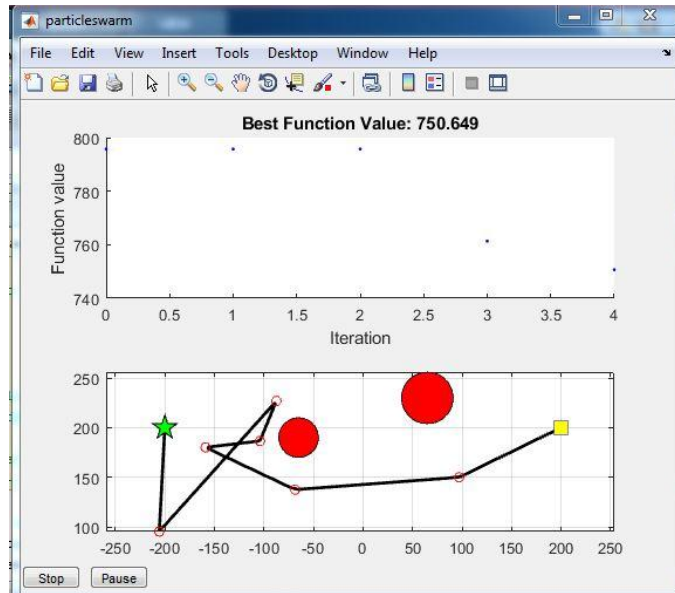
Εικ. 7.21 Εύρεση της βέλτιστης λύσης 2η φορά(Run)

Επίσης , ενώ τρέχει ο PSO από την αφετηρία μέχρι και τον τερματισμό, έχουμε τη δυνατότητα στο command window του Matlab να βλέπουμε τις τιμές που παίρνουν οι σερβο-κινητήρες σε κάθε σημείο.

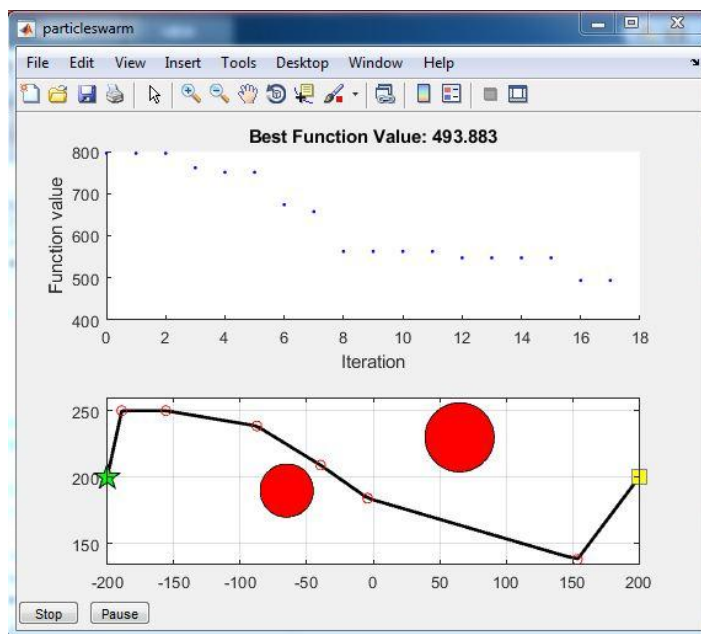


Εικ. 7.22 command window

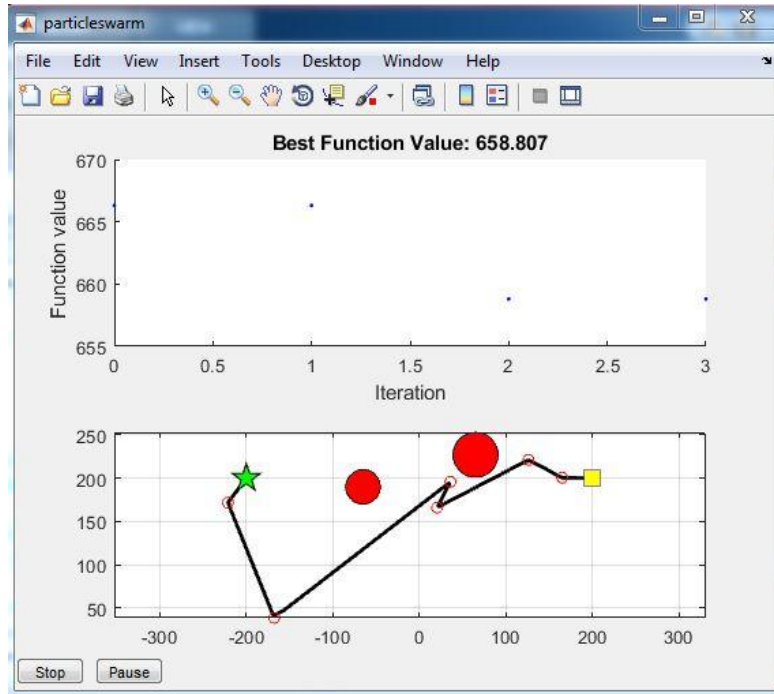
7.3.4 Επίλυση με 6 σημεία καρπής



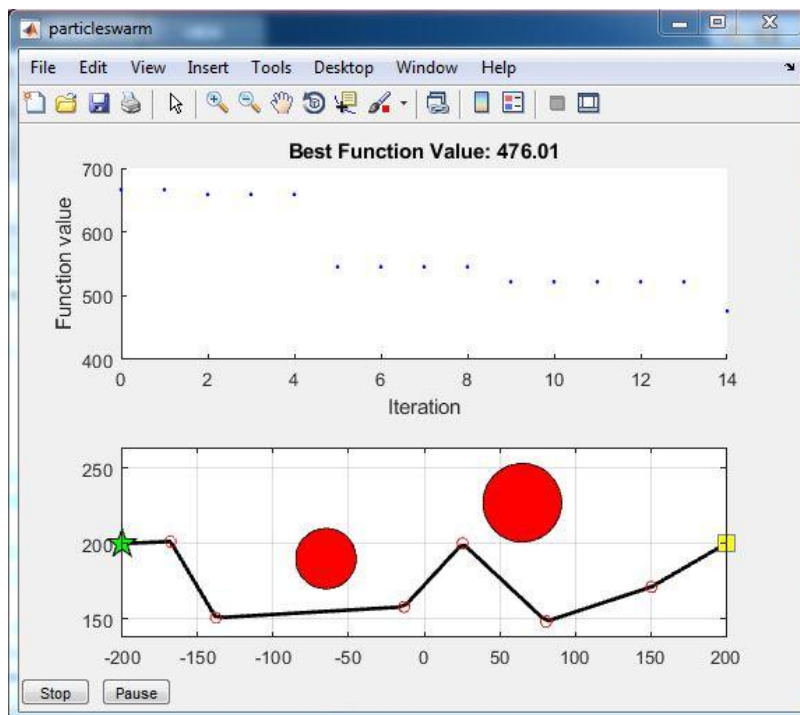
Εικ. 7.23 Στάδιο αναζήτησης βέλτιστης λύσης



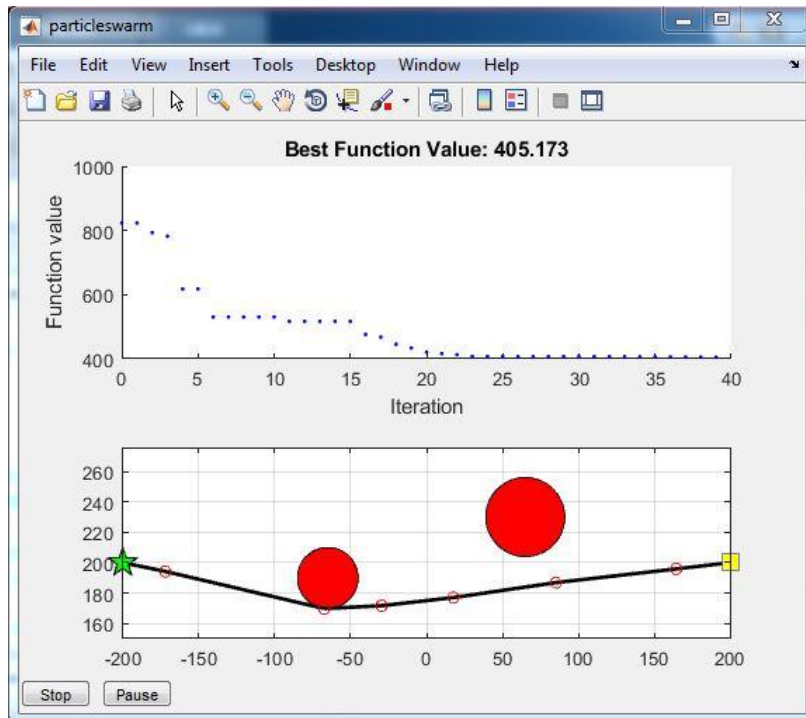
Εικ. 7.24 Στάδιο αναζήτησης βέλτιστης λύσης



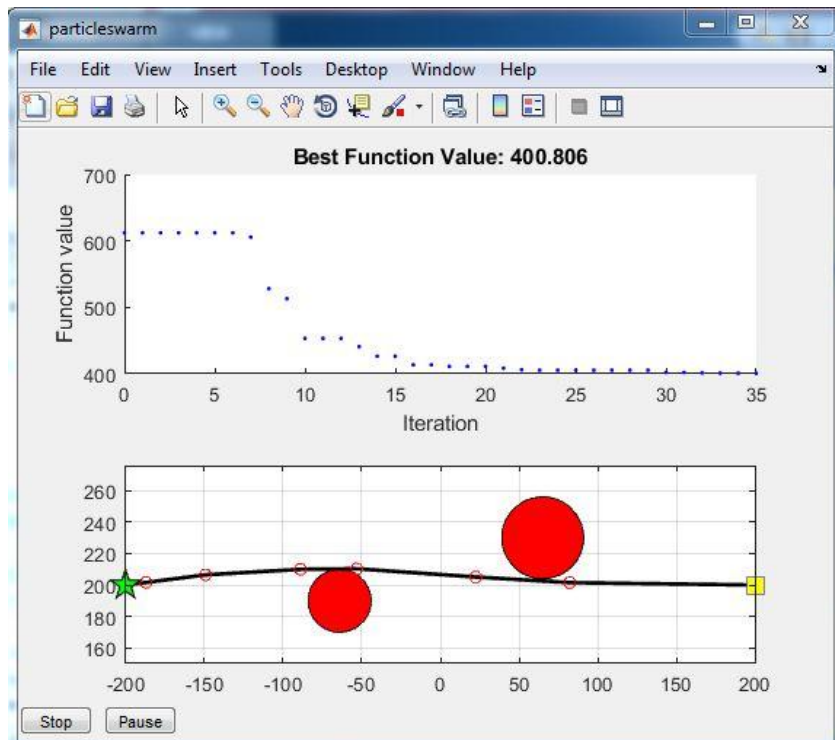
Εικ. 7.25 Στάδιο αναζήτησης βέλτιστης λύσης



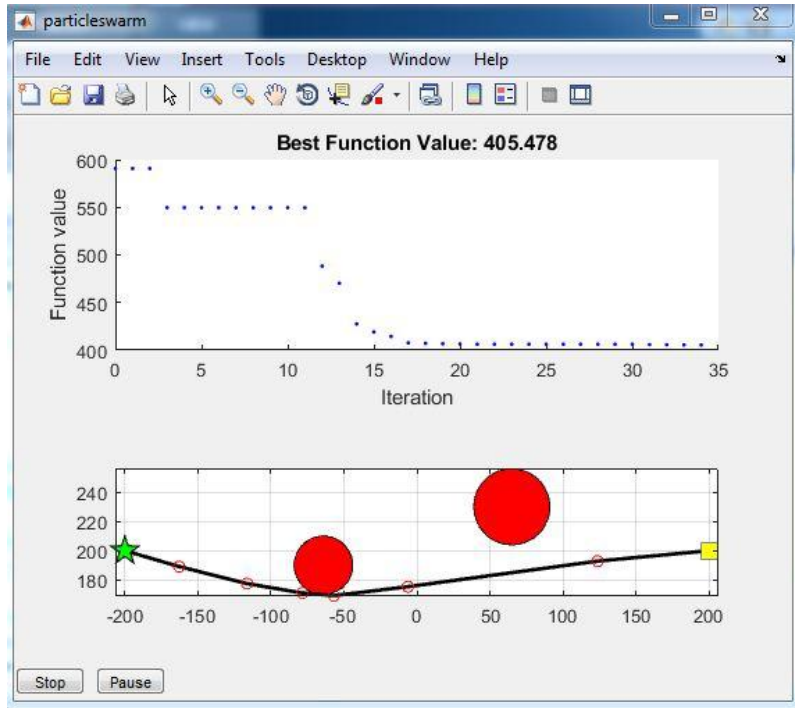
Εικ. 7.26 Στάδιο αναζήτησης βέλτιστης λύσης



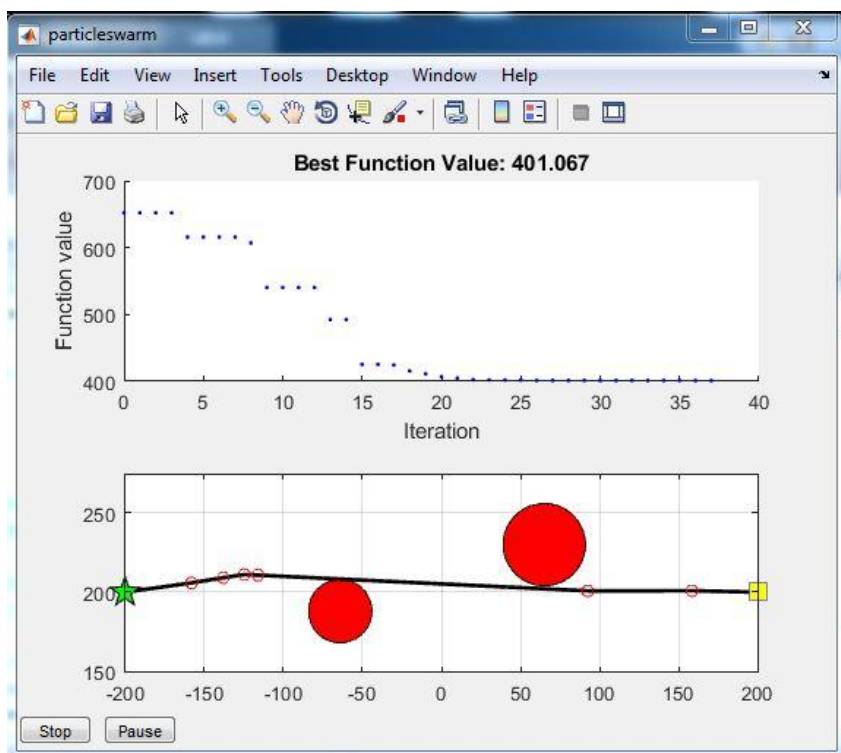
Εικ. 7.27 Εύρεση βέλτιστης λύσης 1^η φορά(Run)



Εικ. 7.28 Εύρεση βέλτιστης λύσης 2^η φορά(Run)



Εικ. 7.29 Εύρεση βέλτιστης λύσης 3^η φορά(Run)



Εικ. 7.30 Εύρεση βέλτιστης λύσης 4^η φορά(Run)

Παρατηρήσεις αποτελέσματος επίλυσης με 6 σημεία καμπής

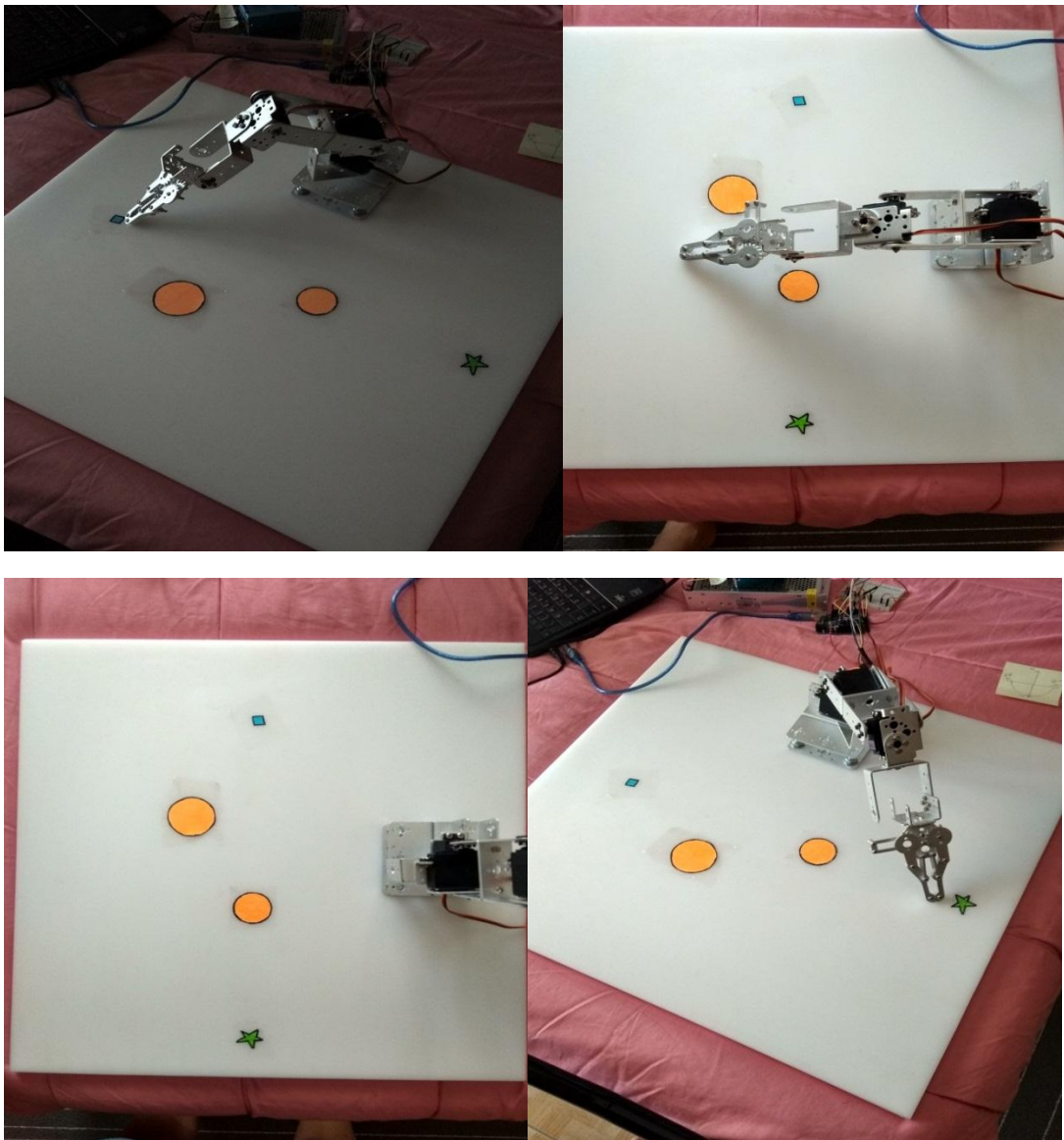
Το αποτέλεσμα της επίλυσης με 6 σημεία καμπής (δηλ. $n=6$), επιτρέπει στον PSO να δώσει μια καλύτερη λύση σε σχέση με τις προηγούμενες, όσο αναφορά την βελτίωση της απόστασης που ενώνει τη αφετηρία με τον προορισμό, βέβαια όμως επηρεάζει άμεσα το χρόνο που χρειάζεται ο αλγόριθμος για να δώσει την τελική λύση, καθυστερεί αρκετά και μερικές φορές είναι πολύ πιθανό να φτάσει σε πρόωρη λύση η οποία δεν είναι η βέλτιστη. Επίσης παρατηρούμε ότι τις περισσότερες φορές τα σημεία καμπής τοποθετούνται πολύ κοντά στην περιμέτρο των εμποδίων, κι αυτό βοηθάει πολύ στην αναζήτηση, το σχεδιασμό και τελικά την εύρεση της βέλτιστης διαδρομής, με την ταυτόχρονη αποφυγή τους, που είναι και ο απώτερος σκοπός.



Εικ. 7.31 Μέθοδος PSO

7.3.5 Τοποθέτηση πραγματικών σημείων στο χώρο

Θεωρήθηκε αναγκαία η τοποθέτηση πραγματικών εμποδίων στο χώρο κίνησης του βραχίονα ώστε να μπορούμε να παρακολουθήσουμε ζωντανά το σχεδιασμό διαδρομής, όπως επίσης και το σημείο αφετηρίας και τερματισμού. Στις παρακάτω εικόνες 7.32 και 7.35 , έχουμε τη δυνατότητα να δούμε όσα περιγράφηκαν παραπάνω.



Εικόνες 7.32,7.33,7.34,7.35 Τοποθέτηση των κυκλικών εμποδίων στο χώρο όπως και αφετηρίας και τερματισμού

Κεφάλαιο 8

Εταιρίες κατασκευής Ρομποτικών Βραχιόνων-Λογότυπα



Κεφάλαιο 9

Μελλοντικές Επεκτάσεις- Συμπεράσματα

9.1 Μαθήματα που χρησιμοποιήθηκαν από το πρόγραμμα σπουδών του ΠΜΣ

Για την ολοκλήρωση της διπλωματικής εργασίας ήταν απαραίτητη η κατανόηση και χρήση των παρακάτω μαθημάτων , των οποίων οι διαλέξεις πραγματοποιήθηκαν στη διάρκεια των 2 εξαμήνων του ΠΜΣ της Ρομποτικής του τμήματος Μηχανικών Πληροφορικής του ΤΕΙ κεντρικής μακεδονίας.Αυτά είναι:

- **Ενσωματωμένα συστήματα** με διδάσκοντες τους: Dr.Καζαρή Σπυρίδων και Dr.Καλόμοιρο Ιωάννη, για τα ηλεκτρονικά μέρη την τροφοδοσία, τη χρήση του arduino και τον έλεγχο των servo.
- **Εισαγωγή στη Ρομποτική** με διδάσκοντες τους: Dr. Μπαλουκτσή Αναστάσιο , Dr. Φαχαντίδη Νικόλαο και Dr. Τσαγκάρη Απόστολο, για την κινηματική ανάλυση.
- **Σχεδίαση και προσομείωση Ρομποτικών συστημάτων** βασισμένη στο ROS-Robotic Operation system με διδάσκοντα τον Dr. Σταύρο Βολογιαννίδη, για την κινηματική ανάλυση και προσομείωση.
- **Αυτόνομα Ρομποτικά οχήματα** με διδάσκοντες τους: Dr.Καζαρή Σπυρίδων Dr.Καλόμοιρο Ιωάννη και Dr. Θεόδωρο Παχίδη ,για την κατασκευή αλγορίθμου αποφυγής εμποδίων και εύρεση βέλτιστης διαδρομής , για την εισαγωγή ευφυΐας.
- **Μηχανική ευφυΐα** με διδάσκοντες τους: Dr.Καζαρή Σπυρίδων Dr.Βολογιαννίδη Σταύρο και Dr. Αντωνίου Ευστάθιο , για την κατασκευή αλγορίθμου αποφυγής εμποδίων και εύρεση βέλτιστης διαδρομής, για την εισαγωγή ευφυΐας στο σύστημα.

9.2 Μελλοντικές Επεκτάσεις-Βελτιώσεις

Η παρούσα μεταπτυχική Διπλωματική εργασία θα μπορούσε να επεκταθεί με διάφορους τρόπους βελτίωσης , για παράδειγμα : θα μπορούσε να γίνει επίλυση της βέλτιστης διαδρομής και αποφυγής εμποδίων με διάφορους μεθόδους διαφορετικών αλγορίθμων , η αποφυγή των εμποδίων στο χώρο θα μπορούσε να γίνει με την προσθήκη μιας κάμερας στο βραχίονα , όπως επίσης θα μπορούσε να γίνει και με την προσθήκη ενός αισθητήρα μέτρησης απόστασης. Για ακόμη μεγαλύτερη επέκταση της εργασίας θα μπορούσαν να μπουν επιπλέον βαθμοί ελευθερίας στο βραχίονα , όπως επίσης θα μπορούσε να τοποθετηθεί κι ένα συγκεκριμένο εργαλείο στο άκρο και να εκτελεί ο βραχίονας συγκεκριμένη εργασία.

Ακόμη θα μπορούσε να γίνει μια κατασκευή κλίμακας βιομηχανικού ρομποτικού βραχίονα, ικανή να παρέχει υπηρεσίες σε βιομηχανικό περιβάλλον, όπως και η χρήση υδραυλικών ή πνευματικών ενεργοποιητών.

9.3 Συμπεράσματα-Δυσκολίες κατά την ολοκλήρωση

Κατά την διάρκεια της Διπλωματικής εργασίας είδαμε πώς είναι σε μια μικρή οικιακή θα λέγαμε κατασκευή η σχεδίαση και η ανάπτυξη ενός συστήματος ρομποτικού βραχίονα, η οποία κάλιστα θα μπορούσε να επεκταθεί και σε βραχίονα κανονικών διαστάσεων Βιομηχανικού Επιπέδου. Παρατηρήσαμε επίσης την ποικιλία των υλικών που υπάρχουν στο εμπόριο για την ανάπτυξη τέτοιων εφαρμογών, όπως servo-κινητήρες και γενικά ενεργοποιητές, τροφοδοτικά και μετατροπείς ρεύματος, πλακέτες και διάφορα ενσωματωμένα συστήματα, software και hardware στη διάθεση όσων επιθυμούν την ενασχόλησή τους με το αντικείμενο της Ρομποτικής. Για την ανάπτυξη ενός ρομποτικού βραχίονα υπάρχουν πλέον πολλοί και διαφορετικοί τρόποι κατασκευής και κριτήρια, στη συγκεκριμένη Διπλωματική εργασία η κατασκευή του βραχίονα έγινε με κριτήριο το χαμηλό κόστος και την υψηλή απόδοση. Η χρήση του λογισμικού Matlab και συγκεκριμένα της έκδοσης Matlab 2017b καθόλη την διάρκεια των σπουδών ήταν ένας σημαντικός παράγοντας που συνέβαλε στην ανάπτυξη προγραμμάτων για την λειτουργία του βραχίονα, όπως και η χρήση της πλακέτας Arduino και του προγραμματιστικού περιβάλλοντος IDE. Τα προβλήματα που προέκυψαν ήταν διαφόρων ειδών και είχαν σχέση με: **το μηχανικό μέρος**, όπου χρησιμοποιήθηκαν σερβο-κινητήρες υψηλής ροπής και ελαφριά κατασκευή αλουμινίου, με την τροφοδοσία, όπου έγινε χρήση εξωτερικής σταθερής τροφοδοσίας και μετατροπέα DC to DC, απότομες επιταχύνσεις και καταπόνηση των σερβο-κινητήρων προκάλεσαν ανησυχία, έτσι έγινε χρήση της βιβλιοθήκης VarSpeedServo.h, η πολύπλοκη επίλυση των 2 κινηματικών ευθέως και ανάστροφου κινηματικού, όπου αντιμετωπίστηκε αποτελεσματικά με ένα αρχείο excel για έλεγχο των αποτελεσμάτων, έπειτα μεγάλο βαθμό δυσκολίας περιείχε το προγραμματιστικό μέρος στο περιβάλλον του Matlab, όπου κατασκευάστηκε ειδικό πρωτόκολλο σειριακής επικοινωνίας Matlab-Arduino, για την κίνηση του βραχίονα, όπως και η δημιουργία Αλγορίθμων υψηλού επιπέδου για την αποφυγή εμποδίων επίσης στο περιβάλλον του Matlab2017b. Αρκετά βοήθησε το πακέτο προσομείωση rvctools, δημιουργός του οποίου είναι ο Peter Corke 1992-2017 (<http://www.petercorke.com>). Η σωστή καθοδήγηση του καθηγητή σε συνδυασμό με την επιμονή και την υπομονή, συνετέλεσαν στην ολοκλήρωση της Διπλωματικής εργασίας.

Κεφάλαιο 10

Κώδικες που αναπτύχθηκαν

Για τον έλεγχο των servo στο περιβάλλον IDE του Arduino

Για την οδήγηση των servo και το χαμηλό επίπεδο εντολών στο περιβάλλον IDE του Arduino , σε γλώσσα προγραμματισμού wiring χρησιμοποιήθηκε το παρακάτω αρχείο με το όνομα Dummy.

```
#include <VarSpeedServo.h> // Εισαγωγή της Βιβλιοθήκης VarSpeedServo.h(η οποία περιλαμβάνει έλεγχο της ταχύτητας των servo).  
#include <stdlib.h> // Εισαγωγή της βιβλιοθήκης stdlib.h  
#include<string.h> //Εισαγωγή της βιβλιοθήκης συμβολοσειρών string.h  
#include<math.h> // Εισαγωγή της βιβλιοθήκης math.h, περιλαμβάνει ένα μεγάλο αριθμό μαθηματικών λειτουργιών.
```

```
VarSpeedServo servo1; // Δήλωση του πρώτου σερβοκινητήρα  
VarSpeedServo servo2; // Δήλωση του δεύτερου σερβοκινητήρα  
VarSpeedServo servo3; // Δήλωση του τρίτου σερβοκινητήρα  
const int servo1Pin = 3; //Δήλωση σταθεράς-1  
const int servo2Pin = 6; //Δήλωση σταθεράς-2  
const int servo3Pin = 9; //Δήλωση σταθεράς-3  
int recValue; // Ορισμός ακέραιας μεταβλητής η οποία θα λαμβάνει τιμές από το Matlab  
String incoming_byte;  
int v1,v2,v3 = 0; // Τιμές που θα δώσει ο χρήστης για τις γωνίες  
boolean enable = LOW; // Αν είναι HIGH τότε ο χρήστης έχει δώσει σωστά τις τιμές στην σειριακή.
```

```
//Servo Servo1; // Δημιουργία Servo Object  
//Servo Servo2; // Δημιουργία Servo Object  
//Servo Servo3; // Δημιουργία Servo Object
```

```
void setup() // Εκτελείται μόνο μια φορά  
{  
Serial.begin(115200); // Ορίζει την ταχύτητα δεδομένων σε bits ανά δευτερόλεπτο (baud) για τη μετάδοση σειριακών δεδομένων.  
servo1.attach(servo1Pin);  
servo2.attach(servo2Pin);  
servo3.attach(servo3Pin);  
//Servo1.attach(servo1Pin); //Δήλωση του pin με το οποίο θα παίρνει εντολή το servo  
//Servo2.attach(servo2Pin); //Δήλωση του pin με το οποίο θα παίρνει εντολή το servo  
//Servo3.attach(servo3Pin); //Δήλωση του pin με το οποίο θα παίρνει εντολή το servo
```

```
pinMode(13,OUTPUT);
}
```

```
void loop()
{
if(Serial.available() > 0)
{
incoming_byte = Serial.readString(); // Διαβάζει την συμβολοσειρά από την σειριακή θύρα
Serial.println(incoming_byte);
int buffer_length = incoming_byte.length(); // Βρίσκει το μήκος της
char incoming_byte_string[buffer_length]; // Ορίζουμε ένα πίνακα χαρακτήρων με μήκος όσο
αυτό της συμβολοσειράς
incoming_byte.toCharArray(incoming_byte_string,buffer_length+1); // Μετατροπή
συμβολοσειρά σε χαρακτήρες για την sscanf
if (sscanf(incoming_byte_string,"%d,%d,%d",&v1,&v2,&v3) == 3) enable = HIGH;
else enable = LOW;
// Διαβάζει τους χαρακτήρες και τους συγκρίνει με ένα δείγμα που πρέπει να έχουν πχ
XX,XX,XX. Αν ισχύει αυτό τότε αποθηκεύει τις τιμές
// από την σειριακή στα v1 ,v2, v3. Ο χρήστης πρέπει δηλαδή να δώσει input με συγκεκριμένη
μορφή για να λειτουργήσει το πρόγραμμα
}
if (enable == HIGH) // Μόνο αν τα στοιχεία στην σειριακή είναι σωστά θα κινηθεί ο βραχίονας
{
//Servo1.write(v1);
//Servo2.write(v2);
//Servo3.write(v3);
servo1.write(v1,20); //speed 20 Καθορισμός της ταχύτητας περιστροφής του servo1
servo2.write(v2,20); //speed 20 Καθορισμός της ταχύτητας περιστροφής του servo2
servo3.write(v3,20); //speed 20 Καθορισμός της ταχύτητας περιστροφής του servo3
//servo1.wait();
//servo2.wait();
//servo3.wait();
Serial.println(v1);
Serial.println(v2);
Serial.println(v3);
enable = LOW;
//digitalWrite(13,HIGH);
//delay(1000);
//digitalWrite(13,LOW);
//delay(1000);
}
}
```

Για την προσομείωση του βραχίονα , το αρχείο `startup_rvc.m`

```
disp('Robotics, Vision & Control: (c) Peter Corke 1992-2017  
http://www.petercorke.com')
```

```
if verLessThan('matlab', '7.0')  
    warning('You are running a very old (and unsupported) version of MATLAB. You  
    will very likely encounter significant problems using the toolboxes but you are on  
    your own with this');  
end
```

```
tb = false;  
rvcpath = fileparts( mfilename('fullpath') );
```

```
robotpath = fullfile(rvcpath, 'robot');
```

```
if exist(robotpath,'dir')  
    addpath(robotpath);  
    tb = true;  
    if exist('startup_rtb') == 2  
        startup_rtb  
    end  
end
```

Για την επίλυση του ανάστροφου κινηματικού, το αρχείο
inverse_kinematic.m

```
function [theta1,theta2,theta3] = inverse_kinematic(PX,PY,PZ,elbow)
%This function solves the inverse kinematic model
% Inputs: PX = desired value for x-axis
%         PY = desired value for y-axis
%         PZ = desired value for z-axis
%         elbow =
% Outputs: theta1 = angle for joint 1
%         theta2 = angle for joint 2
%         theta3 = angle for joint 3

I1 = 114; % legth of first arm
I2 = 104; % length of second arm
I3 = 223; % length of third arm

if elbow == 1
    number = 1;
elseif elbow == 0
    number = -1;
end

theta1= atan2(PY,PX);
r=sqrt(PX^2+PY^2) * number;
D=sqrt((PZ-I1)^2+r^2) * number;
theta3=acos((D^2 - I3^2 - I2^2) / (2 * I2 * I3));
theta2=atan2(r,PZ - I1) - atan2(I2 + (I3 * cos(theta3)), I3 * sin(theta3));
theta1=rad2deg(theta1);
if abs(theta1) <= 180 && theta1 < 0
    theta1 = 180 + theta1;
end
theta2=rad2deg(theta2);
if abs(theta2) <= 180 && theta2 < 0
    theta2 = 180 + theta2;
end
theta3=rad2deg(theta3)+90;
if abs(theta3) <= 180 && theta3 < 0
```

```
theta3 = 180 + theta3;
```

```
end
```

```
end
```

Ο βασικός Κώδικας με ονομασία main.m

```
%%
% Κλείσιμο όλων παραθύρων, καθαρισμός μεταβλητών και καθαρισμός του
command
% window
clear all
close all
clc

global GlobalBest model xs ys zs number_of_points;

%%-----
%%
% Έλεγχος αν η COM θύρα είναι ανοικτή και αν είναι την κλείνουμε
if ~isempty(instrfind)
    fclose(instrfind);
    delete(instrfind);
end
%-----
%%
s = serial('COM6','BAUD',115200); % Ορισμός σειριακής στα 9600
fopen(s); % Ενεργοποίηση σειριακής
data_cell = []; % Δημιουργία array για αποθήκευση τιμών που θα σταλούν στο
Arduino

% 3 διαφορετικές μεταβλητές για 3 διαφορετικά mode λειτουργίας
angle = 0; % Mode 1 -> Λειτουργία με είσοδο γωνίες
xyz = 0; % Mode 2 -> Λειτουργία με είσοδο σημείων
move = 0; % Mode 3 -> Λειτουργία κίνησης σε προκαθορισμένα σημεία
move2 = 0; % Mode 4 -> Path Planning with PSO

% Μήκος των τμημάτων του βραχίονα
l1 = 114;
l2 = 104;
l3 = 223;

% Για να λειτουργήσει το μοντέλο προσομοίωσης πρέπει πρώτα να εκτελεστεί
% το αρχείο startup_rvc.m
% Αρχικοποίηση μεταβλητών για το μοντέλο προσομοίωσης του βραχίονα
% Στην προσομοίωση ο βραχίονας έχει 5 μέλη
L1 = Link([0 l1 0 0]); % Ορισμός 1ου μέλους
L2 = Link([0 0 0 pi/2]); % Ορισμός 2ου μέλους
```

```

L3 = Link([0 0 l2 0]); % Ορισμός 3ου μέλους
L4 = Link([0 0 0 -pi/2]); % Ορισμός 4ου μέλους
L5 = Link([0 l3 0 0]); % Ορισμός 5ου μέλους
W = [-500,500,-500,500,0,500]; % Ορισμός χώρου εργασίας βραχίονα
robot = SerialLink([L1 L2 L3 L4 L5]); % Ένωση όλων των μελών του βραχίονα
robot.name = 'Inferno'; % Ονομασία του βραχίονα

% Το πρόγραμμα αποτελείται από 2 while. Στην πρώτη ο χρήστης επιλέγει mode
% λειτουργίας και στην δεύτερη γίνεται η εκτέλεση της εντολής του χρήστη.
while 1 % Επιλογή mode λειτουργίας από χρήστη
    ans = input('Press 1 for angles, 2 for points, 3 for a simple movement, 4 for path
    planning:', 's'); % Η επιλογή του χρήστη αποθηκεύεται σε μια συμβολοσειρά με
    όνομα ans
    ans = str2num(ans); % Μετατροπή συμβολοσειράς σε αριθμό
    if ans == 1 | ans == 2 | ans == 3 | ans == 4 % Έλεγχος για το mode που έχει
    επιλέξει ο χρήστης

        % Τις μεταβλητές angle,xyz και move τις χρησιμοποιούμε στο 2ο
        % while. Σε αυτό το while ανάλογα με το mode που έχει επιλέξει ο
        % χρήστης κάνουμε 1 την αντίστοιχη μεταβλητή

        % Έλεγχος επιλογής mode 1
        if ans == 1
            angle = 1;
            break % έξοδος από while

        % Έλεγχος επιλογής mode 2
        elseif ans == 2
            xyz = 1;
            break % έξοδος από while

        % Έλεγχος επιλογής mode 3
        elseif ans == 3
            move = 1;
            break % έξοδος από while

        % Έλεγχος επιλογής mode 4
        elseif ans == 4
            move2 = 1;
            break % έξοδος από while
        end

    else continue; % Για οποιοδήποτε input του χρήστη διαφορετικό από αυτό που
    έχουμε
        % ορίσει το πρόγραμμα αγνοεί την είσοδο και περιμένει νέα με
    συγκεκριμένη μορφή
        % για να συνεχίσει στο 2ο while

    end
end
%-----

```

```
%-----
%-----

%%
while 1 % Σε αυτό το while λειτουργούν τα διαφορετικά mode

%#####
###

%#####
###

    if angle == 1 % Αν έχει επιλέξει ο χρήστης mode 1
        data = input('Give angles for joints th1,th2,th3 from 16-165 each: ','s'); %
        Εισαγωγή γωνιών θ1,θ2,θ3 από χρήστη σε μορφή συμβολοσειράς
        disp(data); % εκτύπωση εισόδου χρήστη
        data_cell = strsplit(data,','); % διαχωρισμός συμβολοσειράς χρήστη στο ',' και
        αποθήκευση στο data_cell
        if length(data_cell) == 3 % έλεγχος αν μετά το διαχωρισμό με ',' έχουμε 3
        τμήματα συμβολοσειράς

            % Αν έχουμε τρία τμήματα τότε μετατρέπουμε τα σύμβολα σε
            % αριθμούς και περιορίζουμε τις γωνίες από 15 μέχρι 165 μοίρες
            data_cell = max(16,min(165,str2double(data_cell)));
            robot.plot([0,deg2rad(data_cell(1,1)),deg2rad(data_cell(1,2)),-
            deg2rad(data_cell(1,3)),0],'floorlevel',0,'notiles','workspace',W)
            hold on
            data_cell = num2cell(data_cell); % Μετατρέπουμε το array data_cell σε κελί
            και το
            fprintf(s,strjoin(cellfun(@num2str,data_cell,'un',0),',')); % % στέλνουμε στο
            arduino με μορφή Θ1,Θ2,Θ3

        else % Αν δεν έχουμε τρία τμήματα συμβολοσειράς τότε ο χρήστης έχει κάνει
        λάθος εισαγωγή στοιχείων
            if strcmp(data,'exit') % έλεγχος αν ο χρήστης έχει πληκτρολογήσει exit
                fclose(s); % Αν έχει πληκτρολογήσει exit τότε κλείνουμε την σειριακή και
                break; % τερματίζουμε το πρόγραμμα
            end
            continue; % Οποιαδήποτε άλλο input του χρήστη απλά το αγνοούμε
        end

%#####
###

%#####
###

elseif xyz == 1 % Αν έχει επιλέξει ο χρήστης mode 2
```



```

data = input('Give xyz points: ','s'); % Εισαγωγή συμβολοσειράς με τα σημεία
x,y,z
disp(data); % εκτύπωση εισόδου χρήστη
data_cell = strsplit(data,',' ); % διαχωρισμός συμβολοσειράς χρήστη στο ',' και
αποθήκευση στο data_cell
if length(data_cell) == 3 % % έλεγχος αν μετά το διαχωρισμό με ',' έχουμε 3
τμήματα συμβολοσειράς
    data_cell = str2double(data_cell); % μετατροπή χαρακτήρων σε αριθμούς
    % Με το try - catch μπορούμε να εκτελέσουμε μια εντολή και αν
    % αυτή βγάλει σφάλμα τότε εκτελείται το κομμάτι του κώδικα που
    % βρίσκεται μέσα στο catch.
    try
        [data_cell(1),data_cell(2),data_cell(3)] =
inverse_kinematic(data_cell(1),data_cell(2),data_cell(3),0); % Επίλυση ανάστροφου
κινηματικού με elbow down.
    catch
        warning('Problem with solving inverse kinematic. Imaginary solution'); % Αν
το ανάστροφο κινηματικό δεν έχει λύση τότε βγάζει αυτό το σφάλμα χωρίς να
σταματά το πρόγραμμα και
        % περιμένει από τον χρήστη να εισάγει νέα σημεία
        continue; % Μετά από ανεπιτυχή προσπάθεια επίλυσης του ανάστροφου
κινηματικού το continue οδηγεί το πρόγραμμα στην αρχή του else if
    end
    disp(data_cell) % Εκτύπωση αποτελεσμάτων από επίλυση ανάστροφου
κινηματικού
    robot.plot([0,deg2rad(data_cell(1,1)),deg2rad(data_cell(1,2)),-
deg2rad(data_cell(1,3)),0], 'floorlevel',0, 'notiles', 'workspace',W)
    hold on
    data_cell(2) = max(5,data_cell(2)); % Περιορισμός 2 σέρβου στις 5 μοίρες
    data_cell = num2cell(round(data_cell)); % Στρογγυλοποίηση αποτελέσματος
και μετατροπή του array σε cell
    fprintf(s,strjoin(cellfun(@num2str,data_cell,'un',0),',')); % Προετοιμασία και
αποστολή δεδομένων στο arduino
else
    if strcmp(data,'exit') % έλεγχος αν ο χρήστης έχει πληκτρολογήσει exit
        fclose(s); % Αν έχει πληκτρολογήσει exit τότε κλείνουμε την σειριακή και
        break; % τερματίζουμε το πρόγραμμα
    end
    continue; % Οποιαδήποτε άλλο input του χρήστη απλά το αγνοούμε
end

%#####
####
%#####
####
% Το mode3 ουσιαστικά κάνει 2 κινήσεις των 10 σημείων.

elseif move == 1 % Αν έχει επιλέξει ο χρήστης mode 2

    % 1η κίνηση των 10 σημείων

```

```

% Για αλλαγή των 10 σημείων αλλάζουμε το τελευταίο νούμερο της
παρένθεσης και το
% κάνουμε X
space = linspace(-150,150,15);
x = space;% σχηματισμός της ευθείας γραμμής παράλληλης με τον άξονα_x
y = 265 * ones(1,length(space));% σχηματισμός της ευθείας γραμμής 265mm
στον άξονα_y
z = 100 * ones(1,length(space)); % σχηματισμός της ευθείας γραμμής σε
100mm από τον άξονα_Z
i = 1; % Δείχνει σε ποιο σημείο της κίνησης είμαστε

% Επίλυση του ανάστροφου κινηματικού για τα σημεία της πρώτης
% κίνησης και αποθήκευση τους στον πίνακα data_cell
for i = 1:size(x,2);
    try
        [data_cell(i,1),data_cell(i,2),data_cell(i,3)] =
inverse_kinematic(x(i),y(i),z(i),0);

        catch

% [data_cell(i,1),data_cell(i,2),data_cell(i,3)] =
inverse_kinematic(round(x(i)),round(y(i)),round(z(i)),0);
        end
    end

for i = 1:size(data_cell,1)
    % Plot την θέση του βραχίονα στον υπολογιστή
    robot.plot([0,deg2rad(data_cell(i,1)),deg2rad(data_cell(i,2)),-
deg2rad(data_cell(i,3)),0],'floorlevel',0,'notiles','workspace',W)
    hold on
    plot3(x(1,i),y(1,i),z(1,i),'b-o');
    hold on
% data_cell(i,2) = max(5,data_cell(i,2)); % Περιορισμός 2 σέρβου στις 5
μοίρες
    value = num2cell(round(data_cell(i,:))); % αποθήκευση των τιμών για το
arduino σε μια προσωρινή μεταβλητή
    fprintf(s,strjoin(cellfun(@num2str,value,'un',0),',')); % προετοιμασία τους
και αποστολή στο arduino
    disp(value) % Εκτύπωση θ1,θ2,θ3
    pause(1.5); % Αναμονή 2 sec για ολοκλήρωση κίνησης του βραχίονα
end
fclose(s); % κλείσιμο σειριακής
break

%#####
####
%#####
####

```

elseif move2 == 1

```

% Αρχικοποίηση της δομής GlobalBest που θα κρατά την βέλτιστη λύση
%GlobalBest.Cost=inf;
%GlobalBest.Position=[];
%GlobalBest.Sol=[];
%% Καθορισμός του χώρου και των εμποδίων
%model=CreateModel(x_init,y_init,z_init,number_of_points);%

model=CreateModel();

PSO; % εκτέλεση του PSO

% Εύρεση βέλτιστης λύσης x,y,z
%x_new = GlobalBest.Sol.xx;
%y_new = GlobalBest.Sol.yy;
%z_new = GlobalBest.Sol.zz;
x_new = GlobalBest.Sol.XS;
y_new = GlobalBest.Sol.YS;
%z_new = GlobalBest.Sol.ZS;
z_new=5*ones(size(x_new));

%close all % κλείσιμο όλων των παραθύρων
for i = 1:size(x_new,2);
    try
        %[data_cell(i,1),data_cell(i,2),data_cell(i,3)] =
inverse_kinematic(x_new(i),y_new(i),z_new(i),0);
        [data_cell(i,1),data_cell(i,2),data_cell(i,3)] =
inverse_kinematic(x_new(i),y_new(i),z_new(i),0); % επίλυση ανάστροφου
κινηματικού από την λύση που κατέληξε ο PSO
    catch
        %[data_cell(i,1),data_cell(i,2),data_cell(i,3)] =
inverse_kinematic(round(x_new(i)),round(y_new(i)),round(z_new(i)),0);
    end
end

% Plot των σημείων που πρέπει να πάει ο βραχίονας
for i = 1:size(data_cell,1)
    data_cell(i,2) = max(5,data_cell(i,2)); % Περιορισμός 2 σέρβου στις 5
μοίρες
    value = num2cell(round(data_cell(i,:))); % αποθήκευση των τιμών για το
arduino σε μια προσωρινή μεταβλητή % num2cell μετατρέπει πίνακα σε κελί
    fprintf(s,strjoin(cellfun(@num2str,value,'un',0),',')); % προετοιμασία τους
και αποστολή στο arduino

```

```

disp(value); % Εκτύπωση 01,02,03
pause(1.5); % Αναμονή 1.5 sec για ολοκλήρωση κίνησης του βραχίονα
end
fclose(s); % κλείσιμο σειριακής
break

end
end

```

Για τον αλγόριθμο PSO χρησιμοποιήθηκαν οι κώδικες :

Για την κατασκευή της συνάρτησης δημιουργίας του μοντέλου του χώρου το αρχείο CreateModel.m

```

function model=CreateModel()

% Αφετηρία
xs=200; ys=200;

% Τερματισμός
xt=-200; yt=200;

% Συντεταγμένες κυκλικών εμποδίων
xobs=[65.0 -65.0]
yobs=[230.0 188.0]

% Ακτίνες κυκλικών εμποδίων
robs=[26.0 20.0]

% Αριθμός σημείων καμπής διαδρομής
n=3;

% Όρια του x
xmin= -280; xmax= 280;

% Όρια του y
ymin = 0 ; ymax= 250;

% Κατασκευή της δομής (structure) "model"

```

```

model.xs=xs;
model.ys=ys;
model.xt=xt;
model.yt=yt;
model.xobs=xobs;
model.yobs=yobs;
model.robs=robs;
model.n=n;
model.xmin=xmin;
model.xmax=xmax;
model.ymin=ymin;
model.ymax=ymax;

```

end

Κατασκευή της συνάρτησης για ανάπτυξη και έλεγχο της εκάστοτε λύσης ,
το αρχείο «ParseSolution.m»

```

function sol=ParseSolution(position,model)
% Κατασκευή ανυσμάτων συντεταγμένων ολόκληρης της διαδρομής
XS=[model.xs position.x model.xt];
YS=[model.ys position.y model.yt];
%ZS=[model.zs position.z model.zt];
% Κατασκευή της διαδρομής με 100 σημεία [xx,yy,zz] με γραμμική παρεμβολή
k=numel(XS);
TS=linspace(0,1,k);
tt=linspace(0,1,100);
xx=interp1(TS,XS,tt);
yy=interp1(TS,YS,tt);
%zz=interp1(TS,ZS,tt);
% Υπολογισμός των αποστάσεων μεταξύ των σημείων
dx=diff(xx);
dy=diff(yy);
%dz=diff(zz);
% Υπολογισμός της συνολικής απόστασης της διαδρομής
%L=sum(sqrt(dx.^2+dy.^2+dz.^2));
L=sum(sqrt(dx.^2+dy.^2));
% Έλεγχος επαφής με τα εμπόδια
nobs = numel(model.xobs); % Αριθμός εμποδίων
Violation = 0; % ποινή επαφής με εμπόδια
for k=1:nobs % βρόχος για όλα τα εμπόδια
% Υπολογισμός αποστάσεων των σημείων της διαδρομής από το κέντρο του
εμποδίου
%d=sqrt((xx-model.xobs(k)).^2+(yy-model.yobs(k)).^2 + (zz-model.zobs(k)).^2);
d=sqrt((xx-model.xobs(k)).^2+(yy-model.yobs(k)).^2 );
% Έλεγχος για επαφή με το εμπόδιο
v=max(1-d/model.robs(k),0);
% Καθορισμός ποινής
Violation=Violation+mean(v);
end
% Κατασκευή της δομής sol που επιστρέφει η συνάρτηση

```

```

sol.TS=TS; % k σημεία μεταξύ 0 και 1
sol.XS=XS; % x συντεταγμένες των σημείων αφετηρίας,καμπής και
τερματισμού
sol.YS=YS; % y συντεταγμένες των σημείων αφετηρίας,καμπής και
τερματισμού
%sol.ZS=ZS; % z συντεταγμένες των σημείων αφετηρίας,καμπής και
τερματισμού
sol.tt=tt; % 100 σημεία μεταξύ 0 και 1
sol.xx=xx; % x συντεταγμένες 100 σημείων της διαδρομής
sol.yy=yy; % y συντεταγμένες 100 σημείων της διαδρομής
%sol.zz=zz; % z συντεταγμένες 100 σημείων της διαδρομής
sol.dx=dx; % x αποστάσεις μεταξύ των 100 σημείων της διαδρομής
sol.dy=dy; % y αποστάσεις μεταξύ των 100 σημείων της διαδρομής
%sol.dz=dz; % z αποστάσεις μεταξύ των 100 σημείων της διαδρομής
sol.L=L; % το μήκος της διαδρομής
sol.Violation=Violation; % η ποινή για επαφή με εμπόδια

sol.IsFeasible=(Violation==0); % boolean πεδίο για επαφή με εμπόδια
end

```

Κατασκευή της συνάρτησης ποιότητας για τις λύσεις του προβλήματος , το αρχείο με όνομα «PathPlanning.m»

```

function scores=PathPlanning(pop)
global model GlobalBest;
% Ανάκτηση των συντεταγμένων των σημείων καμπής
position.x=pop(1:model.n);
position.y=pop(model.n+1:2*model.n);
%position.z=pop(13:18);
% Κατασκευή της διαδρομής και έλεγχος αποφυγής εμποδίων
sol=ParseSolution(position,model);
% Υπολογισμός της ποιότητας της λύσης
beta=100; % συντελεστής ποινής για επαφή με εμπόδια
scores=sol.L*(1+beta*sol.Violation);
% Ενημέρωση του GlobalBest
if scores<GlobalBest.Cost
GlobalBest.Cost=scores;
GlobalBest.Sol=sol;
GlobalBest.Position=position;
end
end

```

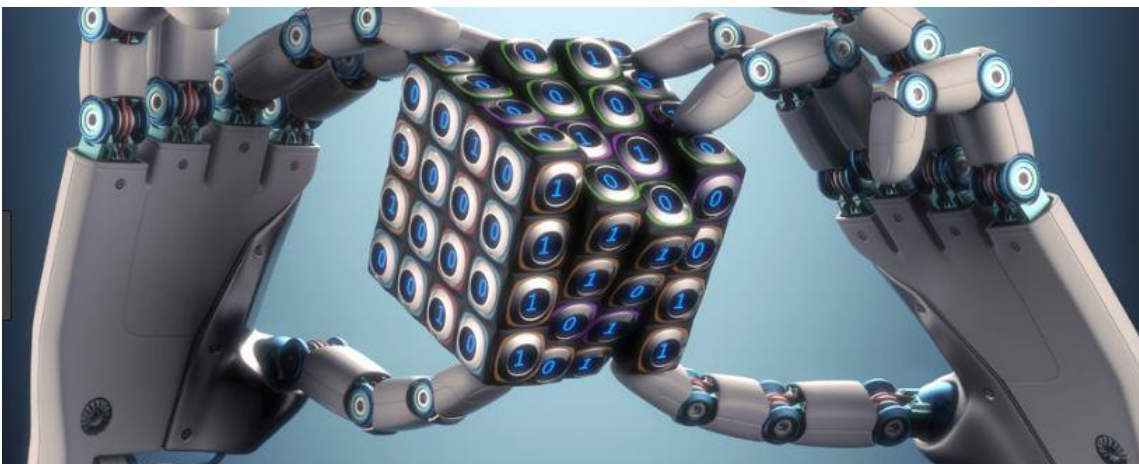
Κατασκευή της συνάρτησης που θα σχεδιάζει τις λύσεις του προβλήματος με όνομα αρχείου «PathPlot.m»

```
function stop = PathPlot(optimValues,state)
% δήλωση των global μεταβλητών
global GlobalBest model %W xs ys zs;
stop = false;
% παραγωγή 100 τιμών από 0 έως 2π
theta=linspace( 0 , 2*pi , 100 );
% Εμφάνιση των κυκλικών εμποδίων
for k=1 : numel(model.xobs)
    fill(model.xobs(k)+model.robs(k)*cos(theta),...
        model.yobs(k)+model.robs(k)*sin(theta),'r');
    hold on;
end
% Εμφάνιση της διαδρομής
plot(GlobalBest.Sol.xx,GlobalBest.Sol.yy,'k','LineWidth',2);
% Εμφάνιση των σημείων καμπής
plot(GlobalBest.Sol.XS,GlobalBest.Sol.YS,'ro');
% Εμφάνιση της αφηρησίας
plot(model.xs,model.ys,'bs','MarkerSize',12,'MarkerFaceColor','y');
% Εμφάνιση του προορισμού
plot(model.xt,model.yt,'kp','MarkerSize',16,'MarkerFaceColor','g');
hold off;
grid on;
axis equal;

end
```

Κατασκευή του κυρίως προγράμματος για Βέλτιστη σχεδίαση Διαδρομής με χρήση του αλγορίθμου PSO, το αρχείο με όνομα «PSO.m»

```
% Πρόγραμμα επίλυσης προβλήματος σχεδιασμού διαδρομής
% Global Μεταβλητές
global GlobalBest model;
% Αρχικοποίηση της γεννήτριας τυχαίων αριθμών
GlobalBest.Cost=inf;
GlobalBest.Position=[ ];
GlobalBest.Sol=[ ];
%rng default % για αναπαραγωγή των αποτελεσμάτων
rng('shuffle'); % για τυχαιότητα
% Καθορισμός options: "PlotFcn" και "Display"
%options = optimoptions(@particleswarm,'PlotFcn',{@pswplotbestf,@PathPlot},...
%'Display','iter');
options =
optimoptions(@particleswarm,'PlotFcn',{@pswplotbestf,@PathPlot},'Display','iter',
ToIFun',0.05,'SwarmSize',500);
% Καθορισμός άνω και κάτω ορίων των μεταβλητών
%lb= [ ones(1,model.n)*model.xmin ones(1,model.n)*model.ymin
ones(1,model.n)*model.zmin];
%ub= [ ones(1,model.n)*model.xmax ones(1,model.n)*model.ymax
ones(1,model.n)*model.zmax];
lb= [ ones(1,model.n)*model.xmin ones(1,model.n)*model.ymin];
ub= [ ones(1,model.n)*model.xmax ones(1,model.n)*model.ymax];
% Εκτέλεση του PSO
[x,fval,exitflag,output] = particleswarm(@PathPlanning,2*model.n,lb,ub,options);
```



Κεφάλαιο 11

Βιβλιογραφία

- [1] “Ανάλυση, Έλεγχος και Προγραμματισμός Ρομποτικών Χειριστών Σταθερής Βάσης”, Ιωάννης Μπούταλης, 978-960-93-7111-7, 2015, Αυτοέκδοση.
- [2] “Κινηματική, δυναμική και έλεγχος αρθρωτών βραχιόνων”, Δουλγέρη Ζωή, 978-960-218-502-5, 2007, ΕΚΔΟΣΕΙΣ ΚΡΙΤΙΚΗ ΑΕ
- [3] Introduction to autonomous mobile robots”, Roland Siegwart, Illah R. Nourbakhsh, and Davide Scaramuzza. - 2nd ed., Cambridge, Mass. : MIT Press, 2011
- [4] “Autonomous Land Vehicles”, K. Berns, E. von Puttkamer, Vieweg and Teubner.
- [5] “Designing Autonomous Mobile Robots, Inside the mind of an intelligent machine”, John Holland, Newness
- [6] “Probabilistic Robotics”, M. Thrun, MIT Press
- [7] ROS Robotics By Example, Carol Fairchild, Thomas L. Harman, 2016, Packt Publishing
- [8] VREP – User manual (<http://www.coppeliarobotics.com/helpFiles/>)
- [9] Gazebo – User manual and tutorial (<http://gazebo.org/tutorials>)
- [10] “Introduction to the design of small-scale embedded systems”, Tim Wilmshurst, Palgrave.
- [11] “Συστήματα Μικροϋπολογιστών II: Μικροελεγκτές AVR και PIC”, Κιαμάλ Πεκμεσιζή. Εκδόσεις Συμμετρία.
- [12] “Ανάπτυξη Εφαρμογών με το Arduino”, Παναγιώτης Παπάζογλου, Σ.-Π. Λιώνης, Εκδόσεις Τζιόλα.
- [13] . “18+ Random nerd Tutorial Projects”, Rui Santos
- [14] “Embedded System Design, A Unified hardware/Software Introduction”, Frank Vahid, Tony Givargis, John Wiley & Sons.
- [15] “Designing Embedded Systems with PIC Microcontrollers”, Tim Wilmshurst, Newnes.
- [16] “Getting Started with Python and Raspberry Pi”, Dan Nixon, Packt Publishing (open source).
- [17] “Αρχιτεκτονική και Προγραμματισμός του PIC16F877”, Ι. Καλόμοιρου, Σημειώσεις.
- [18] <http://nefeli.lib.teicrete.gr/browse/sefe/hlk/2009/TsontosDimitrios/attached-document-1263376508-952463-3039/2009tsontos.pdf>
- [19] <https://www.youtube.com/watch?v=3ZcYSKVDIOc>
- [20] <https://www.youtube.com/watch?v=3ZcYSKVDIOc>
- [21] http://users.sch.gr/imarinakis/electric_engines.htm
- [22] Coley, D. (1999), “An Introduction to Genetic Algorithms for Scientists and Engineers”, World Scientific Press, New York
- [22] Denavit, J., Hartenberg, R.S. (1955), “A Kinematic Notation for Lower Pair Mechanisms Based on Matrices”, Transactions of the ASME, Journal of Applied Mechanics, E22, pp.215-222
- [23] <https://forum.arduino.cc/index.php?topic=82903.0>
- [24] <https://microbotlabs.com/armuno-arduino-schematic.html>
- [25] https://www.researchgate.net/publication/317277584_ROBOT_ARM_CONTROL_WITH_ARDUINO

- [26] Mohammed_and_Sunar_2015_Kinematics_modeling_of_a-4-DOF_robotic_arm.pdf
- [27] <http://matlabarduinorobotics.blogspot.com/2014/07/matlab-arduino-serial-communication.html>
- [28] <https://www.allaboutcircuits.com/projects/arduino-interface-with-matlab/>
- [28] <https://www.youtube.com/watch?v=ldxeo49szW0>
- [29] <https://www.youtube.com/watch?v=D93iQVoSScQ>
- [30] <http://www.sml.ee.upatras.gr/UploadedFiles/InverseKinematics.pdf>
- [31] http://www.academia.edu/Documents/in/Inverse_Kinematics
- [32] <https://robotacademy.net.au/lesson/inverse-kinematics-for-a-2-joint-robot-arm-using-geometry/>
- [33] <https://el.wikipedia.org/wiki/%CE%A4%CF%81%CE%B9%CE%B3%CF%89%CE%BD%CE%BF%CE%BC%CE%B5%CF%84%CF%81%CE%AF%CE%B1>
- [34] https://el.wikipedia.org/wiki/%CE%91%CE%BD%CF%84%CE%AF%CF%83%CF%84%CF%81%CE%BF%CF%86%CE%B5%CF%82_%CF%84%CF%81%CE%B9%CE%B3%CF%89%CE%BD%CE%BF%CE%BC%CE%B5%CF%84%CF%81%CE%B9%CE%BA%CE%AD%CF%82_%CF%83%CF%85%CE%BD%CE%B1%CF%81%CF%84%CE%AE%CF%83%CE%B5%CE%B9%CF%82
- [35] <http://forum.arduino.cc/index.php?topic=165107.0>
- [36] <http://www.m3.tuc.gr/PROSWPIKO/DIPLWMATIKES/delikonstantinou%20vassilis.pdf>
- [37] <https://www.youtube.com/watch?v=HRypuxXxEcg>
- [38] <http://www.allaboutcircuits.com/projects/arduino-interface-with-matlab/>
- [39] https://github.com/thomashiemstra/robot_arm_with_vision
- [40] https://books.google.gr/books?hl=el&lr=&id=d4EkDwAAQBAJ&oi=fnd&pg=PR7&dq=obstacle+avoidance+robot+arms+matlab&ots=tmgdpPIJkS&sig=rIFgHwBymwHLQDhBC0-iavwhBLM&redir_esc=y#v=onepage&q=obstacle%20avoidance%20robot%20arms%20matlab&f=false
- [41] http://www.arnjournals.org/jeas/research_papers/rp_2017/jeas_0417_5860.pdf
- [42] <https://eclass.teicrete.gr/modules/document/file.php/TM118/1%20-%20%CE%98%CE%95%CE%A9%CE%A1%CE%99%CE%91%20/1%20-%20%CE%94%CE%B9%CE%B1%CF%86%CE%AC%CE%BD%CE%B5%CE%B9%CE%B5%CF%82%20%CE%B4%CE%B9%CE%B1%CE%BB%CE%AD%CE%BE%CE%B5%CF%89%CE%BD/%CE%92%CE%91%CE%A3%CE%99%CE%9A%CE%95%CE%A3%20%CE%95%CE%9D%CE%9D%CE%9F%CE%99%CE%95%CE%A3%20.pptx.pdf>
- [43] <http://apothesis.teicm.gr/xmlui/bitstream/handle/123456789/814/sefi.pdf?sequence=1>
- [44] <http://apothesis.teicm.gr/xmlui/bitstream/handle/123456789/2560/Mentesidis.pdf?sequence=1&isAllowed=y>
- [45] <https://appliedgo.net/roboticarm/>
- [46] https://www.mathworks.com/help/fuzzy/modeling-inverse-kinematics-in-a-robotic-arm.html?searchHighlight=forward%20kinematic&s_tid=doc_srchtile
- [47] <https://www.youtube.com/watch?v=xF1KaINQwa8>
- [48] <https://www.youtube.com/watch?v=apPJtJqCu44>
- [49] https://www.researchgate.net/publication/281237691_Modeling_Simulation_and_Position_Control_of_3DOF_Articulated_Manipulator
- [50] <http://users.softlab.ntua.gr/~ktzaf/Courses/robotics-l-shmmy-kinematics-1.pdf>
- [51] http://kontopodis.edu.gr/Ready/Excel_HTML/Fun_Math.htm
- [52] <http://www.sml.ee.upatras.gr/UploadedFiles/InverseKinematics.pdf>
- [53] <https://www.scribd.com/doc/271891148/Arduino-MATLAB-MATuino-MATLAB-ARDUINO-serial-communication>

- [54] <https://www.mathworks.com/matlabcentral/fileexchange/47418-matlab-abd-arduino-uno-serial-comm-led-blinking>
- [55] https://www.mathworks.com/help/fuzzy/modeling-inverse-kinematics-in-a-robotic-arm.html?searchHighlight=forward%20kinematic&s_tid=doc_srchttitle
- [56] <https://www.robotshop.com/community/tutorials/show/arduino-5-minute-tutorials-lesson-5-servo-motors>
- [57] <https://mechatroface.com/arduino/servo-motor>
- [58] <https://www.youtube.com/watch?v=H1KPlms7s90&list=PLn8PRpmsu08p0r0uLo8vcxvQqHUIY6lkQ&index=19>
- [59] <http://cnx.org/content/m12062/latest>
- [60] <https://www.mathworks.com/matlabcentral/answers/126737-how-to-run-in-real-time-five-servos-connected-to-arduino-board>
- [61] https://www.mathworks.com/help/supportpkg/arduinoio/examples/control-servo-motors.html?s_tid=srchtitle
- [61] https://www.mathworks.com/matlabcentral/fileexchange/66164-moverobot?s_tid=srchtitle
- [63] https://www.mathworks.com/matlabcentral/answers/12732-pwm-signal-simulink-on-serial-port?s_tid=srchtitle
- [64] https://www.mathworks.com/search/site_search.html?c%5B%5D=entire_site&q=control+servo+using+serial+port
- [65] <http://forums.trossenrobotics.com/tutorials/how-to-diy-128/complete-control-of-an-arduino-via-serial-3300/?page=2>
- [66] <http://allaboutee.com/2011/07/04/how-to-send-data-from-the-arduino-to-matlab/>
- [67] <http://forum.arduino.cc/index.php?topic=95509.0>
- [68] <https://arduino.stackexchange.com/questions/54050/how-can-i-read-servo-motor-readings-via-serial-communication>
- [69] <https://www.projehocam.com/arduino-matlab-haberlesmesi-ile-servo-motor-kontrolu>
- [70] <https://www.mathworks.com/examples/matlab/community/38023-experiment-8-using-a-servo-motor>
- [71] Inverse%20Kinematics%20for%20Robot%20Arm
- [72] inverse-paper
- [73] kinematics-pdf
- [74] Diplomatiki-papazoglou
- [75] ΔΗΜΟΚΡΙΤΕΙΟ-αντίστροφη κινηματική χειριστών
- [76] (Intelligent Robotics and Autonomous Agents series)
- [77] arduino_notebook_v1-1
- [78] arduino_paliouras.pdf
- [79] arm matlab-arduino.pdf
- [80] Arpatzani_Final.pdf
- [81] ΠΡΟΓΡΑΜΜΑ ΕΠΙΣΤΗΜΟΝΙΚΩΝ ΜΕΛΕΤΩΝ 2011 Ρομποτικά Συστήματα για Σκοπούς Εξερεύνησης και Διάσωσης.pdf
- [82] delikonstantinou robot arms.pdf
- [83] Eisagwgh_Sto_Matlab.pdf
- [84] intro-robots.pdf
- [85] ΒΕΛΤΙΣΤΟΣ ΓΕΩΜΕΤΡΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ ΧΩΡΙΚΟΥ ΒΡΑΧΙΟΝΑ RRR ΜΕ ΧΡΗΣΗ ΥΒΡΙΔΙΚΟΥ ΑΛΓΟΡΙΘΜΟΥ
Δ. Σαγρής, Σ. Μήτση, Κ.-Δ. Μπουζάκης, Γκ. Μανσούρ
Εργαστήριο Εργαλειομηχανών και Διαμορφωτικής Μηχανολογίας, Τμήμα Μηχανολόγων Μηχανικών, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Ελλάδα

- [86] Sagris, D., Mitsi, S., Bouzakis, K.-D., Mansour, G. (2004), "Geometric Design Optimization of Spatial RR Robot Manipulator Using a Hybrid Algorithm", Acta Technica Napocensis, Series: Applied Mathematics and Mechanics, No 47, pp.717-722
- [87] Υλοποίηση ρομποτικού βραχίονα με χρήση της πλατφόρμα Arduino. Πτυχιακή Εργασία
Σέφη Τατιανή-Μπανάβου Νίκη
- [88] Ρομποτικός βραχίονας με χρήση λογισμικού MATLAB & ARDUINO-ΣΜΑΡΑΓΔΑΚΗΣ ΕΜΜΑΝΟΥΗΛ
- [89] ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΤΑΣΚΕΥΗ ΚΑΙ ΕΛΕΓΧΟΣ ΚΑΤΑΚΟΡΥΦΟΥ ΑΡΘΡΩΤΟΥ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ ΤΕΣΣΑΡΩΝ ΒΑΘΜΩΝ ΕΛΕΥΘΕΡΙΑΣ
- [90] Kinematics <https://en.wikipedia.org/wiki/Kinematics>
- [91] ΕΤΗ Ζυριχή ASL Autonomous Systems Lab
- [92] Σημειώσεις Δρ.Καζαρλή Σπυρίδων ΠΜΣ Ρομποτικής Μάθημα Ρ-201
- [93] Σημειώσεις Δρ.Καζαρλή Σπυρίδων ΠΜΣ Ρομποτικής Μάθημα Ρ-202
- [94] Σημειώσεις ΠΜΣ Ρομποτικής Μάθημα Ρ-101 Ενσωματωμένα συστήματα
- [95] Σημειώσεις ΠΜΣ Ρομποτικής Μάθημα Ρ-102 Εισαγωγή στη Ρομποτική-κινηματική
- [96] Σημειώσεις ΠΜΣ Ρομποτικής Μάθημα Ρ-102 ROS Σχεδίαση&προσομείωση ρομποτικών συστημάτων
- [97] Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942–1948.
doi:10.1109/ICNN.1995.488968
- [98] Jump up to: a b Shi, Y.; Eberhart, R.C. (1998). "A modified particle swarm optimizer". Proceedings of IEEE International Conference on Evolutionary Computation. pp. 69–73.
- [99] Jump up^ Taherkhani, M.; Safabakhsh, R. (2016). "A novel stability-based adaptive inertia weight for particle swarm optimization". Applied Soft Computing. 38: 281–295.
doi:10.1016/j.asoc.2015.10.004.
- [100] https://nereus.mech.ntua.gr/courses/robotics/robotics_pdf/intro.pdf
- [101] https://nereus.mech.ntua.gr/courses/robotics/robotics_pdf/Ch2.pdf
- [102] https://nereus.mech.ntua.gr/courses/robotics/robotics_pdf/Ch3.pdf
- [103] Σημειώσεις εργαστηρίου του κ.Φαχαντίδη Μάθημα Ρ-103 ΠΜΣ Ρομποτικής
- [104] Σημειώσεις εργαστηρίου τει σερρών τμήματος μηχανολογίας Συστημάτων Παραγωγής και Ρομποτική Δρ. Δημητρίου Σαγρή
- [105] Σημειώσεις Φασουλά Ιωάννη τμήματος Μηχανολογίας ΤΕΙ Κρήτης Εισαγωγή στη Ρομποτική
- [106] Denavit and Hartenberg (DH) Parameters
- [107] Excerpt from Chapter 5 of the book "Introduction to Robotics" by S.K. Saha, Tata McGraw-Hill, New Delhi, (2008) July 28, 2010
- [108] Βιβλίο Κινηματικής αρθρωτών Βραχιόνων Ζωή Δουλιγέρη
- [109] https://www.researchgate.net/publication/282648910_Kinematic_Modeling_of_Robotic_Manipulator
- [110] https://www.academia.edu/5944982/Kinematics_for_the_SCARA_and_the_cylindrical_manipulators
- [111] https://www.researchgate.net/publication/280111963_Inverse_Kinematic_Models_for_Mobile_Manipulators
- [112] https://www.researchgate.net/publication/264212870_An_Analytical_Solution_of_the_Inverse_Kinematics_Problem_of_Industrial_Serial_Manipulators_with_an_Orthoparallel_Basis_and_a_Spherical_Wrist
- [113] Mathworks

- [114] Arduino_cc
- [115] ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ ΤΟΥ ΠΑΧΙΔΗ ΘΕΟΔΩΡΟΥ ΕΛΕΓΧΟΣ ΤΡΟΧΙΑΣ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ ΜΕ ΤΗ ΧΡΗΣΗ ΨΕΥΔΟΣΤΕΡΕΟΣΚΟΠΙΚΗΣ ΤΕΧΝΗΤΗΣ ΟΡΑΣΗΣ
- [116] Διδακτορική Διατριβή του ΝΙΚΟΛΑΟΥ ΦΑΧΑΝΤΙΔΗ Υβριδικός έλεγχος θέσης/δύναμης ρομποτικού βραχίονα
- [117] Διάλεξη Ιωάννη Φασουλά ΠΜΣ Ρομποτικής ΤΕΙ ΚΜ 2017
- [118] ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ ΠΛΙΑΚΟΣ ΒΥΡΩΝ OFF-LINE ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΒΙΟΜΗΧΑΝΙΚΟΥ ΒΡΑΧΙΟΝΑ ΜΕ ΕΞΙ ΑΡΘΡΩΣΕΙΣ ΠΕΡΙΣΤΡΟΦΗΣ KAWASAKI RS-005L
- [119] <https://sci-hub.tw>
- [120] Διδακτορική διατριβή ΔΗΜΗΤΡΙΟΣ ΣΤ. ΣΑΓΡΗΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΤΟΠΟΘΕΤΗΣΗΣ ΚΑΙ ΚΙΝΗΣΕΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΡΟΜΠΟΤ ΜΕΣΩ ΥΒΡΙΔΙΚΗΣ ΜΕΘΟΔΟΥ
- [121] Δ. Εμίρης, Ρομποτική, Εκδόσεις Άνωση, Αθήνα, 1999.
- [122] Ph.J. McKerrow, Introduction on robotics, Addison – Wesley Publishing Company, Wokingham, 1993.
- [123] J.J. Craig, Introduction to robotics, mechanics and control, Addison-Wesley, Reading, 1986.
- [124] J. Denavit, R.S. Hartenberg, A kinematic notation for lower pair mechanisms based on matrices, ASME Journal of Applied Mechanisms, E 22, pp.215-221, 1955.
- [125] M. Prinz, H.C. Liu, B.O. Nnaji, T. Lueth, From CAD-based kinematic modeling to automated robot programming, Robotics and Computer Integrated Manufacturing, Vol. 12, No. 1, pp.99-109, 1996
- [126] Σ. Μήτση, Χωρικοί μηχανισμοί – βιομηχανικά ρομπότ, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, 1995
- [127] Δ. Εμίρης, Ρομποτική, Εκδόσεις Άνωση, Αθήνα, 1999
- [128] Ph.J. McKerrow, Introduction on robotics, Addison – Wesley Publishing Company, Wokingham, 1993
- [129] J.J. Craig, Introduction to robotics, mechanics and control, Addison-Wesley, Reading, 1986.
- [130] F.L. Lewis et.al, Mechanical engineering handbook “Robotics”, Ed. Frank Kreith, Boca Raton, CRC Press LLC, 1999
- [131] S.L. Chiu, Task compatibility of manipulator postures, International Journal of Robotics Research, Vol.7, No.5, pp.13-21, 1988
- [132] J.F. Petiot, P. Chedmail, J.Y. Hascoet, Contribution to the scheduling of trajectories in robotics - mathematics, Programming and control, Robotics and Computer-Integrated Manufacturing, Vol.14, No.3, pp.237-251, 1998.
- [133] L. Zlajpah, On time optimal path control of manipulators with bounded joint velocities and torques, In Proceedings IEEE Int. Conf. on Robotics and Automation, Minneapolis, 1996.
- [134] <https://arduinobots.wordpress.com>
- [135] <https://www.pemptousia.gr/2012/12/ta-rompot-stin-ipiresia-tou-anthropou>
- [136] <https://www.youtube.com/watch?v=GRNKYtz0jxQ>

- [137] <https://www.youtube.com/watch?v=ZySGP4AwGCY>
[138] https://www.youtube.com/watch?v=f_uRbiGT_fU
[139] <https://www.instructables.com/id/ArduinoServoPotentiometer>
[140] <https://www.arduino.cc/en/Reference/Servo>
[141] Lynxmotion's SSC-32
[142] <https://www.instructables.com/id/Controlling-3-servo-motors-with-3-potentiometers>
[143] <https://www.allaboutcircuits.com/projects/servo-control-with-arduino-through-matlab>
[144] <https://eclass.teiwm.gr/modules/document/file.php/BSMM.pdf>
[145] <https://www.youtube.com/watch?v=Y8ueTjqC>
[146] http://okeanis.lib.teipir.gr/xmlui/bitstream/handle/123456789/1601/aut_00467.pdf
[147] <https://blog.robotiq.com/how-to-calculate-a-robots-forward-kinematics-in-5-easy-steps>
[148] CRAIG_BOOK.pdf
[149] <https://eclass.teicrete.gr/modules/document/file.php/TM118/1>
[150] http://nemertes.lis.upatras.gr/jspui/bitstream/10889/10438/6/Nemertes_Valsamos%28aer%29.pdf [151] http://www.emo.org.tr/ekler/4f226e3f94cd5bb_ek.pdf _YILDIZ
TECHNICAL UNIVERSITY
ELECTRICAL AND ELECTRONICS FACULTY DEPARTMENT OF COMPUTER
ENGINEERING SENIOR PROJECT
KINEMATIC ANALYSIS FOR ROBOT ARM

Τιμοκατάλογος Υλικών

Υλικό	Χαρακτηριστικά/διαστάσεις	Τιμή
Βάση Ερταλόν	500mm x 500mm x 20mm	100 ευρώ
6 x Servo	MG996R	22 ευρώ
Μπουλόνια πάκτωσης βάσης	M8 x 60	4 ευρώ
Ημιέτοιμη κατασκευή αλουμινίου για το σκελετό	Επιπλέον παξιμάδια και βίδες	4 ευρώ
6 x Metal discs για τα Servo	Επιπλέον παξιμάδια, βίδες, ροδέλλες.	7 ευρώ
Εξωτερικό τροφοδοτικό	CP-1027 12V/10A-120W	12 ευρώ
Adjustable converter	DC-DC converter 12A/200w	10 ευρώ
Bread board	1600 οπές	8 ευρώ
Πλακέτα Arduino	Arduino Uno R3	8 ευρώ
Καλώδια	Διάφορα μεγέθη	5 ευρώ
6 x Ποτενσιόμετρα	10 Ω	1,20 ευρώ

ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ

Σχήμα 2.1. Στροφική άρθρωση	σελ.33
Σχήμα 2.2. Περιστροφική άρθρωση	σελ.33
Σχήμα 2.3. Κυλινδρική άρθρωση	σελ.34
Σχήμα 2.4. Άρθρωση Κύληση.....	σελ.34
Σχήμα 2.5 Ελεύθερη Άρθρωση.....	σελ.35
Σχήμα 2.6 Σφαιρική άρθρωση.....	σελ.35
Σχήμα 2.7 Κάθε αριθμός κι ένας βαθμός ελευθερίας.....	σελ.37
Σχήμα 2.8 AC-motor.....	σελ.40
Σχήμα 2.9 Οριακές τιμές κινητήρα servo.....	σελ.45
Σχήμα 2.10 Αναπαράσταση εσωτερικού κυκλώματος οδήγησης servo-κινητήρα.....	σελ.47
Σχήμα 2.11 οι γωνίες RPY (roll – pitch – yaw).....	σελ.58
Σχήμα 4.2 Σφαίρα Επαναληπτικότητας και Ακρίβειας του άκρου.....	σελ.78
Σχήμα 4.3 Κινηματική ανάλυση με τριγωνομετρία.....	σελ.81
Σχήμα 4.4 Βραχίονας περιστροφικών αρθρώσεων 3 βαθμών ελευθερίας.....	σελ.82
Σχήμα 4.5 Επιβεβαίωση παρ.1 στο σχεδιαστικό πρόγραμμα.....	σελ.85
Σχήμα 4.6 Επιβεβαίωση παρ. 3 στο σχεδιαστικό πρόγραμμα.....	σελ.86
Σχήμα 4.7 Επιβεβαίωση παρ.4 στο σχεδιαστικό πρόγραμμα.....	σελ.87
Σχήμα 4.8 Επιβεβαίωση παρ. 5 στο σχεδιαστικό πρόγραμμα.....	σελ.88
Σχήμα 4.9 Επιβεβαίωση παρ. 6 στο σχεδιαστικό πρόγραμμα.....	σελ.89
Σχήμα 4.10 Επιβεβαίωση παρ.7 στο σχεδιαστικό πρόγραμμα.....	σελ.90
Σχήμα 4.11 Βραχίονας 3 Dof.....	σελ.93
Σχήμα 4.12 Άξονες κίνησης του βραχίονα.....	σελ.94
Σχήμα 4.13 Χρήση νόμων γεωμετρίας(α).....	σελ.95
Σχήμα 4.14 Χρήση νόμων γεωμετρίας(β).....	σελ.95
Σχέδιο 4.15 Διάταξη βραχίονα για την εξαγωγή εξισώσεων ανάστροφης κινηματικής.....	σελ.97
Σχήμα 4.16 τριγωνομετρική συνάρτηση atan2.....	σελ.98
Σχήμα 4.17 παράδειγμα 1.....	σελ.102
Σχήμα 4.18 παράδειγμα 3.....	σελ.105

Σχήμα 4.19 παράδειγμα 4.....σελ.107
Σχήμα 4.20 παράδειγμα 5.....σελ.109
Σχήμα 4.21 παράδειγμα 6.....σελ.110
Σχήμα 4.22 παράδειγμα 7.....σελ.112
Σχήμα 4.23 παράδειγμα 8.....σελ.114
Σχήμα 4.24 παράδειγμα 9.....σελ.115
Σχήμα 5.1 Όλα τα μέρη που απαρτίζουν την πλακέτα Arduino Uno.....σελ.122

Συνομογραφίες-Επεξηγήσεις

DOF(Degrees Of Freedom) = Βαθμοί ελευθερίας

DOM(Degrees Of Mobility) = Βαθμοί κινητικότητας

PSO (Partical Swarm Optimization)=Βελτιστοποίηση σμήνους σωματιδίων

Robot arm=Ρομποτικός Βραχίονας/ Manipulator=χειριστής-βραχίονας

Inverse kinematic =Ανάστροφο κινηματικό

Forward kinematic =Ευθύ κινηματικό

Path planning=σχεδιασμός διαδρομής

Servo=Σερβοκινητήρας

Link= Σύνδεσμος

Joint= Άρθρωση

Actuators=Ενεργοποιητές

Microcontroller=Μικροελεγκτής

Obstacle Aavoidance=Αποφυγή εμποδίων

End-effector =Τελικό σημείο δράσης

Atan2=τόξο εφαπτωμένης

acos=ανάστροφο συνημίτονο